

Prueba de programación C++

Sistema de captura de datos meteorológicos

Debemos implementar un programa “meteo-top” capaz de leer datos periódicos de una API HTTP de terceros y mostrarlos periódicamente por la salida estándar.

Requisitos:

1. Es un programa iterativo. Una vez en ejecución solo se puede parar por interacción de usuario (SIGINT, SIGTERM).
2. El programa debe conectarse a la API de terceros debe ser Open Meteo. Es pública y gratuita (<https://open-meteo.com/en/docs>).
3. Se deben publicar valores actuales de temperatura, humedad y velocidad del viento (`current_weather`). Ejemplo de la web con `curl`:

```
curl "https://api.open-meteo.com/v1/forecast
?latitude=52.52&longitude=13.41
&current_weather=true
&hourly=temperature_2m, relativehumidity_2m, windspeed_10m"

{
  "current_weather": {
    "time": "2022-01-01T15:00"
    "temperature": 2.4, "weathercode": 3,
    "windspeed": 11.9, "winddirection": 95.0,
  },
  "hourly": {
    "time": ["2022-07-01T00:00", "2022-07-01T01:00", ...]
    "windspeed_10m": [3.16, 3.02, 3.3, 3.14, 3.2, 2.95, ...],
    "temperature_2m": [13.7, 13.3, 12.8, 12.3, 11.8, ...],
    "relativehumidity_2m": [82, 83, 86, 85, 88, 88, 84, 76, ...],
  }
}
```

4. Con una periodicidad configurable por el usuario. El programa mostrará los tres valores procedentes de la API por la salida estándar (stdout). Se debe escoger un formato estructurado de representación de datos **diferente a JSON**.

Consideraciones:

Es posible que exista algún tipo de limitación del número o de la frecuencia de consultas a la API pública. Esto deberá tenerse en cuenta para cumplir los requisitos del sistema.

Evaluación:

1. Debe utilizarse el compilador gcc y como mínimo el estándar C++11 (<https://gcc.gnu.org/projects/cxx-status.html#cxx11>). Se ejecutará y evaluará el programa en un sistema Linux x86_64.
2. El programa se entregará a través de un repositorio público (Github, Bitbucket, Gitlab).
3. Debe incluir un “readme” con instrucciones para preparar el entorno de desarrollo, las librerías externas (si hubiese), construir el programa, configurarlo y ejecutarlo.
4. Además de cumplir requisitos se valorará:
 - a. Claridad y limpieza de código
 - b. Estructura de proyecto
 - c. Utilización de mecanismos y funcionalidades propias de C++: threads, smart pointers, vectors, strings, classes, templates...
 - d. Claridad y limpieza en los históricos de cambios de git
5. Responder a las preguntas del final del documento.
6. Si surge cualquier duda sobre la prueba asume la respuesta de forma justificada e indícalo.
7. Se debe especificar el tiempo total dedicado a esta evaluación.

Preguntas adicionales:

1. ¿Qué formato has escogido para serializar los datos de meteo en stdout? ¿Por qué?
2. ¿Qué pasaría si la API no responde?
3. ¿Has utilizado alguna librería externa? Justifica tu elección.
4. ¿Cómo de complicado sería migrar el sistema para que utilizase otra forma de comunicación en lugar de stdout?
5. Si tuvieras que diseñar este sistema teniendo en cuenta sólo los requisitos. ¿Utilizarías C++ u otra tecnología? ¿Qué te haría elegir C++ o esa tecnología?