# Ejercicios propuestos en Python

1. **Variables y Tipos de Datos:**
   1. ***Nivel de dificultad: Fácil***
   * Escribe un programa que declare una variable llamada "nombre" y otra variable llamada "edad", y luego imprima un mensaje que diga "Hola, [nombre], tienes [edad] años."
   1. ***Nivel de dificultad: Moderado***
   * Crea un programa que calcule el área de un triángulo. Pide al usuario que ingrese la base y la altura, y luego muestra el área.
   1. ***Nivel de dificultad: Moderado***
   * Escribe un programa que convierta grados Celsius a grados Fahrenheit. Pide al usuario que ingrese la temperatura en grados Celsius y muestra la temperatura equivalente en grados Fahrenheit.
2. **Operadores Aritméticos, Asignación y Lógicos:**
   1. ***Nivel de dificultad: Fácil***
   * Crea un programa que tome un número entero como entrada y determine si es par o impar. Imprime un mensaje apropiado.
   1. ***Nivel de dificultad: Moderado***
   * Escribe un programa que pida al usuario un número y luego determine si es divisible por 3 y por 5 al mismo tiempo. Muestra un mensaje que indique si es divisible o no.
3. **Expresiones y Declaraciones:**
   1. ***Nivel de dificultad: Fácil***
   * Crea un programa que calcule el promedio de tres números ingresados por el usuario y muestre el resultado.
   1. ***Nivel de dificultad: Moderado***
   * Desarrolla un programa que calcule el factorial de un número ingresado por el usuario. El factorial de un número entero positivo "n" se calcula multiplicando todos los enteros positivos desde 1 hasta "n".
4. **Comentarios y Buenas Prácticas de Codificación:**
   1. ***Nivel de dificultad: Fácil***
   * Escribe un programa que imprima una serie de números del 1 al 10 en orden ascendente y luego en orden descendente. Añade comentarios que expliquen el propósito de cada parte del código.
   1. ***Nivel de dificultad: Moderado***
   * Crea un programa que determine si un número ingresado por el usuario es primo. Agrega comentarios que expliquen la lógica detrás de la verificación de números primos.
5. **Salida (print) y Entrada (input) de Datos:**
   1. ***Nivel de dificultad: Fácil***
   * Haz un programa que solicite al usuario su nombre y su edad, y luego imprima un mensaje personalizado que incluya esta información.
6. **Condicionales (if, else, elif):**
   1. ***Nivel de dificultad: Fácil***
   * Escribe un programa que pida al usuario ingresar un número e indique si es positivo, negativo o cero.
   1. ***Nivel de dificultad: Moderado***
   * Desarrolla un programa que solicite al usuario ingresar tres números y luego determine cuál de los tres es el mayor.
7. **Bucles (while, for):**
   1. ***Nivel de dificultad: Fácil***
   * Crea un programa que utilice un bucle **for** para imprimir los números del 1 al 10.
   1. ***Nivel de dificultad: Moderado***
   * Escribe un programa que use un bucle **while** para calcular la suma de todos los números enteros desde 1 hasta un número ingresado por el usuario.
8. **Uso de break y continue en Bucles:**
   1. ***Nivel de dificultad: Moderado***
   * Implementa un programa que use un bucle **for** para imprimir todos los números del 1 al 20, pero omite los números que son múltiplos de 4. Utiliza la instrucción **continue** para lograr esto.
