

Trabajo Práctico Obligatorio (TPO)	2" Cuatrimestre 2024
---	-----------------------------

- **Fecha de entrega:** Tres días antes de la primera fecha de exposición y defensa.
- **Fechas de exposición y defensa:** Ver cronograma del curso.
- **Forma de entrega:** Entrega a través de las tareas de Microsoft Teams. No es necesario que lo entreguen todos los integrantes del equipo; un representante por grupo será suficiente.
- **Desarrollo:** Grupal, utilizando los mismos grupos de trabajo en los que habitualmente se desarrolla el trabajo en clase. No se admitirán trabajos individuales, ya que uno de los objetivos es fomentar el trabajo en equipo.
- **Material a entregar:** Archivo python (.py). Ejemplo: TPO_Grupo7.py. También deberá entregarse cualquier código adicional o archivo de datos que sea necesario para la correcta ejecución del programa.
- **Criterio de evaluación:** Se evaluarán los siguientes aspectos:
 - Fidelidad respecto a lo solicitado
 - Funcionamiento del programa
 - Técnicas de programación
 - Prolijidad del código y de la interfaz de usuario
 - La calificación final recibida por cada integrante dependerá no solo del trabajo presentado sino también de la defensa ejercida.
 - El desarrollo podrá llevarse a cabo durante el tiempo de clase dedicado a la ejercitación o fuera del horario de la misma.

-
- **Objetivo:** El Trabajo Práctico consistirá en el desarrollo de un programa Python que responda a la consigna elegida por el grupo y autorizada por el docente. En el mismo deberán aplicarse las técnicas de programación tratadas en la materia y detalladas más abajo:

Funciones
Matrices
Cadenas de caracteres
Tuplas, conjuntos y/o diccionarios (uno de los tres será suficiente).
Excepciones
Archivos
Recursividad

Todos estos temas tendrán que ser incluidos en el programa. La ausencia de alguno de ellos reducirá significativamente la calificación. Los temas podrán ser añadidos a medida que sean abordados en clase.

Exposición: Se realizará en dos etapas. La primera etapa representa al primer parcial de la materia, y sólo se podrán incluir aquellos contenidos tratados en clase con anterioridad a la fecha de primera entrega. La segunda etapa representa el examen final. En los días de exposición todos los miembros del grupo deberán participar de la defensa del trabajo presentado. Si algún integrante no participa de la misma se lo considerará ausente al segundo parcial y deberá recuperarlo en la fecha prevista para el recuperatorio. El recuperatorio consistirá en un examen individual, sin relación con el TPO.

Nota: No se permite el uso de módulos externos no tratados en clase como CSV, pygame, freegames, numpy, tkinter, etc. Sin embargo, los módulos time, os y colorama pueden ser utilizados si fuera necesario.

Algunos ejemplos de temas para la elaboración del Trabajo Práctico Obligatorio

1. **Sistema de Gestión de Inventarios**

Implementar un sistema que gestione inventarios utilizando matrices para categorías de productos. Utilizar Git para el control de versiones del proyecto. Incluir funciones avanzadas de listas, diccionarios y manejo de archivos para almacenar y procesar datos de inventario. Implementar recursividad para cálculos complejos y pruebas unitarias para validar el sistema.

2. **Aplicación de Seguimiento de Proyectos de Desarrollo de Software**

Crear una aplicación que gestione el seguimiento de proyectos de software. Usar matrices para asignar tareas, Git para control de versiones, listas y diccionarios para gestionar tareas y equipos, manejo de excepciones para la robustez del sistema, archivos para almacenamiento de datos y recursividad para generación de reportes.

3. **Plataforma de Análisis de Datos de Ventas**

Desarrollar una plataforma que lea datos de ventas desde archivos CSV y JSON. Utilizar matrices para representar datos de ventas, listas avanzadas y diccionarios para análisis y reportes. Implementar manejo de excepciones y recursividad para cálculos avanzados. Controlar el proyecto con Git y realizar pruebas unitarias para asegurar la calidad.

