

Plantilla para Trabajos Integradores – Programación I

Datos Generales

- **Título del trabajo**: (Debe reflejar con claridad el tema central, por ejemplo: "Algoritmos de Búsqueda y Ordenamiento en Python")
- Alumnos: (Nombre completo Correo electrónico institucional)
- Materia: Programación I
- **Profesor/a**: (Nombre completo del docente)
- Fecha de Entrega: (Formato: día/mes/año)

Indice

- 1. Introducción
- 2. Marco Teórico
- 3. Caso Práctico
- 4. Metodología Utilizada
- 5. Resultados Obtenidos
- 6. Conclusiones
- 7. Bibliografía
- 8. Anexos

1. Introducción

- ★ En esta sección se debe presentar el tema del trabajo y su relevancia dentro del área de estudio. El texto debe responder:
 - ¿Por qué se eligió este tema?
 - ¿Qué importancia tiene en la programación?
 - ¿Qué objetivos se propone alcanzar con el desarrollo del trabajo?

Se recomienda un desarrollo claro, no muy extenso (entre 1/2 y 1 carilla), y escrito en tercera persona.

2. Marco Teórico



- Este apartado contiene la fundamentación conceptual del tema tratado. Debe incluir definiciones, clasificaciones, jerarquías, estructuras y sintaxis si corresponde.
 - Incluir gráficos, esquemas o tablas si son útiles para la comprensión.
 - Mencionar autores o documentación oficial si se toman definiciones o explicaciones.
 - Para temas técnicos, se recomienda describir cómo se implementa el concepto en el lenguaje de programación trabajado (ej. Python, Java, etc.).

3. Caso Práctico

Aquí se debe presentar un problema o situación concreta que haya sido desarrollada o simulada para aplicar el contenido del trabajo.

Incluye:

- Breve descripción del problema a resolver.
- Código fuente comentado.
- Capturas de pantalla si corresponde.
- Explicación de decisiones de diseño (por ejemplo: ¿por qué se eligió un método de ordenamiento sobre otro?).
- Validación del funcionamiento.

El código debe estar identado correctamente y con comentarios que expliquen los pasos importantes.

4. Metodología Utilizada

Describe los pasos seguidos durante el desarrollo del trabajo.

Se pueden incluir:

- Investigación previa (fuentes utilizadas).
- Etapas de diseño y prueba del código.
- Herramientas y recursos utilizados (IDE, librerías, control de versiones, etc.).
- Trabajo colaborativo (reparto de tareas si se realizó en grupo).

5. Resultados Obtenidos



ii Detalla qué se logró con el caso práctico, qué aspectos funcionaron correctamente y qué dificultades se presentaron.

Se pueden incluir:

- Casos de prueba realizados.
- · Errores corregidos.
- Evaluación de rendimiento (si aplica, por ejemplo, comparar el tiempo de ejecución entre algoritmos).
- Enlace a repositorio si el trabajo está subido a GitHub u otra plataforma.

6. Conclusiones

- **©** Reflexión final del grupo de trabajo. Aquí se recomienda incluir:
 - Qué se aprendió al hacer el trabajo.
 - Qué utilidad tiene el tema trabajado para la programación o para otros proyectos.
 - Posibles mejoras o extensiones futuras.
 - Dificultades que surgieron y cómo se resolvieron.

7. Bibliografía

Listado de fuentes consultadas, utilizando normas básicas APA u otro formato consistente.

Se sugiere incluir:

- Libros.
- Documentación oficial.
- Sitios web con fecha de acceso.
- Artículos o materiales recomendados por el docente.

Ejemplo:

- Python Software Foundation. (2024). Python 3 Documentation. https://docs.python.org/3/
- Sweigart, A. (2019). Automate the Boring Stuff with Python. No Starch Press.

8. Anexos



Material adicional que no va en el cuerpo principal del trabajo pero que aporta valor.

Pueden ser:

- Capturas del programa funcionando.
- Enlace al video explicativo.
- Código completo como archivo externo o adjunto.
- Cuadros comparativos.
- Documentos auxiliares (como diagramas de flujo).

👲 Recomendaciones para la presentación

- Formato del archivo: .docx o .pdf
- Tipografía: Arial o Calibri, tamaño 11 o 12
- Interlineado: 1,5
- Márgenes estándar (2.5 cm)
- Portada opcional con el logo de la institución (si se requiere)
- Entrega digital mediante plataforma institucional o por correo