9. **Uso de la función range en el bucle for:**
   1. ***Nivel de dificultad: Fácil***
   * Crea un programa que use un bucle **for** y la función **range** para imprimir todos los números pares del 2 al 20.
   1. ***Nivel de dificultad: Moderado***
   * Desarrolla un programa que use un bucle **for** y la función **range** para calcular la suma de todos los números impares entre 1 y un número ingresado por el usuario.
10. **Listas y su Manipulación:**
    1. ***Nivel de dificultad: Fácil***
    * Crea un programa que defina una lista de números enteros y luego imprima la suma de todos los elementos en la lista.
    1. ***Nivel de dificultad: Moderado***
    * Escribe un programa que tome una lista de palabras como entrada y muestre la palabra más larga de la lista.
11. **Tuplas y sus Características:**
    1. ***Nivel de dificultad: Fácil***
    * Desarrolla un programa que declare una tupla con nombres de frutas y permite al usuario buscar si una fruta específica se encuentra en la tupla.
    1. ***Nivel de dificultad: Moderado***
    * Crea un programa que tome dos tuplas, una con nombres de estudiantes y otra con sus calificaciones, y muestre los nombres de los estudiantes que obtuvieron calificaciones por encima de un valor especificado.
12. **Diccionarios:**
    1. ***Nivel de dificultad: Fácil***
    * Escribe un programa que defina un diccionario con nombres de ciudades como clave y sus respectivos países como valor. Luego, permite al usuario ingresar el nombre de una ciudad y muestra el país al que pertenece.
13. **Definición y Llamada de Funciones:**
    1. ***Nivel de dificultad: Fácil***
    * Crea una función llamada "saludo" que tome un nombre como argumento y muestre un mensaje de saludo personalizado, por ejemplo, "Hola, [nombre]".
    1. ***Nivel de dificultad: Moderado***
    * Desarrolla una función llamada "es\_palindromo" que tome una cadena como argumento y devuelva True si la cadena es un palíndromo (se lee igual de izquierda a derecha y de derecha a izquierda) y False en caso contrario.
14. **Parámetros y Argumentos:**
    1. ***Nivel de dificultad: Fácil***
    * Escribe una función llamada "calcular\_promedio" que tome una lista de números como argumento y calcule el promedio de esos números.
    1. ***Nivel de dificultad: Moderado***
    * Crea una función llamada "encontrar\_mayor" que tome una lista de números como argumento y devuelva el número más grande de la lista.
15. **Retorno de Valores y Ámbito de Variables:**
    1. ***Nivel de dificultad: Moderado***
    * Implementa una función llamada "factorial" que tome un número entero como argumento y devuelva su factorial. Asegúrate de que la función maneje el caso especial del factorial de 0 y 1.
16. **Trabajar con Archivos de Texto:**
    1. ***Nivel de dificultad: Fácil***
    * Crea un programa que abra un archivo de texto llamado "mi\_archivo.txt", escriba algunas líneas de texto en él y luego cierre el archivo.
    1. ***Nivel de dificultad: Moderado***
    * Escribe un programa que abra un archivo de texto llamado "mi\_archivo.txt", lea su contenido y muestre el número de palabras en el archivo.

