

## Rúbrica de evaluación

### Actividad 2.8 – Procesando datos binarios con bucles (Java)

 Evaluación por ---

marp: true

title: Rúbrica – Actividad 2.8: Programa con nodos

theme: default

\_class: lead

paginate: true




backgroundColor: #ffffff

## Rúbrica de Evaluación




## Criterios generales

| Criterio                 | Peso aproximado |
|--------------------------|-----------------|
| Funcionalidad principal  | 30%             |
| Uso de listas enlazadas  | 20%             |
| Modularidad y POO        | 15%             |
| Presentación estructural | 15%             |
| Opcionales / creatividad | 10%             |
| Estilo y documentación   | 10%             |




## Funcionalidad básica

| Nivel   | Descripción   |
|---|---|
|  Bajo  | El programa no compila o tiene funciones incompletas    |
|  Medio | Funciona parcialmente: inserta pero no busca o elimina  |
|  Alto  | Inserta, busca, elimina y lista canciones correctamente |

## Uso de listas enlazadas




| Nivel   | Descripción  |
|---|--|
|  Bajo  | No se usan estructuras enlazadas, o están mal implementadas          |
|  Medio | Listas enlazadas por género, pero con errores o duplicación          |
|  Alto  | Estructura por género clara y correcta, con enlaces bien gestionados |

## Diseño modular y POO




| Nivel   | Descripción   |
|---|---|
|  Bajo  | Todo el código está en <code>main</code> o en una sola clase            |
|  Medio | Uso básico de clases ( <code>Cancion</code> , <code>Nodo</code> , etc.) |
|  Alto  | Clases bien definidas, separación lógica y métodos reutilizables        |






## Visualización estructural

| Nivel   | Descripción   |
|---|---|
|  Bajo  | No muestra estructura ni cambios en consola                       |
|  Medio | Muestra parcial, sin claridad estructural                         |
|  Alto  | Representación clara de los cambios en listas tras cada operación |

## Ampliaciones (opcionales)

| Nivel   | Descripción  |
|---|--|
|  Bajo  | No se incluye ningún añadido                               |
|  Medio | Añade ordenación, estadísticas o menú                      |
|  Alto  | Interfaz gráfica, múltiples mejoras o interacción avanzada |

## Estilo y documentación

| Nivel   | Descripción  |
|---|--|
|  Bajo  | Código sin comentarios ni estructura                   |
|  Medio | Comentarios básicos y organización mínima              |
|  Alto  | Comentarios claros, nombres adecuados y código legible |



## Evaluación final




✓ Debes demostrar que comprendes:

- El uso de **estructuras dinámicas**
- El diseño de **programas modulares en Java**
- Cómo manipular nodos y listas en memoria

 ¡Crea una biblioteca de canciones viva, organizada y funcional!




criterios del programa desarrollado por el alumnado.

## **12 34** Criterio 1: Conversión numérica

| Nivel  | Descripción  |
|--|--|
|  Bajo     | Solo convierte a una base o usa métodos automáticos                          |
| <br>Medio | Convierte a varias bases, pero con errores menores                           |
|  Alto    | Conversión manual a binario, octal y hexadecimal usando bucles correctamente |






## Criterio 2: Codificación ASCII




| Nivel   | Descripción   |
|---|---|
|  Bajo  | Codificación incorrecta o incompleta                                  |
|  Medio | Codificación correcta pero sin ajustar a 8 bits                       |
|  Alto  | Codificación binaria de cada carácter correctamente alineada a 8 bits |






## Criterio 3: Bit de paridad

| Nivel   | Descripción   |
|---|---|
|  Bajo  | No se implementa o el resultado es incorrecto               |
|  Medio | Lógica correcta pero sin justificar o explicar el resultado |
|  Alto  | Función correcta, bien integrada y documentada              |




## Criterio 4: Hash XOR

| Nivel   | Descripción   |
|---|---|
|  Bajo  | No implementado o sin coherencia  |
|  Medio | Funciona, pero sin validación clara del resultado                       |
|  Alto  | XOR secuencial bien aplicado y explicado como verificador de integridad |

## Criterio 5: Uso de bucles y estructuras

| Nivel   | Descripción   |
|---|---|
|  Bajo  | Bucles ausentes o mal utilizados  |
|  Medio | Bucles presentes, pero poco optimizados   |
|  Alto  | Uso adecuado de <code>for</code> / <code>while</code> según la necesidad lógica |




## Criterio 6: Modularidad del código

| Nivel   | Descripción  |
|---|--|
|  Bajo  | Todo el código en <code>main</code> , sin funciones                  |
|  Medio | Algunas funciones separadas, pero poco reutilizables                 |
|  Alto  | Funciones bien organizadas, reutilizables y con nombres descriptivos |









## Criterio 7: Presentación de resultados

| Nivel   | Descripción  |
|---|--|
|  Bajo  | Salidas confusas o escasa legibilidad                              |
|  Medio | Salidas funcionales pero poco estructuradas                        |
|  Alto  | Salidas claras, alineadas, con información relevante por cada paso |

## Criterio 8: Ampliaciones y creatividad (opcional)

| Nivel   | Descripción  |
|---|--|
|  Bajo  | Solo los requisitos mínimos  |
|  Medio | Añade una mejora: menú, entrada en bucle, validación de datos                |
|  Alto  | Ampliación significativa: gestión de errores, simulación de fallos, interfaz |

## Evaluación final

- ✓ Se valorará el uso correcto de estructuras de control
  - ✓ La claridad en el diseño y modularidad del código
  - ✓ La comprensión del tratamiento binario aplicado
  - ✓ El uso de buenas prácticas de codificación en Java
-  ¡Entrega un simulador funcional y bien estructurado!