



# Introducción a la Programación Segura

Guía de ejercicios unidad 3: Colecciones y librerías en Python.



# Ejercicio 1 – Generador de Reportes de Gastos Mensuales

- **Descripción:** Desarrolle un programa que genere un reporte de los gastos mensuales de un usuario, clasificados por categorías. El programa debe permitir al usuario ingresar los gastos de cada categoría y mostrar un resumen al final.
- **Entrada:** Los gastos de cada categoría (números reales).
- **Salida:** Un reporte de los gastos mensuales por categoría.

## Pasos:

1. Ingresar Gastos por Categoría: Solicite al usuario que ingrese los gastos correspondientes a cada categoría.
2. Almacenar Gastos: Guarde los gastos ingresados en una estructura de datos adecuada, como un diccionario donde las claves son las categorías y los valores son las listas de gastos.
3. Mostrar Reporte: Itere sobre las categorías y muestre los gastos correspondientes.

## Restricciones:

1. Los gastos deben ser números reales positivos.

## Ejercicio 2 – Gestor de Tareas por Hacer con Recordatorios

- **Descripción:** Desarrolle un programa que gestione las tareas por hacer de un usuario, permitiéndole agregar nuevas tareas, establecer fechas límite y recibir recordatorios cuando se acerque la fecha límite de una tarea.
- **Entrada:** Descripción de la tarea (cadena de caracteres). Fecha límite de la tarea (fecha en formato YYYY-MM-DD).
- **Salida:** Recordatorio cuando se acerca la fecha límite de una tarea.

### Pasos:

1. Agregar Tarea por Hacer: Solicite al usuario que ingrese la descripción de la tarea y su fecha límite.
2. Almacenar Tareas: Guarde las tareas ingresadas en una estructura de datos adecuada, como un diccionario donde las claves son las fechas límite y los valores son las listas de tareas correspondientes.
3. Comprobar Fechas Límite: Compruebe si alguna tarea tiene la fecha límite cercana.
4. Enviar Recordatorio: Si se encuentra una tarea con fecha límite cercana, por ejemplo, una semana, muestre un recordatorio al usuario.

### Restricciones:

1. La fecha límite debe ser posterior a la fecha actual.
2. El formato de fecha debe ser YYYY-MM-DD.

# Ejercicio 3 – Sistema de Reservas de Mesas en un Restaurante

- **Descripción:** Desarrolle un programa que gestione las reservas de mesas en un restaurante. El programa debe permitir a los clientes reservar mesas para una fecha y hora específicas, así como cancelar reservas si es necesario.
- **Entrada:** Reservas de mesas (fecha, hora, número de personas).v
- **Salida:** Confirmación de reserva o cancelación.

## Pasos:

1. Realizar Reserva: Solicite al usuario que ingrese la fecha, hora y número de personas para la reserva.
2. Guardar Reserva: Almacene la reserva en una estructura de datos adecuada, como una lista de tuplas.
3. Verificar Disponibilidad: Compruebe si hay mesas disponibles para la fecha y hora especificadas.
4. Confirmar Reserva: Si hay disponibilidad, confirme la reserva y actualice la lista de mesas disponibles.
5. Cancelar Reserva: Permita al usuario cancelar una reserva existente, liberando la mesa reservada.

## Restricciones:

1. Las reservas deben realizarse con anticipación.
2. El restaurante tiene un número limitado de mesas disponibles.

# Ejercicio 4 – Sistema de Gestión de Citas para Pruebas de Manejo

- **Descripción:** Desarrolle un programa para gestionar las citas de pruebas de manejo en un concesionario de vehículos. El programa debe permitir a los clientes programar citas para realizar pruebas de manejo y recibir confirmaciones de reserva.
- **Entrada:** Fecha y hora de la cita. Tipo de vehículo para la prueba de manejo. Información de contacto del cliente.
- **Salida:** Confirmación de reserva de la cita.

## Pasos:

1. Seleccionar Fecha y Hora: Permita al cliente seleccionar una fecha y hora para la cita de la prueba de manejo, según la disponibilidad del concesionario.
2. Elegir Tipo de Vehículo: Permita al cliente elegir el tipo de vehículo que desea probar durante la cita.
3. Ingresar Información de Contacto: Solicite al cliente que ingrese su información de contacto para recibir la confirmación de la reserva.
4. Confirmar Reserva: Muestre un mensaje de confirmación de reserva al cliente, indicando la fecha, hora y detalles de la cita de la prueba de manejo..

## Restricciones:

1. Debe tener en cuenta la disponibilidad de vehículos y de personal para realizar las pruebas de manejo.
2. Debe proporcionar opciones de fecha y hora que se ajusten a la agenda del concesionario.



# Ejercicio 5 – Juego de Adivinanza de Palabras

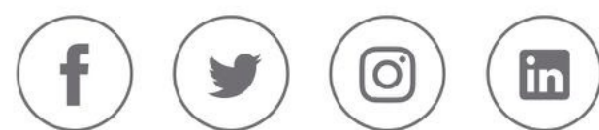
- **Descripción:** Desarrolle un juego en el que el jugador debe adivinar una palabra secreta, letra por letra. El jugador tiene un número limitado de intentos para adivinar la palabra completa.
- **Entrada:** Letra propuesta por el jugador como adivinanza.
- **Salida:** Estado actual de la palabra enmascarada, mostrando las letras adivinadas y las letras aún no adivinadas. Indicación de si la letra propuesta por el jugador está en la palabra secreta.

## Pasos:

1. Generar Palabra Secreta: El sistema elige aleatoriamente una palabra secreta de una lista predefinida.
2. Mostrar Palabra Enmascarada: El sistema muestra al jugador la palabra enmascarada, con todas las letras ocultas excepto la primera.
3. Pedir Adivinanza al Jugador: El sistema solicita al jugador que adivine la siguiente letra de la palabra, mostrando cuántos intentos le quedan (5 intentos máximo).
4. Verificar Adivinanza: El sistema compara la letra propuesta por el jugador con la palabra secreta. Si la letra está en la palabra secreta, el sistema actualiza la palabra enmascarada para mostrar la letra adivinada en su posición correcta. Si la letra no está en la palabra secreta, el sistema informa al jugador y reduce el número de intentos restantes.
5. Repetir hasta adivinar o agotar Intentos: El proceso de adivinanza se repite hasta que el jugador adivine la palabra secreta o agote todos sus intentos.

## Restricciones:

1. El juego debe ser capaz de manejar solo una palabra, si el texto ingresado es mayor a una palabra devolver error.



inacap.cl