

**FACULTAD DE INGENIERIA Y CIENCIAS EXACTAS**  
**Departamento de Tecnología Informática**

## **FUNDAMENTOS DE INFORMÁTICA**

### **Guía de Trabajos Prácticos**

#### **EJERCICIOS ADICIONALES v1.0**

**Objetivos:** Que el alumno pueda resolver problemas de forma simple y práctica.

- Plantear estrategias de resolución de problemas creando algoritmos
- Entender y saber construir un programa estructurado
- Diferenciar los distintos tipos de estructuras de control
- Identificar módulos y definir funciones
- Utilizar estructuras de datos unidimensionales y bidimensionales

#### **Bibliografía sugerida:**

##### **Básica**

- Joyanes Aguilar, Luis: Fundamentos de Programación. Editorial Mc.Graw-Hill. ISBN 978-84-481-6111-8. EAN 9788448161118.

##### **Complementaria**

- Pilgrin, Mark: Inmersión en Python 3 [en línea] Traducido por José Miguel González Aguilera. Disponible gratuitamente (bajo licencia Creative Commons 3.0). [consultas: 17/02/21]  
<http://www.jmgaguilera.com/libro/python/traducci%C3%B3n/latex/2016/08/19/inmersion-python.html>
- Marzal Varó, Andrés - Gracia Luengo, Isabel, García Sevilla, Pedro: Introducción a la programación con Python 3 [En línea] Disponible gratuitamente bajo licencia Creative Commons. [consultas: 17/02/21]  
<https://openlibra.com/es/book/introduccion-a-la-programacion-con-python-3>
- The Python Language Reference. [consultas: 17/02/21]. Español:  
<https://docs.python.org/3/reference/>
- Documentos varios en Python.org.ar [consultas: 17/02/21]. Inglés / Español:  
<https://wiki.python.org.ar/aprendiendopython/>

#### **Autor:**

Equipo docente de cátedra.

## Trabajo Práctico 1: Ejercicios Adicionales Estructuras de Control

### Objetivos:

- Ser capaz de escribir algoritmos identificando los pasos a realizar y las estructura de control a utilizar
- Manipular información en diferentes tipos de datos.

*Para los siguientes ejercicios, identifique las entradas y salidas. Diseñe el algoritmo en pseudocódigo y luego escriba el programa en lenguaje Python.*

- Ejercicio 1:** Ingresar la edad de dos hermanos e indicar quién es el mayor o si son mellizos.
- Ejercicio 2:** Ingresar la edad de una persona e indicar si es mayor de edad (mayor o igual a 18 años). Utilizar una variable de tipo booleana para guardar el valor Verdadero si la persona tiene 18 años o más y Falso en caso contrario.
- Ejercicio 3:** Ingresar los nombres de dos visitantes al Parque de Diversiones y sus respectivas edades y alturas. Decidir quién/es pueden acceder a una montaña rusa cuyo uso requiere tener una edad mínima de 15 años y una altura superior a 1,5m.
- Ejercicio 4:** Un albañil necesita contar con un sistema que le permita ingresar las medidas de un ambiente rectangular, tamaño de una puerta y una ventana, y el tamaño de los dos lados de una cerámica y le informe:
- a. Cantidad de cerámicas requeridas para colocar en las cuatro paredes del ambiente.
  - b. Ingresando el costo por paquete y la cantidad de cerámicas por paquete, mostrar el costo total en cerámicas.
- Ejercicio 5:** **Hora de jugar:** Se requiere desarrollar un juego de preguntas, en que se puede responder "si" o "no". Gana quien responde correctamente 3 preguntas. Si se responde mal alguna, no se formula la siguiente y termina el juego (las preguntas son libres, las piensa Ud.)
- Ejercicio 6:** Leer tres números correspondientes al día, mes y año de una fecha e imprimir un mensaje indicando si la fecha es válida o no.
- Ejercicio 7:** La fecha de Pascua para un año cualquiera X puede determinarse de la siguiente forma:
- Calcular el resto de dividir X sobre 19 y llamarlo A.
  - Calcular el resto de dividir X sobre 4 y llamarlo B.
  - Calcular el resto de dividir X sobre 7 y llamarlo C.
  - Multiplicar A por 19, sumarle 24 y calcular el resto de dividir este resultado por 30. Este resto recibirá el nombre D.
  - Multiplicar B por 2, C por 4 y D por 6. Sumar los tres resultados y sumarle 5. Calcular el resto de dividir este último resultado por 7 y llamarlo E.
  - La fecha de Pascua se obtiene sumando los valores D y E, más la constante 22.
  - El resultado se expresa como una fecha dentro del mes de Marzo. Si el número es mayor que 31, entonces ese año Pascua se celebrará en el mes de Abril. Ejemplo: Un resultado 35 significa que Pascua cae el 4 de Abril (35 menos los 31 días del mes de Marzo es igual a 4).
- Preparar un programa que permita ingresar el año y calcule la fecha de Pascua.
- Ejercicio 8:** Se desea ingresar la nota de N alumnos y mostrar su promedio
- Ejercicio 9:** Un videojuego lleva registro del mejor puntaje indicando nombre y cantidad de puntos obtenidos. Escribir un algoritmo que permita ingresar el nombre de cada jugador y su puntaje obtenido. Para indicar que no hay más jugadores se ingresa la palabra "fin" como nombre. ¿Cuántos valores existen entre el menor y el mayor?

