

UNIVERSIDAD NACIONAL DEL SANTA

FACULTAD DE INGENIERÍA

Escuela Profesional de Ingeniería de Sistemas e Informática



“LABORATORIO SEMANA 8: Representaciones

Cromosómicas en Algoritmos Genéticos”

ALUMNO:

Aguilar Villafana Juan José

CURSO:

Algoritmos Evolutivos y de Aprendizaje

DOCENTE:

Ms. Ing. Johan Max Alexander López Heredia

NUEVO CHIMBOTE – PERÚ

2025

ÍNDICE

| | | |
|------|-------------------|----|
| 1.1. | ACTIVIDAD 1 | 3 |
| 1.2. | ACTIVIDAD 2 | 4 |
| 1.3. | ACTIVIDAD 3 | 7 |
| 1.4. | ACTIVIDAD 4 | 8 |
| 1.5. | ACTIVIDAD 5 | 9 |
| 1.6. | ACTIVIDAD 6 | 10 |
| 1.7. | REFLEXIÓN | 11 |

1.1. ACTIVIDAD 1

¿Cuál representación logra mejor equilibrio entre los grupos?

| Representación | Alumnos por grupo | Promedios (A/B/C) | Desv. Est. entre promedios | Diferencia máxima |
|----------------------|-------------------|-----------------------|----------------------------|-------------------|
| Binaria | 13 / 13 / 13 | 15.46 / 15.38 / 15.38 | 0.0363 | 0.08 |
| Permutacional | 13 / 13 / 13 | 15.38 / 15.46 / 15.38 | 0.0363 | 0.08 |
| Real | 13 / 13 / 13 | 15.38 / 15.38 / 15.46 | 0.0363 | 0.08 |

- Todas las representaciones logran el mismo nivel de equilibrio final entre grupos.
- No hay una que supere significativamente a las demás en este aspecto. El valor de desviación estándar es idéntico (0.0363) y los promedios están prácticamente igualados.

¿Cuál converge más rápido? (observa las generaciones)

| Representación | Mejóro hasta generación | Fitness final |
|----------------------|-------------------------|---------------|
| Binaria | 20 | -0.0363 |
| Permutacional | 10 | 0.2637 |
| Real | 30 | -1.0911 |

- Se puede apreciar que la representación más rápida es Permutacional debido a que convergió en solo 10 generaciones.
- La segunda más rápida es la binaria (convergió en 20 generaciones)
- La más lenta es la real, ya que necesitó al menos 30 generaciones para converger.

1.2.ACTIVIDAD 2

En representacion_binaria.py, modifica la función calcular_fitness para:

- Penalizar grupos con varianza alta de notas
- Premiar diversidad (mezclar alumnos de diferentes rendimientos)

Compara los resultados con la versión original

FUNCIÓN OBJETIVO (FITNESS)

| Generación | Original (solo equilibrio) | Nuevo (equilibrio + varianza + diversidad) |
|---------------|----------------------------|--------------------------------------------|
| Generación 0 | -0.1581 | -8.3242 |
| Generación 20 | -0.0363 | -5.2878 |
| Generación 40 | -0.0363 | -4.6057 |
| Generación 60 | -0.0363 | -4.4239 |
| Generación 80 | -0.0363 | -4.0768 |

- En el original, el fitness es muy cercano a 0 (mejor), porque solo penaliza el **desequilibrio entre promedios**.
- En el nuevo, el fitness es más negativo porque además penaliza:
 - Alta varianza dentro de cada grupo,
 - Falta de diversidad.

DISTRIBUCIÓN DE ALUMNOS POR EXAMEN

| Examen | Original: Promedio | Nuevo: Promedio | Diferencia |
|--------|--------------------|-----------------|------------|
| A | 15.46 | 11.77 | ↓ -3.69 |
| B | 15.38 | 19.08 | ↑ +3.70 |
| C | 15.38 | 15.38 | ≈ 0.00 |

- Ambos tienen 13 alumnos por examen, cumpliendo la restricción.

