

<b>Fecha:</b> 16/07/2024	<b>Duración:</b> 2 HORAS	
<b>Evaluación:</b> EXAMEN	<b>Uso de calculadora</b> NO	
<b>Materia:</b> PROGRAMACION II	<b>Uso de Material:</b> NO	<b>Página 1 de 2</b>
<b>Turno:</b> UNICO	<b>Puntaje Máximo:</b> 100 Puntos	<b>V4</b>

“ElectroCartel” diseña y repara carteles luminosos. Cada cartel tiene una matriz rectangular, donde en cada posición hay un luz, que puede estar prendida o apagada. La imagen mostrada en el cartel se desplaza en cada segundo una posición a la derecha (en forma circular, ver ejemplos más adelante).

En el sistema debe ser posible:

- Registrar técnico:** se indica nombre y dirección
- Registrar cliente:** se indica nombre, rut y teléfono
- Registrar cartel:** se indica técnico que lo realiza, cliente, diseño (matriz booleana que representa el cartel inicial, cada posición representa: true: prendido, false: apagado), descripción y costo.
- Simulación de desplazamiento.** dada una matriz que contiene un diseño inicial de cartel y una cantidad de segundos, retornar una matriz que representa cómo se visualiza el cartel luego de esos segundos. En cada segundo se desplaza una posición a la derecha y al llegar al borde, retoma el diseño desde la izquierda, en forma circular.

Ejemplo:

Dado este diseño inicial de cartel:	si se solicita el cartel al segundo 1 (o al segundo 8, etc)	si se solicita el cartel al segundo 2 (o al segundo 9, ...):

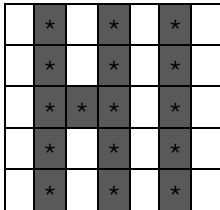
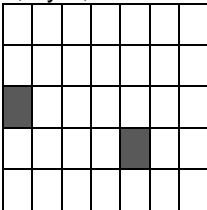


- Análisis de luces.** Se dispone de dos archivos: “Cartel.txt”: contiene la matriz inicial de un cartel y “Roturas.txt”: contiene las posiciones rotas. Se desea procesar los dos archivos. Asumiendo disponible y **utilizando** el método de simulación de desplazamiento de d), teniendo en cuenta las posiciones rotas retornar cuál es el primer segundo (segundo 0, 1, 2,...) en el cual no se verá el cartel completo correctamente o -1 si se verá siempre bien. Para verse completamente bien, todas las posiciones indicadas como “prendidas” del cartel deben estar funcionando. Asumir no hay inconsistencias.

Archivos:

<p>“Cartel.txt”:</p> <p>La primera línea contiene:</p> <p>CantidadDeFilas # CantidadDeColumnas</p> <p>Luego, por cada línea del cartel, se indica una secuencia de “P” (corresponde a prendido) y “A” (apagado), en el segundo 0.</p> <p>Ejemplo:</p> <p>5#7</p> <p>APAPAPA</p> <p>APAPAPA</p> <p>APPPAPA</p> <p>APAPAPA</p> <p>APAPAPA</p>	<p>“Roturas.txt”</p> <p>Contiene en cada línea la indicación de la fila y columna de una luz que no funciona, separadas por “#”</p> <p>Ejemplo:</p> <p>3#4</p> <p>2#0</p>
---	---

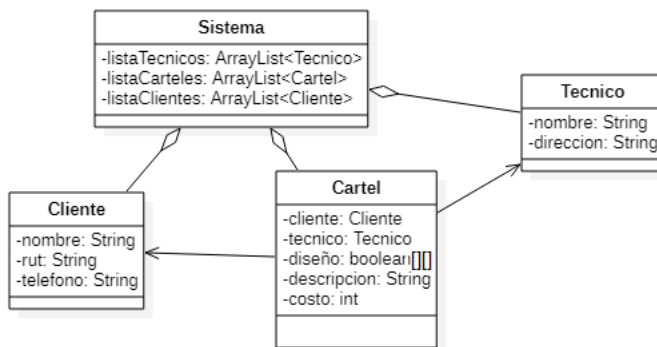
<b>Fecha:</b> 16/07/2024	<b>Duración:</b> 2 HORAS	
<b>Evaluación:</b> EXAMEN	<b>Uso de calculadora</b> NO	
<b>Materia:</b> PROGRAMACION II	<b>Uso de Material:</b> NO	<b>Página 2 de 2</b>
<b>Turno:</b> UNICO	<b>Puntaje Máximo:</b> 100 Puntos	<b>V4</b>

Ejemplos:

Dado este cartel, en el segundo 0: 	Si las posiciones rotas son: 3,4 y 2,0: 	El cartel se ve correctamente en el segundo 0, pero en el segundo 1 hay luces que deben prenderse y no funcionan.
Si el cartel fuera: 	Y las posiciones rotas son 1,0; 1,1 y 1,2: 	Retornaría -1, pues puede mostrarse completamente todo el tiempo.

- f) **Consulta de costos.** Se indica monto mínimo y monto máximo, al presionar el botón de consulta debe mostrarse en una lista todos los carteles cuyo costo esté en el rango indicado (considerando ambos montos). Debe visualizarse el nombre del cliente, la descripción del cartel y costo. La lista debe verse ordenada crecientemente por monto. Si no hay datos para mostrar, indicarlo con una ventana emergente.

A partir de realizar el análisis del problema, se diseñó la siguiente jerarquía en UML donde se asumen disponibles métodos de acceso y modificación de atributos simples y constructores sin parámetros. **NO** pueden agregarse atributos.



Se pide:

- (25 puntos) Implementar en Java en la clase Sistema el pedido d) simulación de desplazamiento.  
**firma: public boolean[][] simular(boolean[][] mat, int segundos)**
- (40 puntos) Implementar en Java en la clase Sistema el método del pedido e) Análisis de luces.  
**firma: public int analisisLuces()**
- (30 puntos) Implementar el pedido f) Consulta de costos, según la interfaz:

Carteles

Monto minimo:

Monto maximo:

- ☐ cajaMinimo [JTextField]
- ☐ cajaMaximo [JTextField]
- ☒ botonConsultar [JButton]
- ☒ jScrollPane1 [JScrollPane]
- ☐ listaCarteles [JList]

Se asume disponible la variable Sistema modelo.

Anotar en Java el constructor de la ventana y el código completo asociado a los eventos a procesar.

**Lógica, claridad, comentarios, presentación: 5 puntos.** Si se usan métodos auxiliares, implementarlos.