

## Trabajo Práctico Final - Programación 1 con Python

### Consigna General del Proyecto

#### Descripción

El trabajo práctico final consiste en el **desarrollo de un sistema de gestión** para un área específica elegida por los estudiantes y aprobada por el docente. El sistema deberá implementar funcionalidades CRUD (Crear, Leer, Actualizar, Eliminar), gestionar y procesar información utilizando estructuras de datos adecuadas, y aplicar buenas prácticas de programación. Además, deberá incluir validaciones, almacenamiento de datos en archivos, muestra de datos por consola y control de versiones con Git.

El desarrollo del proyecto se realizará en equipos de trabajo, fomentando la colaboración, el uso de herramientas de programación profesional y aplicando metodologías ágiles.

#### Organización

Se formarán grupos de 5 (cinco) estudiantes en la primera clase<sup>1</sup>. Durante la segunda semana se definirá la temática y el líder de equipo. El grupo se mantendrá durante toda la cursada. No se admitirán trabajos individuales, ya que uno de los objetivos es fomentar el trabajo en equipo. Cada estudiante es responsable de contribuir activamente al desarrollo del proyecto y su entrega.

Durante las clases subsiguientes se destinarán momentos para el desarrollo del proyecto, aclaración de dudas y consultas. Se realizará un seguimiento personalizado semanal/quincenal de cada grupo, en el cual deberán exponer brevemente:

- a) Avances realizados hasta el momento.
- b) Trabajo en curso y desafíos actuales.
- c) Próximos pasos y planificación futura.

El desarrollo requiere un trabajo en equipo con **comunicación constante**, asegurando la **participación activa de todos los integrantes**, tanto durante las clases como fuera de ellas. Esto es fundamental para cumplir con las metas semanales del proyecto en tiempo y forma.

#### Requisitos del Proyecto (A: primera entrega; B: segunda entrega)

##### 1. Gestión de datos (CRUD), consulta y estadística (A/B)

- Implementar funciones para agregar, leer, actualizar y eliminar los elementos relevantes del sistema (ejemplo: usuarios, productos, tareas, reservas, etc.).
- Implementar funciones para consultar por un registro completo y datos relacionados (ejemplo: consultar por un cliente, mostrando sus datos y las compras que realizó).
- Implementar funciones de estadística básica que incluyan:
  - Cálculo de promedio general y por categoría (por ejemplo: promedio de ventas, notas, reservas, etc.).
  - Conteo de registros totales y discriminados según categorías específicas.
  - Porcentajes relativos de categorías sobre el total general (por ejemplo: porcentaje de usuarios activos, productos más vendidos, reservas realizadas, etc.).
  - Identificación de valores máximos y mínimos en los registros (por ejemplo: la mayor o menor venta, el estudiante con mejor o peor promedio, etc.).

<sup>1</sup> En caso de que la cantidad total de estudiantes no sea múltiplo de 5, se permitirán excepciones con grupos más pequeños, a definir por el docente.

- Resumen estadístico simple: total, promedio, valor máximo, valor mínimo y conteos específicos.
- 2. **Estructuras de datos**
  - o Utilizar **matrices** para organizar y procesar datos de manera eficiente. (A)
  - o Usar **listas** para gestionar información adicional, como tareas pendientes o historiales. (A/B)
  - o Implementar **diccionarios** para almacenar información estructurada (ejemplo: datos de usuarios, inventarios, configuraciones). (A/B)
  - o Utilizar **tuplas** cuando se requiera almacenar datos inmutables o devolver múltiples valores en una función. (A/B)
  - o Aplicar **conjuntos** para evitar duplicados y realizar operaciones de intersección o diferencia en estructuras de datos. (A/B)
- 3. **Manejo de excepciones (B)**
  - o Validar entradas del usuario para evitar errores y mejorar la robustez del sistema.
- 4. **Recursividad (B)**
  - o Utilizar recursividad para resolver problemas que requieran cálculos repetitivos o búsquedas avanzadas.
- 5. **Control de versiones con Git (A/B)**
  - o Usar Git para el control de versiones, asegurando un historial claro de commits.
- 6. **Manejo de archivos JSON/CSV (B)**
  - o Almacenar y recuperar datos utilizando archivos JSON y/o CSV para persistencia de información.
- 7. **Pruebas unitarias (B)**
  - o Desarrollar y ejecutar pruebas unitarias para verificar el correcto funcionamiento de las funciones implementadas.
- 8. **Expresiones regulares (A/B)**
  - o Utilizar expresiones regulares para validar y buscar patrones en los datos (ejemplo: correos electrónicos, números de teléfono, códigos, etc.).

