

Programación en Lenguaje Java

Práctica 4.2. Mareógrafo



Michael González Harbour
Mario Aldea Rivas

Departamento de Matemáticas,
Estadística y Computación

Este tema se publica bajo Licencia:

[Creative Commons BY-NC-SA 4.0](https://creativecommons.org/licenses/by-nc-sa/4.0/)

Práctica 4-2: Mareógrafo

Objetivos: Practicar con arrays o `ArrayList`

Descripción: Se desea escribir parte del software de análisis de los datos obtenidos de un mareógrafo, que es un sistema que mide la altura del nivel del mar en un punto concreto

El mareógrafo proporciona de forma periódica una medida que se guarda en un objeto de la clase `Medida`, que ya está implementada

Se pide implementar la clase `Mareografo`, que tiene una lista de medidas y métodos para analizarlas, y el programa principal

Clase Medida

Atributos:

- **horaGMT**: es un texto con la fecha y hora referidos al meridiano de Greenwich
- **nivelDelMar**: en metros
- **alturaOlas**: en metros

Métodos:

- Constructor al que se le pasan los datos de la medida excepto la fecha y hora, que toma del reloj del sistema
- Observadores, uno para cada atributo

Medida
-String horaGMT -double nivelDelMar -double alturaOlas
+Medida (double nivelDelMar, double alturaOlas) +double nivelDelMar() +double alturaOlas() +String horaGMT()

Clase Mareografo

Atributos:

- **lista**: guarda una lista de medidas obtenidas en instantes consecutivos a lo largo del tiempo
 - El primer elemento de la lista es el más antiguo, y el último el más nuevo
- **codigo**: un String que identifica el mareógrafo
- **cadencia**: el tiempo entre cada medida, en minutos

Mareografo
-ArrayList<Medida> lista -String codigo -int cadencia
+Mareografo(String codigo, int cadencia) +int numMedidas() +Medida medida(int num) +double alturaMediaOlas(int intervalo) +boolean hayAlturaMayor(double alt) +muestraGrafica() +conecta()

Clase Mareografo: métodos

- *Constructor*: copia en los respectivos atributos el `codigo` y la `cadencia`. Además, crea el atributo `lista` vacío
- `numMedidas()`: retorna el número de medidas que contiene la lista
- `medida()`: retorna la medida que ocupa la posición `num` en la lista (se numeran desde cero). Si `num` indica una medida no existente en la lista se retornará `null`
- `alturaMediaOlas()`: Retorna la altura media de las olas en el último intervalo expresado por el parámetro `intervalo`, en minutos
 - Para ello se puede recorrer la lista de medidas desde atrás sumando las alturas de las olas en una variable
 - En total debe recorrer un número de casillas igual a `intervalo/cadencia`
 - Finalizado el bucle retorna la media de las alturas de las olas
 - Si el tamaño de la lista es inferior a `intervalo/cadencia` retorna `Double.NaN`

Clase Mareografo: métodos (cont.)

- `hayAlturaMayor()`: retorna un booleano que indica si entre las medidas hay alguna cuya suma del nivel del mar y de altura de olas supere el parámetro `alt`
- `muestraGrafica()`: parte avanzada: muestra una gráfica de la evolución de la altura de las olas y del nivel del mar
- `conecta()`: se conecta por satélite con el mareógrafo y obtiene sus medidas, guardándolas en la lista
 - Este método no se pide. Se da una simulación que lee los datos del fichero `mar.txt`, que debe estar en la carpeta del proyecto