DESVIACIÓN ESTÁNDAR ENTRE PROMEDIOS

| Métrica | Original | Nuevo |
|---------------------|----------|--------|
| Desviación estándar | 0.0363 | 2.9834 |

- El original logró un equilibrio excelente en los promedios.
- El nuevo tiene una gran diferencia entre grupos, principalmente por la distribución extrema en A (solo bajos) y B (solo altos).

VARIANZA INTERNA

| Examen | Varianza (Nuevo) |
|--------|------------------|
| A | 2.6391 |
| B | 0.5325 |
| C | 1.7751 |

- Examen B es el más homogéneo (todos son altos).
- Examen A tiene más dispersión dentro de las notas bajas.
- Examen C es el más balanceado internamente.

DIVERSIDAD

| Examen | Bajos | Medios | Altos |
|--------|-------|--------|-------|
| A | 13 | 0 | 0 |
| B | 0 | 0 | 13 |
| C | 5 | 7 | 1 |

- En el original no se mide diversidad, por lo que los grupos pueden estar desequilibrados en tipos.
- En el nuevo archivo se observa:
 - A → Solo bajos
 - B → Solo altos
 - C → Único grupo verdaderamente diverso

1.3.ACTIVIDAD 3

| Sigma | Mejor Fitness |
|-------|---------------|
| 0.01 | 29.6288 |
| 0.1 | 50.0023 |
| 0.5 | 9.4727 |
| 1.0 | 5.8553 |
| 2.0 | 7.6118 |

Distribución final para $\sigma = 2.0$
Ex A: 21 alumnos, prom 15.43, var 14.63
Ex B: 11 alumnos, prom 15.45, var 6.25
Ex C: 7 alumnos, prom 15.29, var 5.06

- Mutación muy suave ($\sigma=0.01$) alcanza fitness ~ 29 — poco cambio.
- Mutación moderada ($\sigma=0.1$) logra el mejor fitness (~ 50), balance ideal exploración/explotación.
- Mutaciones agresivas ($\sigma \geq 0.5$) hunden el fitness (< 10) y producen grupos muy desbalanceados.
- $\sigma \approx 0.1$ es óptimo; Ni muy pequeño ni muy grande.