**Importante:** Todos estos temas tendrán que ser incluidos en el programa. La ausencia de alguno de ellos reducirá significativamente la calificación. Los temas serán añadidos al trabajo a medida que sean abordados en clase (ver temas por entregas A/B).

### Criterios de Evaluación para el trabajo y la exposición

El **trabajo práctico** será evaluado en función de los siguientes aspectos fundamentales:

1. **Funcionalidad:** El sistema debe cumplir con todos los requisitos planteados en la consigna, asegurando la correcta implementación de las operaciones CRUD (Crear, Leer, Actualizar, Eliminar), búsqueda y estadística, según lo requerido.
2. **Calidad del Código:** Se valorará el uso de buenas prácticas de programación, incluyendo claridad, organización, comentarios adecuados y eficiencia en la implementación de las soluciones.
3. **Uso de Git:** Se deberá mantener un historial de commits que refleje el desarrollo progresivo del proyecto, evidenciando una correcta gestión de versiones y trabajo en equipo. Cada integrante deberá realizar aportes al repositorio y evidenciar su trabajo en los commits.
4. **Documentación:** Las funciones deberán estar debidamente documentadas a través del uso de docstrings.

5. **Informe:** Se evaluará la claridad y completitud del informe entregado, asegurando que describa adecuadamente el sistema, su funcionamiento, problemas encontrados, soluciones implementadas Backlog y breve detalle de los Sprints.
6. **Pruebas Unitarias:** El código debe incluir pruebas unitarias que validen el correcto funcionamiento de las funciones implementadas, con una cobertura adecuada y resultados documentados.
7. **Fidelidad a la consigna:** El desarrollo del proyecto debe alinearse con lo solicitado, asegurando que toda las funcionalidades y requisitos sean implementados correctamente. Importante: Todos los temas especificados en "Requisitos del Proyecto" deben estar presentes en la implementación. La ausencia de cualquiera de ellos impactará significativamente en la calificación.
8. **Prolijidad y Presentación:** Tanto el código como la interfaz de usuario deben ser organizados, legibles y de fácil comprensión, favoreciendo la usabilidad y el mantenimiento del sistema.

**Importante:** No se aceptarán trabajos que incluyan funciones, bibliotecas externas o herramientas de Python que no hayan sido tratadas en clase, excepto que exista autorización expresa del docente. Ante cualquier duda o interés en incorporar funcionalidades adicionales, consultar previamente.

El trabajo para exponer deberá ser el mismo que se envió el día de la fecha de entrega.

### Exposición del proyecto

La **exposición** del proyecto será evaluada considerando los siguientes aspectos:

1. **Participación de todos los integrantes / Claridad y estructura de la presentación**
  - **Todos** los miembros del grupo deben exponer una parte del trabajo y demostrar conocimiento sobre el proyecto completo.
  - La calificación individual dependerá tanto de la presentación grupal como del desempeño personal durante la exposición, así como de su capacidad para responder preguntas al finalizar la presentación. Se evaluará la claridad, seguridad, velocidad y asertividad en sus respuestas. Una demora significativa al responder o la necesidad frecuente de buscar fragmentos de código por desconocimiento afectará negativamente la evaluación individual. Por este motivo, integrantes de un mismo grupo pueden recibir calificaciones diferentes.
  - Si un integrante no participa en la defensa del trabajo, se lo considerará ausente y deberá recuperarlo en la fecha prevista, independientemente de su desempeño individual en la validación de conocimientos escrita (*ver Calificación*).
  - La exposición debe ser clara, organizada y concisa, evitando la lectura literal del código.
2. **Defensa del proyecto**
  - El código debe ejecutarse sin errores y deben demostrarse las principales funcionalidades. Se sugiere enfáticamente que el código sea probado previamente en la computadora que se va a utilizar para exponer.
  - Se debe explicar el enfoque de resolución, las técnicas utilizadas y los desafíos encontrados.
  - No es necesario leer línea por línea el código, sino describir su funcionamiento y lógica general.

3. **Capacidad de argumentación y resolución de problemas**

- Se evaluará la capacidad del estudiante para explicar y justificar sus decisiones técnicas.
- Se podrán realizar preguntas sobre posibles modificaciones o mejoras del código.

4. **Trabajo en equipo**

- Se considerará cómo se distribuyeron las responsabilidades dentro del grupo.
- Se valorará si hubo un trabajo equitativo y colaborativo en el desarrollo del proyecto.

**Criterios de Evaluación Individual en la Exposición**

Durante la exposición cada estudiante será evaluado según el siguiente esquema:

- **Cumple (A):** Demuestra una comprensión sólida y explica correctamente el proyecto.
- **Cumple parcialmente (B):** Tiene algunos conocimientos, pero su explicación es confusa, incompleta o presenta errores.
- **No cumple (C):** No demuestra comprensión del tema o no participa en la exposición.

