

Ejercicio 1: Biblioteca Virtual ERP

Descripción

La universidad *KnowledgeLand* desea implementar un sistema ERP para administrar la **Biblioteca Virtual** y llevar el control total de estudiantes, libros, préstamos y reportes.

Actualmente, el proceso se hace de forma manual, lo cual genera confusión en préstamos, sanciones por mora y disponibilidad de libros.

Se contrata un equipo de programadores para desarrollar un programa en **Python**, modularizado, con persistencia en **JSON**, y que funcione mediante **menús y submenús**.

Especificaciones del sistema

Roles del sistema

El programa debe permitir tres tipos de roles con sus funcionalidades:

1. **Estudiante**
 - Consultar catálogo de libros.
 - Reservar libros.
 - Realizar préstamos.
 - Devolver libros.
 - Ver estado de cuenta (libros prestados, sanciones, reservas pendientes).
2. **Bibliotecario**
 - Registrar nuevos libros.
 - Actualizar stock de libros.
 - Registrar devoluciones y aplicar sanciones.
 - Cambiar estado de estudiantes (Activo, Sancionado, Retirado).
3. **Coordinador académico**
 - Generar reportes consolidados.
 - Revisar estudiantes sancionados.
 - Listar libros más prestados.
 - Gestionar estadísticas de uso de la biblioteca.

Información a manejar

- **Estudiantes:** ID, nombres, apellidos, carrera, semestre, teléfono, correo, estado (Activo, Sancionado, Retirado).
- **Libros:** ID, título, autor, categoría, ISBN, cantidad disponible.
- **Préstamos:** estudiante, libro, fecha de inicio, fecha de entrega, estado (Activo, Devuelto, Vencido).
- **Reservas:** estudiante, libro, fecha, estado (Pendiente, Activa, Cancelada).

- **Sanciones:** estudiante, motivo, fecha inicio, fecha fin.

Menús y Submenús

1. Menú Principal

- Estudiante
- Bibliotecario
- Coordinador académico
- Salir

2. Menú Estudiante

- Consultar catálogo de libros.
- Reservar libro.
- Prestar libro.
- Devolver libro.
- Estado de cuenta.
- Volver al menú principal.

3. Menú Bibliotecario

- Registrar nuevo libro.
- Actualizar stock de libros.
- Registrar devolución y sanciones.
- Cambiar estado de estudiante.
- Volver al menú principal.

4. Menú Coordinador Académico

- Reporte: estudiantes sancionados.
- Reporte: libros más prestados.
- Reporte: préstamos activos.
- Reporte: estadísticas de uso (total préstamos, devoluciones, reservas).
- Volver al menú principal.

Reglas de negocio

- Un estudiante no puede reservar más de **3 libros al mismo tiempo**.
- Un estudiante sancionado no puede hacer préstamos ni reservas.
- Si un préstamo excede la fecha límite → pasa a estado "**Vencido**" y genera sanción automática.
- Un libro no puede prestarse si no hay stock disponible.
- El coordinador académico debe poder exportar los reportes en formato **JSON** o **TXT**.

Ejercicio 2: Sistema de Gestión Hospitalaria ERP

Descripción

El Hospital *VidaSalud* desea crear un sistema ERP que permita gestionar pacientes, doctores, citas médicas y reportes administrativos. Actualmente, las citas se asignan en papel, lo que provoca pérdida de información, citas duplicadas y dificultad para dar seguimiento a diagnósticos y tratamientos.

Se contrata un equipo de programadores para diseñar un programa en **Python**, modularizado, con almacenamiento en **JSON**, que funcione mediante **menús y submenús**.

Especificaciones del sistema

Roles del sistema

1. **Paciente**
 - Registrarse en el sistema.
 - Solicitar citas médicas.
 - Consultar historial de citas.
 - Cancelar citas.
2. **Doctor**
 - Consultar agenda de citas programadas.
 - Atender citas y registrar diagnóstico y tratamiento.
 - Marcar citas como **Atendidas** o **Canceladas**.
3. **Administrador hospitalario**
 - Registrar nuevos doctores.
 - Modificar horarios disponibles de cada doctor.
 - Asignar especialidades a los doctores.
 - Generar reportes estadísticos.

Información a manejar

- **Pacientes:** ID, nombre, apellidos, edad, dirección, teléfono, correo, historial médico, estado (Activo, Inactivo).
- **Doctores:** ID, nombre, apellidos, especialidad, horarios disponibles, estado (Activo, Retirado).
- **Citas:** paciente, doctor, especialidad, fecha, hora, estado (Programada, Atendida, Cancelada).
- **Consultas:** cita asociada, diagnóstico, tratamiento, observaciones.ç

Menús y Submenús

1. Menú Principal

- Paciente
- Doctor
- Administrador
- Salir

2. Menú Paciente

- Registrarse.
- Solicitar cita médica.
- Consultar historial de citas.
- Cancelar cita.
- Volver al menú principal.

