

Evaluación Parcial N°3

Sigla	Nombre Asignatura	Tiempo Asignado	% Ponderación
FPY1101	Fundamentos de Programación	3 h	30%

1. Situación evaluativa

<input checked="" type="checkbox"/>	Ejecución práctica
-------------------------------------	--------------------

2. Agente evaluativo

<input checked="" type="checkbox"/>	Heteroevaluación	<input type="checkbox"/>	Coevaluación	<input type="checkbox"/>	Autoevaluación
-------------------------------------	------------------	--------------------------	--------------	--------------------------	----------------

3. Tabla de Especificaciones

Resultado de Aprendizaje	Indicador de Logro (IL)	Indicador de Evaluación (IE)	Ponderación Indicador Logro	Ponderación Indicador de Evaluación ¹
RA3 Construye estructuras de datos temporales, utilizando arreglos para almacenar datos, con el fin de darle persistencia a éstos y trabajar con problemas que necesiten guardar información temporalmente.	IL 3.1 Identifica colecciones y arreglos que permitan el almacenamiento de datos según los requerimientos del problema planteado.	IE 3.1.1 Identifica colecciones y arreglos que permitan el almacenamiento de datos según los requerimientos del problema planteado.	20%	20%
	IL 3.2 Utiliza arreglos y matrices para la inserción, eliminación, modificación y búsqueda de datos temporales para cumplir con los requerimientos del problema planteado.	IE 3.2.1 Utiliza arreglos y matrices para la inserción, eliminación, modificación y búsqueda de datos temporales para cumplir con los requerimientos del problema planteado.	20%	20%
	IL 3.3 Utiliza archivos para lograr la persistencia de los datos de la aplicación según el caso planteado.	IE 3.3.1 Utiliza archivos para lograr la persistencia de los datos de la aplicación según el caso planteado.	10%	10%
RA4 Modulariza programas, mediante funciones con y sin parámetros, para simplificar y darle eficiencia a las soluciones construidas.	IL 4.1 Utiliza librerías de sistema para la optimización del código según el caso planteado.	IE 4.1.1 Utiliza librerías de sistema para la optimización del código según el caso planteado.	20%	20%
	IL 4.2 Programa funciones que permitan la reutilización de código según el problema planteado.	IE 4.2.1 Programa funciones que permitan la reutilización de código según el problema planteado.	20%	20%
	IL 4.3 Integra funciones invocadas desde el programa principal para dar solución al problema planteado.	IE 4.3.1 Integra funciones invocadas desde el programa principal para dar solución al problema planteado.	10%	10%
Total			100%	100%

¹ La ponderación del Indicador de Logro, debe ser distribuida como puntaje entre las preguntas que lo conformen.

4. Rúbrica

Para ayudar su desarrollo, le entrego la rúbrica de cómo será revisada en detalle su evaluación.

Criterio	Puntos
Lectura de Archivos	TOTAL 10 ptos
Función “leer archivo”	10
Funciones	TOTAL 40 ptos
Función “contar likes”	10
Función “contar comentarios”	10
Función “total interacciones”	10
Función “calcular edad”	10
Listas y Arreglos	TOTAL 10 ptos
Identificación de top 3 usuarios	10
Escritura de archivos	TOTAL 20 ptos
Función “escribir reporte”	20
Funcionamiento General	TOTAL 30 ptos
Código ejecuta sin errores	10
Código cumple con los requisitos	20
TOTAL	110 Puntos

5. Instrucciones generales para el/la estudiante

Esta es una evaluación que corresponde a una prueba de ejecución práctica y tiene un **30%** de ponderación sobre la nota final de la asignatura.

El **tiempo** para desarrollar esta evaluación es de **180 minutos** y se realiza de manera **individual** en **laboratorio**

La evaluación consiste en:

- Desarrollo de una aplicación en Python y en entorno de desarrollo Visual Studio Code
- Uso de colecciones
- Uso de archivos de texto
- Uso de librerías estándar de Python

6. Evaluación

Estimado/a estudiante,

Esta prueba tiene como objetivo evaluar tu capacidad para leer y escribir archivos, utilizar funciones, y manejar listas y arreglos en Python. El tema de esta prueba será el análisis de interacciones en una red social ficticia.

