

TAREA PRÁCTICA SOBRE LA UNIDAD 5 – PARTE (II): AMPLIACIÓN Y UTILIZACIÓN DE LA CLASE DEPOSITO

1. Método `abrirGrifo`:

Sobrecargar el método `abrirGrifo` con una nueva versión que sea capaz de recibir dos parámetros de tipo `LocalTime` que indicarán la **hora de apertura y de cierre del grifo**.

- Este método hará lo mismo que su homónimo salvo que en lugar de recibir directamente el número de segundos, recibirá dos instantes (objetos de tipo `LocalTime` que indican hora, minuto y segundo) y a partir de ellos se podrá calcular nuevamente el número de segundos que se desea abrir el grifo.
- Dado que hay que hacer lo mismo en ambos métodos, una vez calculada la cantidad de segundos que ha de estar abierto el grifo, este método debería llamar al otro para **no contener código redundante ni incoherente**.
- Debe tenerse en cuenta que en este caso podría darse **una situación de error** que debería lanzar una **excepción si la hora de apertura es posterior a la hora de cierre**. En tal caso no se haría nada y se lanzaría una excepción de tipo `IllegalArgumentException` con el mensaje "Error: Parámetros de tiempo inválidos (el instante de inicio es posterior al de fin).".
- También se debería lanzar la excepción `IllegalArgumentException` con el mensaje "Error: Parámetros de tiempo inválidos (null).". si alguno de los parámetros es null, para así evitar una `NullPointerException`.
- Por último hay que tener en cuenta que este método también debe **devolver la cantidad de litros** que se han vertido durante el tiempo que el grifo ha estado abierto



2. Constructor copia:

Implementar un **constructor copia** que instance un clon exacto del objeto deposito que se pase como parámetro, con sus **mismas características y estado**, y que ello afecte de manera colateral a todos los valores históricos tanto del propio objeto como del global de depósitos, así como a la cantidad de depósitos completamente llenos y/o vacíos en el momento actual.

Si el parámetro recibido es null debe lanzarse una `NullPointerException`.



3. Programa de pruebas:

Hacer funcionar la clase `Deposito` con el programa principal de pruebas que se proporciona.

Rúbrica de corrección (Parte II):

El método sobrecargado <code>abrirGrifo</code> de la clase <code>Deposito</code> se ha implementado apropiadamente y funciona correctamente.	1,0
El constructor copia de la clase <code>Deposito</code> se ha implementado apropiadamente y funciona correctamente.	1,0
El programa de prueba proporcionado para hacer el test de la clase <code>Deposito</code> funciona correcta y apropiadamente.	1,0