

EJERCICIO 16: Intervalo De Tiempo

En Java, las fechas se representan normalmente con instancias de la clase [java.time.LocalDate](#).

Se pueden crear con varios métodos "static" como por ejemplo `LocalDate.now()`.

Investigar:

1) cómo hacer para crear una fecha determinada, por ejemplo 15/09/1972.

2) cómo hacer para determinar si la fecha de hoy se encuentra entre las fechas 15/12/1972 y 15/12/2032.

Sugerencia: vea los métodos permiten comparar `LocalDates` y que retornan booleans.

3) cómo hacer para calcular el número de días entre dos fechas.

Lo mismo para el número de meses y de años

Sugerencia: vea el método until.

Tenga en cuenta que los métodos de `LocalDate` colaboran con otros objetos que están definidos a partir de enums, clases e interfaces de `java.time`; por ejemplo `java.time.temporal.ChronoUnit.DAYS`.

Tareas:

1) Implemente:

Implemente la clase **DateLapse** (Lapso de tiempo).

Un objeto `DateLapse` representa el lapso de tiempo entre dos fechas determinadas.

La primera fecha se conoce como "*from*" y la segunda como "*to*".

Una instancia de esta clase entiende los siguientes mensajes:

// Retorna la fecha de inicio del rango.

public LocalDate getFrom()

// Retorna la fecha de fin del rango.

public LocalDate getTo()

// Retorna la cantidad de días entre la fecha 'from' y la fecha 'to'.

public int sizeInDays()

// Recibe un objeto `LocalDate` y retorna true si la fecha está entre el from y el to del receptor y false en caso contrario.

public boolean includesDate(LocalDate other)

2) Pruebas automatizadas:

- Diseñe los casos de prueba teniendo en cuenta los conceptos de valores de borde y particiones equivalentes vistos en la teoría.

- Implemente utilizando JUnit los tests automatizados diseñados en el punto anterior.