

FACULTAD DE INGENIERIA Y CIENCIAS EXACTAS
Departamento de Tecnología Informática

FUNDAMENTOS DE INFORMÁTICA

Guía de Trabajos Prácticos

V10.50.00

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 básicas

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: 9/07/20]
<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: 9/07/20]
<https://openlibra.com/es/book/introduccion-a-la-programacion-con-python-3>
- The Python Language Reference. [consultas: 9/07/20]. Español:
<https://docs.python.org/3/reference/>
- Documentos varios en Python.org.ar [consultas: 9/07/20]. Inglés / Español:
<http://www.python.org.ar/aprendiendo-python/>

Autor:

Ricardo Thompson.

Actualizada Julio 2020:

Verónica Galati, Ramiro Escalante Leiva, Guillermo Maquieira, Manuel Quintana.

Trabajo Práctico 1: Algoritmos elementales

Objetivo:

- Resolver problemas creando algoritmos.
- Definir la estructura básica de un algoritmo: Entrada-Proceso-Salida.
- Conocer Modelo Top-Down.

Ejercicio 1: *Escribir la secuencia de acciones necesarias para lograr realizar el objetivo planteado. Identificar el estado inicial y el estado final para cada situación.*

- Llegar a la vereda desde la posición en la que se encuentra.
- Cruzar una calle en una esquina con semáforos.
- Lavarse las manos.
- Preparar un mate.
- Destapar una botella de cerveza.
- Poner una alarma en el reloj del celular.
- Pintar una pared.
- Comprar una entrada de cine por Internet.
- Resolver la siguiente cuenta: $4 + 2 * 3$
- Separar el cuatro de copas de un conjunto de naipes.
- Buscar la menor carta de un mazo de naipes, que puede no estar completo.

Trabajo Práctico 2: Entrada y Salida de Datos

Objetivos:

- Conocer los distintos tipos de datos y la gestión de los datos a través de variables.
- Identificar los distintos operadores aritméticos y el orden de precedencia y cálculos básicos.

Ejercicio 1: Diseñar el algoritmo para resolver los siguientes problemas y escriba el programa correspondiente en lenguaje Python.

- Mostrar el mensaje "Hola Mundo".
- Ingresar el nombre del usuario del programa y saludarlo. Ejemplo: Si el usuario se llama Juan, se debe mostrar el mensaje "Hola Juan".
- Ingresar dos números y mostrar la suma y la diferencia.
- Ingresar tres números y mostrar la suma y el promedio.
- Ingresar el monto de una factura y calcular el IVA (21%).

Ejercicio 2: Calcular el valor de las siguientes expresiones aritméticas, respetando el orden establecido, indicar el orden que ejecuta el intérprete Python cada una de ellas. Ejemplo: $12*4+4*5 = 68$ orden: multiplicación, suma:

- | | | |
|----|--------------------|---------------|
| a. | $12 * 4 + 4 * 5$ | (68) |
| b. | $12 * (4 + 4) * 5$ | (480) |
| c. | $5 * 4 / 2$ | (10.0) |
| d. | $5 * (4 / 2)$ | (10.0) |
| e. | $24 / 2 ** 2$ | (6.0) |
| f. | $(24 / 2) ** 2$ | (144.0) |
| g. | $- 9 ** 2$ | (-81) |
| h. | $(- 9) ** 2$ | (81) |
| i. | $\sqrt{81}$ | (9) |
| j. | $\sqrt[3]{27}$ | (3) |
| k. | $10 / 3$ | (3.333333...) |
| l. | $10 // 3$ | (3) |
| m. | $12 \% 5$ | (2) |

Ejercicio 3: Indicar cuál es el cálculo que se debe realizar para lograr obtener la información solicitada:

- Promedio con decimales 15, 20, 35 (23.33...)
- Promedio sin decimales 15, 20, 35 (23)
- Porcentaje Notas aprobadas (mayores o iguales a 4): 10, 4, 2, 8, 2, 5, 8, 9 (75%)

Ejercicio 4: Las variables guardan datos de diferente tipo y permiten cambiar su valor. Dadas dos variables numéricas A y B, que el usuario debe ingresar por teclado, se pide:

- Duplicar el valor que poseen y mostrarlo por pantalla.
- Intercambiar los valores de ambas variables y mostrar cuánto valen al final las dos variables.

