

# Compresión No Estadística



# AGENDA

- Formas de compresión no estadística
- Compresor LZ77
- Compresor LZ78
- Compresor LZH
- Compresor LZW
- Aplicaciones
- Referencias

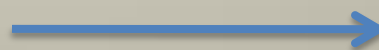
# FORMAS DE COMPRESIÓN NO ESTADÍSTICA

- Compresión por **predicción**
  - Se basan en la experiencia previa para predecir el próximo carácter
  - Si aciertan generan un código pequeño
- Compresión por **sustitución o basada en diccionario**
- Compresión **run-length**

# COMPRESIÓN RUN-LENGTH

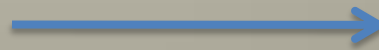
- Detecta **repeticiones de símbolos** y las reemplaza por pares (*símbolo, longitud*)
- Se basa en el desplazamiento de una ventana sobre el archivo (sólo una pasada por el archivo)
- Funciona **bien** cuando hay secuencias de símbolos que se repiten mucho
- Ejemplos:

ABCABCABC  
1A1B1C1A1B1C1A1B1C



Estadístico: **Mal**  
Run-length: **Mal**

AAABBBCCC  
3A3B3C



Estadístico: **Igual de mal**  
Run-length: **Bien!**

- Notar que aumenta el tamaño si **no** se repiten los caracteres

# COMPRESIÓN POR SUSTITUCIÓN

- Consisten en sustituir una cadena de varios símbolos por un puntero a la entrada en un diccionario
- El diccionario puede ser **estático** o **adaptable** (dinámico)
- Un mismo símbolo no siempre se sustituye con lo mismo ( $\neq$  codificación)

## Ejemplo simple de compresión

“Esto es una clase de organización de  
datos. Cátedra servetto”

