

## FORMATO DE FECHAS

Puedes formatear las fechas con diversos métodos, aquí te propongo dos ejemplos probados:

1.- En este ejemplo se ve usar una clase en java SimpleDateFormat que tiene un constructor que recibe el formato de fechas con el que trabajará la app, las líneas te darán la fecha de hoy con el formato correspondiente 10-10-2023 por ejemplo.

```
SimpleDateFormat sdf = new SimpleDateFormat ("dd-MM-YYYY");
Date d = new Date();
System.out.println("sdf, format{d}"); // de fecha a string
```

2.- de String a Date con el método parse, que puede usar una excepción, en este ejemplo, saltará cuando no coincida con el formato que definimos anteriormente.

```
String s = "10-10-2023";
try
{
    d= sdf.parse(s);
}
catch (ParseException e)
{
    System.out.println("Hubo un error con la fecha")
}
System.out.println(d.toString());
```

Por ejemplo en una clase de prueba queda implementado de la siguiente manera

```

public class PruebaFechas {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        SimpleDateFormat sdf = new SimpleDateFormat("dd-MM-yyyy");
        Date d = new Date();
        System.out.println(sdf.format(d));

        String s = "10-10-2010";
        try
        {
            d = sdf.parse(s);
        }
        catch (ParseException e)
        {
            System.out.println("Hubo un error con la fecha");
        }

        System.out.println(d.toString());
    }
}

```

La salida sería algo parecido a esto

```

Output - PruebaFechas (run)
run:
16-10-2015
Sun Oct 10 00:00:00 CEST 2010
BUILD SUCCESSFUL (total time: 0 seconds)

```

Ahora inténtalo integrar con el ejercicio anterior de clientes:

```

private Date fechaAlta;
private String provincia;
private SimpleDateFormat sdf = new SimpleDateFormat("dd-MM-yyyy");

```

y en

```

public String[] toArrayString()
{
    String[] s = new String[4];
    s[0] = nombre;
    s[1] = apellidos;
    s[2] = sdf.format(fechaAlta);
    s[3] = provincia;
    return s;
}

```