Trabajo Práctico 3: Estructura secuencial

Objetivos:

- Crear los primeros algoritmos en pseudocódigo con estructura secuencial.
- Ser capaz de escribir algoritmos secuenciales en lenguaje Python.

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

- Ejercicio 1:** Calcular el promedio de las notas que obtiene un alumno al rendir los dos parciales.
- Ejercicio 2:** Tres personas invierten dinero para fundar una empresa (no necesariamente en partes iguales). Calcular qué porcentaje invirtió cada una.
- Ejercicio 3:** Leer una medida en metros e imprimir esta medida expresada en centímetros, pulgadas, pies y yardas. Los factores de conversión son:
- 1 pie = 12 pulgadas
 - 1 yarda = 3 pies
 - 1 pulgada = 2,54 cm.
 - 1 metro = 100 cm.
- Ejercicio 4:** Una inmobiliaria paga a sus vendedores un salario base, más una comisión fija por cada venta realizada, más el 5% del valor de esas ventas. Realizar un programa que imprima el número del vendedor y el salario que le corresponde en un determinado mes. Se leen por teclado el número del vendedor, la cantidad de ventas que realizó y el valor total de las mismas.
- Ejercicio 5:** Una persona quiere invertir su capital en un banco y quiere saber cuánto dinero gana en un mes si el banco paga 2% mensual. ¿Cuánto ganará en seis meses si deja su dinero invertido?
- Ejercicio 6:** Un banco necesita para sus cajeros de la sucursal un programa que lea una cantidad de dinero que desea retirar el cliente e imprima a cuántos billetes equivale, considerando que existen billetes de \$1000, \$500, \$200, \$100, \$50, \$20 y el resto en monedas. Desarrollar dicho programa de tal forma que se minimice la cantidad de billetes entregados por el cajero.
- Ejercicio 7:** Desarrollar un algoritmo que, dados el valor de una hora de trabajo y la cantidad de horas trabajadas por día, calcule y muestre el valor del sueldo semanal, asumiendo que se trabajan todos los días hábiles y media jornada los sábados. Ejemplo: Ingresa 120 y 8, debe devolver 5280.
- Ejercicio 8:** Leer un período en segundos e imprimirlo expresado en días, horas, minutos y segundos. Por ejemplo, 200000 segundos equivalen a 2 días, 7 horas, 33 minutos y 20 segundos.
- Ejercicio 9:** Escribir un programa para convertir un número binario de 4 cifras en un número decimal. Se ingresa como un solo número entero de cuatro dígitos.

Ayuda: Dado un número binario, se toma de a un dígito a la vez, de menor a mayor comenzando por el índice 0 y se multiplica por (2^x) donde x corresponde a la posición del dígito. Estos resultados se suman.

Ejemplo: $1\ 0\ 1\ 1_2 = 1 * 2^0 + 1 * 2^1 + 0 * 2^2 + 1 * 2^3 = 11_{10}$

Índices: 3 2 1 0

Trabajo Práctico 4: Estructura condicional

Objetivos:

- Crear algoritmos en pseudocódigo combinando estructura secuencial y condicional.
- Ser capaz de escribir algoritmos creados en lenguaje Python.
- Comenzar a utilizar variables que guardan valores booleanos (True/False).

