

Sintaxis y Semántica del Lenguaje

Trabajo Practico TAD

Se formarán grupos de hasta CUATRO personas.

Cada grupo deberá desarrollar en Python un ejercicio del Trabajo Practico, que será asignado por el encargado de la práctica.

La fecha de entrega del trabajo será acordada con el encargado de la Práctica, según cronograma.

Es necesaria la APROBACION del trabajo antes de la fecha límite, para la aprobación de la cursada

Se deberá entregar simultáneamente:

a) La especificación e implementación de cada TAD y de la aplicación realizada.

b) Enviar por mail el trabajo (Código de los TADs y de la aplicación)

c) Cada grupo deberá presentar el trabajo de forma presencial, y demostrar la correcta implementación y funcionamiento.

Resolver las siguientes situaciones usando abstracción de datos. Desarrollar un menú en la aplicación para resolver cada punto del ejercicio

- 1) Desarrollar un sistema de gestión de expedientes para un estudio jurídico, implementando un archivero digital que permita organizar, administrar y manipular los documentos legales de forma eficiente. El sistema debe ejecutarse mediante un menú de opciones que habilite la interacción con el usuario y brinde soporte para diversas operaciones sobre los expedientes, respetando las siguientes especificaciones:
 - a. **Alta de Expedientes:**
Implementar la funcionalidad para agregar nuevos expedientes al archivero. Cada expediente deberá incluir los siguientes campos: número de expediente (único), titular, tipo de trámite y fecha de presentación (con hora).
 - b. **Modificación y Eliminación Individual:**
Permitir la modificación de los datos de un expediente existente a partir de su número, así como su eliminación del archivero en caso de que ya no sea necesario.
 - c. **Visualización General:**
Desarrollar una opción que muestre un listado completo y ordenado de todos los expedientes almacenados en el archivero, detallando todos sus datos.
 - d. **Actualización Masiva de Hora:**
Implementar una función que permita modificar la **hora** de presentación de todos los expedientes cuya **fecha** se encuentre dentro de un rango definido por el usuario.
 - e. **Generación de Pila Filtrada por Rango de Fechas:**
Crear una nueva pila que contenga **solo los expedientes** presentados entre dos fechas determinadas. Esta nueva pila debe ser impresa automáticamente en pantalla.
 - f. **Depuración por Mes:**
Incorporar una función que elimine del archivero original todos los expedientes cuya fecha de presentación corresponda a un **mes específico** indicado por el usuario.

- 2) Desarrollar una aplicación de consola que gestione un grupo de procesos que requieren acceso a la CPU. Cada proceso deberá registrar los siguientes datos: identificador (PID), nombre, tipo de proceso, tamaño, prioridad, y fecha y hora de su última modificación. El sistema debe presentar un menú interactivo que permita ejecutar las operaciones requeridas para la administración eficiente de los procesos.
- a. **Agregar nuevos procesos:**
Implementar la función para agregar procesos al grupo, respetando el orden de llegada. Se deberán ingresar todos los datos del proceso: PID, nombre, tipo, tamaño, prioridad, fecha y hora de última modificación.
 - b. **Modificación de prioridad individual:**
Permitir la actualización del campo **prioridad** de un proceso específico, identificado por su PID.
 - c. **Terminar proceso (atención de procesos):**
Implementar la opción para retirar del grupo el proceso que se encuentra al frente (el primero en llegar), simulando su atención por la CPU.
 - d. **Visualización completa de procesos:**
Diseñar una función que muestre todos los procesos, desplegando de manera ordenada y clara cada uno de sus atributos.
 - e. **Modificación masiva de prioridad por mes:**
Dado un mes ingresado por el usuario, actualizar la prioridad a “baja” para todos los procesos cuya fecha de última modificación corresponda a ese mes.
 - f. **Filtrado por tipo y por intervalo horario:**
 - o a) Eliminar todos los procesos cuyo tipo coincida con el valor ingresado por el usuario.
 - o b) Crear una **cola** con los procesos cuya **hora de última modificación** se encuentre dentro de un intervalo definido por el usuario, e imprimir su contenido automáticamente.
- 3) Desarrollar una aplicación que permita a un médico gestionar su agenda de citas utilizando los Tipos Abstractos de Datos (TADs) que se consideren apropiados. Cada cita deberá contener la siguiente información: DNI del paciente, nombre, obra social, teléfono, fecha de la cita y hora de la cita. El sistema deberá brindar un menú de opciones mediante el cual el usuario pueda realizar distintas operaciones sobre las citas almacenadas en la agenda.
- a. **Alta de citas:**
Implementar la funcionalidad para registrar nuevas citas médicas, ingresando todos los datos correspondientes al paciente.
 - b. **Modificación de fecha y hora de una cita:**
Permitir modificar la fecha y/o la hora de una cita ya existente, identificando al paciente mediante su número de DNI.
 - c. **Eliminación de citas individuales:**
Implementar una opción para eliminar una cita específica de la agenda, ya sea por cancelación o modificación del cronograma del médico.
 - d. **Listado completo de citas:**
Desarrollar una funcionalidad que permita mostrar todas las citas almacenadas, desplegando ordenadamente todos los datos de cada paciente citado.

