

# Vacante programador C++/Perl/Python - Meteorológica S.A.

27 de mayo de 2019

## 1. Notas generales

Para continuar con el proceso de selección te pedimos que nos envíes resuelto por lo menos un ejercicio de los que te proponemos a continuación. Puedes enviar más de uno, pero con uno es suficiente. Si tienes dudas sobre cual abordar, te recomendamos el primero.

Te pedimos que resuelvas el ejercicio en C++, Perl o Python, y utilizando las bibliotecas que consideres oportunas, la única condición es que sean de código abierto para que podamos reproducirlo.

Por favor no envíes binarios ya que no los vamos a ejecutar, sólo código junto con lo necesario para compilar y/o ejecutar correctamente lo que has mandado. Es decir, si tu solución depende de 1000 bibliotecas, debes proporcionar algún método para que podamos reproducirlo de forma portable (si sólo funciona en una plataforma quizás no podamos reproducirlo). Por ejemplo en C++ el `CMakeLists.txt` correspondiente o puede que con un `Makefile` sea suficiente o nada si es un sólo archivo. Si utilizas Visual Studio, puedes mandar los archivos `sln` y `vcxproj`, pero nosotros utilizamos sistemas Linux, asique preferiríamos que utilizaras CMake (Visual Studio 2017 ya lo soporta)

Si habitualmente utilizas Git puedes enviarnos el contenido del repositorio y lo tendremos en cuenta. Basta con que incluyas el directorio `.git` junto con el resto del código.

Una de las tecnologías que utilizamos y valoramos es *Docker*, asique valoraremos que incluyas un `Dockerfile` (o similar) con las instrucciones necesarias para obtener un contenedor listo para ejecutar cada ejercicio.

Algunos de los ejercicios son ambiguos de forma intencionada, queremos que nos indiques qué decisiones has tomado en aquellas cuestiones que no tenías claras y el porqué.

## 2. Ejercicios

### 2.1. Sistema de predicción eólica sencillo

El ejercicio consiste en implementar un sistema de predicción de generación de energía eólica mediante la relación entre la producción de energía de un parque eólico y el viento. Esta relación podrá ser diferente para cada parque eólico, por lo que se necesita un sistema automático para obtener y evaluar la relación para diferentes parques eólicos.

#### 2.1.1. Método de predicción

En este caso vamos a suponer que hubiera una relación lineal entre módulo de velocidad de viento y la producción de energía. A través de los datos de producción de energía y el módulo de velocidad de

viento se trata de obtener una recta que proporcione la producción de energía a partir de la velocidad de viento. El ajuste de esta recta se hará utilizando una regresión Deming <sup>1</sup>.

### 2.1.2. Resultados a obtener

El ajuste obtenido se utilizará para dos cosas:

- Calcular los errores medio absoluto y cuadrático medio por mes que se está cometiendo con ese ajuste
- Realizar nuevas predicciones

### 2.1.3. Entrada de datos

La entrada de datos consistirá en dos bloques. El primero comenzará con la palabra observaciones. En la siguiente línea tendremos la fecha en formato YYYY-MM-DD hh:mm y la observación de producción junto con el módulo de la velocidad de viento. Como máximo se incluirán las observaciones de un año, pero pueden ser menos. El segundo bloque estará delimitado por la palabra predicciones y contendrá líneas con la fecha y hora y el módulo de la velocidad de viento. La entrada terminará con el carácter de fin de archivo. Las observaciones y las predicciones de entrada pueden no estar ordenadas.

Ejemplo:

```
observaciones
2008-01-25 10:00 1200 8
2008-01-15 11:00 1150 7.5
predicciones
2009-10-25 23:00 4
2009-10-20 14:00 5
```

En el archivo adjunto `Meteorologica_vacante_ProgC_ProblemaDatos_20190318.txt` hay unos datos de ejemplo como entrada del programa.

### 2.1.4. Salida de datos

La salida de datos consistirá en varias partes. La primera parte será la recta de ajuste, primero la pendiente y luego la ordenada en el origen, con dos decimales de precisión. La segunda parte consistirá en el error por mes que se está cometiendo en ese ajuste respecto a los datos de producción proporcionados, con dos decimales de precisión. Primero aparecerá el error medio absoluto en porcentaje de producción y después el error cuadrático medio en porcentaje de producción<sup>2</sup>. La última parte consistirá en la predicción que se daría de producción para cada uno de los valores introducidos en la parte predicciones de la entrada, pero de forma ordenada, sin decimales.

Ejemplo:

```
100.00 400.00
Mes1 0.00 0.00
2009-10-20 14:00 900
2009-10-25 23:00 800
```

---

<sup>1</sup>Las fórmulas de esta regresión se pueden consultar en el apéndice A

<sup>2</sup>Las fórmulas para calcular estos errores se pueden consultar en el apéndice B

### 2.1.5. Limitaciones de implementación

Para el cálculo del error medio absoluto se debe utilizar la función proporcionadas en el archivo de código fuente adjunto.

La función para el cálculo del error cuadrático medio la debe implementar el candidato.