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 dos números enteros e indicar si son iguales o distintos.
- Ejercicio 2:** Ingresar la edad de dos hermanos e indicar quién es el mayor o si son mellizos.
- Ejercicio 3:** Leer un número entero e imprimir un mensaje indicando si es par o impar.
- Ejercicio 4:** 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 5:** Ingresar las notas de los dos parciales de un alumno e indicar si promocionó, aprobó o debe recuperar.
- Se promociona cuando las notas de ambos parciales son mayores o iguales a 7.
 - Se aprueba cuando las notas de ambos parciales son mayores o iguales a 4.
 - Se debe recuperar cuando al menos una de las dos notas es menor a 4.
- Ejercicio 6:** 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 7:** Una editorial determina el precio de un libro según la cantidad de páginas que contiene. El costo básico del libro es de \$125, más \$2,20 por página con encuadernación rústica. Si el número de páginas supera las 300 la encuadernación debe ser en tela, lo que incrementa el costo en \$80. Además, si el número de páginas sobrepasa las 600 se hace necesario un procedimiento especial de encuadernación que incrementa el costo en \$136. Desarrollar un programa que calcule el costo de un libro dado el número de páginas.
- Ejercicio 8:** Una remisería requiere un sistema que calcule el precio de un viaje a partir de la cantidad de km que desea recorrer el usuario. Tiene la siguiente tarifa:
- Viaje mínimo \$50
 - Si recorre entre 0 y 10km: \$20/km
 - Si recorre 10km o más: \$15/km
- Ejercicio 9:** 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 10:** **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 11: Leer un número correspondiente a un año e imprimir un mensaje indicando si es bisiesto o no. Se recuerda que un año es bisiesto cuando es divisible por 4. Sin embargo, aquellos años que sean divisibles por 4 y también por 100 no son bisiestos, a menos que también sean divisibles por 400. Por ejemplo, 1900 no fue bisiesto, pero sí el 2000.

Ejercicio 12: 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 13: Crear un programa que pida un número de mes (por ejemplo, 4) y escriba el nombre del mes en letras ("abril"). Verificar que el mes sea un número de mes válido e informar en caso que no lo sea.

Ejercicio 14: 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 15: Diseñar un programa que calcule y muestre el sueldo neto de un empleado en base a su sueldo básico y su antigüedad en años. Si es soltero se le incrementa el sueldo en 5% del salario bruto por cada año de antigüedad, mientras que si es casado se le incrementa el sueldo en 7% del bruto por cada año de antigüedad. También se le realizan los siguientes descuentos: Jubilación: 11%, Obra Social: 3%, Sindicato: 3%. Como datos de entrada se ingresa por teclado el sueldo básico, antigüedad y estado civil ('s' o 'c'). Se debe informar: (reemplazar los 9 por los valores que correspondan)

Estado Civil: Soltero/Casado

| | | | |
|---------------|-------------|-------------|----------|
| Sueldo básico | Antigüedad | Descuentos | Importe |
| \$ 999.99 | 99 años | | + 999.99 |
| | | Jubilación | - 999,99 |
| | | Obra Social | - 999,99 |
| | | Sindicato | - 999,99 |
| | | | ----- |
| | Sueldo Neto | | 999,99 |

Ejercicio 16: 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).

Trabajo Práctico 5: Estructura iterativa

Objetivos:

- Crear algoritmos en pseudocódigo combinando estructura secuencial, condicional e iterativa.
- Ser capaz de escribir algoritmos creados en lenguaje Python.

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

- Ejercicio 1:** Escribir un algoritmo que muestre los primeros 25 números naturales.
- Ejercicio 2:** Calcular e imprimir la suma de los números comprendidos entre 42 y 176.
- Ejercicio 3:** Mostrar las tablas de multiplicar (entre 1 y 10) del número 4. ¿Cómo cambiaría el algoritmo para que el usuario pueda decidir la tabla de multiplicar a mostrar?
- Ejercicio 4:** Se desea ingresar la nota de 5 alumnos y mostrar su promedio.
- Ejercicio 5:** Leer 10 números enteros e imprimir el mayor y en qué orden fue ingresado.
- Ejercicio 6:** Escribir un algoritmo que lea números enteros hasta que se ingrese un 0, y muestre el máximo, el mínimo (sin considerar el 0) y la media (promedio) de todos ellos. Piense cómo debe inicializar las variables.
- Ejercicio 7:** Ingresar números, hasta que la suma de los números pares supere 100. Mostrar Cuántos números en total se ingresaron.
- Ejercicio 8:** Un negocio desea saber el importe total recaudado al fin del día, desea contar con un programa que pueda ingresar el importe de cada venta realizada. Para indicar que finalizó el día se ingresa -1. ¿Cuál fue el monto total vendido y cuántas ventas se realizaron? El importe de cada venta realizada debe ser un valor positivo.
- Ejercicio 9:** Cada cliente que va al banco Express, indica su número de documento y aguarda a ser atendido, cuando finaliza la atención del día se ingresa -1 para indicar que no hay más clientes para ser atendidos. El banco desea saber quién fue el primer cliente atendido y quién fue el último.
- Ejercicio 10:** "El Veloz", una pequeña agencia de automóviles, cuenta con un vendedor que cobra un salario base y una comisión del 1,5% sobre el precio de cada unidad vendida. Ingresar el nombre del vendedor, el salario base y según la cantidad de autos vendida en el mes, ingresar el valor de cada unidad para calcular el salario bruto del mes.
- Ejercicio 11:** La agencia de autos "El Veloz" necesita ampliar su negocio por lo que contrata a tres vendedores. ¿Cómo cambiaría el algoritmo del ejercicio anterior para poder calcular el salario bruto de cada uno de ellos?
- Ejercicio 12:** 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 13:** Se desea analizar cuántos autos circulan con patente que tiene numeración par y cuántos con numeración impar en un día. Le solicitan escribir un algoritmo que permita ingresar la terminación de la patente (entre 0 y 9) hasta ingresar -1 e informar cuántas se ingresaron con numeración par y cuántas con numeración impar.
- Ejercicio 14:** Leer dos números A y B (enteros positivos). Calcular el producto $A * B$ por sumas sucesivas e imprimir el resultado. Ejemplo: $4 * 3 = 4 + 4 + 4$ (4 sumado 3 veces).