4. **Sistema de Reservas de Salas de Reunión**

Crear un sistema que gestione reservas de salas de reuniones utilizando matrices. Implementar control de versiones con Git, listas y diccionarios para gestión de usuarios y reservas, manejo de archivos para almacenamiento y recuperación de datos, manejo de excepciones para robustez y recursividad para búsqueda de disponibilidad.

5. **Aplicación de Gestión de Contactos**

Implementar una aplicación de gestión de contactos utilizando diccionarios para almacenar datos, listas para manejar contactos, matrices para organizar datos en vistas, manejo de excepciones para validación, recursividad para búsquedas, y Git para control de versiones. Incluir lectura y escritura de archivos.

6. **Sistema de Evaluación Académica**

Crear un sistema de evaluación que utilice matrices para calificaciones, listas y diccionarios para gestionar estudiantes y materias, manejo de excepciones para validar datos, recursividad para cálculos de promedios, y Git para el control de versiones. Almacenar y recuperar datos utilizando archivos.

7. **Sistema de Gestión de Biblioteca**

Desarrollar un sistema que gestione una biblioteca utilizando matrices para organizar libros, listas y diccionarios para gestión de usuarios y préstamos, manejo de excepciones para validación, recursividad para búsqueda de libros, y Git para el control de versiones. Implementar lectura y escritura de datos con archivos.

8. **Sistema de Registro de Asistencia**

Implementar un sistema de registro de asistencia utilizando matrices para organizar datos, listas y diccionarios para gestión de estudiantes y sesiones, manejo de excepciones para validar entradas, recursividad para análisis de datos, y Git para el control de versiones. Utilizar archivos para almacenamiento de datos.

9. Aplicación de Análisis de Texto

Desarrollar una aplicación que analice textos utilizando expresiones regulares para búsqueda de patrones, listas y diccionarios para almacenamiento y gestión de datos, matrices para organizar resultados, manejo de excepciones para validación, recursividad para análisis de textos, y Git para el control de versiones. Utilizar archivos para almacenamiento de datos.

10. Sistema de Gestión de Eventos

Implementar un sistema que gestione eventos utilizando matrices para organizar datos, listas y diccionarios para la administración de participantes y actividades, manejo de excepciones para validación, recursividad para cálculos de asistencia y participación, y Git para control de versiones. Utilizar archivos para almacenamiento y recuperación de datos.

11. Sistema de Venta de Pasajes

Implementar un sistema que gestione las ventas de pasajes de micro a distintos destinos y en diferentes fechas, utilizando matrices, listas y diccionarios para mantener la información. Esta debe almacenarse en archivos para permitir su posterior recuperación. Aplicar GIT para control de versiones y recursividad para realizar las búsquedas.

12. Sistema de Venta de Entradas para Espectáculos

Implementar un sistema que gestione las ventas de entradas para distintos espectáculos a realizarse en un estadio, utilizando matrices, listas y diccionarios para mantener la información, almacenándola en archivos para permitir su posterior recuperación. Aplicar GIT para control de versiones y recursividad para realizar las búsquedas.

13. Sistema de Reserva de Turnos Médicos

Implementar un sistema que gestione la reserva de turnos médicos para los profesionales de las distintas especialidades que atiende un centro de salud, utilizando matrices, listas y diccionarios para mantener la información, almacenándola en archivos para permitir su posterior recuperación. Aplicar GIT para control de versiones y recursividad para realizar las búsquedas.

14. Sistema de Búsqueda y Reserva de Paquetes Turísticos

Implementar un sistema que permita realizar la búsqueda y reserva de paquetes turísticos para los distintos destinos que comercializa una agencia de viajes, utilizando matrices, listas y diccionarios para mantener la información, almacenándola en archivos para permitir su posterior recuperación. Aplicar GIT para control de versiones y recursividad para realizar búsquedas.

Importante:

Cualquier otro tema puede ser propuesto y considerado,
sujeto a la aprobación del docente.