233/4 219/5 579/2 121/9 182/1  
393/7 182/1 180/1 99/15 servetto

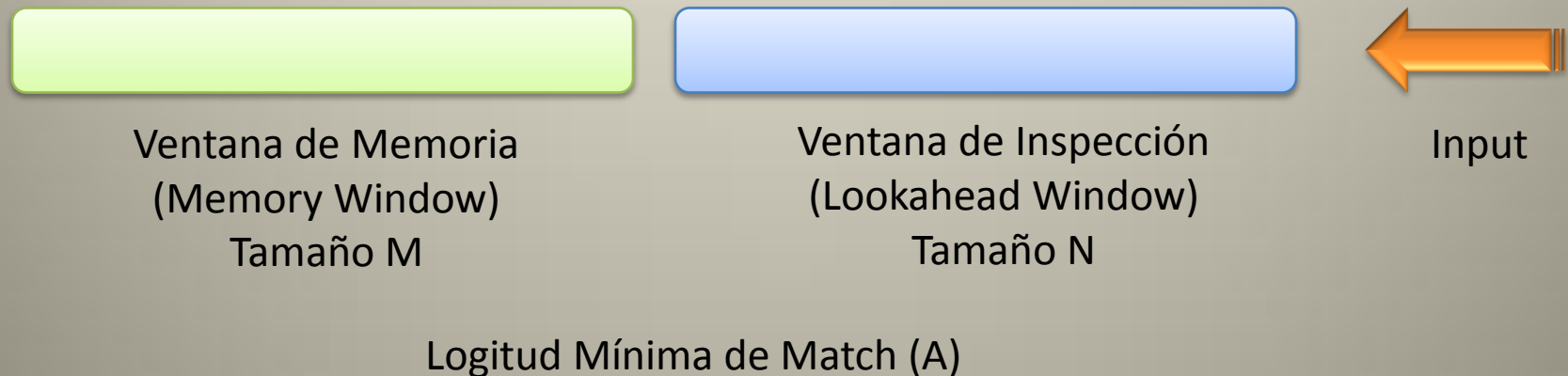
*palabra*

*(página, número)*



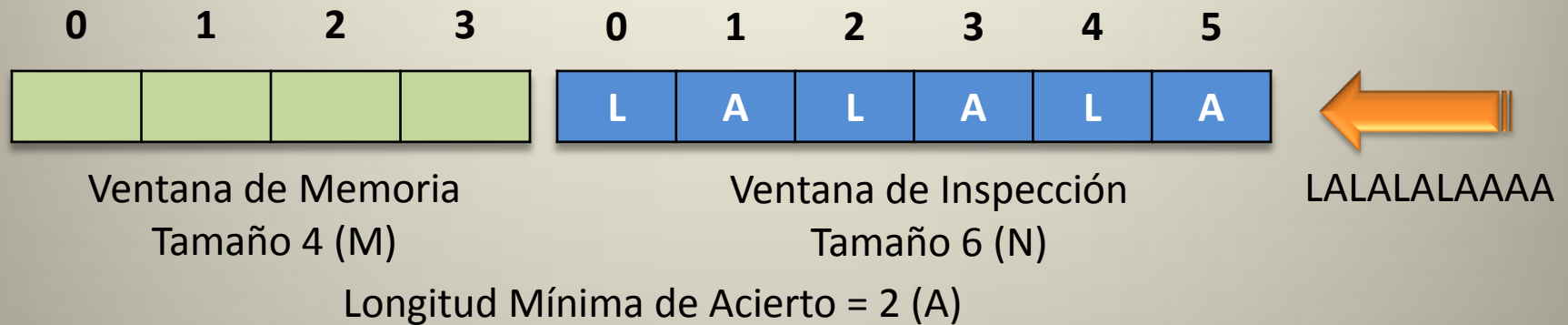
# LZ77 (SLIDING WINDOWS)

- Abraham Lempel y Jacob Ziv en 1977
- El diccionario es la ventana de memoria (adaptativo)



- Emito una letra o un par (posición, longitud)

# LZ77 (EJEMPLO)



Siempre me quedo con el acierto más grande.

Un bit más para distinguir la codificación de un **par ordenado** de un **carácter**.

Para codificar posiciones y longitudes:

- 3 posiciones posibles (0,1,2): 2 bits  $\rightarrow \log_2 (M - A + 1)$
- 3 longitudes posibles (2,3,4): 2 bits  $\rightarrow \log_2 (\text{Min}(M,N) - A + 1)$
- + 1 bit de diferencia entre carácter y dupla = **5 bits**


• **Original:** 11 letras x 8 bits = **88 bits**

• **Comprimido:** 3 letras x 9 bits + 3 duplas x 5 bits = **42 bits**

# LZ78

- Abraham Lempel y Jacob Ziv en 1978
- También utiliza un diccionario adaptativo, pero de todos los símbolos anteriores

0	1	2	3	4	5	6	7	8	9	10
L	A	L	A	L	A	L	A	A	A	A



Ubicación	Emisión	Agrego al dicc.
0	L	256 – LA
1	A	257 – AL
...	...	...

## Diccionario

Código	Símbolo
0	'0'
...	...
255	'255'
256	LA
257	...

- La cantidad de bits emitidos dependen del tamaño de la tabla
- Original:** 11 letras x 8 bits = **88 bits**
- Comprimido:** 1 código x 8 bits + 6 códigos x 9 bits = **62 bits**



# LZ78

- Descompresión


- El tamaño de la tabla indica la cantidad de bits que tengo que leer
- Se empieza con el diccionario vacío y se va regenerando
- Siempre se almacena lo anterior más la primera letra del próximo

- Caso Especial

- Cuando para agregar una entrada en la tabla tengo que leer un símbolo que no tengo