- Ejercicio 15:** Leer dos números naturales A y B. Calcular A^B mediante productos sucesivos y mostrar el resultado. Ejemplo: $4^3 = 4 * 4 * 4$ (4 multiplicado 3 veces).
- Ejercicio 16:** 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 17:** 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 18:** Ingresar un número n, validar que sea positivo. Luego:
- Mostrar los primeros n números impares hasta el número ingresado.
Ejemplo:
 - Si se ingresa 5, se debe mostrar 1 3 5
 - Si se ingresa 8, se debe mostrar 1 3 5 7
 - Si se ingresa -5, se debe pedir otro número.
 - Informar la suma de esos números impares.

Trabajo Práctico 6: Ejercicios Integradores

Objetivos:

- Plantear estrategias de resolución de problemas creando algoritmos.
- Entender y saber construir un programa estructurado.
- Diferenciar los distintos tipos de estructuras de control.

En los ejercicios siguientes identifique las entradas y salidas, diseñe el algoritmo utilizando pseudocódigo o diagrama de flujo.

Describa con datos de ejemplos al menos tres conjuntos de datos para las entradas posibles y la salida que se debe obtener en cada uno de los ejemplos.

Desarrolle el programa en lenguaje Python.

Ejercicio 1: Leer números que representan edades de un grupo de personas, finalizando la lectura cuando se ingrese el número 999. Imprimir cuántos son menores de 18 años, cuántos tienen 18 años o más y el promedio de edad de ambos grupos. Descartar valores que no representan una edad válida. (Se considera válido una edad entre 0 y 100).

Ejercicio 2: Escribir un algoritmo que permita ingresar los números de legajo de los alumnos de un curso y su nota de examen final. El fin de la carga se determina ingresando un -1 en el legajo. Informar para cada alumno si aprobó o no el examen considerando que se aprueba con nota mayor o igual a 4. Se debe validar que la nota ingresada sea entre 1 y 10. Terminada la carga de datos, informar:

- Cantidad de alumnos que aprobaron con nota mayor o igual a 4.
- Cantidad de alumnos que desaprobaron el examen. Nota menor a 4.
- Porcentaje de alumnos que están aplazados (tienen 1 en el examen).

Ejercicio 3: Una empresa aplica el siguiente procedimiento en la comercialización de sus productos:

- Aplica el precio base a la primera docena de unidades.
- Aplica un 10% de descuento a todas las unidades entre 13 y 100.
- Aplica un 25% de descuento a todas las unidades por encima de las 100.

Por ejemplo, supongamos que vende 230 unidades de un producto cuyo precio base es 100. El cálculo resultante sería:

$$100 * 12 + 90 * 88 + 75 * 130 = 18870, \text{ y el precio promedio será } 18870 / 230 = 82,04$$

Escribir un algoritmo que lea la cantidad solicitada de un producto y su precio base, y muestre el valor total de la venta y el precio promedio por unidad. El fin de carga se determina ingresando -1 como cantidad solicitada.

