



Lógica con Fechas (Programación)

Recuerda aplicar la lógica a cada una de las funciones que se plantean, reflexionando sobre que se se pide, y en un contexto **NO INFORMÁTICO** lo resolverías.

Ejercicio 1: Diferencia Exacta Entre Fechas

Calcula la diferencia exacta en años, meses, días, horas, minutos y segundos entre dos fechas dadas: el 29 de febrero de 2020 a las 23:45:30 y el 1 de enero de 2025 a las 08:15:45.

Ejercicio 2: Último Viernes de Cada Mes

Escribe un algoritmo para encontrar el último viernes de cada mes durante el año 2023. Devuelve una lista de las fechas.

Ejercicio 3: Ajustar Horarios por Zonas Horarias

Dada la fecha y hora `2025-03-15T14:30:00` en la zona horaria de Nueva York, conviértela a las zonas horarias de Tokio, Londres y Sídney. Asegúrate de manejar correctamente los horarios de verano.

Ejercicio 4: Validación de Fechas Bisiestas

Crea un algoritmo que recorra los próximos 50 años y determine cuántos días 29 de febrero habrá durante ese período. Devuelve también las fechas específicas.

Ejercicio 5: Edad en Milisegundos

Calcula cuántos milisegundos ha vivido una persona nacida el 15 de julio de 1990 a las 03:30:00 (hora local). Devuelve también cuántos milisegundos ha vivido en años bisiestos.

Ejercicio 6: Próximo Eclipse Solar

Dada una lista de eclipses solares históricos y futuros con sus fechas y horarios en UTC, encuentra el próximo eclipse solar que ocurrirá después del 1 de enero de 2025. Calcula también cuánto tiempo falta exactamente para ese eclipse en años, meses, días, horas, minutos y segundos.

Salida esperada

Tests run: 34, Failures: 0, Errors: 0, Skipped: 0

--- jacoco-maven-plugin:0.8.10:report (report) @ logica-fechas-2 ---
Loading execution data file /Users/jpexposito/Downloads/logica-fechas/target/jacoco.exec
Analyzed bundle 'logica-fechas-2' with 6 classes
=====

BUILD SUCCESS

logica-fechas-2 > es.ies.puerto

Source Files Sessions

es.ies.puerto

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods	Missed	Classes
Ejercicio1	<div></div>	63 %	<div></div>	100 %	2	4	6	17	2	3	0	1
Ejercicio4	<div></div>	70 %	<div></div>	50 %	7	11	2	11	2	3	0	1
Ejercicio6	<div></div>	96 %	<div></div>	n/a	1	4	1	16	1	4	0	1
Ejercicio5	<div></div>	96 %	<div></div>	92 %	2	11	1	17	1	4	0	1
Ejercicio2	<div></div>	94 %	<div></div>	100 %	1	5	1	11	1	2	0	1
Ejercicio3	<div></div>	91 %	<div></div>	100 %	1	3	1	8	1	2	0	1
Total	59 of 417	85 %	9 of 40	77 %	14	38	12	80	8	18	0	6

Created with JaCoCo 0.8.10.202304240956

Programación defensiva

Realiza un diseño de programación defensiva.

Con ella lograrás un comportamiento adecuado ante parámetros de entrada no esperados.

Ejemplos

- [LocalDate](#).

Dudas

- Puede ser posible y recomendable hacer uso de funciones entre ejercicios.
- Durarion, Period, etc de la clase 'java.time' serán necesario.
- Si tienes alguna duda más guíate por los test unitarios. Serán tu mejor aliado.