Los aspectos a evaluar durante la exposición oral serán los siguientes:

- **Defensa del proyecto:** Capacidad del estudiante para explicar su código y justificar decisiones técnicas.
- **Resolución de problemas:** Habilidad para responder preguntas sobre posibles modificaciones o mejoras del código.
- **Trabajo en equipo:** Se evaluará el grado de participación de cada miembro, demostrado en la exposición y la forma en que se distribuyeron las responsabilidades.

**Ejemplos de Temas para el Proyecto**

Ver archivo *Ejemplos de temas TPO Final P1 - AyED 2C 2025.pdf*

**Formato y Fechas de Entrega**

Se entregará una carpeta comprimida con el nombre TPO\_GrupoX (reemplazar X por el número de grupo), que contendrá los siguientes archivos:

- Código fuente:** Archivos de Python (.py) organizados y debidamente documentados.
- Enlace al Repositorio Git:** Con historial de commits y ramas utilizadas.
- Informe (primera entrega):** Descripción del proyecto, estructura de menús, entidades involucradas, problemas encontrados y soluciones implementadas. Además, el informe deberá tener los datos del equipo y el Scrum Master y el desarrollo del Backlog y un breve detalle de los Sprints organizados por el equipo. Se compartirá un modelo de informe en las sucesivas clases.
- Pruebas unitarias (segunda entrega):** Casos de prueba y resultados de las pruebas ejecutadas, agregadas al informe.

**Fecha de entrega:** Tres días antes de cada fecha de exposición y defensa (primera y segunda entrega).

**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.

## Calificación

La calificación responde a una **evaluación integral** entre el desempeño individual y el desempeño grupal, representado a través del Trabajo Práctico y su defensa.

En cada instancia (correspondientes al primer / segundo parcial), la **calificación** de los estudiantes se conformará de la siguiente manera:

- **Evaluación individual:** La evaluación individual consta de dos partes:
  - Una validación escrita de conocimientos previos a la exposición, que evaluará los contenidos vistos hasta el momento de la entrega.
  - La participación en la exposición y defensa del trabajo, conforme a los criterios establecidos en *Criterios de Evaluación Individual en la Exposición*.

**Importante:** Para aprobar, el estudiante deberá obtener una calificación mínima de 4 en la validación escrita y una calificación A o B en la defensa del proyecto (ver *Criterios de Evaluación Individual en la Exposición*).
- **TP grupal:** Se considerarán los aspectos desarrollados en *Criterios de Evaluación para el trabajo y la exposición* y en *Requisitos del Proyecto*. Todos los miembros del grupo deberán participar en la defensa.

**Importante:** Para poder considerarse aprobado (4 o más), el estudiante deberá aprobar **ambas instancias** (individual y grupal).

**¡Éxitos en el desarrollo del proyecto!**

### Uso responsable de inteligencia artificial en el Proyecto

La inteligencia artificial, particularmente herramientas como ChatGPT, puede ser una excelente aliada para facilitar la comprensión de conceptos, aclarar dudas y agilizar el proceso de aprendizaje en la programación. Sin embargo, su uso indebido, limitado al simple copiado y pegado de soluciones automáticas, perjudica directamente el aprendizaje y la adquisición real de conocimientos necesarios para la carrera. Es **fundamental que cada integrante del grupo comprenda a fondo el código que presenta**, ya que la evaluación oral no solo revisará el funcionamiento técnico, sino especialmente la capacidad individual de explicar y defender las decisiones tomadas durante el desarrollo del proyecto. Recurrir a la IA como reemplazo del esfuerzo personal limitará significativamente su capacidad para afrontar preguntas, resolver problemas imprevistos, y justificar las soluciones implementadas durante la exposición oral.

Por ello, **se invita enfáticamente a utilizar estas herramientas de manera responsable y orientada al aprendizaje**: planteen preguntas concretas, soliciten ejemplos para entender cómo abordar los problemas, y profundicen en la lógica detrás de cada solución. Prioricen el proceso de razonamiento individual y grupal sobre la obtención rápida de respuestas. De este modo, no solo estarán mejor preparados para la evaluación oral, sino que también desarrollarán habilidades críticas y técnicas esenciales que perdurarán a lo largo de su carrera académica y profesional.

En caso de detectarse un uso indebido de herramientas de inteligencia artificial, evidenciado por la incapacidad del equipo para explicar o fundamentar adecuadamente las decisiones técnicas o el código presentado, **el trabajo práctico será automáticamente desaprobado**. Asimismo, se elevará un informe detallado a la Dirección Académica, que evaluará la posible aplicación de sanciones disciplinarias a los integrantes involucrados.