Al finalizar informar:

- a) Cantidad de ventas realizadas total.
- b) Cantidad de ventas que se aplicaron un 10% de descuento.
- c) Cantidad de ventas que SOLO se aplicó el precio base, no se le realizaron descuentos.

Ejercicio 4: Una empresa factura a sus clientes el último día de cada mes. Si el cliente paga su factura dentro de los primeros 10 días del mes siguiente, tiene un descuento de \$120 o del 2% de la factura, lo que resulte más conveniente para el cliente. Si paga en los siguientes 10 días del mes deberá pagar el importe original de la factura, mientras que si paga después del día 20 deberá abonar una multa de \$150 o del 10% de su factura, lo que resulte mayor. Escriba un algoritmo que lea el número del cliente y el total de la factura, y emita un informe donde conste el número del cliente y los tres importes que podrá abonar según la fecha de pago.

Ejercicio 5: El factorial de un número entero N mayor que cero se define como el producto de todos los enteros X tales que $0 < X \leq N$. Desarrollar un programa para calcular el factorial de un número dado. Deberán rechazarse las entradas inválidas (menores que 0).

Ejercicio 6: Leer tres números D, M y A correspondientes al día, mes y año de una fecha, y un número entero N que representa una cantidad de días. Realizar un programa que imprima la nueva fecha que resulta de sumar N días a la fecha dada. Tener en cuenta los años bisiestos tal como se detalla en el ejercicio 9 de la práctica 3.

Ejercicio 7: Una empresa cuenta con N empleados, divididos en tres categorías A, B y C. Por cada empleado se lee su legajo, categoría y salario. Se solicita elaborar un informe que contenga:

- Importe total de salarios pagados por la empresa.
- Cantidad de empleados que ganan más de \$20000.
- Cantidad de empleados que ganan menos de \$5000, cuya categoría sea "C".
- Legajo del empleado que más gana.
- Sueldo más bajo.
- Importe total de sueldos por cada categoría.
- Salario promedio.

Ejercicio 8: Calcular las calificaciones de un grupo de alumnos. La nota final de cada alumno se calcula según el siguiente criterio: la parte práctica vale el 10%; la parte de problemas vale el 50% y la parte teórica el 40%. El algoritmo leerá el nombre del alumno, las tres notas, escribirá el resultado y volverá a pedir los datos del siguiente alumno hasta que el nombre sea una cadena vacía. Las notas deben estar entre 0 y 10.

Ejercicio 9: Una reconocida cadena de comida rápida desea obtener información de las compras realizadas por sus clientes. Para ello necesita crear un programa que estará centralizado y que permitirá ingresar la información generada por sus distintos locales en el siguiente orden:

- Zona geográfica del local (1: CABA, 2: GBA: 2)
- Sexo del cliente (1: M, 2: F)
- Numero de menú (1: Menú \$250, 2: Menú \$400, 3: Menú \$500)

Se supone que todos los datos ingresados serán numéricos. Ejemplo: zona= 1 será CABA, sexo= 2 será F, menú= 1 será el de \$250, etc. El programa finaliza cuando se ingresa una zona igual a -1. Se pide: validar que las opciones de sexo y menú estén dentro de los valores permitidos. Considerar que existe una promoción que aplica a todos los locales y en donde las mujeres tienen un 25% de descuento. Informar el promedio de ticket de compra de hombres (en \$), el total de personas que compraron el menú de \$250, la recaudación total de la cadena (en \$). Para el promedio y la recaudación total se deberá tener en cuenta el descuento aplicado.

Ejercicio 10: **Hora de jugar:** Se está eligiendo al youtuber favorito del año. Para ello se realiza una votación de los 3 con más visitas recibidas. Ingresar el nombre y URL de cada uno de los candidatos. La cantidad de votantes no se conoce ya que la inscripción es libre. Se debe ingresar el nombre del youtuber votado hasta ingresar un nombre vacío. Luego indicar el resultado mostrando para cada candidato: nombre, URL, cantidad de votos, porcentaje de votos recibido. Puede haber votos en blanco. ¿Cuál fue el youtuber con mayor cantidad de votos? Hay que considerar que puede haber empate.

Ejercicio 11: Ingresar por teclado la cantidad de términos a generar de la siguiente serie:

1 7 19 43 91 187 379 763 1531 3067 6139

El primer término es el 1 y cada término se genera como el doble del término anterior más 5. Mostrar la serie por pantalla e informar la suma de los términos generados.

