# Conversión de objetos de fechas-horas en cadenas de caracteres y viceversa.

## Conversión de objetos fechas-horas en cadenas de caracteres.

Declaración de los métodos **format** que serán utilizados.

Los métodos format dado que siempre convierten hacia el mismo tipo de dato (String) y siempre se recibe como parámetro un tipo de dato diferente para cada método (Calendar, Date, LocalDateTime), la nomenclatura del método será siempre la misma, solo la palabra **format**, de esta manera el programador no deberá preocuparse por el método a ejecutar, siempre y cuando utilice un tipo de dato correcto. Los métodos format tendrán dos especificaciones, una en la cual se reciben dos parámetros el objeto a convertir en cadena de caracteres y el patrón de formato y la otra especificación en la cual solo se recibirá un solo parámetro el objeto a convertir en cadena de caracteres, utilizando un patrón de formato por defecto, de esta manera si el programador necesita utilizar diferentes patrones de formato, podrá realizarlo a través de un solo método, o en cambio si necesita utilizar siempre el mismo patrón de formato, solo debe utilizar el método que modifique el patrón por defecto y a partir de ahí utilizar el método format que recibe un solo parámetro.

* Format (Calendar, String): String: convierte un objeto de tipo Calendar, en una cadena de caracteres, dado un patrón de conversión.
* Format (Calendar): String: convierte un objeto de tipo Calendar, en una cadena de caracteres, utilizando el patrón de conversión por defecto.
* Format (Date, String): String: convierte un objeto de tipo Date, en una cadena de caracteres, dado un patrón de conversión.
* Format (Date): String: convierte un objeto de tipo Date, en una cadena de caracteres, utilizando el patrón de conversión por defecto.
* Format (LocalDateTime, String): String: convierte un objeto de tipo LocalDateTime, en una cadena de caracteres, dado un patrón de conversión.
* Format (LocalDateTime): String: convierte un objeto de tipo LocalDateTime, en una cadena de caracteres, utilizando el patrón de conversión por defecto.

## Conversión de cadenas de caracteres en objetos de fechas-horas

Declaración de los métodos **parse** que serán utilizados.

Los métodos parse siempre devuelven un objeto de diferente tipo de dato (Calendar, Date LocalDateTime) y siempre reciben como parámetro el mismo tipo de dato (String), por tanto los métodos parse tendrán una nomenclatura especifica **parseTo** *nombre de la clase del tipo de dato a convertir.* Para cada método parse existirán dos especificaciones una en la cual se reciben dos parámetros una cadena de caracteres que será la fecha a convertir en objeto y el otro parámetro una cadena de caracteres que será el patrón de formato de la fecha, la segunda especificación será un método parse que recibirá un solo parámetro la cadena de caracteres que será la fecha a convertir y utilizara el patrón por defecto para realizar la conversión, de est manera si el programador necesita utilizar diferentes formatos, puede utilizar la primera especificación o de lo contrario si siempre utilizara el mismo formato, puede establecer un formato por defecto y después utilizar la segunda especificación.

* parseTo(String, String):Calendar: convierte un cadena de caracteres en un objeto de tipo Calendar, dado un patrón de formato.
* parseTo(String):Calendar: convierte un cadena de caracteres en un objeto de tipo Calendar, utilizando el patrón de formato por defecto.
* parseTo (String, String):Date: convierte un cadena de caracteres en un objeto de tipo Date, dado un patrón de formato.
* parseTo(String): Date: convierte un cadena de caracteres en un objeto de tipo Date, utilizando el patrón de formato por defecto,
* parseTo(String, String):LocalDateTime: convierte un cadena de caracteres en un objeto de tipo LocalDateTime, dado un patrón de formato.
* parseTo(String): LocalDateTime: convierte un cadena de caracteres en un objeto de tipo LocalDateTime, utilizando el patrón de formato por defecto.