

Lo que vamos a ver:

* Digitalización
* Compresión
* Teléfonos digitales Packet Tracert

**Teorema de Nyquist:** Afirma que la velocidad de muestreo fs debe ser mayor que el doble del componente de interés de frecuencia más alto en la señal medida.

“Para que la voz se pueda digitalizar y sea perfecta la frecuencia tiene que ser el doble”

Frecuencia de muestreo 8000 Hz / 8KHz (a la que tienes que hacer grabaciones)

↪ en 1s 8000 direcciones de memoria



Teléfono digital: 2 puertos RJ-45 uno conectado al PC y otro a la RED

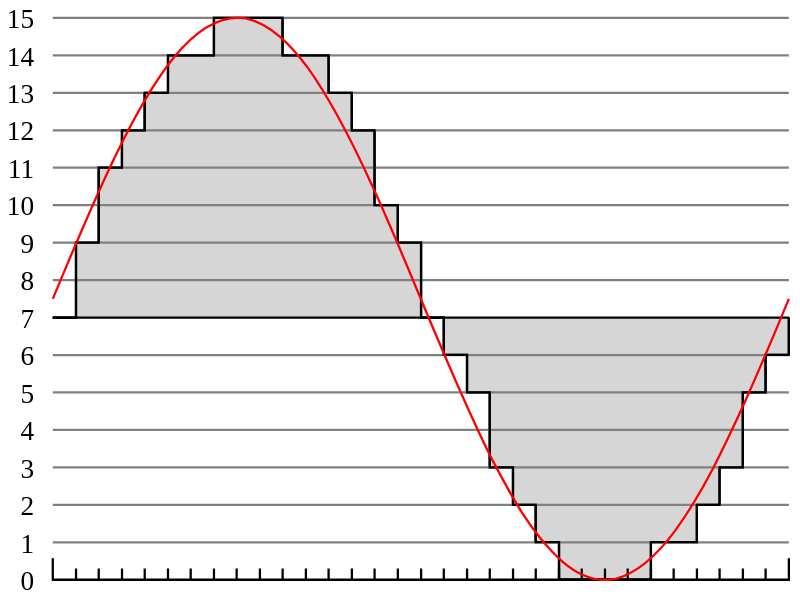
↪DHCP para el PC, hay una instrucción para darle otra IP al teléfono.

* Fue inventado por el italiano **Antonio Meucci,** quien en 1854 construyó su primer prototipo, aunque no formalizó su patente por dificultades económicas, presentando solo una breve descripción de su invento en la Oficina de Patentes de Estados Unidos el año de 1871.
  + Años después, en 1876, el escocés **Alexander Graham** Bell fue el primero en patentar formalmente, y durante ¡’ muchos años fue considerado el inventor del teléfono.

*“Normalmente se suele utilizar 8 bits (hexadecimal) en telefonía”*

**Tipos de modulación:**

* **PCM (Pulse Code Modulation):** modulación de impulsos codificados
  + En una señal analógica hacer como escalones
  + Muestreo: velocidad en la que se rastrea
  + Cuantos más bits (+escalones) los escalones van a ser más pequeños
  + Convertidores de 8 bits 256, por más que le pongas más velocidad (muestreos) no va a coger más de 256, no va a coger más info.