**17. Leer y Escribir Archivos CSV:**

* 1. ***Nivel de dificultad: Moderado***
  + Desarrolla un programa que lea un archivo CSV que contenga información de empleados (nombre, salario) y calcule el salario promedio de todos los empleados.

**18. Leer y Escribir Archivos JSON:**

* 1. ***Nivel de dificultad: Moderado***
  + Implementa un programa que lea un archivo JSON que contenga una lista de personas con información personal (nombre, edad, dirección, etc.) y muestre la información de todas las personas en un formato legible.
  1. ***Nivel de dificultad: Moderado***
  + Escribe un programa que tome información sobre libros (título, autor, año de publicación, etc.) y la guarde en un archivo JSON llamado "biblioteca.json". Luego, crea una función que lea ese archivo y muestre la lista de libros disponibles en la biblioteca.

**19. Clases y Objetos:**

* 1. ***Nivel de dificultad: Fácil***
  + Crea una clase llamada "Persona" con atributos como nombre, edad y género. Luego, crea un objeto de esta clase y muestra la información de la persona.
  1. ***Nivel de dificultad: Moderado***
  + Desarrolla una clase llamada "Rectángulo" con atributos como longitud y ancho. Implementa métodos para calcular el área y el perímetro del rectángulo.

**20. Atributos y Métodos de Clase:**

* 1. ***Nivel de dificultad: Moderado***
  + Escribe una clase llamada "CuentaBancaria" que tenga un atributo "saldo" y métodos para depositar y retirar dinero de la cuenta. Asegúrate de que el saldo no pueda ser negativo.

**21. Encapsulación y Herencia Básica:**

* 1. ***Nivel de dificultad: Moderado***
  + Implementa una clase base llamada "Vehiculo" con atributos como marca y modelo, y un método para imprimir la información del vehículo. Luego, crea una clase derivada llamada "Coche" que herede de "Vehiculo" y tenga atributos adicionales como año de fabricación y tipo de motor.
  1. ***Nivel de dificultad: Moderado***
  + Crea una clase base llamada "Animal" con atributos como nombre y edad, y un método para emitir un sonido. Luego, define clases derivadas como "Perro" y "Gato" que hereden de "Animal" y tengan métodos para sonidos característicos de cada uno.

**22. Importación y Uso de Módulos:**

* 1. ***Nivel de dificultad: Fácil***
  + Escribe un programa principal que importe el módulo **math** y utilice sus funciones para calcular la raíz cuadrada de un número y el valor de Pi.
  1. ***Nivel de dificultad: Moderado***
  + Crea un módulo llamado "calculadora" que contenga funciones para realizar operaciones matemáticas básicas como suma, resta, multiplicación y división. Luego, importa este módulo en un programa principal y realiza cálculos con él.

**23. Creación de Paquetes Personalizados:**

* 1. **Nivel de dificultad: Moderado**
  + Diseña un paquete personalizado llamado "geometria" que contenga módulos para calcular el área y el perímetro de figuras geométricas como círculos, cuadrados y triángulos. Luego, crea un programa principal que importe y utilice estos módulos para realizar cálculos.
  1. **Nivel de dificultad: Moderado**
  + Crea un paquete personalizado llamado "miapp" que contenga módulos relacionados con una aplicación de notas. Los módulos pueden incluir funciones para agregar, editar, eliminar y mostrar notas. Luego, crea un programa principal que utilice estos módulos para administrar notas.

**24. Entender las Excepciones Comunes en Python:**

* 1. ***Nivel de dificultad: Fácil***
  + Escribe un programa que intente dividir un número por cero y maneje la excepción de división por cero para que no cause un error.
  1. ***Nivel de dificultad: Moderado***
  + Desarrolla un programa que solicite al usuario ingresar un número y maneje la excepción si el usuario ingresa una cadena en lugar de un número.

**25. Bloques try, except y finally:**

* 1. ***Nivel de dificultad: Moderado***
  + Crea un programa que intente abrir un archivo que no existe y maneje la excepción de archivo no encontrado utilizando un bloque **try** y **except**. Luego, asegúrate de que el archivo se cierre correctamente utilizando un bloque **finally**.
  1. ***Nivel de dificultad: Moderado***
  + Escribe un programa que lea un número entero desde el usuario y maneje las excepciones que puedan ocurrir si el usuario ingresa un valor no válido, como una cadena. Utiliza un bloque **try** y **except** para garantizar que el programa no falle.

**26. Sintaxis Básica de las Expresiones Regulares:**

* 1. ***Nivel de dificultad: Fácil***
  + Escribe un programa que utilice una expresión regular para encontrar todas las direcciones de correo electrónico en un texto proporcionado por el usuario.
  1. ***Nivel de dificultad: Moderado***
  + Crea un programa que utilice una expresión regular para validar si una cadena dada es un número de teléfono en un formato específico, como "123-456-7890".

**27. Coincidencias y Búsquedas Avanzadas:**

* 1. ***Nivel de dificultad: Moderado***
  + Desarrolla un programa que utilice una expresión regular para buscar y contar todas las palabras de tres letras en un texto proporcionado por el usuario.

