

# DOCUMENTO DE PROFUNDIZACIÓN

---

Teoría de la Información y la Comunicación

Recurso 2

## Recurso 2

### Explicación del algoritmo LZ77

Consideremos la cadena: ABCABCABC

#### Paso 1: Inicio

Al inicio:

- No hay datos previos
- La ventana está vacía

Entonces NO puede encontrar coincidencias

Por eso:

(0,0,A)

(0,0,B)

(0,0,C)

#### ¿Qué significa (0,0,A) (offset, longitud, símbolo)?

- 0 → no hay desplazamiento (no hay pasado)
- 0 → no hay coincidencia
- A → símbolo literal

Los primeros símbolos siempre se envían sin compresión porque no hay historial.

Después de leer ABC el sistema ya tiene en memoria ABC y esta se convierte en la ventana de búsqueda

#### Paso 2: Detectar repetición

Ahora aparece la parte restante de la cadena: ABCABC

Empieza desde la siguiente posición y busca: ¿Cuál es la coincidencia MÁS LARGA en el pasado?

En la ventana tenemos: ABC

Y lo que viene es: ABCABC

Coincidencia encontrada: ABC

Entonces construye **(offset, longitud) (3,3)**

Offset = 3: La coincidencia empieza 3 posiciones atrás

Longitud = 3: Porque la coincidencia es ABC → 3 caracteres  
(3,3) significa: “ve 3 posiciones atrás y copia 3 caracteres”.

Repite el proceso con los siguientes caracteres

En la ventana tenemos ABCABC  
Y lo que viene es ABC

Coincidencia encontrada: ABC

Nuevamente:

- offset = 3
- longitud = 3

(3,3)

### **Paso 3: Resultado final**

Representación: A B C (3,3) (3,3)

Ejercicio de aplicación

Texto: “ABABABAB”

Aplicar LZ77

Solución:

LZ77 busca:

- Patrones repetidos
- En la parte ya leída (ventana)

Paso 1

Entrada: A

No hay coincidencias

Salida: (0,0,A)

Paso 2

Entrada: B

No hay coincidencias

Salida: (0,0,B)

Estado actual

Ventana: AB

Paso 3

Entrada: ABABAB

Búsqueda de coincidencia

En la ventana: AB

En el buffer: ABABAB

Coincidencia más larga: ABAB

- longitud = 4
- offset = 2

Salida: (2,4)

Resultado final: (A)(B)(2,2)(4,4)

Conclusión: reducción significativa.

Conclusión

- LZ77 logra alta compresión en datos repetitivos
- No necesita probabilidades
- Se basa en estructura

### Explicación del algoritmo LZ78

Supongamos que tenemos la cadena ABAABABAAB

Estado inicial

- Diccionario vacío
- Índice inicia en 1

#### Paso 1

Entrada: A

- "A" NO está en el diccionario

Salida: (0, A)

Se agrega:

| Paso | Cadena | Diccionario | Salida |
|------|--------|-------------|--------|
| 1    | A      | {1:A}       | (0,A)  |

Significado:

- 0 → no hay patrón previo
- A → símbolo nuevo

## Paso 2

Entrada: B

- “B” tampoco está

Salida: (0, B)

Se agrega:

| Paso | Cadena | Diccionario | Salida |
|------|--------|-------------|--------|
| 1    | A      | {1:A}       | (0,A)  |
| 2    | B      | {2:B}       | (0,B)  |

## Paso 3

Entrada: A

- “A” YA está en el diccionario (índice 1)

No se emite nada todavía

Se intenta extender el patrón

| Paso | Cadena | Diccionario | Salida |
|------|--------|-------------|--------|
| 1    | A      | {1:A}       | (0,A)  |
| 2    | B      | {2:B}       | (0,B)  |
| 3    | A      | ya existe   | —      |

## Paso 4

Ahora vemos: AA

- “AA” NO existe

Pero:

- “A” sí existe → índice 1

Entonces:

- patrón previo = A (índice 1)
- símbolo nuevo = A

Salida: (1, A)

Se agrega:

| Paso | Cadena | Diccionario | Salida |
|------|--------|-------------|--------|
| 1    | A      | {1:A}       | (0,A)  |
| 2    | B      | {2:B}       | (0,B)  |
| 3    | A      | ya existe   | —      |
| 4    | AA     | {3:AA}      | (1,A)  |

## Paso 5

Entrada: B

- “B” ya existe (índice 2)

No se emite aún

Se intenta extender

| Paso | Cadena | Diccionario | Salida |
|------|--------|-------------|--------|
| 1    | A      | {1:A}       | (0,A)  |
| 2    | B      | {2:B}       | (0,B)  |

|   |    |           |       |
|---|----|-----------|-------|
| 3 | A  | ya existe | —     |
| 4 | AA | {3:AA}    | (1,A) |
| 5 | B  | ya existe | —     |

### Paso 6

Ahora vemos: AB

- “AB” NO existe

Pero:

- “A” sí existe → índice 1
- “B” es el nuevo símbolo

Salida: (2, B)

Se agrega:

| Paso | Cadena | Diccionario | Salida |
|------|--------|-------------|--------|
| 1    | A      | {1:A}       | (0,A)  |
| 2    | B      | {2:B}       | (0,B)  |
| 3    | A      | ya existe   | —      |
| 4    | AA     | {3:AA}      | (1,A)  |
| 5    | B      | ya existe   | —      |
| 6    | AB     | {4:AB}      | (2,B)  |

### Resultado final

(0,A) (0,B) (1,A) (2,B)