3. Menú Doctor

- Consultar agenda de citas.
- Registrar diagnóstico y tratamiento.
- Marcar cita como atendida o cancelada.
- Volver al menú principal.

4. Menú Administrador

- Registrar nuevo doctor.
- Modificar horarios de doctores.
- Asignar especialidad.
- Reportes:
 - a. Listar pacientes atendidos por cada doctor.
 - b. Listar citas canceladas.
 - c. Especialidades más solicitadas.
 - d. Reporte general de productividad (número de citas atendidas, canceladas y pendientes).
 - e. Volver al menú principal.

Reglas de negocio

- Un paciente solo puede tener **máximo 2 citas programadas activas** al mismo tiempo.
- Un doctor no puede tener citas fuera de su horario registrado.
- Si un paciente cancela 3 citas consecutivas, su estado cambia a **"Inactivo"**.
- Cada cita debe estar asociada a una **especialidad médica válida** (Cardiología, Pediatría, Medicina General, etc.).
- El administrador debe poder exportar reportes en **JSON** o **CSV**.

Ejercicio 3: Tienda Online ERP

Descripción

La empresa *ShopEasy* desea implementar un sistema ERP para la gestión de su **tienda virtual**, con el fin de organizar productos, clientes, órdenes de compra y reportes de ventas.

Actualmente, la información se maneja en hojas de cálculo, lo que provoca errores en inventario, ventas duplicadas y dificultad para identificar a los clientes frecuentes.

Se contrata un equipo de programadores para desarrollar un sistema en **Python**, modularizado, con persistencia en **JSON**, que funcione mediante **menús y submenús**.

Especificaciones del sistema

Roles del sistema

1. **Cliente**

- Registrarse en la tienda.
- Consultar catálogo de productos.
- Realizar órdenes de compra.
- Consultar estado de sus órdenes.
- Descargar factura.

2. **Vendedor**

- Registrar nuevos productos.
- Actualizar precios y stock.
- Consultar órdenes pendientes de envío.
- Cambiar estado de una orden (Pendiente → Enviado → Entregado).

3. **Administrador**

- Gestionar clientes (activar, desactivar).
- Gestionar vendedores.
- Generar reportes de ventas.

Información a manejar

- **Clientes:** ID, nombre, apellidos, correo, dirección, teléfono, estado (Activo, Inactivo).
- **Productos:** ID, nombre, categoría, precio, stock disponible.
- **Órdenes:** cliente, lista de productos, fecha, total, estado (Pendiente, Enviado, Entregado).
- **Facturas:** número de factura, cliente, detalle de la orden, valor total, fecha.

Menús y Submenús

1. Menú Principal

- Cliente
- Vendedor
- Administrador
- Salir

2. Menú Cliente

- Registrarse.
- Consultar catálogo de productos.
- Realizar orden de compra.
- Consultar estado de mis órdenes.
- Descargar factura.
- Volver al menú principal.

3. Menú Vendedor

- Registrar nuevo producto.
- Actualizar precio o stock.
- Consultar órdenes pendientes.
- Cambiar estado de orden.
- Volver al menú principal.

4. Menú Administrador

- Gestionar clientes (activar/desactivar).
- Registrar vendedor.
- Reportes:
 - a. Ventas por mes.
 - b. Producto más vendido.
 - c. Clientes con mayor número de compras.
 - d. Ventas por vendedor.
- Volver al menú principal.

Reglas de negocio

- Un cliente solo puede tener una orden activa en estado **Pendiente** antes de realizar otra compra.
- Si un producto tiene stock = **0**, no puede incluirse en nuevas órdenes.
- El sistema debe calcular automáticamente el **total de cada orden**.
- Toda orden debe generar una **factura digital**.
- Los reportes deben poder exportarse en **JSON** o **TXT**.

Ejercicio 4: Sistema de Cine ERP

Descripción

La cadena de cines *CineMax* desea implementar un sistema ERP que permita administrar películas, funciones, salas y reservas de boletos.

Actualmente, las reservas se hacen en ventanilla de manera manual, lo que genera problemas de sobreventa, pérdida de información y dificultades para conocer las películas más exitosas.

Se contrata un equipo de programadores para diseñar un sistema en **Python**, modularizado, con almacenamiento en **JSON**, que funcione mediante **menús y submenús**.

Especificaciones del sistema

Roles del sistema

1. Cliente

- Registrarse en el sistema.
- Consultar cartelera de películas.
- Reservar boletos para una función.
- Cancelar reservas.
- Consultar historial de reservas.

2. Cajero

- Registrar nuevas funciones (película, sala, fecha, hora).
- Registrar ventas directas de boletos.
- Validar boletos al ingreso.
- Cancelar funciones en caso necesario.