**28. Sustituciones y Grupos de Captura:**

* 1. ***Nivel de dificultad: Moderado***
  + Implementa un programa que utilice una expresión regular para buscar y reemplazar todas las fechas en un texto con un formato específico, como "DD/MM/AAAA", y cambie el formato a "AAAA-MM-DD".
  1. ***Nivel de dificultad: Moderado***
  + Escribe un programa que utilice grupos de captura en expresiones regulares para extraer nombres y direcciones de correo electrónico de una lista de contactos en un formato específico, y luego imprime la información extraída en un formato legible.

# Proyectos finales con Python

## Proyecto 1: Calculadora de Promedio

* **Descripción:** Este proyecto consiste en crear una calculadora que permita a un usuario ingresar una lista de números y calcule el promedio de esos números. La aplicación debe utilizar funciones y control de excepciones para manejar situaciones inesperadas, como la entrada de datos no numéricos.
* **Temas a Usar:**
  + Funciones
  + Control de Excepciones
  + Listas
* **Recomendaciones:**
  + Define una función para calcular el promedio de una lista de números.
  + Utiliza un bucle para permitir que el usuario ingrese múltiples números hasta que decida detenerse.
  + Maneja posibles errores al convertir la entrada del usuario a números, como ValueError.
  + Proporciona una interfaz de usuario amigable que permita al usuario ingresar los números y ver el resultado.

## Proyecto 2: Gestión de Tareas

* **Descripción:** En este proyecto, crearás una aplicación de línea de comandos que permite a un usuario agregar tareas, listar tareas y eliminar tareas. La información de las tareas se almacenará en un archivo de texto.
* **Temas a Usar:**
  + Archivos de Texto
  + Funciones
  + Listas
* **Recomendaciones:**
  + Define funciones para agregar, listar y eliminar tareas.
  + Utiliza un archivo de texto para almacenar la información de las tareas.
  + Implementa una interfaz de línea de comandos que permita al usuario interactuar con la aplicación.
  + Utiliza control de excepciones para manejar situaciones como la eliminación de tareas que no existen.

## Proyecto 3: Buscador de Palabras Clave

* **Descripción:** En este proyecto, crearás un programa que permite al usuario buscar palabras clave en un archivo de texto. Utilizarás expresiones regulares para realizar la búsqueda y resaltar las coincidencias encontradas en el texto.
* **Temas a Usar:**
  + Expresiones Regulares
  + Archivos de Texto
  + Funciones
* **Recomendaciones:**
  + Define una función que utilice expresiones regulares para buscar y resaltar palabras clave en un archivo de texto.
  + Permite al usuario ingresar las palabras clave a buscar y el nombre del archivo.
  + Muestra las coincidencias encontradas en el texto con un formato especial, como texto resaltado.

## Proyecto 4: Calculadora de Gastos

* **Descripción:** En este proyecto, crearás una aplicación que permita a los usuarios ingresar sus gastos diarios y calcular el gasto total en un mes. Utilizarás archivos CSV para almacenar y cargar datos.
* **Temas a Usar:**
  + Archivos CSV
  + Funciones
  + Listas
* **Recomendaciones:**
  + Define funciones para agregar gastos, calcular el gasto total y mostrar un resumen mensual.
  + Utiliza un archivo CSV para almacenar los gastos diarios, con detalles como la fecha y el monto.
  + Permite al usuario ingresar gastos en diferentes categorías (alimentación, transporte, entretenimiento, etc.).
  + Muestra un resumen de gastos mensuales por categoría.

## Proyecto 5: Gestor de Contactos

* **Descripción:** En este proyecto, desarrollarás un programa que permita a un usuario agregar, editar y eliminar contactos. Utilizarás un diccionario para almacenar la información de los contactos y guardarás los datos en un archivo JSON.
* **Temas a Usar:**
  + Archivos JSON
  + Funciones
  + Diccionarios
* **Recomendaciones:**
  + Define funciones para agregar, editar, eliminar y listar contactos.
  + Utiliza un archivo JSON para almacenar la información de los contactos.
  + Implementa una interfaz de línea de comandos que permita al usuario interactuar con la aplicación.
  + Asegúrate de manejar situaciones como la edición o eliminación de contactos que no existen.

