

Tabla de Contenidos



- Fechas y Tareas
- Responsabilidades y fechas
- Calendar y DateFormat

Cuando empezamos a programar en Java y tenemos pocos conocimientos intentamos hacer las cosas de la forma mas sencilla posible y a veces nos llevamos algunas sorpresas. Este es el caso del manejo de Java y Fechas . Normalmente todo el mundo busca una clase que se llame “Date” o algo similar. Una vez que la hemos encontrado la usamos de la siguiente manera.

```
public class Principal {  
  
    public static void main (String args[]) {  
  
        Date fecha= new Date();  
        int horas=fecha.getHours();  
        System.out.println(horas);  
  
    }  
}
```

Sin embargo si vemos el código dentro de eclipse podremos darnos cuenta de que el método `getHours()` se encuentra deprecado y no se recomienda su uso.

```
public static void main (String args[]) {  
    Date fecha= new Date();  
    int horas=fecha.getHours();  
    System.out.println(horas);  
  
}
```

Fechas y Tareas

Muchas personas se sorprenden de esta casuística ya que parece no existir una forma mas sencilla de solicitar la hora . Sin embargo si pensamos un poco mas en cuales son las tareas con las que nosotros solemos trabajar con fechas nos encontraremos con algo como lo siguiente.

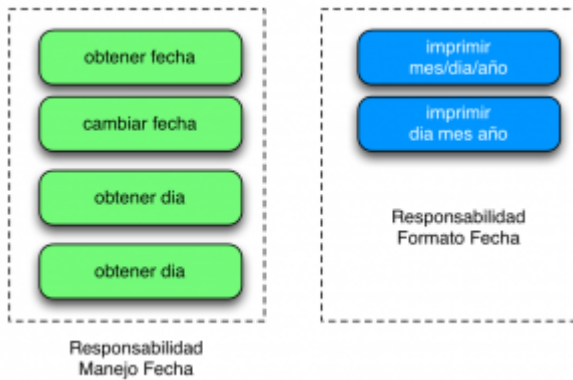


Como podemos ver son muchos tipos de tareas a realizar .Ahora bien estas tareas pueden agruparse de la siguiente forma.



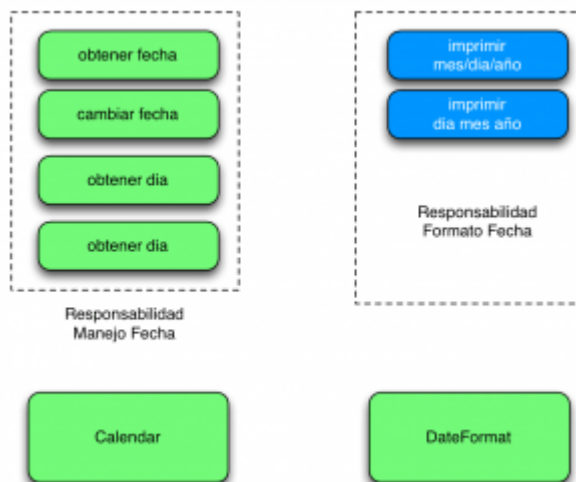
Responsabilidades y fechas

Las tareas ahora aparecen en distintos colores ya que se agrupan por “Tipo” de responsabilidad. Unas tareas están orientadas a gestionar las fechas y hacer cálculos como obtener la hora ,cambiar la fecha etc .Mientras por otro lado existen otras tareas orientadas a imprimir las fechas en distintos formatos.



Calendar y DateFormat

De esta forma al encontrarnos con dos responsabilidades distintas para el manejo de fechas en java . El JDK nos proveerá de dos clases para solventar este tipo de problemática . La clase “Calendar” para el manejo de fechas y la clase DateFormat para la gestión de los



formatos.

Asi pues cuando trabajemos con fechas y queramos realizar operaciones puramente de gestión usaremos la clase Calendar en donde se ubican todos los métodos que necesitamos

```
public class PrincipalCalendario {
```

```

public static void main (String args[]) {

    Calendar calendario=Calendar.getInstance();
    Date fecha=calendario.getTime();
    calendario.setTime(fecha);
    System.out.println(calendario.get(Calendar.HOUR_OF_DAY));

}
}

```

Por otro lado cuando queramos formatear una fecha con por ejemplo dia/mes/año usaremos la clase `DateFormat` y `SimpleDateFormat` nos ayudarán a gestionar estas operaciones.

```

public class PrincipalCalendarioDateFormat {

    public static void main (String args[]) {

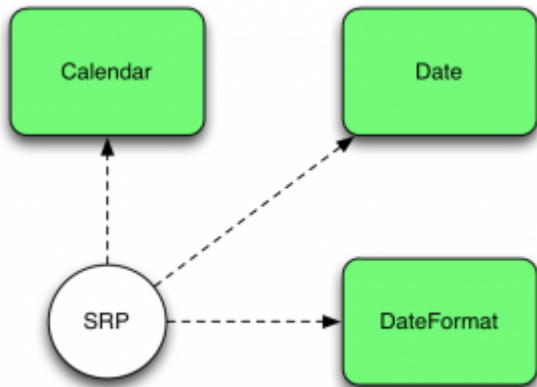
        Calendar calendario=Calendar.getInstance();
        Date fecha=calendario.getTime();
        calendario.setTime(fecha);
        System.out.println(calendario.get(Calendar.HOUR_OF_DAY));
        DateFormat formato= new SimpleDateFormat("dd/MM/yyyy");
        System.out.println(formato.format(fecha));

    }
}

```

De esta forma el JDK cumple con el principio SRP de asignar a cada clase una única

responsabilidad.



Puedes encontrar mas detalles sobre este principio en mi libro.