

Rúbrica de Evaluación – TPI Programación 1

Gestión de Datos de Países en Python: filtros, ordenamientos y estadísticas

Reglas excluyentes (antes de puntuar)

- Sin **video** (o fuera de 10–15 min, o sin ambos integrantes a cámara al inicio): calificación final = **0**.
- Repositorio **no público** o sin CSV base: **0**.
- Proyecto no ejecutable (código con errores que impiden correrlo): **0**.

Rúbrica con puntajes por nivel

I. Marco Teórico e Investigación (30%)

Criterio	Excelente (100%)	Bueno (75%)	Satisfactorio (50%)	Insuficiente (25%)	No presentado (0%)
1. Conceptos Fundamentales (15)	Define con precisión y profundidad todos los conceptos teóricos (listas, diccionarios, funciones, condicionales, ordenamientos, estadísticas, archivos CSV), relacionándolos claramente con el proyecto.	Define la mayoría de los conceptos, con alguna omisión menor o falta de profundidad.	Define los conceptos de forma básica y general, con conexiones limitadas al proyecto.	Definición incorrecta, vaga o sin relación con el proyecto.	No presenta.
2. Fuentes Bibliográficas (10)	Presenta ≥ 3 fuentes relevantes, académicas o técnicas, citadas correctamente.	Presenta 2–3 fuentes, con pequeños errores de citación.	Presenta 1–2 fuentes, no del todo pertinentes o mal citadas.	No presenta fuentes o las presentadas son irrelevantes.	No presenta.



3. Conclusiones (5)	Conclusiones claras y profundas que muestran aprendizajes significativos sobre estructuras, modularidad y estadísticas.	Conclusiones adecuadas pero con poca profundidad en algunos aspectos.	Conclusiones básicas o muy generales.	Conclusiones incorrectas o mínimas, sin aprendizaje relevante.	No presenta.
---------------------	---	---	---------------------------------------	--	--------------

II. Desarrollo del Software (60%)

Criterio	Excelente (100%)	Bueno (75%)	Satisfactorio (50%)	Insuficiente (25%)	No presentado (0%)
1. Funcionalidad (20)	Todas las funcionalidades (carga CSV, búsqueda, filtrado, ordenamiento y estadísticas) operan correctamente, robustas ante distintos casos.	La mayoría funciona bien, con pequeños errores o casos límite no contemplados.	Solo operan funciones básicas, con errores frecuentes.	Funcionalidad muy limitada o incompleta.	No presenta.
2. Estructuras de Datos (10)	Uso óptimo de listas y diccionarios, coherente con el modelo de países.	Uso correcto pero con margen de optimización.	Uso básico o con algunas inconsistencias.	Uso incorrecto o inadecuado.	No presenta.
3. Modularidad y Funciones (10)	Código perfectamente modularizado: cada función cumple una tarea específica y clara.	Modularización adecuada pero con algunas funciones sobrecargadas.	Modularización parcial, funciones poco cohesionadas.	Código sin modularización real.	No presenta.
4. Manejo de Archivos (CSV) (5)	Lectura robusta, conversión correcta de tipos y manejo de excepciones.	Lectura funcional pero con validaciones limitadas.	Errores frecuentes o falta de manejo de excepciones.	No implementado o falla.	No presenta.



5. Lógica Condicional y Ordenamiento (5)	Filtros precisos y ordenamientos eficientes en múltiples criterios.	Lógica mayormente correcta, con pequeños errores.	Lógica incompleta o con fallas en ordenamientos.	Lógica incorrecta o ausente.	No presenta.
6. Legibilidad y Comentarios (5)	Código claro, bien comentado, con nombres significativos.	Legible con comentarios adecuados, aunque mejorable.	Legibilidad limitada o pocos comentarios.	Código confuso, sin comentarios.	No presenta.
7. Manejo de Errores (5)	Manejo completo de errores con mensajes claros para el usuario.	Manejo parcial de errores, pero funcional.	Manejo limitado o poco claro.	Ausencia de manejo de errores.	No presenta.

III. Entregables y Presentación (10%)

Criterio	Excelente (100%)	Bueno (75%)	Satisfactorio (50%)	Insuficiente (25%)	No presentado (0%)
1. Carpeta Digital (3)	Completa: marco teórico, código, capturas y conclusiones.	Contiene casi todos los elementos, con mínimos faltantes.	Faltan varios elementos importantes.	Carpeta incompleta o mal organizada.	No presenta.
2. Repositorio GitHub (4)	Público, completo, con README detallado (instrucciones, ejemplos, integrantes).	Proyecto y README funcionales, aunque mejorables.	Repositorio básico o README escueto.	Repositorio deficiente o inaccesible.	No presenta.
3. Video Tutorial (3)	Entre 10–15 min, ambos integrantes a cámara, explica teoría y muestra programa funcionando.	Cumple parcialmente con los requisitos, explicación aceptable.	Explicación superficial o tiempo inadecuado.	Video deficiente o incompleto.	No presenta.

Escala de puntuación

- **90–100% (Sobresaliente):** Proyecto supera expectativas, implementación robusta y bien documentada.
- **70–89% (Notable):** Cumple la mayoría de los criterios con buena calidad.
- **50–69% (Aprobado):** Cumple los mínimos requeridos, aunque con deficiencias claras.
- **0–49% (Reprobado):** No alcanza los mínimos, implementación incompleta o incorrecta.