## Proyecto 6: Generador de Contraseñas

* **Descripción:** Este proyecto consiste en crear un generador de contraseñas que permita a los usuarios generar contraseñas seguras basadas en sus preferencias, como longitud y tipos de caracteres (mayúsculas, minúsculas, números, símbolos).
* **Temas a Usar:**
  + Generación de Contraseñas
  + Funciones
  + Listas
* **Recomendaciones:**
  + Define una función que genere contraseñas aleatorias con base en las preferencias del usuario.
  + Permite al usuario especificar la longitud de la contraseña y qué tipos de caracteres debe incluir.
  + Asegúrate de que las contraseñas generadas sean seguras y difíciles de adivinar.

## Proyecto 7: Motor de Búsqueda de Archivos

* **Descripción:** En este proyecto, crearás un motor de búsqueda de archivos que permita a un usuario buscar archivos en un directorio específico y sus subdirectorios. Utilizarás expresiones regulares para buscar patrones de nombres de archivos.
* **Temas a Usar:**
  + Búsqueda de Archivos
  + Expresiones Regulares
  + Funciones
* **Recomendaciones:**
  + Define una función que realice la búsqueda de archivos en un directorio y sus subdirectorios.
  + Permite al usuario especificar patrones de nombres de archivos utilizando expresiones regulares.
  + Muestra los resultados de la búsqueda, incluyendo la ubicación de los archivos coincidentes.

## Proyecto 8: Sistema de Registro y Autenticación

* **Descripción:** En este proyecto, desarrollarás un sistema de registro y autenticación de usuarios que almacene credenciales en un archivo CSV. Los usuarios podrán registrarse, iniciar sesión y cambiar sus contraseñas.
* **Temas a Usar:**
  + Archivos CSV
  + Autenticación de Usuarios
  + Funciones
* **Recomendaciones:**
  + Define funciones para registrar usuarios, iniciar sesión y cambiar contraseñas.
  + Utiliza un archivo CSV para almacenar las credenciales de los usuarios, incluyendo nombres de usuario y contraseñas encriptadas.
  + Asegúrate de que el sistema maneje la autenticación de manera segura y que evite el acceso no autorizado.

## Proyecto 9: Calculadora Científica

* **Descripción:** En este proyecto, crearás una calculadora científica que permita a los usuarios realizar operaciones matemáticas avanzadas, como raíces cuadradas, logaritmos y funciones trigonométricas. Utilizarás la biblioteca **math** para las operaciones matemáticas.
* **Temas a Usar:**
  + Funciones Matemáticas Avanzadas
  + Biblioteca **math**
  + Funciones
* **Recomendaciones:**
  + Define una función que permita a los usuarios seleccionar el tipo de operación matemática que desean realizar.
  + Utiliza la biblioteca **math** para implementar las operaciones matemáticas avanzadas.
  + Proporciona una interfaz de usuario que permita a los usuarios ingresar números y operaciones matemáticas.

## Proyecto 10: Gestión de Inventarios

* **Descripción:** En este proyecto, crearás una aplicación que permita a un usuario administrar un inventario de productos. El programa debe permitir agregar, editar, eliminar y listar productos, y almacenar la información en un archivo CSV.
* **Temas a Usar:**
  + Archivos CSV
  + Funciones
  + Listas
* **Recomendaciones:**
  + Define funciones para agregar, editar, eliminar y listar productos en el inventario.
  + Utiliza un archivo CSV para almacenar la información de los productos, incluyendo detalles como nombre, precio y cantidad disponible.
  + Asegúrate de que el programa gestione adecuadamente las operaciones de inventario, como la actualización de la cantidad disponible y la manipulación de productos.