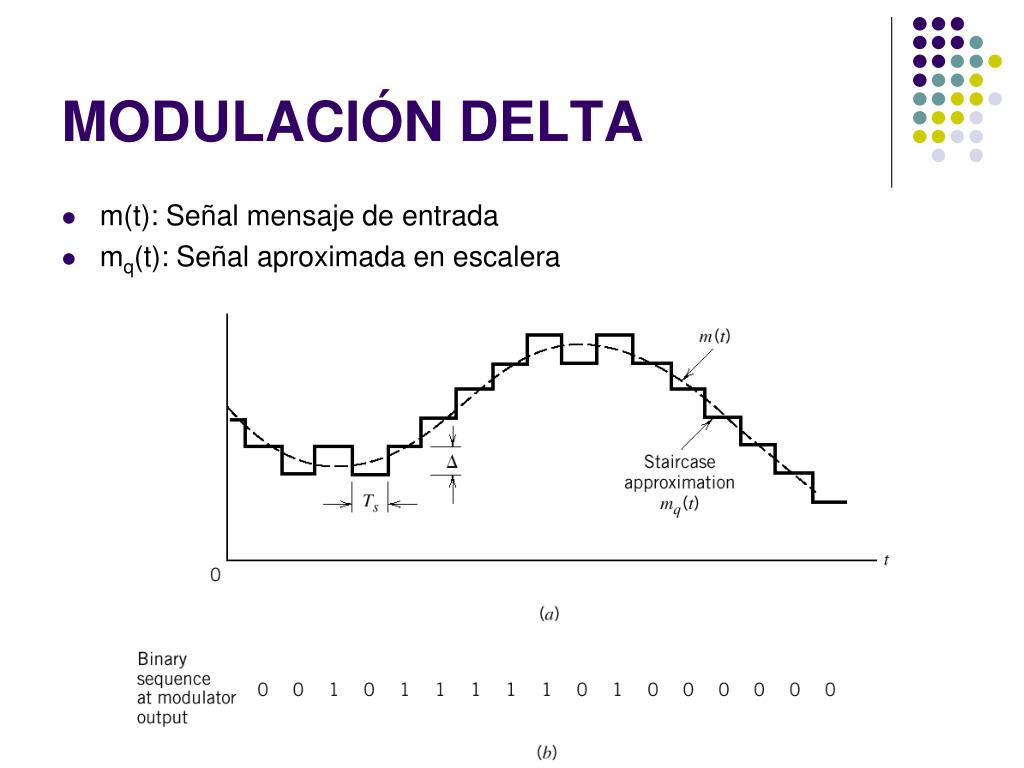
[Explicación de PCM: Pulse Code Modulation](https://youtu.be/dGjzetQy76Y)

PCM por cada muestro da un código de bits igual al codificador (8 bits)

**VS**

Delta por cada muestreo da un código de 1 bit

* **Delta** 
  + Para conseguir la máxima velocidad de muestreos, aumenta la cantidad de muestreos.
  + Aquí se va añadiendo un bit: 1 si es superior a la anterior y si es inferior se añade un bit: 0.
  + Por tanto utilizar menos memoria



Ejercicios:

1. Qué dice la ley de Nyquist

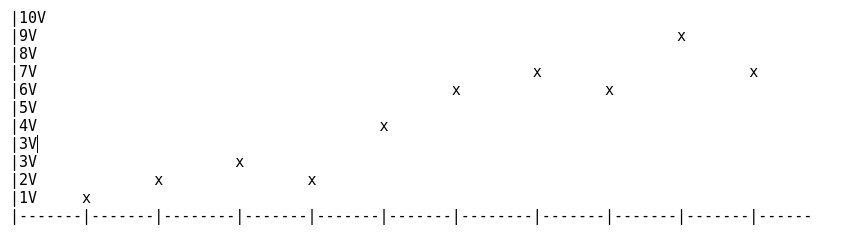
Que la frecuencia de muestreo sea el doble de la frecuencia de habla

2. Se busca en internet

3. Vref del convertidor A/D=10v, 3 bits, 2 elevado a 3 8 combinaciones (este era muy sencillito, 3 bits)

\_

V\_| = (Vref / 2 elevado a n.º de bits)= 10/2³= 10/8= 1,25v



| COMBI | 000 | 001 | 010 | 011 | 100 | 101 | 110 | 111 |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| MUEST | 0,8 | 1,5v | 3,8v | 2,4v | 5,8v | 7,6v | 6,1v |  |
| INTERV | 0-1,25v | 1,25v | 2,5v | 3,75v | 5v | 6,25v | 7,5v | 8,75v |