- e. **Traslado de citas de un día a otro:**
Permitir al usuario mover todas las citas de una fecha determinada a una nueva fecha, facilitando la reorganización de la agenda ante imprevistos.
 - f. **Depuración y generación de lista filtrada:**
 - o a) Eliminar todas las citas asociadas a una obra social específica, ingresada por el usuario.
 - o b) Generar una nueva **cola** que contenga únicamente los **nombres y obras sociales** de los pacientes citados en un día específico, e imprimirla inmediatamente en pantalla.
- 4) Diseñar e implementar una aplicación que permita a una empresa proveedora de servicios de Internet gestionar la información de sus clientes suscriptos. Cada cliente debe almacenar los siguientes datos: número de cliente, apellido, nombre, fecha de alta, tipo de servicio contratado y precio del servicio. El sistema deberá estar basado en estructuras de datos adecuadas (TADs) y ofrecer un menú interactivo que permita realizar diversas operaciones sobre la base de datos de clientes.
- a. **Registro de nuevos clientes:**
Implementar una función que permita agregar nuevos clientes a la base de datos, solicitando e ingresando todos los campos obligatorios.
 - b. **Modificación de datos de clientes:**
Desarrollar la funcionalidad para actualizar cualquier información de un cliente existente, identificándolo mediante su número de cliente.
 - c. **Eliminación de clientes:**
Permitir la eliminación de un cliente de la base de datos en caso de baja del servicio, garantizando la integridad de los datos restantes.
 - d. **Visualización general de clientes:**
Implementar una función que muestre un listado completo de todos los clientes registrados, detallando claramente sus datos personales y de servicio.
 - e. **Operaciones por tipo de servicio y antigüedad:**
 - o a) Eliminar todos los clientes que tengan un tipo de servicio específico, ingresado por el usuario.
 - o b) Aplicar un descuento sobre el precio del servicio a aquellos clientes cuya antigüedad sea superior a tres años, calculada a partir de la fecha de alta.
 - f. **Listado de clientes en período promocional:**
Generar un listado que contenga únicamente a los clientes que se encuentren dentro de sus primeros **tres meses desde la fecha de alta**, indicando que aún están dentro del período promocional.
- 5) Desarrollar una aplicación que permita gestionar una **agenda personal de citas**, utilizando los Tipos Abstractos de Datos (TADs) que el estudiante considere apropiados. Cada cita deberá contener: actividad a realizar, nivel de importancia, fecha y hora de inicio. La aplicación deberá presentar un menú interactivo para que el usuario realice distintas operaciones de administración sobre las citas registradas.
- a. **Alta de citas:**
Implementar la opción para agregar nuevas citas a la agenda, solicitando todos los datos requeridos: actividad, nivel de importancia, fecha y hora de inicio.

- b. **Modificación de una cita existente:**
Permitir la actualización completa o parcial de los datos de una cita ya registrada, identificándola mediante la fecha y hora de inicio.
 - c. **Eliminación de citas individuales:**
Desarrollar la funcionalidad para eliminar una cita específica de la agenda, liberando su espacio en el calendario personal.
 - d. **Listado general de citas:**
Implementar una función que muestre todas las citas almacenadas, detallando actividad, importancia, fecha y hora de cada una.
 - e. **Reorganización y depuración por fecha:**
 - o a) Trasladar todas las citas de una fecha determinada a una nueva fecha ingresada por el usuario.
 - o b) Eliminar todas las citas correspondientes a un día específico, permitiendo limpiar completamente la agenda para esa fecha.
 - f. **Generación de cola filtrada por día:**
Crear una **nueva cola** que contenga únicamente las **actividades** y su **nivel de importancia** de un día específico. Esta cola deberá generarse y mostrarse automáticamente en pantalla.
-
- 6) Diseñar e implementar una aplicación que permita a una farmacia gestionar la información relacionada con las **ventas de medicamentos**, utilizando los Tipos Abstractos de Datos (TADs) que el estudiante considere adecuados. Cada venta debe registrar: código del medicamento, nombre, droga, obra social, plan, importe, fecha y hora de la venta. El sistema deberá contar con un menú interactivo que permita realizar distintas operaciones sobre los datos registrados.
- a. **Registro de nuevas ventas:**
Implementar la opción para ingresar nuevas ventas de medicamentos, cargando todos los datos requeridos para cada registro.
 - b. **Modificación y eliminación de ventas específicas:**
 - o a) Permitir la modificación de una venta existente, identificada por el nombre del medicamento.
 - o b) Permitir la eliminación de una venta a partir del **código** del medicamento.
 - c. **Visualización completa de las ventas:**
Mostrar un listado detallado de todas las ventas registradas, incluyendo todos los datos asociados a cada medicamento vendido.
 - d. **Informes y actualizaciones por obra social y plan:**
 - o a) Generar un informe con el **total recaudado por cada obra social**, en base a las ventas realizadas.
 - o b) Aplicar un descuento del 20% en el importe de las ventas correspondientes a un **plan específico**, ingresado por el usuario.
 - e. **Eliminación por droga y fecha reciente:**
Eliminar todas las ventas del **último mes** que correspondan a medicamentos cuya **droga coincida** con la especificada por el usuario.
 - f. **Filtrado por obra social y resumen diario por hora:**
 - o a) Generar una **cola** con el nombre, droga y fecha de venta de los medicamentos vendidos por una **obra social determinada**, y mostrarla en pantalla.