Ejercicio 12: Ingresar por teclado la cantidad de términos a generar de la siguiente serie:

$$S = \frac{1}{2^0} \frac{1}{2^1} \frac{1}{2^2} \frac{1}{2^3} \frac{1}{2^4} \dots$$

Mostrar la serie por pantalla e informar el promedio de los términos generados. Verificar que la cantidad de términos a generar sea mayor a 0. Ejemplo: si N= 5, donde N: cantidad de términos a generar, se muestra por pantalla:

Serie: 1 0.5 0.25 0.125 0.0625

El promedio de los términos generados es: 0.3875

Trabajo Práctico 7: Sub-Algoritmos: Funciones

Objetivo:

- Diseñar algoritmos que permitan modularizar e independizar sub-algoritmos reutilizarlos.

Diseñe el sub-algoritmo planteado en cada enunciado y desarrolle la función en lenguaje Python. Debe crear un programa principal para ejecutar la función solicitada.

Ejercicio 1: Diseñar una función que reciba dos parámetros numéricos enteros, calcule y devuelva el resultado de la multiplicación de ambos utilizando sólo sumas.

Ejercicio 2: Diseñar una función que reciba dos números enteros como parámetros enteros A y B, y permita obtener A^B mediante multiplicaciones sucesivas.

Ejercicio 3: Diseñar una función para mostrar un título filas de asteriscos, la longitud de la fila de asteriscos y el texto del título se recibe como parámetro. Ejemplo: titulo("Ejercicio 3", 15) muestra:

```
*****  
Ejercicio 3  
*****
```

Ejercicio 4: 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 5: Diseñar una función que devuelva el máximo entre dos números recibidos como parámetros.

Ejercicio 6: Escribir la función obtener_mes_en_texto, que devuelva una cadena que representa un mes expresado en letras según un número entero entre 1 y 12 recibido como parámetro. Si el parámetro no es válido, devolver una cadena vacía. Ejemplo: Se invoca obtener_mes_en_texto(4) → devuelve "Abril".

Ejercicio 7: Diseñar una función para verificar si un número es par o impar, devolviendo True o False.

Ejercicio 8: Ingresar números hasta ingresar -1. Informar cuántos números pares se ingresaron y su suma. Utilizar la función del ejercicio 7 para resolverlo.

Ejercicio 9: Extraer un dígito de un número. La función recibe como parámetros dos números enteros, uno será del que se extraiga el dígito y el otro indica qué cifra se desea obtener. La cifra de la derecha se considera la número 0. Retornar el valor -1 si no existe el dígito solicitado. Tener en cuenta que el número puede ser positivo o negativo. Ejemplo: extraerdigito(12345, 1) devuelve 4, y extraerdigito(12345, 8) devuelve -1.

Ejercicio 10: Devolver los últimos N dígitos de un número entero. Devolver el número completo si N es demasiado grande. Ejemplo: ultimosdigitos(12345,3) devuelve 345, y ultimosdigitos(12345,8) devuelve 12345.

Ejercicio 11: Diseñar una función que solicite por teclado un número y lo retorne solo si el número ingresado es natural, caso contrario la función deberá seguir solicitando el número por teclado.

Ejercicio 12: Hora de jugar: Una calculadora tiene cuatro operaciones básicas (a saber: sumar, restar, multiplicar, dividir). Desarrolle una función para realizar cada operación, que reciba como parámetros dos números ingresados por el usuario y devuelva el resultado de la operación. Resuelva la división por restas sucesivas (investigar cómo se resuelve). Incluya un menú que permita realizar una operación y posea una opción para "Salir". Luego de cada operación realizada se debe volver a presentar el menú.

Trabajo Práctico 8: Estructuras de datos Listas o Arreglos

Desarrollar las siguientes funciones para crear listas con valores según lo solicitado:

Ejercicio 1: Escribir una función que solicite ingresar una serie de números entre 1 y 20 y guardarlos en una lista. En caso de ingresar un valor fuera de rango el programa mostrará un mensaje de error y solicitará un nuevo número. Para finalizar la carga se deberá ingresar -1. La función no recibe ningún parámetro, y devuelve la lista cargada (o vacía, si el usuario no ingresó nada) como valor de retorno.

