# Declaración de clases y métodos.

Clase DateUtils

## Generalización de métodos:

* Métodos convert:
  + Los métodos que comiencen con la palabra **convert** será los métodos utilizados para realizar la conversión de objetos de fechas hacia otros objetos de fechas.
  + El nombre de los métodos tendrá la siguiente convención: **convert From** *nombre de la clase* de la cual se va a realizar la conversión **To** *nombre de la clase* a la cual se va a realizar la conversión.
  + Para cada método convert se le creara di es necesario su método convert que sirva como reciproco, o sea, realice la conversión opuesta. Si el método no existe, esto se debe a que la conversión puede realizarse de manera directa sin necesidad de usar la librería DateUtils.
* Métodos format:
  + Los métodos que comiencen con la palabra **format** serán los métodos encargados de realizar la conversión de objetos de fechas hacia cadena de caracteres.
  + El nombre de los métodos **format** siempre tendrá la misma convención, solo tendrán la palabra format, dado que siempre recibirán como parámetros un objeto de fechas siempre diferente, el sistema lo reconocerá como métodos diferentes, además el usuario no se tiene que preocupar por el tipo específico de dato para realizar la conversión, esto porque los métodos se llamarían iguales.
  + Para cada método format tendrá dos convenciones en cuanto a los parámetros, uno de los métodos el cual recibirá como parámetro el objeto de fecha a convertir y una cadena de caracteres que especifique el patrón de la fecha al cual se va a convertir. La segunda convención es que existirá el mismo método format para el mismo tipo de dato, pero este no recibirá el patrón, sino que utilizara uno por defecto.
* Métodos parse:
  + Los métodos que comiencen con la palabra **parse**, serán los métodos encargados de convertir cadenas de caracteres en objetos de fechas. Estos métodos realizan la operación opuesta a los métodos **format**.
  + La convención de los métodos **parse**, será la siguiente: **parseTo** *nombre de la clase* a la cual se va a realizar a conversión.
* Método **setDefaultPattern**:
  + El método **setDefaultPattern** recibirá como parámetro una cadena de caracteres, el cual será el patrón de formato por defecto a utilizar en caso de que el programador no proporcione uno.
  + El método permite al programador en caso de siempre utilizar el mismo patrón de formato establecer el suyo propio y a partir de ahí, poder realizar las conversiones pertinentes de una manera sencilla.

## Objetos de tipo fecha-hora a utilizar:

* Calendar.
* Date.
* LocalDateTime.

## Objetos que no se utilizaran en la conversión:

* LocalDate: La clase no se utilizara en la conversión dado que puede ser devuelto de manera directa por un objeto de tipo LocalDateTime.
* LocalTime: La clase no se utilizara en la conversión dado que puede ser devuelto de manera directa por un objeto de tipo LocalDateTime.
* Java.sql.Date: La clase no se utilizara en versiones iniciales en la conversión, pero es posible que en próximas versiones, si se generen métodos para permitir la conversión de objetos hacia java.sql.Date.