


Rúbrica de Evaluación




Actividad 2.8: Procesando datos binarios con bucles

 Evaluación del trabajo en Java basado en conversión numérica, codificación binaria, paridad y verificación con hash.

Criterios Generales




Criterio	Peso aproximado
Corrección técnica	40%
Uso de estructuras (bucles)	20%
Modularidad y organización	15%
Presentación de resultados	15%
Creatividad / ampliación	10%

Conversión decimal → binario, octal y hexadecimal

Nivel	Descripción
 Bajo	Conversión incorrecta o usa funciones automáticas
 Medio	Conversión correcta solo para binario
 Alto	Conversión manual correcta a las tres bases






Codificación de texto carácter a carácter en binario




Nivel	Descripción
 Bajo	Codificación incompleta o errónea
 Medio	Codificación válida pero no ajustada a 8 bits
 Alto	Codificación precisa, 8 bits por carácter, texto completo






Cálculo del bit de paridad

Nivel	Descripción
 Bajo	No se calcula o el resultado es incorrecto
 Medio	Cálculo correcto pero sin indicar si es par o impar
 Alto	Cálculo correcto, razonado, y bien explicado




Verificación con hash XOR simplificado

Nivel	Descripción
 Bajo	No implementado o sin sentido lógico
 Medio	Implementado pero no explica utilidad
 Alto	Bien aplicado, con interpretación clara del resultado

Uso de bucles




Nivel	Descripción
 Bajo	No se usan bucles; lógica repetitiva
 Medio	Se usan bucles pero de forma poco eficiente
 Alto	Uso adecuado, claro y funcional de <code>for</code> / <code>while</code>

Modularidad y organización del código




Nivel	Descripción
 Bajo	Código desorganizado o todo en <code>main</code>
 Medio	Algunas funciones separadas pero mal estructuradas
 Alto	Código modular, funciones bien definidas, legible



Presentación de resultados

Nivel	Descripción
 Bajo	Salidas incompletas o mal formateadas
 Medio	Información mostrada pero sin claridad
 Alto	Salida bien presentada, alineada, clara y explicativa

Creatividad y ampliaciones

Nivel	Descripción
 Bajo	Solo lo mínimo requerido
 Medio	Añade algún detalle extra: menú, validación, bucles anidados
 Alto	Simula errores, añade interfaz de usuario, visualiza binarios



Autoevaluación (opcional)


- ¿Tu código está estructurado?
- ¿Has probado entradas variadas?
- ¿Tu salida es clara?
- ¿Has comentado tus funciones?
- ¿Has probado alguna mejora más allá del enunciado?



Reflexionar también cuenta.

Conclusión

Un buen trabajo debe demostrar:

- ✓ Comprensión del binario y su procesamiento
 - ✓ Capacidad de dividir el problema en partes
 - ✓ Uso eficaz del lenguaje Java
 - ✓ Claridad en el diseño, en los resultados y en el código
-  ¡Evalúate y mejora tu solución antes de entregarla!