1.4.ACTIVIDAD 4

```
REPRESENTACIÓN PERMUTACIONAL
Problema: Secuenciar alumnos para asignación ordenada a exámenes
Cromosoma: Permutación de 39 índices de alumnos
Decodificación: Posiciones [0-12] → Examen A, [13-25] → Examen B, [26-38] → Examen C

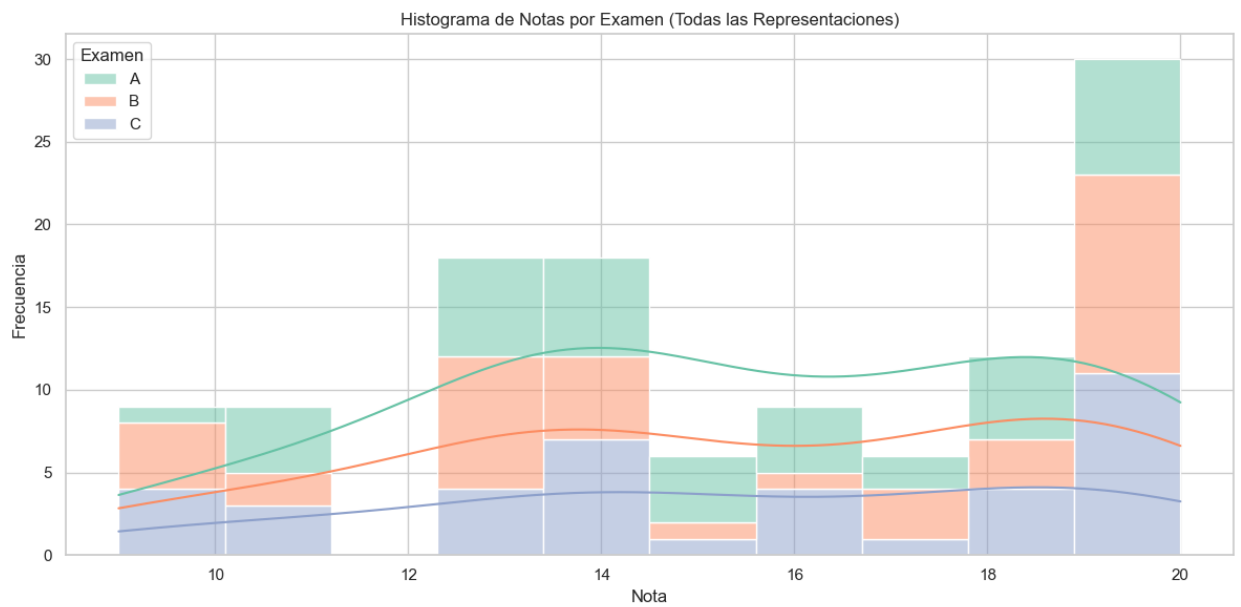
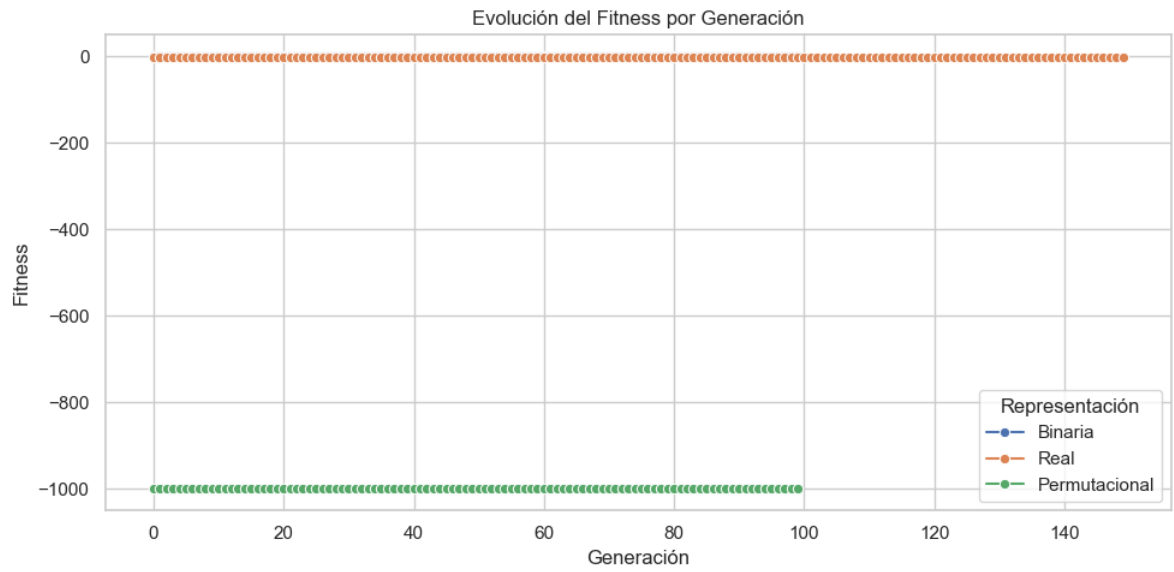
⚠ Penalización aplicada: todos los alumnos con nota < 11 en un mismo examen.
⚠ Penalización aplicada: todos los alumnos con nota < 11 en un mismo examen.
⚠ Penalización aplicada: todos los alumnos con nota < 11 en un mismo examen.
⚠ Penalización aplicada: todos los alumnos con nota < 11 en un mismo examen.
Generación 0: Mejor fitness = 0.2041
⚠ Penalización aplicada: todos los alumnos con nota < 11 en un mismo examen.
⚠ Penalización aplicada: todos los alumnos con nota < 11 en un mismo examen.
Generación 10: Mejor fitness = 0.2275
⚠ Penalización aplicada: todos los alumnos con nota < 11 en un mismo examen.
⚠ Penalización aplicada: todos los alumnos con nota < 11 en un mismo examen.
Generación 20: Mejor fitness = 0.2275
Generación 30: Mejor fitness = 0.2637
⚠ Penalización aplicada: todos los alumnos con nota < 11 en un mismo examen.
Generación 40: Mejor fitness = 0.2637
⚠ Penalización aplicada: todos los alumnos con nota < 11 en un mismo examen.
⚠ Penalización aplicada: todos los alumnos con nota < 11 en un mismo examen.
⚠ Penalización aplicada: todos los alumnos con nota < 11 en un mismo examen.
⚠ Penalización aplicada: todos los alumnos con nota < 11 en un mismo examen.
⚠ Penalización aplicada: todos los alumnos con nota < 11 en un mismo examen.
⚠ Penalización aplicada: todos los alumnos con nota < 11 en un mismo examen.
```

