

ICI 116 Fundamentos de Programación

Proyecto

Fecha de entrega: **11 de julio, a las 12.00 hrs.**

Cantidad de integrantes: **2 personas máximo**

Resultados de aprendizaje a evaluar:

- **CE1.N1.RA1.** Identifica estructuras abstractas de programación, con el fin de ser utilizadas en la solución de problemas y desarrollo de sistemas.
- **CE1.N1.RA4.** Usa lenguajes de programación para resolver problemas algorítmicos de forma eficiente.
- **CE1.N1.RA5.** Identifica, analiza e implementa posibles soluciones con el objetivo de resolver problemas algorítmicos básicos.

Enunciado

En la actualidad uno de los desafíos es la gestión de la gran cantidad de información a la que día a día se ve expuesta la población mundial. Sin embargo, esta problemática ya había sido expuesta en los años 70, con el término *information overload*, en el libro Future Shock, que predijo el aumento de influencia que tendrían los computadores en la vida diaria y le puso nombre al sentimiento de estar saturado con datos y conocimiento. Incluso, algunos expertos, advierten que vivimos un tsunami informativo.

A pesar de la creencia de que esto presume un problema, no es así. Existen diversas iniciativas que apuntan a utilizar estos grandes volúmenes de datos, en información relevante para la gestión y toma de decisiones en organizaciones. Un claro ejemplo de ello es la iniciativa de Datos.gob (<https://datos.gob.cl>). Dicha entidad se reconoce como un portal donde es posible encontrar datos y estadísticas del sector público, para distintos fines, tales como aplicaciones, visualizaciones, investigación, etc.

Para apoyar el proceso de manipulación de grandes volúmenes de datos, se le ha solicitado generar una implementación en el lenguaje Python, que contenga las siguientes características:

1. Utilizar **una** fuente de datos externa ([Datos.gob](#), [Ministerio de Ciencias](#), [kaggle](#), [Awesomedata](#), [Google](#), o similar).
2. Poseer un menú que permita al usuario seleccionar entre al menos 4 opciones (algunas restricciones).
 - a. Presentar los datos por rango (ej: de fecha, edad, etc.), ingresado por el usuario.
 - b. Desplegar los datos, por alguna característica del dataset (edad, región, provincia, distrito, comuna, etc.)
 - c. Desplegar una o más visualizaciones (revisar documentación de visualización).
 - d. Exportar un subconjunto de datos en formato(s) seleccionado(s) por el usuario (xls, csv, txt, etc.)
3. Desarrollar implementación en al menos **3 funciones**, de las cuales, al menos una de ella debe recibir argumentos y/o retornar un resultado.
4. Ejecutarse mientras el usuario así lo estime.

Apoyo:

Ejemplo librería para graficar.

Matplotlib.pyplot

Gráfico de líneas.

```
import matplotlib.pyplot as plt
#grafico de lineas
ejex=[4,8,13,17,20]
ejey=[54, 67, 98, 78, 45]
plt.plot(ejex,ejey)
plt.show()
```

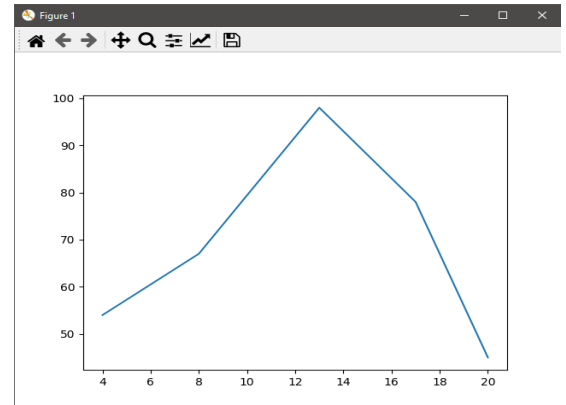
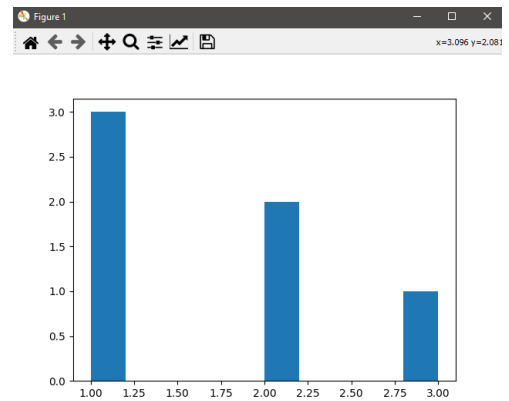


