utn-tp-integracion

Proyecto de Integración

Trabajo Integrador: Matemáticas y Programación en Python

Alumnos - Grupo N° 2

- **Agustin Emiliano Sotelo Carmelich**
- **Gabriel Valdez Arce**
- **Bruno Giuliano Vapore**
- **Daiana Judith Velasquez Torrez**
- **Jose Gabriel Torres**

Evaluación: Modalidad "Cumple / No Cumple" (La rúbrica se entregará en un documento aparte).
Consultas: Disponibles en ambas materias para resolver dudas.

---

1. Equipos y Organización

- Equipos de hasta **5 integrantes**.
- **Todos los miembros** deben participar activamente en la **explicación del proyecto**.

---

2. Selección y Desarrollo del Proyecto

- Elijan una de las actividades propuestas o propongan una alternativa relacionada a su interés.
- El proyecto debe utilizar **únicamente los conceptos de programación ya aprendidos**.
- Debe integrar **aspectos matemáticos** como álgebra de Boole, sistema binario, etc., y su **aplicación en Python**.

---

3. Consultas Sincrónicas

- Pueden participar en sesiones de consultas sincrónicas de **matemáticas y programación**.
- La **asistencia es fundamental** para el desarrollo correcto del proyecto.

---

4. Requerimientos Técnicos

Código en Python

- Crear un programa claro, bien documentado y funcional, que **resuelva un problema o simule un fenómeno matemático**.

Video Explicativo

- Grabar un video que muestre el funcionamiento del programa.
- Explicar el proceso de desarrollo.
- **Cada integrante debe explicar una parte del proyecto**.

Entrega

- Subir el video a **YouTube**.
- Entregar en la plataforma:
  - Enlace del video.
  - Breve descripción del proyecto.
  - Código fuente en Python.

---

5. Evaluación

La evaluación será **"Cumple / No Cumple"** de acuerdo a la rúbrica proporcionada.
Se evaluarán:

- Aplicación correcta de **conceptos matemáticos y de programación**.
- **Calidad del código**.
- **Claridad del video** explicativo.
- **Participación activa** en las consultas sincrónicas.

---

6. Uso de Inteligencia Artificial

Integración en el Proceso

- Usar herramientas de IA en todas las fases del proyecto: generación de ideas, análisis, refinamiento.

Iteraciones y Refinamiento

- Aplicar mejoras progresivas a través de iteraciones.
- Documentar **cada paso** del proceso.

Evidencia y Justificación

- Presentar evidencia del uso de IA (capturas, reportes, registros).
- Justificar cómo contribuyó a optimizar el proyecto.

> **Objetivo:** Aplicar de manera práctica lo aprendido, trabajar en equipo y comunicar ideas de forma clara y precisa.
> ¡Mucho éxito en el desarrollo del proyecto!

---

7. Propuestas de Proyectos

A continuación, algunas ideas que pueden elegir o adaptar para desarrollar el proyecto:

Simulación de Puertas Lógicas Básicas

- Programa en Python que simule las puertas **AND**, **OR** y **NOT**.
- Solicitar al usuario ingresar valores binarios (0 o 1).
- Mostrar el resultado de cada operación.
- **Extensión:** Agregar puertas **NAND**, **NOR**, **XOR** si lo desean.

Conversión de Números

- Convertir números **decimales a binarios**.
- Opcional: también **binario a decimal**.
- **Extensión:** Validar entradas y mostrar mensajes de error ante datos incorrectos.

Contador Binario

- Usar un ciclo para contar de 0 a 15.
- Mostrar cada número en su **representación binaria**.
- **Extensión:** Simular un circuito usando `time.sleep()` como retardo.

Generador de Tabla de Verdad

- Crear una tabla de verdad para una expresión booleana como **"A AND B"**.
- **Extensión:** Permitir al usuario elegir entre distintas operaciones lógicas.

Comparador de Expresiones Booleanas

- Permitir que el usuario ingrese **dos expresiones booleanas simples**.
- Comparar sus resultados evaluando todas las combinaciones posibles de valores.

Calculadora de Operaciones Bit a Bit

- Recibir dos números y aplicar operaciones **bit a bit (AND, OR, XOR)**.
- Mostrar resultados en **formato decimal y binario**.

Simulador de Sumador de 1 Bit

- Programar un **sumador de 1 bit** usando lógica booleana.
- Mostrar el **bit de suma** y el **carry** (acarreo).

Juego de Adivinanza en Binario

- Mostrar un número en binario y desafiar al usuario a adivinar su **equivalente decimal**, o viceversa.
- Refuerza la conversión entre ambos sistemas.

Simulador de Circuito Combinacional Básico

- Combinar varias puertas lógicas para resolver un problema simple.
- Ejemplo: determinar si un número binario es **par o impar** (basado en el último dígito).