Trabajo Integrador 2: Matemática y Programación

Objetivo

Profundizar la integración entre los contenidos de Matemática (conjuntos y lógica) y Programación (estructuras condicionales, repetitivas y funciones), fortaleciendo también el trabajo en equipo, la comunicación clara y la responsabilidad individual en proyectos colaborativos.

Trabajo en grupo

El trabajo debe hacerse en grupo y todos los integrantes deben pertenecer a la misma comisión.

La conformación de grupos tiene como objetivo fomentar la colaboración entre pares, una habilidad fundamental que todo programador debe desarrollar para integrarse eficazmente en proyectos de gran envergadura.

Cada integrante debe asumir responsabilidades específicas dentro del proyecto, explicar su parte en el video y entregar por escrito una descripción de las tareas que realizó.

Parte 1 – Desarrollo Matemático (Conjuntos y Lógica)

1. Cada integrante debe anotar su número de DNI.

2. A partir de los DNIs, se deben formar tantos conjuntos de dígitos únicos como integrantes tenga el grupo.

3. Realizar entre esos conjuntos las siguientes operaciones: unión, intersección, diferencia (entre pares) y diferencia simétrica.

4. Para cada una de estas operaciones, se debe realizar un diagrama de Venn (a mano o digital), que debe incluirse en la entrega.

5. Redactar al menos dos expresiones lógicas en lenguaje natural, que puedan luego implementarse en Python y escribir en la documentación que van a presentar cual seria el resultado con los conjuntos que tienen.

Ejemplos de expresiones lógicas:

- Si todos los conjuntos tienen al menos 5 elementos, entonces se considera que hay una alta diversidad numérica.·

- Si el conjunto A tiene más elementos que el conjunto B y el conjunto C contiene al menos un número impar, entonces se cumple la condición de combinación amplia.·
- Si ningún conjunto tiene el número 0, entonces se considera un grupo sin ceros.·
- Si algún dígito aparece en todos los conjuntos, se marca como dígito común.·
- Si hay más conjuntos con cantidad par de elementos que con cantidad impar, entonces se etiqueta como "grupo par".·
- Si la intersección entre todos los conjuntos tiene exactamente un elemento, se considera un dígito representativo del grupo.

Estas expresiones deben incluirse en el archivo PDF de la parte teórica y se espera que al menos una de ellas se implemente directamente como lógica en el programa Python.

Parte 2 – Desarrollo del Programa en Python

El programa debe implementar varias de las ideas trabajadas en papel. Debe incluir:

A. Operaciones con DNIs

· Ingreso de los DNIs (reales o ficticios).

· Generación automática de los conjuntos de dígitos únicos.

· Cálculo y visualización de: unión, intersección, diferencias y diferencia simétrica.

· Conteo de frecuencia de cada dígito en cada DNI utilizando estructuras repetitivas.

· Suma total de los dígitos de cada DNI.

· Evaluación de condiciones lógicas (condicionales), vinculadas con las expresiones escritas.

Ejemplos:

· Si un dígito aparece en todos los conjuntos, mostrar "Dígito compartido".

· Si algún conjunto tiene más de 6 elementos, mostrar "Diversidad numérica alta".

B. Operaciones con años de nacimiento

· Ingreso de los años de nacimiento (Si dos o mas integrantes del grupo tienen el mismo año, ingresar algún dato ficticio, según el caso).

· Contar cuántos nacieron en años pares e impares utilizando estructuras repetitivas.

· Si todos nacieron después del 2000, mostrar "Grupo Z".

· Si alguno nació en año bisiesto, mostrar "Tenemos un año especial".

· Implementar una función para determinar si un año es bisiesto.

· Calcular el producto cartesiano entre el conjunto de años y el conjunto de edades actuales.

Parte 3 – Video de Presentación

Duración estimada entre 5 y 10 minutos. Todos los integrantes deben presentarse en cámara, mostrar el programa funcionando y explicar la parte que realizaron. También deben comentar brevemente qué aprendieron al combinar matemática y programación.

Entrega final

1. Archivo PDF con: desarrollo de conjuntos y operaciones, todos los diagramas de Venn, expresiones lógicas redactadas, y tareas de cada integrante explicadas por escrito.
2. Archivo con extensión .py que contenga el programa en Python.
3. Video grupal subido en lo posible a YouTube.
4. Documento adicional con los nombres de los integrantes, descripción de lo que hizo cada uno y la relación entre las expresiones lógicas escritas y el código implementado.

Calificación

La calificación será numérica y para aprobar deberá ser mayor o igual a **6.**

E J E M P L O

Supongamos que en el grupo hay 3 integrantes, con los siguientes DNIs:

- Persona A: 23456789
- Persona B: 22334455
- Persona C: 98765432

De cada DNI se toman los dígitos únicos y se arma un conjunto (sin repetir elementos).