Gráfico de barras.

```
import matplotlib.pyplot as plt
#grafico de barras
x=[1,1,1,2,2,3]
plt.hist(x)
plt.show()
```



La elección de la librería para graficar es libre, puede basarse en cualquiera que se acomode los datos que posee. [Aquí puede encontrar algunas referencias.](#)

Formato de entrega:

Un integrante de cada equipo deberá subir al aula virtual, los siguientes elementos, a más tardar el día **11 de julio, a las 12:00 hrs.** Además del aplicativo (código Python), deberá realizar un informe/documento que incluya al menos:

1. Presentación del proyecto a abordar (mínimo 200 palabras - máx 300).
2. Esquema general de la solución en un diagrama de alto nivel.
3. Lista de funcionalidades a incorporar (descripción de qué realiza cada funcionalidad).

A continuación, se presentan las rúbricas de evaluación:

Rúbrica informe			
Ítem	Cumple (100%)	Suficiente (60%)	Insuficiente (24%)
Resumen Proyecto (30%)	30% El resumen incluye contexto, problema y propuesta de solución.	18% El resumen no incluye contexto, problema o propuesta de solución. (1 ausencia).	7% 2 o más ausencias.
Esquema General de la Solución (40%)	40% El esquema permite identificar fuentes de datos externas, funcionalidades, y salidas.	24% Presenta problemas en el manejo de estructuras repetitivas (1 a 2 errores).	10% Más de 2 casos presentan problemas.
Lista de Funciones (30%)	30% Presenta 3 funciones, de las cuales, al menos una de ella recibe argumentos y/o retorna un resultado.	18% Presenta menos de 3 funciones, o no identifica qué argumentos recibe o qué retornan.	7% Presenta menos de 3 funciones, y no identifica qué argumentos recibe o qué retornan.

Rúbrica Aplicativo			
Ítem	Cumple (100%)	Suficiente (60%)	Insuficiente (15%)
Manejo de estructuras selectivas (10%)	10% Identifica y aplica correctamente estructuras selectivas	5% Presenta problemas en la identificación y manejo de estructuras selectivas (1 a 2 errores).	1% Más de 2 casos presentan problemas.
Manejo de estructuras repetitivas (10%)	10% Identifica y aplica correctamente estructuras repetitivas	5% Presenta problemas en el manejo de estructuras repetitivas (1 a 2 errores).	1% Más de 2 casos presentan problemas.
Manejo de listas (20%)	20% Identifica y aplica correctamente listas	10% Presenta problemas en el manejo de listas (1 a 2 errores).	5% Más de 2 casos presentan problemas.
Manejo de funciones (20%)	20% Identifica y aplica correctamente funciones	10% Presenta problemas en el manejo de funciones (1 a 2 errores).	5% Más de 2 casos presentan problemas.
Manejo de archivos (10%)	10% Identifica y aplica correctamente archivos	5% Presenta problemas en el manejo de archivos (1 a 2 errores).	2% Más de 2 casos presentan problemas.
Entrega resultado esperados (25%)	25% Entrega el resultado esperado	10% Los resultados no son los esperados. No obstante entrega similares.	5% Entrega resultados incorrectos.
Comenta código (5%)	5% Agrega comentarios claves para la correcta comprensión del programa	2% Comenta código, no obstante no son determinantes para entender el programa	0% Comenta segmentos de código totalmente irrelevantes

Si no se evidencia ítem, el puntaje es 0%.