

1) (DEBE RESOLVERSE EN LENGUAJE PYTHON)

LinkedIn

LinkedIn, la red social laboral por excelencia, nos solicita confeccionar un software capaz de proveer estadísticas de interés respecto a los perfiles cargados en su plataforma. Los perfiles están compuestos por datos sencillos como un nombre de usuario, Nombre y Apellido de la persona reflejada en el perfil, información respecto a sus estudios, a su experiencia laboral, a sus aptitudes y el dominio de las mismas, a su participación en la red y también referencias de otros usuarios que han trabajado con la persona durante su desarrollo profesional.

Para ello, LinkedIn nos brinda un archivo donde se encuentra un conjunto de información respecto de cada uno de los perfiles, a saber:

perfiles.csv

Campos:

Nombre de Usuario, Nombre y Apellido, Título de Grado (S/N), Título de Posgrado (S/N), Cantidad de posteos en los últimos 6 meses, Aptitud 1, Dominio 1, ... , Aptitud N, Dominio N

Valores:

costaguido, Guido Costa, S, N, 7, Gestión de Proyectos, E, Liderazgo de Equipos, E, Python, A, OLAP, I
garciagisela, Gisela Garcia, S, S, 38, Java, E, SQL, E, Machine Learning, B
gonzalezarely, Arely Gonzalez, S, N, 12, Python I, Datawarehousing B

Donde las valoraciones del dominio de una aptitud son las siguientes:

- “E” = Experto
- “A” = Avanzado
- “I” = Intermedio
- “B” = Básico

A su vez nos proporcionan un archivo donde podemos visualizar las referencias positivas sobre una aptitud de un usuario a otro:

Valoraciones.csv

Campos:

Nombre de Usuario Referente, Nombre de Usuario Referido, aptitud valorada

Valores:

costaguido,garciagisela,Java
costaguido,garciagisela,SQL
garciagisela,gonzalezarely,Datawarehousing
gonzalezarely,garciagisela,Java
gonzalezarely,costaguido,Python

El software debe permitir hacer lo siguiente:

1. Se debe poder ingresar un nuevo perfil con la información correspondiente incluyendo las aptitudes y respectivos dominios (se debe controlar que no se dupliquen usuarios). *La información debe quedar almacenada en el archivo al finalizar la ejecución.*
2. Se debe poder ingresar una nueva referencia entre usuarios (se debe controlar que los usuarios sean existentes). *La información debe quedar almacenada en el archivo al finalizar la ejecución.*
3. **Reporte de Formación:** Indicar que % de los usuarios tienen un título de Posgrado. De dichos usuarios debe indicarse cuál es el top 3 de aptitudes que posee ese conjunto de perfiles, indicando aptitud y cantidad de perfiles que la dominan sin importar el nivel de dominio (ordenado por cantidad de perfiles de modo descendente).
4. **Reporte de Participación en red:** Dada una aptitud ingresada por el usuario, debe indicar el promedio de posteos según el nivel de dominio que posean ordenado por promedio de posteos descendente. Ej Python:
 - a. Python Básico – 3.2 posteos
 - b. Python Avanzado – 1.2 posteos
 - c. Python Intermedio – 1.1 posteos
 - d. Python Experto – 0 posteos
5. **Reporte de Referencias:** Dada una aptitud ingresada por el usuario, debe armarse el Top 3 de usuarios que mayor puntaje de referencias tienen, valorizando las referencias según el dominio que posean (Experto = 4, Avanzado = 3, Intermedio = 2, Básico = 1). El mismo debe exportarse al archivo {aptitud}.csv indicando nombre de usuario y puntaje obtenido, ordenado de manera ascendente por puntaje.

Aclaración 1: Se debe utilizar funciones para resolver cada uno de los puntos

Aclaración 2: Es obligatorio usar try-except por lo menos en un lugar y justificar su uso

Aclaración 3: Es obligatorio el uso de un diccionario y una lista como mínimo

Aclaración 4: Debe existir un menú para poder llamar a las opciones a gusto del usuario

2) (DEBE RESOLVERSE EN LENGUAJE C)

Escribir un programa en C que permita contar las ocurrencias entre números del 0 al 9. El mismo debe ser resuelto con una función que reciba como parámetro un vector de números e imprima por pantalla el resultado y a su vez debe ejemplificarse con un consumo de dicha función desde el main, donde también debe solicitar los números del vector al usuario. Se supone un límite máximo de 50 números.

Ejemplo:

Input: [0,2,2,3,1,3,1,4,2,2]

Output:

0: 1

1: 2

2: 4

3: 2

4: 1

5: 0

...

9: 0