Ejercicio 2: Escribir una función para crear una lista con N números al azar en un rango de valores que se recibe por parámetro. La función devuelve la lista cargada (o vacía si el rango indicado no es válido).

En los siguientes ejercicios utilice una de las funciones anteriores para ingresar datos en una lista y luego resuelva:

Ejercicio 3: Calcular la suma de los números de una lista.

Ejercicio 4: 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 5: 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 6: Determinar si una lista es capicúa. La lista puede tener cualquier tipo de dato básico que vimos en clase.

Ejercicio 7: Una escuela necesita conocer cuántos alumnos cumplen años en cada mes del año, con el propósito de ofrecerles un agasajo especial en su día. Desarrollar un programa que lea el número de legajo y fecha de nacimiento (día, mes y año) de cada uno de los alumnos que concurren a dicha escuela. La carga finaliza con un número de legajo igual a -1. Emitir un informe donde aparezca (mes por mes) cuántos alumnos cumplen años a lo largo del año. Mostrar también una leyenda que indique cuál es el mes con mayor cantidad de cumpleaños.

Ejercicio 8: Cargar una lista con números aleatorios y reemplazar un valor en una posición determinada por el usuario.

Ejercicio 9: Desarrollar un algoritmo que permita ingresar 5 números pertenecientes a la lista A y otros 5 números pertenecientes a la lista B. Crear una lista C, donde cada posición es el resultado de la suma del número en la misma posición en la lista A con el número en la misma posición en la lista B. Ejemplo: Se ingresa A = [1, 2, 3, 4, 5] y B = [4, 7, 1, 3, 6] → C = [5, 9, 4, 7, 11]

Ejercicio 10: Escribir una función para devolver la posición que ocupa un valor pasado como parámetro, utilizando **búsqueda secuencial** en una lista desordenada. La función debe devolver -1 si el elemento no se encuentra en la lista.

Ejercicio 11: Crear tres listas y ordenarlas en forma ascendente utilizando un método para cada lista: **métodos de selección, inserción y burbujeo**. ¿Qué cambia para ordenar en forma descendente?

Ejercicio 12: Utilizando **búsqueda binaria** sobre una lista ordenada realizar la búsqueda de valores e informar si se encuentran o no en la lista, finalizar las búsquedas con -1 e informar cuantas búsquedas fueron exitosas y en cuantas no se encontró el valor buscado.

- Ejercicio 13:** Rellenar una lista con números enteros entre 0 y 100 obtenidos al azar e imprimir el valor mínimo y el lugar que ocupa. Tener en cuenta que el mínimo puede estar repetido, en cuyo caso deberán mostrarse todas las posiciones que ocupe. La carga de datos termina cuando se obtenga un 0 como número al azar, el que no deberá cargarse en la lista.
- Ejercicio 14:** Ingresar 20 números e indicar cuales son los 5 mayores.
- Ejercicio 15:** Crear una lista de N números generados al azar entre 0 y 100 pero sin elementos repetidos. Validar que N sea menor o igual a 100.
- Ejercicio 16:** 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 17:** Dada una lista ordenada de números llamada A y un nuevo número N, desarrollar un programa que agregue el elemento N dentro de la lista A, respetando el ordenamiento existente. El programa deberá detectar automáticamente si el ordenamiento es ascendente o descendente antes de realizar la inserción.
- Ejercicio 18:** 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 19:** Leer dos listas de números M y N, ambas ordenadas de menor a mayor. Generar e imprimir una tercera lista que resulte de intercalar los elementos de M y N. La nueva lista también debe quedar ordenada, sin utilizar ningún método de ordenamiento.
- Ejercicio 20:** **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.

Trabajo Práctico 9: Resolver Problemas de forma simple y práctica

Objetivos:

- Plantear estrategias de resolución de problemas creando algoritmos.
- Entender y saber construir un programa estructurado.
- Diferenciar los distintos tipos de estructuras de control.

En los ejercicios siguientes identifique las entradas y salidas, diseñe el algoritmo utilizando pseudocódigo o diagrama de flujo. Se debe resolver identificando subproblemas y definiendo funciones.

