**1. Comparación de Representaciones**

En esta sección, compararemos las tres representaciones cromosómicas utilizadas en el algoritmo genético: **representación binaria**, **representación real**, y **representación permutacional**.

* **Representación Binaria**:
  + **Descripción**: En esta representación, cada alumno es asignado a uno de los tres exámenes (A, B, C) usando una codificación binaria. Cada alumno se representa con un vector de 3 bits, donde solo uno de los bits está activo (valor 1) para indicar a qué examen está asignado.
  + **Ventajas**: Es fácil de implementar y tiene una representación compacta. Ideal para problemas de **selección** y **asignación discreta**.
  + **Desventajas**: Dado que solo se puede asignar un examen con un bit activo, es más difícil de manejar cuando el número de exámenes aumenta, como lo hicimos en la **actividad 6** al pasar de 3 a 4 exámenes.
* **Representación Real**:
  + **Descripción**: En esta representación, cada alumno tiene 3 valores reales (uno para cada examen). Los valores son probabilidades que suman 1.0, y cada valor indica la probabilidad de asignar al alumno a un examen determinado.
  + **Ventajas**: Ofrece una mayor flexibilidad en la asignación de los alumnos a los exámenes, ya que los valores no están restringidos a ser enteros. Es ideal para **optimización continua**.
  + **Desventajas**: Aunque es útil en optimización continua, puede ser más difícil de implementar en problemas discretos y puede requerir un ajuste fino de los parámetros de mutación y cruce.
* **Representación Permutacional**:
  + **Descripción**: En esta representación, cada cromosoma es una permutación de 39 índices (uno para cada alumno). La asignación a los exámenes se realiza dividiendo la permutación en tres segmentos: los primeros 13 índices corresponden a examen A, los siguientes 13 a examen B, y los últimos 13 a examen C.
  + **Ventajas**: Es excelente para **problemas de ordenamiento y secuenciación**, donde el orden de los elementos es crucial. Además, permite un reparto equitativo de los alumnos en los exámenes.
  + **Desventajas**: Aunque es útil para algunos problemas de optimización, no siempre es la mejor opción cuando las restricciones o las asignaciones no son estrictamente secuenciales.

**2. Resultados de las Actividades Realizadas**

A continuación, se describen los resultados obtenidos en cada actividad del proyecto:

* **Actividad 1: Análisis Comparativo**:
  + Se ejecutaron los algoritmos para las tres representaciones y se compararon los resultados en términos de equilibrio y velocidad de convergencia.
  + **Conclusión**: Todas las representaciones lograron un buen equilibrio en la asignación de alumnos a los exámenes, con desviaciones estándar bajas entre los promedios. Sin embargo, la representación permutacional fue la que más rápido convergió, mientras que la representación binaria y real tardaron más en estabilizarse.
* **Actividad 2: Modificación de Fitness**:
  + Se modificó la función de fitness de la representación binaria para penalizar los grupos con alta varianza de notas y premiar la diversidad en la asignación de alumnos.
  + **Conclusión**: La modificación de la función de fitness mostró que los grupos con alta varianza eran penalizados, pero el cambio no tuvo un gran impacto en el equilibrio de la distribución final. La diversidad entre los grupos mejoró ligeramente, pero no hubo grandes diferencias en los resultados.
* **Actividad 3: Nuevo Operador Genético**:
  + Se implementó un operador de **mutación gaussiana** en la representación real, y se probaron tres valores de sigma (0.05, 0.1, y 0.2).
  + **Conclusión**: La mutación gaussiana permitió ajustar las probabilidades de asignación a los exámenes, y el valor de sigma afectó la rapidez de convergencia. Con un sigma más alto, la convergencia fue más rápida, pero con mayor variabilidad en los resultados.
* **Actividad 4: Restricciones Adicionales**:
  + Se añadió una restricción en la representación permutacional para evitar que todos los alumnos con notas menores a 11 estuvieran asignados al mismo examen.
  + **Conclusión**: La restricción fue correctamente implementada y penalizó las soluciones que la violaban. La distribución de los alumnos en los exámenes se mantuvo equilibrada, y el algoritmo funcionó correctamente con la restricción añadida.
* **Actividad 5: Visualización (Avanzado)**:
  + Se crearon gráficos para visualizar la evolución del fitness, la distribución de las notas y las comparaciones entre las representaciones.
  + **Conclusión**: Las visualizaciones proporcionaron una forma clara de comparar las representaciones en términos de convergencia del fitness y cómo las notas de los alumnos se distribuyen en los exámenes para cada representación.
* **Actividad 6: Problema Extendido**:
  + Se modificó la representación binaria para distribuir los alumnos en 4 exámenes en lugar de 3.
  + **Conclusión**: La adición de un examen aumentó la complejidad del algoritmo, lo que resultó en una convergencia más lenta debido al mayor número de combinaciones posibles. Esto hizo que la optimización tardara más tiempo en estabilizarse, pero el algoritmo todavía pudo encontrar una solución aceptable.

**3. Conclusiones sobre Cuándo Usar Cada Representación**

1. **Representación Binaria**:
   * **Ideal para**: Problemas discretos de asignación donde el número de opciones es pequeño y la distribución de los valores es binaria.
   * **Cuándo usarla**: Usarla cuando se tiene un número limitado de opciones (como en el caso de los 3 exámenes) y cuando se necesita una representación compacta.
   * **Desventaja**: La adición de más exámenes (como en la actividad 6) puede complicar el algoritmo, ya que el número de bits por alumno aumenta rápidamente.
2. **Representación Real**:
   * **Ideal para**: Problemas de optimización continua, donde los valores reales representan probabilidades o ponderaciones.
   * **Cuándo usarla**: Usarla cuando se necesita una flexibilidad en la asignación (por ejemplo, asignaciones probabilísticas) y cuando se está trabajando con optimización continua.
   * **Desventaja**: Puede ser más compleja de implementar en problemas discretos, ya que los valores no son estrictamente enteros.
3. **Representación Permutacional**:
   * **Ideal para**: Problemas de ordenamiento y secuenciación, donde el orden de los elementos es crucial.
   * **Cuándo usarla**: Usarla cuando se requiere un orden específico o secuencial en la asignación (como en el caso de los exámenes, donde cada alumno debe ser asignado a uno de los exámenes en un orden determinado).
   * **Desventaja**: No es tan flexible como las otras representaciones cuando se trata de asignaciones más dinámicas.

A través de este proyecto, se pudo explorar cómo diferentes representaciones cromosómicas afectan la asignación de alumnos a los exámenes y cómo la convergencia del algoritmo genético se ve afectada por cambios en la estructura del cromosoma y el número de exámenes. Cada representación tiene sus propias ventajas y limitaciones, y la elección de la mejor representación depende del tipo de problema que se esté resolviendo. La representación binaria es simple y eficiente para problemas discretos, la representación real es más flexible pero puede ser compleja de manejar en problemas discretos, y la representación permutacional es ideal para problemas de ordenamiento y secuenciación.