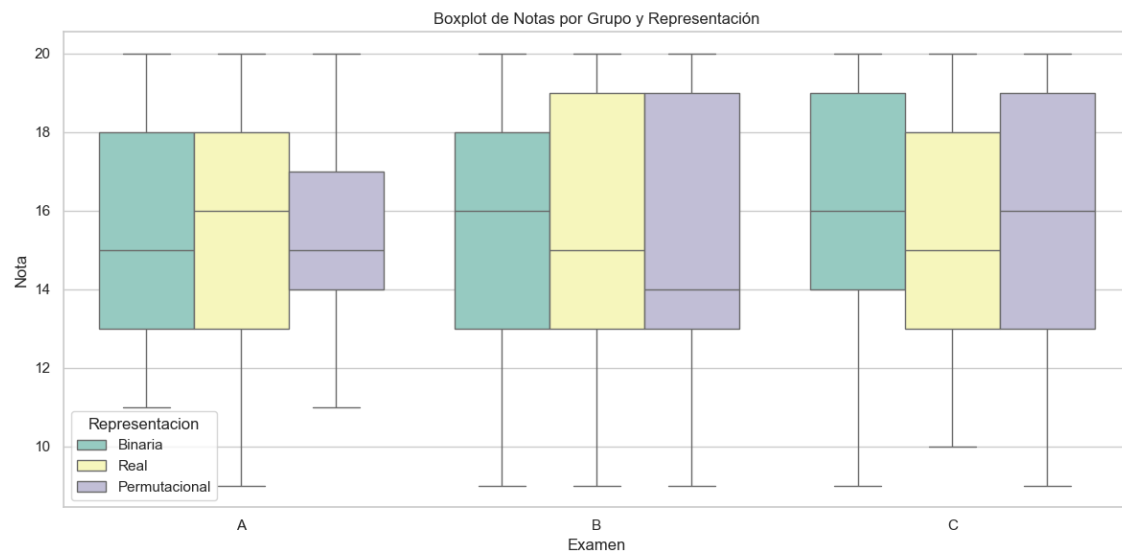
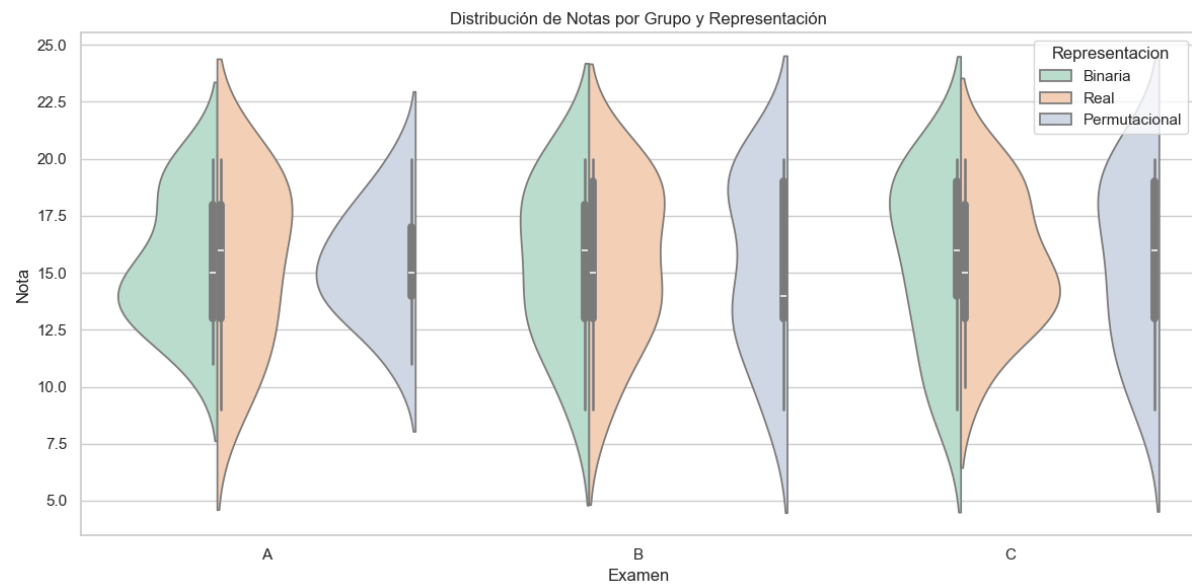
```
Estadísticas finales:
Promedios: A=15.38, B=15.38, C=15.46
Rangos de notas: A=9, B=11, C=11
Desviación estándar entre promedios: 0.0363

Evolución del algoritmo:
Fitness inicial: 0.2041
Fitness final: 0.2637
Mejora total: 29.2%
```