## 2.2. API HTTP de predicciones

Queremos hacer disponibles las predicciones a través de una API HTTP. La idea es que cualquier cliente se pueda conectar a esa API, y pedir la predicción para una ubicación y una fecha. El protocolo que vamos a utilizar en este caso es HTTP, y las peticiones deben ir en JSON, así como la respuesta. Para pedir predicciones realizaremos una petición POST cuyo contenido sea similar a este:

```
{
  "ubicacion": {
    "type": "Feature",
    "geometry": {
      "type": "Point",
      "coordinates": [125.6, 10.1]
    },
    "properties": {
      "name": "Dinagat Islands"
    }
  },
  "fecha": "2019-01-15"
}
```

La ubicación está en formato GeoJSON. Los datos a devolver serán aleatorios, devolveremos la temperatura máxima para la fecha solicitada y los siguientes 10 días. Ejemplo de respuesta:

```
{
  "ubicacion": {
    "type": "Feature",
    "geometry": {
      "type": "Point",
      "coordinates": [125.6, 10.1]
    },
    "properties": {
      "name": "Dinagat Islands"
    }
  },
  "prediccion": [
    {"fecha": "2019-01-15",
     "Tmax": 22.5},
    {"fecha": "2019-01-16",
     "Tmax": 22.5},
    {"fecha": "2019-01-17",
     "Tmax": 22.5},
    {"fecha": "2019-01-18",
     "Tmax": 22.5},
    {"fecha": "2019-01-19",
```

```

    "Tmax": 22.5},
    {"fecha": "2019-01-20",
    "Tmax": 22.5},
    {"fecha": "2019-01-21",
    "Tmax": 22.5},
    {"fecha": "2019-01-22",
    "Tmax": 22.5},
    {"fecha": "2019-01-23",
    "Tmax": 22.5},
    {"fecha": "2019-01-24",
    "Tmax": 22.5}
  ]
}

```

## 2.3. Construcción económica de vallas

Tenemos algunos bosques por el campo, y queríamos construir una valla alrededor para que no entre ningún pirómano. Cada uno de esos bosques tiene los árboles distribuidos de forma irregular, y queremos vallarlos de la forma más económica posible, es decir, minimizando la longitud total de la valla. El ejercicio consiste en hacer un pequeño programa que dada una serie de puntos nos devuelve el polígono que corresponderá a la valla (basta con los vértices de ese polígono). Ejemplo entrada:

```

0 0
0 2
2 0
2 2
1 1

```

Ejemplo salida:

```

0 0
0 2
2 2
2 0

```

Este programa debe funcionar para entradas aleatorias.

Para que puedas comprobar si tu algoritmo está funcionando bien, queremos que hagas una pequeña visualización del resultado. Para hacer esta visualización utilizaremos la biblioteca CppPlotly que genera un html que se puede ver en el navegador. Esta biblioteca consiste sólo de archivos de cabecera y depende únicamente de json11

## A. Fórmulas de regresión Deming

La regresión Deming es un método de regresión ortogonal. Reproducimos aquí las fórmulas necesarias para la implementación:

$$\bar{x} = \frac{1}{n} \sum x_i, \quad (1)$$

$$\bar{y} = \frac{1}{n} \sum y_i, \quad (2)$$

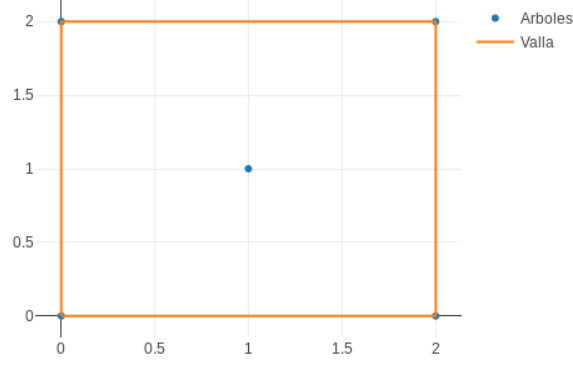


Figura 1: Ejemplo correspondiente a los árboles y valla de la entrada y salida mostrada

$$s_{xx} = \frac{1}{n-1} \sum (x_i - \bar{x})^2, \quad (3)$$

$$s_{xy} = \frac{1}{n-1} \sum (x_i - \bar{x})(y_i - \bar{y}), \quad (4)$$

$$s_{yy} = \frac{1}{n-1} \sum (y_i - \bar{y})^2 \quad (5)$$

$$\hat{\beta}_1 = \frac{s_{yy} - \delta s_{xx} + \sqrt{(s_{yy} - \delta s_{xx})^2 + 4\delta s_{xy}^2}}{2s_{xy}} \quad (6)$$

$$\hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x} \quad (7)$$

Para este problema utilizaremos  $\delta = 1$ .

El candidato interesado puede completar información en: [http://en.wikipedia.org/wiki/Deming\\_regression](http://en.wikipedia.org/wiki/Deming_regression)

## B. Fórmulas de error

$$EMA \% Producción = \frac{100}{\sum_i^n O_i} \cdot \sum_i^n |O_i - P_i| \quad (8)$$

$$ECM \% Producción = \frac{100n}{\sum_i^n O_i} \cdot \sqrt{\frac{\sum_i^n (O_i - P_i)^2}{n}} \quad (9)$$