Describe con datos de ejemplos al menos tres conjuntos de datos para las entradas posibles y la salida que se debe obtener en cada uno de los ejemplos.

Desarrolle el programa en lenguaje Python.

Ejercicio 1: Leer los números de legajo de los alumnos de un curso y su nota de examen final. Cargar el número de legajo en la lista LEGAJOS y la nota en la lista NOTA_FINAL. El fin de la carga se determina ingresando un -1 como legajo. Se debe validar que la nota ingresada sea entre 1 y 10. Terminada la carga de datos, recorrer las listas e informar:

- Cantidad de alumnos que aprobaron con nota mayor o igual a 4
- Cantidad de alumnos que desaprobaron el examen. Nota menor a 4
- Promedio de nota y los legajos que superan el promedio

Luego se solicita, ordenar las listas según el número de legajo, de manera ascendente.

Ejercicio 2: 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:

| Código | Cantidad |
|--------|----------|
| 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 3: 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 4:** Una Administradora de Consorcios necesita un sistema para poder gestionar el cobro de las expensas de un edificio de departamentos de 20 unidades. En 2 listas almacena la siguiente información: número de unidad y superficie en metros cuadrados. Validar que no se ingresen números de unidades duplicadas. Cada unidad paga \$100 de expensas por metro cuadrado por mes. Se pide:
- Informar el promedio de expensas del mes.
 - Ordenar los listados de mayor a menor según la superficie. Mostrar por pantalla el listado ordenado informando Número de Unidad y Superficie en metros cuadrados.
- Ejercicio 5:** 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.
- Ejercicio 6: Hora de jugar:** Desarrollar un programa que genere un número entero al azar de cuatro cifras y proponerle al usuario que lo descubra, ingresando valores repetidamente hasta hallarlo. En cada intento el programa mostrará mensajes indicando si el número ingresado es mayor o menor que el valor secreto. Permitir que el usuario abandone al ingresar -1. Informar la cantidad de intentos realizada al terminar el juego, haciendo que el usuario ingrese su número de documento si mejoró la mejor marca de intentos obtenida hasta el momento. Luego mostrar la lista ordenada de los 5 mejores puntajes (indicando también a quién pertenecen) y preguntar si se desea jugar otra vez, reiniciando el juego en caso afirmativo.
- Ejercicio 7:** Modificar el programa anterior para que las pistas brindadas por el programa no sean del tipo "es mayor" o "es menor" sino "M dígitos correctos y N dígitos aproximados". Se considera que un dígito es correcto cuando tanto su valor como su posición coinciden con los del número secreto, mientras que un dígito es aproximado cuando coincide el valor pero no su posición. Ejemplos:
- Número secreto: 5739
- Intento 1: 1234 -> 1 correcto
 - Intento 2: 5678 -> 1 correcto y 1 aproximado
 - Intento 3: 9375 -> 4 aproximados
- Ejercicio 8:** Construir una lista llamada SECUENCIAS con N números enteros al azar entre 1 y 20. Esta lista se caracterizará porque sus valores deben encontrarse divididos en secuencias de números separadas por ceros, cuya suma no sea mayor que 20. Para eso se deberá agregar un elemento de valor 0 a fin de separar cada secuencia de la siguiente, cuidando que ninguna secuencia sume más de 20. Agregar un 0 adicional al final de la lista y mostrar la lista obtenida por pantalla.
- Ejemplo:
- Lista original:
- | | | | | | | | | | | |
|---|---|---|---|---|----|---|----|----|---|---|
| 5 | 2 | 9 | 6 | 4 | 15 | 3 | 19 | 12 | 1 | 5 |
|---|---|---|---|---|----|---|----|----|---|---|
- Resultado:
- | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|----|---|---|----|---|----|---|---|---|
| 5 | 2 | 9 | 0 | 6 | 4 | 0 | 15 | 3 | 0 | 19 | 0 | 12 | 1 | 5 | 0 |
|---|---|---|---|---|---|---|----|---|---|----|---|----|---|---|---|
- Ejercicio 9:** A partir de la lista SECUENCIAS generada en el ejercicio anterior, mostrar la secuencia más larga almacenada en la misma. Si hubiera varias secuencias con la misma longitud máxima deberán mostrarse todas las que correspondan.