Análisis de Interacciones en Redes Sociales

Contexto: Supongamos que trabajas para una empresa que administra una red social. Tu tarea es analizar los datos de interacción de los usuarios, que se almacenan en un archivo CSV.

Ejemplos de Archivos Proporcionados:

- usuarios.csv:

```
id_usuario,nombre_usuario,fecha_nacimiento
usuario1@example.com,usuario1,1990-01-01
usuario2@example.com,usuario2,1985-05-15
usuario3@example.com,usuario3,1992-07-20
...
```

- interacciones.csv:

```
id_usuario,id_post,tipo_interaccion
usuario1@example.com,post1,like
usuario2@example.com,post2,comment
usuario3@example.com,post1,share
...
```

Requisitos de la Prueba:

Parte 1: Lectura de Archivos

- Crea una función `leer_archivo` que lea un archivo CSV y devuelva los datos en una lista de listas.
 - Utilizar la función para leer un archivo CSV llamado `interacciones.csv`.
 - Utilizar la misma función para que lea el archivo CSV llamado `usuarios.csv`.

Parte 2: Funciones

- Crea una función `contar_likes` que reciba los datos de interacciones y devuelva el número total de "likes" que se han realizado.
- Crea una función `contar_comentarios` que reciba los datos de interacciones y devuelva el número total de "comentarios" que se han realizado.
- Crear una función `total_interacciones` que reciba el total de "comentarios" y el total de "likes" que se han realizado y devuelva el total de interacciones realizadas por el usuario.
- Crea una función `calcular_edad` que reciba una fecha de nacimiento y devuelva la edad actual del usuario.

Parte 3: Listas y Arreglos

- Identifica los 3 usuarios que más interactúan y marca estos usuarios en el archivo de salida (con un booleano `TRUE`).

Parte 4: Escritura de Archivos

- Crea una función `escribir_reporte` que reciba un diccionario con las interacciones por usuario, los datos de usuarios, y escriba un archivo CSV llamado `reporte_interacciones.csv` con el siguiente formato:

<code>id_usuario, nombre_usuario, edad, total_likes, total_comments, numero_interacciones, es_top_usuario</code>
--

Objetivo:

- Utiliza las funciones que has creado para leer los datos de `usuarios.csv` e `interacciones.csv`.

- Cuenta el total de "likes".
- Cuenta el total de "comentarios".
- Cuenta el total de interacciones
- Calcula la edad de los usuarios.
- Identifica los usuarios que más interactúan.
- Cruza la información de ambos archivos para generar el archivo reporte_interacciones.csv con los campos solicitados.

Entrega:

Sube tu código comprimido si tienes más de un archivo o en archivo Python (.py) y asegúrate de incluir el archivo CSV de salida `reporte_interacciones.csv`.

Ejemplo de Salida (reporte_interacciones.csv):

```
id_usuario,nombre_usuario,total_likes,total_comments,numero_interacciones,edad,es_top_usuario
usuario1@example.com,usuario1,2,1,3,34,True
usuario2@example.com,usuario2,1,1,2,39,True
usuario4@example.com,usuario2,1,4,5,25,True
usuario3@example.com,usuario3,0,0,0,31,False
```

Ayudas:

Manejo de datetime

```
from datetime import datetime

fecha_nacimiento_str = '1990-01-01'
fecha_nacimiento = datetime.strptime(fecha_nacimiento_str, '%Y-%m-%d')
```

Cálculo de la edad

```
from datetime import datetime

def calcular_edad(fecha_nacimiento):
    fecha_nacimiento = datetime.strptime(fecha_nacimiento, '%Y-%m-%d')
    hoy = datetime.today()
    edad = hoy.year - fecha_nacimiento.year - ((hoy.month, hoy.day) < (fecha_nacimiento.month, fecha_nacimiento.day))
    return edad
```