- b) Dada una **hora específica**, listar la **cantidad de medicamentos vendidos** y el **monto total recaudado** ese mismo día **hasta dicha hora**.
- 7) Desarrollar una aplicación que permita administrar el ingreso y egreso de vehículos en un estacionamiento perteneciente a un conjunto de oficinas en la ciudad de Buenos Aires. Cada vehículo debe almacenar los datos de patente, hora de ingreso, torre asignada, y eventualmente hora de salida y monto total de estadía. El sistema debe permitir definir el valor de la hora al comienzo de cada jornada y calcular el costo de permanencia, aplicando un descuento del 15% a los vehículos estacionados en la torre 3. El sistema debe construirse utilizando los TADs que el estudiante considere más adecuados y presentar un menú interactivo con las funciones detalladas a continuación.
- a. **Registro y modificación de vehículos ingresados:**
Implementar funciones que permitan ingresar nuevos vehículos al estacionamiento, registrando patente, hora de ingreso y torre; así como modificar los datos de ingreso de un vehículo ya registrado, si fuera necesario.
 - b. **Registro de salida y cálculo de monto a cobrar:**
Registrar la hora de salida de un vehículo y calcular el monto total a cobrar por la estadía, aplicando un 15% de descuento si el vehículo estuvo en la torre 3.
 - c. **Listado completo de vehículos:**
Mostrar todos los vehículos registrados, ya sea que aún estén estacionados o hayan egresado, incluyendo todos sus datos (patente, ingreso, salida, torre, monto).
 - d. **Informe de recaudación por torre:**
Generar un informe que indique el **monto total recaudado** por **cada torre** durante el día.
 - e. **Estadísticas y limpieza por horario y torre:**
 - a) Contar la cantidad de vehículos ingresados durante las **horas pico** (de 7 a 10 h y de 17 a 20 h).
 - b) Eliminar los vehículos que hayan ingresado **después de las 18 h** a una **torre específica**, permitiendo así liberar espacio.
 - f. **Generación y visualización de cola por torre:**
Crear una **cola** con todos los vehículos pertenecientes a una torre específica, y **mostrar automáticamente** en pantalla la información correspondiente de cada uno.
- 8) Desarrollar una aplicación que permita gestionar la información de los **choferes y sus camiones** en una empresa de transporte de carga, utilizando los Tipos Abstractos de Datos (TADs) que el estudiante considere adecuados. De cada chofer se registran: nombre, apellido, legajo, fecha de ingreso, número de camión y zona de trabajo. El sistema debe ofrecer un menú interactivo que permita realizar las operaciones detalladas a continuación.
- a. **Alta y modificación de choferes:**
Implementar funciones para agregar nuevos choferes con todos sus datos, así como modificar la información de un chofer existente (nombre, zona de trabajo, número de camión, etc.).

- b. **Eliminación segura de choferes:**
Permitir eliminar choferes del sistema utilizando su legajo como identificador único, asegurando que los datos se eliminen correctamente.
- c. **Visualización completa:**
Generar y mostrar un listado con **todos los choferes registrados**, incluyendo todos sus datos.
- d. **Cambio masivo de zona por año de ingreso:**
Permitir cambiar la **zona de trabajo** de todos los choferes que ingresaron en un año determinado, reorganizando así la asignación geográfica.
- e. **Depuración por antigüedad:**
Eliminar automáticamente del sistema a los choferes cuya **antigüedad supere los 30 años**, calculada a partir de su fecha de ingreso.
- f. **Cola por zona de trabajo:**
Generar una **cola** con los choferes de una **zona específica**, y mostrar en pantalla los datos de cada uno en el orden correspondiente.