EJERCICIO ESTATICOS

En el proyecto de Estaticos, crear los paquetes:

- com.krakedev.estaticos.ejercicio.test
- com.krakedev.estaticos.ejercicio.entidades
- com.krakedev.estaticos.ejercicio.logica
- com.krakedev.estaticos.ejercicio.utils

Crear una clase Alarma en el paquete entidades, que tiene los atributos dia, hora y minuto, los 3 de tipo entero.

Agregar un constructor que recibe los tres parámetros y los agrega a los atributos correspondientes.

Agregar getters y setters

Sobreescribir el método toString

Crear una clase de constantes llamada DiasSemana, en el paquete utils, con los valores

LUNES 0

MARTES 1

MIERCOLES 2

JUEVES 3

VIERNES 4

SABADO 5

DOMINGO 6

Crear una clase AdminAlarmas en el paquete logica, con un atributo llamado alarmas del tipo ArrayList<Alarma>

Agregar un método agregarAlarma, que recibe una Alarma y la agrega a la lista.

Agregar un método get para el atributo alarmas, no agregar el método set.

TEST

Crear una clase TestAlarmas en el paquete test, con un método main.

En el main:

Instanciar varias alarmas y agregarlas a la lista usando el método agregarAlarma de AdminAlarmas.

IMPORTANTE, cuando instancie la alarma usar las constantes de DiasSemana

Ejemplo

```
new Alarma(DiasSemana.LUNES, 5,45); //en lugar de new Alarma(0,5,45);
```

Recuperar la lista de Alarmas de AdminAlarmas e imprimir la información

```
AdminAlarmas admin = new AdminAlarmas();

//agregar alarmas

ArrayList<Alarma> alarmasActuales = admin.getAlarmas();

System.out.println(alarmasActuales);
```

En este caso, el método println, invoca al método toString de alarmasActuales que es de tipo ArrayList<Alarma> y tiene el método toString sobreescrito, internamente se barre todo el ArrayList e invoca al método toString de cada Alarma que lo sobreescribimos.

Crear una clase llamada Util, en el paquete util

Agregar un método **estático** llamado formatearHora, recibe como parámetro un entero, lo convierte a String de dos caracteres y retorna el valor. La forma más fácil de convertirlo es concatenar con un String, por ejemplo:

```
String valor = numero + "";
```

Si el número es de 1 dígito, se agrega un cero delante, por ejemplo si llega el 5, debe retornar 05

Agregar un método **estático** llamado formatearDia, recibe como parámetro un entero y retorna el String equivalente, así si llega el 0, retorna "Lunes", si es 1, retorna "Martes", etc.

Usar los métodos de Util para dar formato al dia, hora y minutos en la sobreescritura del toString de Alarma. INVOCAR CORRECTAMENTE A LOS MÉTODOS ESTÁTICOS.

El formato en el que se muestra debe ser por ejemplo Lunes, 05:24

Ejecutar nuevamente el Test y verificar que la información se muestra formateada como se solicitó.