- 
- Ejercicio 10:** Una fábrica produce bicicletas de paseo y de montaña. Para construir una bicicleta de paseo se necesita 1.5 kg de acero, y para construir una bicicleta de montaña se necesita 2kg de acero. La fábrica vende las bicicletas de paseo a \$16000 y las de montaña a \$18000. Se pide: ingresar las cantidades de bicicletas a producir (de paseo y montaña). Suponga que la fábrica dispone de 80kg de acero para su producción. Informar si la cantidad de acero disponible en la fábrica es suficiente para las cantidades de bicicletas ingresadas. Si no es suficiente, indicar la cantidad faltante de acero necesaria para producirlas. Verificar que las cantidades ingresadas sean valores mayores o iguales a 0 (cero).
- Ejercicio 11:** Escribir un algoritmo que calcule la raíz cuadrada del número que introduzca el usuario. Si se introduce un número negativo, debe mostrar un mensaje de error y volver a pedirlo (tantas veces como sea necesario).
- Ejercicio 12:** Escribir un algoritmo que solicite un nombre de usuario y una contraseña, y que no permita seguir hasta que se introduzcan correctamente. Deberá repetirse hasta que el usuario sea "admin" y la contraseña sea "1234", y en ese caso mostrar el mensaje "Login exitoso". Tras 3 intentos sin acertar, se deberá mostrar un mensaje indicando que se agotaron esos 3 intentos y finaliza el programa.
- Ejercicio 13:** Diseñar una función que reciba un nombre como parámetro y lo salude por pantalla. Por ejemplo, si recibe el dato "Juan", debe mostrar: "Hola Juan".
- Ejercicio 14:** Diseñar una función que devuelva el máximo entre dos números recibidos como parámetros.
- Ejercicio 15:** Diseñar una función para verificar si un número es par o impar, devolviendo True o False.}
- Ejercicio 16:** Ingresar números hasta ingresar -1. Informar cuántos números pares se ingresaron y su suma.

## Trabajo Práctico 2: Ejercicios Adicionales Estructuras de datos unidimensionales y bidimensionales

### Objetivos:

- Ser capaz de escribir algoritmos identificando los pasos a realizar y las estructuras de control a utilizar
- Manipular información en diferentes tipos de datos y estructuras de datos.