Salida

L	A	256	258
---	---	-----	-----



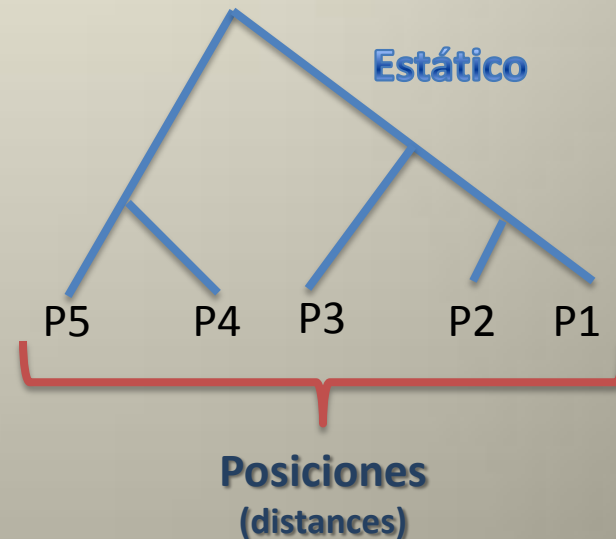
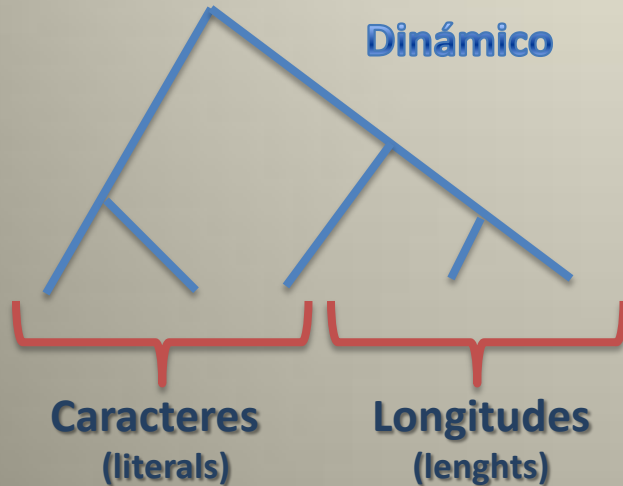
Diccionario

Código	Símbolo
256	LA
257	AL
258	??

¿Cómo lleno este símbolo? LA (256) + L (primera letra del símbolo actual)

# LZH (LZ + HUFFMAN)

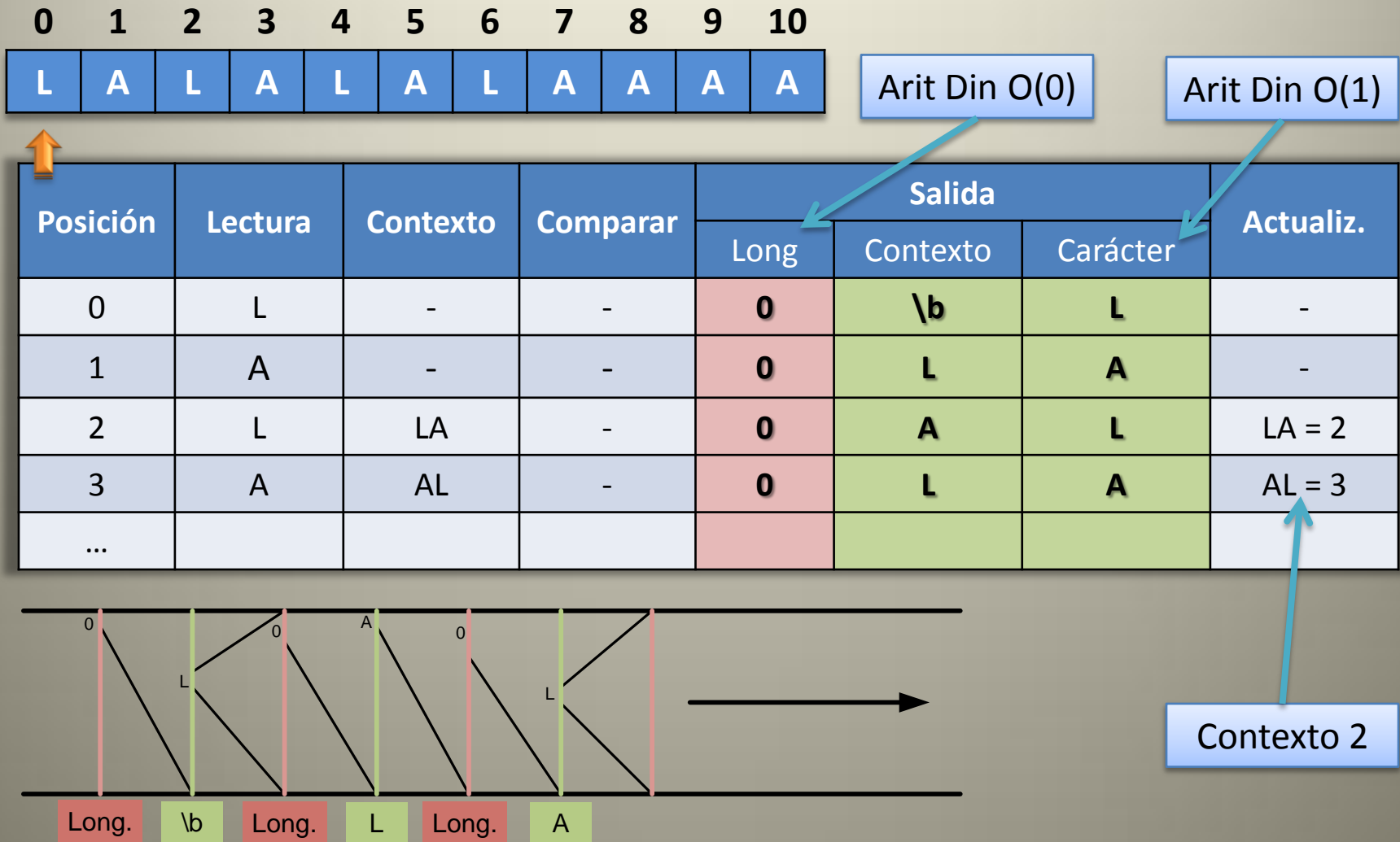
- Es un LZ77 que utiliza árboles de Huffman



