



Condiciones del Proyecto Final: Programación I

El proyecto final debe demostrar la aplicación práctica e integrada de los conocimientos y habilidades adquiridos durante el semestre, incluyendo los temas adicionales de archivos, interfaz y el uso de Python.

Objetivo General del Proyecto

El estudiante debe **diseñar, implementar y documentar una aplicación funcional en Python** que resuelva un problema práctico definido por ellos mismos utilizando los conceptos de **programación estructurada, subprogramas, estructuras de datos simples, manejo de archivos** y una **interfaz de usuario funcional** (de consola o gráfica).

Requisitos Mínimos Obligatorios

El proyecto, implementado en **Python**, debe cumplir estrictamente con los siguientes requisitos funcionales y de código:

Requisito	Detalle	Fundamento Programático
1. Manejo de Archivos	El programa debe leer y escribir datos persistentes en al menos un archivo (texto o binario, según la complejidad del proyecto), no solo para configuración, sino para la lógica central de la aplicación (e.g., guardar registros, historial, o datos de la aplicación).	Tema adicional: Archivos
2. Uso de Subprogramas	Debe implementar y utilizar funciones o procedimientos (subprogramas) propios para modularizar el código, incluyendo el pase de parámetros	Funciones, módulos, paquetes Subprogramas

3. Estructuras de Datos Simples	Debe utilizar al menos una estructura de datos compleja (como una lista o tupla o una colección de registros/diccionarios) para almacenar y manipular la información clave del sistema.	Estructuras de datos simples
4. Interfaz de Usuario	El proyecto debe contar con una interfaz de usuario que permita la interacción clara con el usuario. Esta puede ser: Interfaz de Consola Avanzada: Con menús claros, validación de entradas y mensajes de error. O Interfaz Gráfica (GUI): Utilizando una librería estándar de Python entradas y salidas de datos.	Tema adicional: Interfaz
5. Lógica de Control	Debe emplear las estructuras de decisión (condicionales) y al menos 2 tipos de estructuras de repetición (bucles <code>for</code> y <code>while</code>) para implementar la lógica de control del programa.	Lógica de programación

Componentes a Entregar y Criterios de Evaluación

La evaluación se centrará en la **funcionalidad** , **la calidad del código** y **la documentación** .

1. Implementación (Código Fuente y Ejecución)

- **Funcionalidad** : El programa debe ejecutar sin errores críticos y realizar todas las tareas descritas en el diseño.
- **Presentación y Defensa** : Los estudiantes deben demostrar la aplicación en funcionamiento y ser capaces de explicar las decisiones de diseño y las secciones clave del código, demostrando un entendimiento total.

2. Calidad y Estructura del Código

- **Modularización y Eficiencia:** Uso correcto y estratégico de subprogramas y estructuras de datos para evitar la duplicidad de código.
- **Legibilidad y Estilo :** Uso de convenciones de estilo de Python (PEP 8), nombres de variables y funciones significativos, y comentarios claros que expliquen la lógica compleja.
- **Manejo de Errores :** Inclusión de bloques `try/except` para manejar entradas no válidas y errores de archivos, asegurando que el programa no falle inesperadamente.

3. Documentación (Informe Técnico)

El informe debe ser conciso, profesional y contener:

- **Análisis del Problema :** Descripción clara del problema a resolver y los requisitos funcionales del sistema.
- **Diseño :**
 - Diagrama de flujo o pseudocódigo para las funciones clave, mostrando la lógica de programación.
 - Descripción de las estructuras de datos usadas y el formato del archivo persistente.
- **Conclusiones y Lecciones Aprendidas:** Una sección crítica sobre los desafíos enfrentados y cómo se aplicaron los conocimientos del curso (archivos, interfaz, modularidad) para superarlos.



Plazos y Formato de Entrega

Para asegurar una entrega ordenada y rigurosa, se establece:

- **Entrega:** La entrega debe incluir el código fuente completo, el informe técnico en formato PDF, y cualquier archivo de datos de ejemplo necesario para la ejecución.
 - **Presentación:** Se realizará una presentación obligatoria donde cada grupo demostrará el funcionamiento del proyecto y responderá preguntas sobre el código y el diseño.
 - **Fecha Límite:** 25/11/2025.
 - **Defensa Oral:** 26/11/2025.
-

15 Proyectos Finales de Nivel Intermedio

1. Sistema de gestión de inventario

- Registra productos con nombre, código, cantidad y precio.
- Permite búsqueda, actualización y eliminación.

2. Agenda de eventos personales

- Guarda eventos con fecha, hora, descripción.
- Permite consultar eventos por día o rango de fechas.

3. Simulador de sistema de votación

- Registra candidatos, permite emitir votos y muestra resultados con porcentajes.

4. Gestor de tareas con prioridad

- Crea tareas con prioridad (alta, media, baja).
- Permite marcar como completadas y filtrar por estado.

5. Control de asistencia estudiantil

- Registra estudiantes y su asistencia diaria.
- Calcula porcentajes de asistencia por estudiante.

6. Sistema de reservas de citas médicas

- Agenda citas por fecha y hora.
- Verifica disponibilidad y permite cancelaciones.

7. Simulador de cajero automático

- Permite depósitos, retiros, consulta de saldo y validación de PIN.

8. Conversor de unidades científicas

- Longitud, masa, temperatura, tiempo.
- Incluye validación y menú interactivo.

9. Gestor de biblioteca personal

- **Registra libros con autor, género, estado (prestado/disponible).**
- **Permite búsqueda por título o autor.**

10. Sistema de inscripción a cursos

- **Registra estudiantes y asignaturas.**
- **Verifica conflictos de horario y cupos disponibles.**

11. Analizador de texto

- **Cuenta palabras, caracteres, frecuencia de letras.**
- **Detecta palíndromos y frases repetidas.**

12. Simulador de tienda en línea

- **Catálogo de productos, carrito de compras, cálculo de total.**
- **Simulación de pago y generación de factura.**

13. Gestor de recetas culinarias

- **Registra recetas con ingredientes y pasos.**
- **Permite búsqueda por ingrediente o tipo de comida.**

14. Sistema de control de acceso

- **Registra usuarios con contraseñas.**
- **Verifica credenciales y permite múltiples roles (admin, usuario).**

15. Simulador de sistema de calificaciones

- **Registra estudiantes, asignaturas y notas.**
- **Calcula promedios, clasifica por rendimiento y genera reportes.**