- Ejercicio 1:** Desarrollar un algoritmo que permita mostrar una lista de enteros con sus valores a la mitad.  
Ejemplo: Lista [1,2,3,4], mostrar: 0.5 1.0 1.5 2.0.
- Ejercicio 2:** Llenar una lista con números enteros (números positivos o negativos). Mostrar la cantidad de números positivos y la cantidad de números negativos que hay en dicha lista.
- Ejercicio 3:** Cargar una lista con números aleatorios y reemplazar un valor en una posición determinada por el usuario.
- Ejercicio 4:** Ingresar 20 números e indicar cuales son los 5 mayores.
- Ejercicio 5:** Dadas los siguientes arreglos apareados nombres y edades, se pide ordenar por edad de menor a mayor (recuerde que, entre arreglos relacionados, las posiciones indican la relación que hay entre los datos de estos, así en el ejemplo, el nombre "Juan" se corresponde con la edad 28, "Pablo" con la edad 22, etc.):  
nombres = ['Juan', 'Pablo', 'Damián', 'Manuel', 'Nahuel', 'Lucas']  
edades = [ 28, 22, 12, 18, 25, 43 ]
- Ejercicio 6:** Eliminar de una lista de números enteros los valores que se encuentren en una segunda lista. Mostrar la lista original, la lista de valores a eliminar y como queda la lista original luego de eliminar los valores.
- Ejercicio 7:** **Hora de jugar:** Se modela un juego de tateti con 3 listas, donde cada uno representa una fila del tablero del tatetí.  
list1 = ['-','-', '-'] esta lista representa las posiciones 1, 2, 3  
list2 = ['-','-', '-'] esta lista representa las posiciones 4, 5, 6  
list3 = ['-','-', '-'] esta lista representa las posiciones 7, 8, 9  
El programa debe mostrar alternadamente: 'Juega X' y 'Juega O'.  
En cada jugada, el jugador respectivo deberá ingresar la posición donde quiere colocar su ficha, hasta que un jugador gane o se acaben las posiciones disponibles y se decrete "empate".  
Extra: Si un jugador quiere ocupar una posición ya ocupada, se le debe indicar y solicitarle que ingrese una posición válida. (Lo pasaría a matrices)
- Ejercicio 8:** A partir de la lista SECUENCIAS generada en la práctica de clase, mostrar la secuencia más larga almacenada en la misma. Si hubiera varias secuencias con la misma longitud máxima deberán mostrarse todas las que correspondan.
- Ejercicio 9:** Crear una una matriz cuadrada, obtener la matriz de cofactores.
- Ejercicio 10:** Dada una matriz de  $n*m$ , obtener la suma de los valores de cada fila e ir guardándolos en una lista. Recorrer la lista e informar en que posición se encuentra el mayor valor.
- Ejercicio 11:** Desarrollar una función para recibir una matriz  $A(n*m)$ , transponerla sobre la matriz  $B(m*n)$
- Ejercicio 12:** Crear una matriz de  $M \times N$  con valores al azar positivos entre a y b sin repetir, a continuación un valor de referencia *hasta -1* y recorriendo de forma eficiente la matriz informar si coincide con alguno de los valores de la matriz, indicar fila y columna. Al finalizar informar cuántos valores se encontraron.

**Ejercicio 13:** Desarrollar un programa para crear una lista con números al azar entre 1 y 20. Finalizar la carga al crear el número cero y contemplar además que la lista debe tener al menos 10 elementos cargados. Mostrar la lista por pantalla. Luego se pide: Informar cuántos números NO están repetidos. Desarrollar una función que reciba la lista y retorna cuántos números únicos hay en la lista.

**Ejercicio 14:** Una fábrica de bicicletas desea realizar un control de inventario de sus productos, la empresa fabrica N productos diferentes y quiere mantener en stock un mínimo de 5 unidades por producto. Necesita un programa que ingrese código del producto positivo y simule la cantidad en stock actual (valor entre 1 y 50). Si se vuelve a ingresar un código de producto, se debe rechazar. La carga de datos finaliza con un código de producto igual a -1.

Luego de la carga, informar:

- ✓ Código del producto y cantidad en stock actual. Ordenado por Código de producto de menor a mayor. Con el siguiente formato:

<b>Código</b>	<b>Cantidad</b>
123	20
243	10
450	3
8080	15

- ✓ Código del producto y cantidad en stock actual de los productos que están por debajo del stock mínimo. A continuación, indicar cuál es el total general de unidades faltantes para que todos los productos tengan al menos las 5 unidades en stock.
- ✓ Mayor cantidad en stock y todos los códigos de productos que tienen la mayor cantidad en stock.

Realizar la búsqueda de códigos de producto ingresados por teclado e informar su cantidad en stock con una leyenda si está por debajo del stock mínimo. Finalizar el ingreso de códigos para la búsqueda con -1. Informar cuántos productos de los buscados fueron encontrados.

**Ejercicio 15:** Una empresa de colectivos debe renovar su flota de vehículos que tengan más de 20 años de antigüedad. Para eso necesita un programa que le permita llevar el control de dicha renovación. El programa deberá realizar las siguientes tareas:

- Ingresar el año de renovación (validar que el año sea mayor o igual al año actual).
- Ingresar el número de interno de cada uno de los vehículos y a continuación su modelo (año de fabricación). Finalizar la carga ingresando -1 en el número de unidad.
- Al ingresar los datos del vehículo, validar que el modelo sea mayor a 1980 y menor al año de renovación.
- Mostrar por pantalla un listado con todos los datos de los vehículos que deben ser renovados (número de interno, modelo).
- Informar cuántos vehículos posee la empresa y qué porcentaje de éstos debe ser renovado.
- Informar el año de fabricación de la unidad más antigua de la empresa.