- Ahora no necesito un bit más para distinguir entre duplas y caracteres

# LZP

- Charles Bloom, 1994. Le agrega predicción al algoritmo de los LZ





# APLICACIONES

- RLE → Fax, Windows Bitmap (bmp)
- LZH → DEFLATE → PNG (Imágenes Blanco y Negro) ,  
ZIP, Gzip, HTTP, PPP
- LZW → GIF / Acrobat<sup>®</sup> PDFs



# REFERENCIAS

- Data Compression: The Complete Reference, David Salomon, 3rd Edition, Springer
- The Data Compression Book, Mark Nelson, 2nd Edition, M&T Books, 1995
- Ziv, J. and Lempel, A., “A Universal Algorithm for Secuential Data Compression”, IEEE Trans. Information Theory, vol. 23, pp. 337-343, May 1977.  
(<http://www.stanford.edu/class/ee398a/resources/ziv:77-SDC.pdf> )
- Ziv, J. and Lempel, A., “Compression of Individual Secuencias via Variable-Rate Coding”, IEEE. Transactions on Information Theory, vol. 24, pp. 530-536, 1978.  
([http://www.cs.duke.edu/courses/spring03/cps296.5/papers/ziv\\_lempel\\_1978\\_variable-rate.pdf](http://www.cs.duke.edu/courses/spring03/cps296.5/papers/ziv_lempel_1978_variable-rate.pdf) )
- <http://www.datacompression.info/>
- [http://www.arturocampos.com/ac\\_rle.html](http://www.arturocampos.com/ac_rle.html)
- <http://www.data-compression.com/>
- <http://ccc.inaoep.mx/~mmorales/documents/Compre.pdf>
- <http://www.binaryessence.com/dct/en000003.htm> (Muy bueno!)
- Wikipedia:
  - <http://es.wikipedia.org/wiki/RLE>
  - [http://en.wikipedia.org/wiki/Dictionary\\_coder](http://en.wikipedia.org/wiki/Dictionary_coder)
  - [http://en.wikipedia.org/wiki/LZ77\\_and\\_LZ78](http://en.wikipedia.org/wiki/LZ77_and_LZ78)