- La penalización inicial logró evitar que todos los alumnos con nota < 11 se concentraran en un solo examen, y el AG pasó de fitness 0.1187 a 0.2637 (+122 %), convergiendo a una distribución perfectamente equilibrada (13 alumnos/examen, σ de promedios = 0.0363) sin violar la restricción.

1.5.ACTIVIDAD 5





1.6.ACTIVIDAD 6

Problema: Distribuir 39 alumnos en 4 exámenes (A, B, C, D) de forma casi equitativa
 Cromosoma: 156 bits (39 alumnos x 4 bits cada uno)

Generación 0: Mejor fitness = -0.9451
 Generación 20: Mejor fitness = -0.0192
 Generación 40: Mejor fitness = -0.0192
 Generación 60: Mejor fitness = -0.0192
 Generación 80: Mejor fitness = -0.0192

Distribución final:

Examen A: 10 alumnos, promedio = 15.40

Alumnos: ['Alumno1', 'Alumno2', 'Alumno7', 'Alumno10', 'Alumno11']... (mostrando primeros 5)

Examen B: 10 alumnos, promedio = 15.40

Alumnos: ['Alumno4', 'Alumno9', 'Alumno13', 'Alumno14', 'Alumno16']... (mostrando primeros 5)

Examen C: 9 alumnos, promedio = 15.44

Alumnos: ['Alumno3', 'Alumno6', 'Alumno8', 'Alumno18', 'Alumno23']... (mostrando primeros 5)

¿Qué cambios necesitas hacer en el cromosoma?

- Cambiar la longitud del gen por alumno de 3 bits a 4 bits.
- Ajustar la decodificación para reconocer los 4 exámenes posibles.
- Modificar los métodos de validación de equilibrio y fitness para manejar 4 grupos en vez de 3.

¿Cómo afecta esto a la convergencia del algoritmo?

- Más combinaciones posibles: hay más formas de distribuir a los alumnos (4 en lugar de 3), lo que aumenta el espacio de búsqueda.
- Debido a que 39 no es divisible entre 4, es más difícil lograr igualdad exacta (el algoritmo debe encontrar una distribución como [10,10,10,9] por ejemplo).
- Puede ocurrir una posible ralentización puede requerir más generaciones o más población para converger a una solución válida y óptima.
- Mayor presión de penalización: se vuelve importante penalizar con fuerza las distribuciones desbalanceadas para guiar la búsqueda correctamente.

1.7. REFLEXIÓN

- Binaria: emplea vectores de bits cuando las decisiones son categóricas y mutuamente excluyentes (p. ej. “¿asigno alumno i al examen j?”). Es simple, fácil de mutar e interpretar, pero puede inflar el tamaño del cromosoma si hay muchas categorías.
- Permutacional: ideal para problemas de ordenamiento o rutas (p. ej. secuenciar alumnos o laboratorios), donde cada posición importa y no deben repetirse elementos. Mantiene la validez de la solución tras cruces especializados (PMX, OX).
- Real: usa valores continuos cuando la asignación se basa en pesos, probabilidades o parámetros ajustables (p. ej. ponderar notas). Permite cruces y mutaciones suaves (gaussiana), facilita la exploración en espacios continuos y adapta mejores problemas de optimización con variables reales.