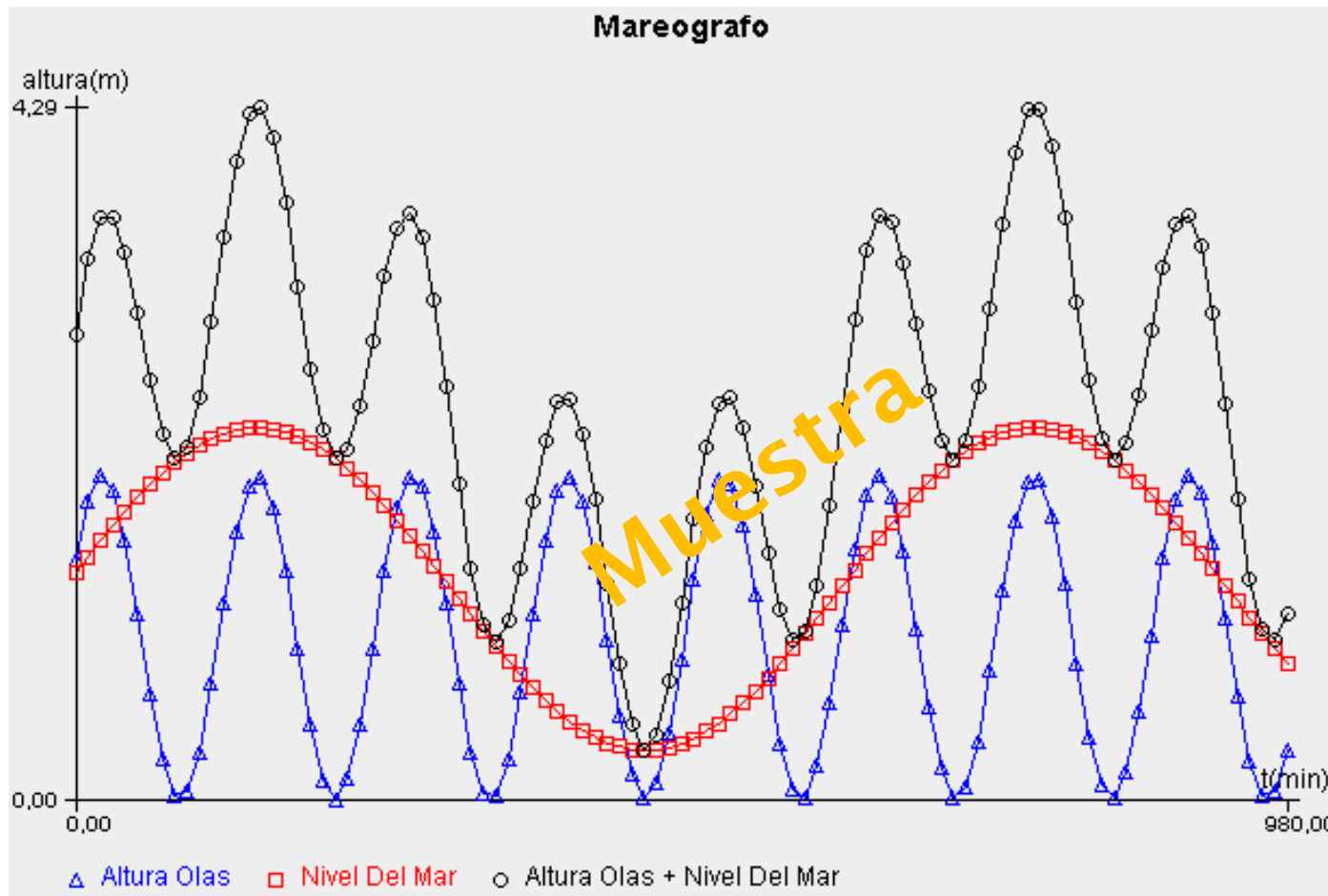
Clase MuestraMedidas

Esta clase contiene el programa principal que hace:

- a) Crear un objeto de la clase `Mareografo` con código `"PUERTO_CHICO_001"` y cadencia `10` minutos
- b) Invocar el método `conecta()` para establecer la conexión con el mareógrafo
- c) Mostrar en pantalla los datos de la primera medida de la lista y de la última
- d) Mostrar en pantalla la altura media de las olas en los últimos `60` minutos
- e) Mostrar en pantalla si hubo altura mayor a `10` metros o no
- f) Mostrar en pantalla si hubo altura mayor a `4` metros o no

Parte avanzada

Añadir a Mareógrafo el método muestraGrafica() para hacer una gráfica como esta, que muestre el nivel, altura de olas y su suma



Entrega

Entregar un informe en pdf que contenga:

- Código de la clase `Mareografo` (5 puntos)
- Código de la clase principal (1 punto)
- Captura de pantalla de los resultados de la ejecución del `main` (2 puntos)
- Parte avanzada: código del método `muestraGrafica()` (1 punto)
- Parte avanzada: captura de pantalla de la gráfica (1 punto)