

 Deusto Facultad de Ingeniería Ingeniaritza Fakultatea	EXAMEN DE PROGRAMACIÓN I	19-ENERO-2023
---	-----------------------------	---------------

Nombre: _____

DNI: _____ **Aula:** _____ **Fila:** _____ **Columna:** _____

Instrucciones

- El ejercicio a pie de máquina se entregará al terminar el examen. Se recogerán en pendrive por parte del profesor.
- NO está permitida la conexión a internet en el examen. Tampoco la utilización de tu propio portátil ni la consulta de apuntes o de otros códigos.
- Descarga el fichero 'hinojosa.csv' de ALUD y guárdalo en una **carpeta** que se llame **"ExamenFinal-NombreApellidos"**, por ejemplo: ExamenFinal-PabloGaraizar.
- Guarda el código de todos los ejercicios en un **único fichero** dentro de la carpeta anterior que se llame **"ExamenFinal-NombreApellidos.py"**, por ejemplo: ExamenFinal-JennyFajardo.py.

Nos escriben desde Instagram preocupados por la salud mental de quienes usan su red social. Quieren que les ayudemos a analizar el estado de ánimo en las publicaciones en castellano y para ello nos sugieren utilizar un estudio científico de Hinojosa y cols. (2015) donde se catalogan 875 palabras con una puntuación de 0 (muy negativa) a 10 (muy positiva). Además, nos piden trabajar con una versión simplificada de su modelo de datos en el que hay estas dos entidades:

Post

Almacena los datos que nos interesan de cada publicación en Instagram para este estudio:

- **uid:** identificativo único de cada post (int), por ejemplo: 83211.
- **text:** texto que acompaña a la publicación (str), por ejemplo: 'Feliz 2023'.
- **hashtags:** lista de hashtags que se han usado en la publicación (lista de str), por ejemplo: ['feliz', '2023', 'alegría']

User

Almacena los datos que nos interesan de cada cuenta de Instagram en este estudio:

- **username:** cadena de caracteres (string), por ejemplo: 'bosanz'.
- **posts:** lista de posts (lista de elementos de la entidad descrita anteriormente).

Un ejemplo de user con varios posts podría ser:

```
user1 = {
  'username': 'ablago',
  'posts': [
    {'uid': 134, 'text': 'Feliz 2023', 'hashtags': ['feliz', '2023', 'alegría']},
    {'uid': 253, 'text': '¡Feliz año!', 'hashtags': ['deseos', 'esperanza']},
    {'uid': 124, 'text': 'Mal examen de inglés', 'hashtags': ['inglés', 'examen', 'tristeza']}
  ]
}
```

Teniendo esto en cuenta, programa las siguientes funciones:

- **datos prueba:** función que recibe una lista de users vacía y la rellena introduciendo al menos 2 users que tengan al menos 3 posts cada uno y 3 hashtags en cada post. Define estos datos con variables locales de esta función y añade los users a la lista recibida (no es necesario solicitar información mediante input). **(0,5 puntos).**

- **cargar palabras:** función que carga el fichero 'hinojosa.csv' en una estructura de datos que puede ser una lista de dos dimensiones (palabra y puntuación) o un diccionario (clave: palabra, valor: puntuación) y devuelve esa estructura de datos. **(1,5 puntos).**

```
lista_palabras = [ ['abandono', 1.47], ['abatir', 3.03], ['abierto', 5.6], ['abismo', 2.77]...]

dic_palabras = {
  'abandono': 1.47,
  'abatir': 3.03,
  'abierto': 5.6,
  'abismo': 2.77,
  ...}
```

- **generar users:** función que recibe la lista de users y las palabras cargadas anteriormente y añade 1000 users a la lista con 20 posts cada uno y 3 hashtags en cada post. Los usernames de cada user serán 'user-0', 'user-1'... 'user-999' y los posts de cada user tendrán como uid un valor aleatorio entre 111111 y 999999 (no puede estar repetido en toda la lista de users), como text 'text-0', 'text-1'... 'text-19' y como hashtags, 3 palabras elegidas aleatoriamente (no pueden estar repetidas dentro del mismo post) de entre las palabras cargadas anteriormente. **(2,5 puntos).**
- **post menor puntos:** función que recibe la lista de users y las palabras cargadas anteriormente y devuelve el post que **menor media de puntos** tenga de entre todos los posts de todos los users. Para calcular la media de puntos de un post es necesario calcular la media de las puntuaciones de cada una de las palabras de sus hashtags. Si una palabra no está entre las palabras cargadas anteriormente, se le asignará una puntuación de 5 para el cálculo de esta media. Por ejemplo, un post cuyos hashtags sean ['feliz', '2023', 'alegría'], tendrá como media de puntos: $(\text{puntos}('feliz') + \text{puntos}('2023') + \text{puntos}('alegría')) / 3 = (5 + 5 + 8.6) / 3 = 6.2$ **(2 puntos).**
- **clasificar users:** función que recibe la lista de users y las palabras cargadas anteriormente y devuelve un diccionario en el que en la clave 'mal' se almacenará una lista de todos los usernames de users que tienen una media de puntuación en todos sus posts menor que 5, en la clave 'bien' se almacenará una lista de todos los usernames de users que tienen una media de puntuación en todos sus posts mayor que 6 y en la clave 'regular' todos los usernames del resto de users. La media de puntos de un post se calcula de la misma forma que en la función anterior **(2 puntos).**

```
dicc = {
  'mal': ['bosanz', 'mlguenaga'],
  'regular': ['ablago', 'garaizar', 'aeguiluz'],
  'bien': ['jfajardo']
}
```

- **guardar clasificacion:** función que recibe el diccionario de la función anterior y almacena en el fichero 'clasifica.csv' la lista de usernames y su clasificación **(1,5 puntos).** Por ejemplo:

```
bosanz;mal;
ablago;regular;
...
```

Tu programa principal debería ser algo así:

```
users = [] # Lista de users vacía

datos_pruebas(users) # Creamos los users de prueba

palabras = cargar_palabras() # Cargar palabras y puntuaciones desde hinojosa.csv

generar_users(users, palabras) # Generamos 1000 users con 3 posts y 3 hashtags

print(post_menor_puntos(users, palabras)) # Mostramos post con menor media

clasificacion = clasificar_users(users, palabras) # Clasificamos a los users

guardar_clasificacion(clasificacion) # Guardamos la clasificación en clasifica.csv
```