- Conjunto A = {2, 3, 4, 5, 6, 7, 8, 9}
- Conjunto B = {2, 3, 4, 5}
- Conjunto C = {2, 3, 4, 5, 6, 7, 8, 9}

Ejemplo A y B:

- Unión (A ∪ B): todos los elementos de A y B sin repetir → {2, 3, 4, 5, 6, 7, 8, 9}

- Intersección (A ∩ B): elementos que están en ambos → {2, 3, 4, 5} ● Diferencia (A - B): elementos de A que no están en B → {6, 7, 8, 9}

- Diferencia simétrica (A Δ B): elementos que están en A o en B pero no en ambos → {6, 7, 8, 9} (En este caso, B no tiene ningún dígito que A no tenga, así que la diferencia simétrica es igual a la diferencia A - B.)

Diagrama de Venn: Hay herramientas online como https://venngage.com o pueden hacerlo en PowerPoint, Paint o a mano en papel.

Expresiones lógicas en lenguaje natural Estas son frases que expresan condiciones que se pueden traducir a código Python.

Ejemplo 1:Lenguaje natural: “Los dígitos que están en A y no están en B”
Resultado con los conjuntos del ejemplo: {6, 7, 8, 9}

Ejemplo 2: Lenguaje natural: “Los dígitos que están en A o en B, pero no en ambos”
Resultado: {6, 7, 8, 9}

Ejemplo 3: “Los dígitos que aparecen en los tres DNIs a la vez”
Resultado: {2, 3, 4, 5}

-

Entrega semana de integración 2Tarea

Por hacer: Hacer un envío**Por hacer:** Recibir una calificación**Por hacer:** Recibir una calificación de aprobado

Apertura: jueves, 15 de mayo de 2025, 00:00

Cierre: viernes, 13 de junio de 2025, 23:59

- Marcar como hecha

Rúbrica de Evaluación: Trabajo Integrador 2: Matemática y Programación

Criterio de Evaluación	**Insuficiente (1-5)**	**Suficiente (6-7)**	**Bueno (8-9)**	**Excelente (10)**	**Ponderación**
Parte 1: Desarrollo Matemático	Presenta errores conceptuales significativos en conjuntos, lógica u operaciones. Diagramas ausentes o incorrectos. Expresiones lógicas poco claras o ausentes.	Comprensión básica de conjuntos y lógica, con algunos errores menores. Diagramas presentes pero con imprecisiones. Expresiones lógicas	Correcta aplicación de conceptos de conjuntos y lógica. Diagramas claros y precisos. Expresiones lógicas bien formuladas.	Dominio de conceptos de conjuntos y lógica. Diagramas impecables. Expresiones lógicas creativas y relevantes.	40%

		rudimentarias.			
Parte 2: Desarrollo en Python	El programa no funciona o presenta errores graves. Faltan implementaciones clave (operaciones, lógicas, años, función bisiesto). Código desorganizado.	El programa funciona parcialmente, con errores. Algunas implementaciones faltan o son incorrectas. Código con poca estructura.	El programa funciona correctamente, implementa la mayoría de los requisitos. Código funcional y con cierta organización.	El programa es robusto, implementa todos los requisitos de forma eficiente. Código bien estructurado, modular y legible.	30%
Parte 3: Video de Presentación	Video ausente o muy por debajo de la duración/calidad esperada. Poca o nula participación de integrantes. Explicaciones confusas.	Video presente, cumple mínimamente con duración. Participación limitada o desigual. Explicaciones básicas de su parte.	Video cumple con duración y calidad razonable. Participación equitativa. Explicaciones claras de su parte.	Video cumple con duración y calidad excelente. Todos los integrantes participan activamente y explican su parte con solvencia. Reflexión sobre la integración.	15%
Entrega Final y Documentación	Entrega incompleta o desorganizada. PDF, .py,	Entrega casi completa, con algunas faltas	Entrega completa y organizada. Documenta	Entrega completa, organizada y	15%

	video o documento adicional faltantes. Descripción de tareas pobre o ausente. Relación lógica-código no clara.	menores en archivos o formato. Descripción de tareas básica. Relación lógica-código mencionada.	ción clara (PDF, tareas, relación lógica-código).	profesional. Documentación detallada, clara y precisa. Excelente descripción de tareas y relación lógica-código bien justificada.	
Trabajo en Equipo y Responsabilidad	No se evidencia trabajo en equipo. No hay claridad sobre la responsabilidad individual.	Trabajo en equipo limitado o con conflictos. Responsabilidad individual poco definida.	Se evidencia trabajo en equipo. Responsabilidad individual identificable.	Excelente colaboración y distribución de tareas. Clara evidencia de responsabilidad individual y apoyo mutuo.	Se tendrá en cuenta en todos los puntos anteriores.

Calificación Final:

- Para cada criterio se asignará un puntaje de 1 a 10 según el nivel alcanzado.
- La calificación final se obtiene ponderando los puntajes con los porcentajes indicados.
- Se aprueba con un promedio ponderado mayor o igual a **6.00**. Se redondea al número entero más próximo.

Notas Adicionales para el Evaluador:

- Observar la claridad de las explicaciones en el video.

- Evaluar la originalidad de las expresiones lógicas propuestas.
- Revisar la limpieza y comentarios del código Python.
- Considerar la prolijidad de los diagramas de Venn.
- Verificar que las tareas descritas por escrito coincidan con lo presentado en el video y el código.