3. Administrador

- Registrar nuevas películas.
- Administrar salas (número de sala, capacidad).
- Generar reportes de ocupación y ventas.

Información a manejar

- **Clientes:** ID, nombres, apellidos, teléfono, correo, estado (Activo, Inactivo).
 - **Películas:** ID, título, género, clasificación, duración.
 - **Salas:** número de sala, capacidad, estado (Activa, Inactiva).
 - **Funciones:** película, sala, fecha, hora, estado (Programada, Cancelada, Finalizada).
 - **Reservas:** cliente, función, número de asientos reservados, estado (Activa, Cancelada).
 - **Ventas:** función, cantidad de boletos, valor total.
-

Menús y Submenús

1. **Menú Principal**

- Cliente
- Cajero
- Administrador
- Salir

2. **Menú Cliente**

- Registrarse.
- Consultar cartelera de películas.
- Reservar boletos.
- Cancelar reserva.
- Consultar historial de reservas.
- Volver al menú principal.

3. **Menú Cajero**

- Registrar nueva función.
- Registrar venta directa de boletos.
- Validar boleto.
- Cancelar función.
- Volver al menú principal.

4. **Menú Administrador**

- Registrar nueva película.
- Administrar salas.
- Reportes:
 - a. Películas más vistas.
 - b. Funciones agotadas.
 - c. Ingresos por película.
 - d. Clientes frecuentes.
- Volver al menú principal.

Reglas de negocio

- No se puede superar la capacidad máxima de la sala al reservar o vender boletos.
- Un cliente puede reservar máximo **5 boletos por función**.
- Una función no puede programarse en una sala **ya ocupada en la misma franja horaria**.
- Una reserva cancelada libera inmediatamente los asientos.
- Los reportes deben poder exportarse en **JSON** o **CSV**.

Ejercicio 5: Sistema de Transporte Público ERP

Descripción

La empresa de transporte *TransLand* desea implementar un sistema ERP que le permita administrar rutas, buses, conductores, pasajeros y la venta de tickets. Actualmente, las ventas se realizan de forma manual en terminales, lo que genera sobreventa de tickets, falta de control en las rutas y dificultad para conocer la rentabilidad de cada trayecto.

Se contrata un equipo de programadores para desarrollar un sistema en **Python**, modularizado, con persistencia en **JSON**, que funcione mediante **menús y submenús**.

Especificaciones del sistema

Roles del sistema

1. **Pasajero**
 - Registrarse en el sistema.
 - Consultar rutas disponibles.
 - Comprar tickets.
 - Cancelar tickets.
 - Consultar historial de viajes.
2. **Conductor**
 - Consultar las rutas asignadas.
 - Marcar el inicio y fin de un viaje.
 - Reportar incidencias en ruta (averías, retrasos, accidentes).
3. **Administrador**
 - Registrar nuevas rutas.
 - Registrar buses y asignar conductores.
 - Gestionar horarios.
 - Generar reportes de uso y rentabilidad.

Información a manejar

- **Pasajeros:** ID, nombres, apellidos, teléfono, correo, estado (Activo, Bloqueado).
- **Rutas:** ID, origen, destino, paradas, horario, estado (Activa, Inactiva).
- **Buses:** placa, capacidad, conductor asignado, estado (Disponible, En mantenimiento).
- **Conductores:** ID, nombre, apellidos, licencia, disponibilidad.
- **Tickets:** pasajero, ruta, fecha, hora, número de asiento, estado (Activo, Cancelado, Usado).
- **Viajes:** ruta, bus, conductor, fecha, hora inicio, hora fin, incidencias reportadas.

Menús y Submenús

1. **Menú Principal**

- Pasajero
- Conductor
- Administrador
- Salir

2. **Menú Pasajero**

- Registrarse.
- Consultar rutas disponibles.
- Comprar ticket.
- Cancelar ticket.
- Consultar historial de viajes.
- Volver al menú principal.

3. **Menú Conductor**

- Consultar rutas asignadas.
- Marcar inicio de viaje.
- Marcar fin de viaje.
- Reportar incidencia.
- Volver al menú principal.

4. **Menú Administrador**

- Registrar nueva ruta.
- Registrar bus y asignar conductor.
- Gestionar horarios.
- Reportes:
 - Rutas más utilizadas.
 - Conductores con mayor número de viajes.
 - Ingresos por ruta.
 - Buses con más incidencias reportadas.
- Volver al menú principal.

Reglas de negocio

- Un bus no puede superar su **capacidad máxima** de pasajeros.
- Un pasajero no puede comprar más de **5 tickets por viaje**.
- Un ticket cancelado libera inmediatamente el asiento correspondiente.
- Un conductor no puede ser asignado a dos rutas que se crucen en el mismo horario.
- Si un pasajero acumula **3 cancelaciones consecutivas**, su estado cambia a **"Bloqueado"**.
- Los reportes deben poder exportarse en **JSON** o **CSV**.