 <b>Deusto</b> Facultad de Ingeniería Ingeniaritza Fakultatea	SIMULACRO PARCIAL 2	NOVIEMBRE 2023
---	---------------------	----------------

**Nombre:** \_\_\_\_\_

**DNI:** \_\_\_\_\_ **Fila:** \_\_\_\_\_ **Columna:** \_\_\_\_\_

## Instrucciones

- El ejercicio a pie de máquina se entregará al terminar el examen. Se recogerán en pendrive por parte del profesor.
- Crea una carpeta con tu nombre y apellido (por ejemplo: XabierCantero) en el Escritorio y guarda ahí todos los ficheros necesarios para resolver el examen.
- 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.

Vamos a hacer dos modelos A y B. Usa un fichero diferente por modelo (modelo\_a.py y modelo\_b.py). Tenemos un fichero CSV con datos de IMDB con las siguientes columnas:

- **Título:** Título de la película.
- **Rating:** Nota media que ha obtenido la película en todas las reviews.
- **Número de reviews:** Número total de usuarios que han valorado esa película.

Este es un ejemplo de cómo se ve el fichero:

```
Carmencita;5.6;1648
Le clown et ses chiens;6.1;198
Pauvre Pierrot;6.5;1352
Un bon bock;6.2;120
```

## Modelo A

### Lectura del fichero

Programa una función **cargar\_datos** que recibe una lista vacía y carga en esa lista los datos del fichero CSV como registros (listas), donde el primer elemento es el título, el segundo el rating y el tercero el nº de reviews. Recuerda que el primer elemento ha de ser un string, el segundo un float y el tercero un entero.

```
peliculas = []
cargar_películas(peliculas)
print(len(peliculas)) # Devuelve 824778
```

### Nota media


Programa una función **nota\_media** que recibe la lista anterior y calcula la nota media de todas las películas.

```
print(nota_media(peliculas)) # Devuelve 6.91
```

### Nota película

Programa una función **obtener\_nota** que reciba la lista anterior y el nombre de una película y devuelva la nota que tiene esa película.

```
print(obtener_nota(peliculas, "War Games")) # Devuelve 8.8
```

 <b>Deusto</b> Facultad de Ingeniería Ingeniaritza Fakultatea	SIMULACRO PARCIAL 2	NOVIEMBRE 2023
---	---------------------	----------------

## Nota media mínimo reviews

Programa una función **nota\_media\_reviews** que reciba la lista anterior y un número, y obtenga la nota media de todas las películas que tengan al menos el número de reviews que se pasa por parámetro.

```
print(nota_media_reviews(peliculas, 600)) # Devuelve 7.06
```

## Película más reviews

Programa una función **mas\_reviews** que reciba la lista anterior y devuelva la película que tiene mayor número de reviews.

```
print(mas_reviews(peliculas)) # Devuelve 'The Shawshank Redemption'
```

## Modelo B

### Lectura del fichero

Programa una función **cargar\_datos** que se encarga de leer el fichero CSV y devuelve un diccionario donde cada la clave es el nombre de la película y el valor es otro diccionario con las claves reviews y nota (y cada clave tiene asociado el respectivo valor)

```
peliculas = cargar_datos()
print(len(peliculas)) # Devuelve 824778
```

### Review media

Programa una función **review\_media** que reciba el diccionario anterior y calcule el número medio de reviews de todas las películas.

```
print(review_media(peliculas)) # Devuelve 670.82
```

### Review película

Programa una función **reviews\_película** que recibe el diccionario anterior y el nombre de una película y obtiene el número de reviews de esa película.

```
print(reviews_película(peliculas, "War Games")) # Devuelve 6
```

### Películas entre notas

Programa una función **peliculas\_según\_notas** que reciba el diccionario anterior y dos notas, nota\_min y nota\_max, y guarda en un fichero (con el mismo formato que el original) las películas cuya nota está entre ambos valores (incluidos).

```
# El fichero peliculas_filtradas.csv tiene 423860 filas
```

### Mejor película

Programa una función **mejor\_pelicula** que reciba el diccionario anterior y obtenga la película con mejor nota.

```
print(mejor_pelicula(peliculas)) # Devuelve 'Behind the Scenes'
```