T. MUESTREO / T. ESCALÓN= 6,1v/1,25v= **4**,~~88~~; cogemos 4 y lo pasamos a hexadecimal, entonces es 100.

**Cuando ya son de más de 8 bits lo pones en hexadecimal.**

*En la vida real hay tanto positivo como negativo, por eso multiplicamos por 2:*

\_

V\_| = 2 \* (Vref / 2 elevado a n.º de bits)=

5. Vreferencia: +-10v; 8 bits, 256 combinaciones

\_

V \_| = 2\*10/2⁸= 0.0782v

\_

**Vm/V\_| =**

M6= 10+5.8 PQ CONTAMOS DESDE -10v / 0.0782= 202.04 a hexadecimal = **CA**

M16= 10-7,3/0.0782= 34.56= en hexa= **22**

**EJEMPLO M1: vref=10v (entre el -10 y +10) muestreo= 0.8v**

\_

V\_| = 2\*10/2⁸= 0,078; [10 (x q partes del -10)+0.8v]/0.078= 138.~~46~~ = a hexadecimal = 8A

Si partiesemos del 0,0 de la gráfica sería sumar 80;

**EJEMPLO CON NEGATIVO: muestreo=-4.4v**

Vref lo calculas

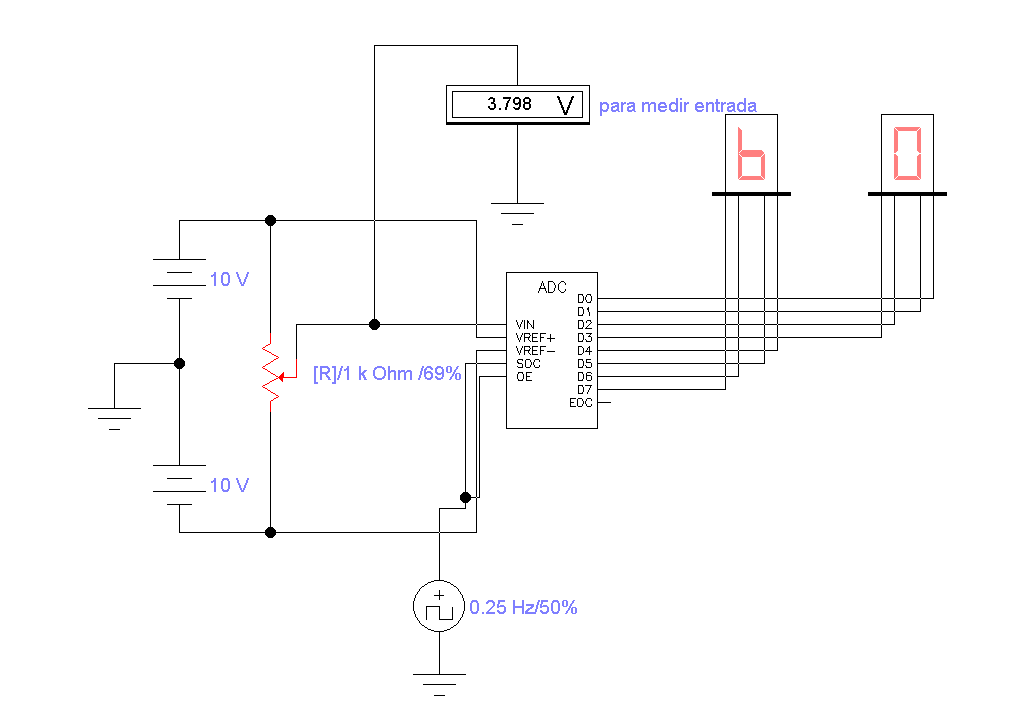
TIenes que restar: 10-4.4/Vref

**Workbench**

Convertidor A/D

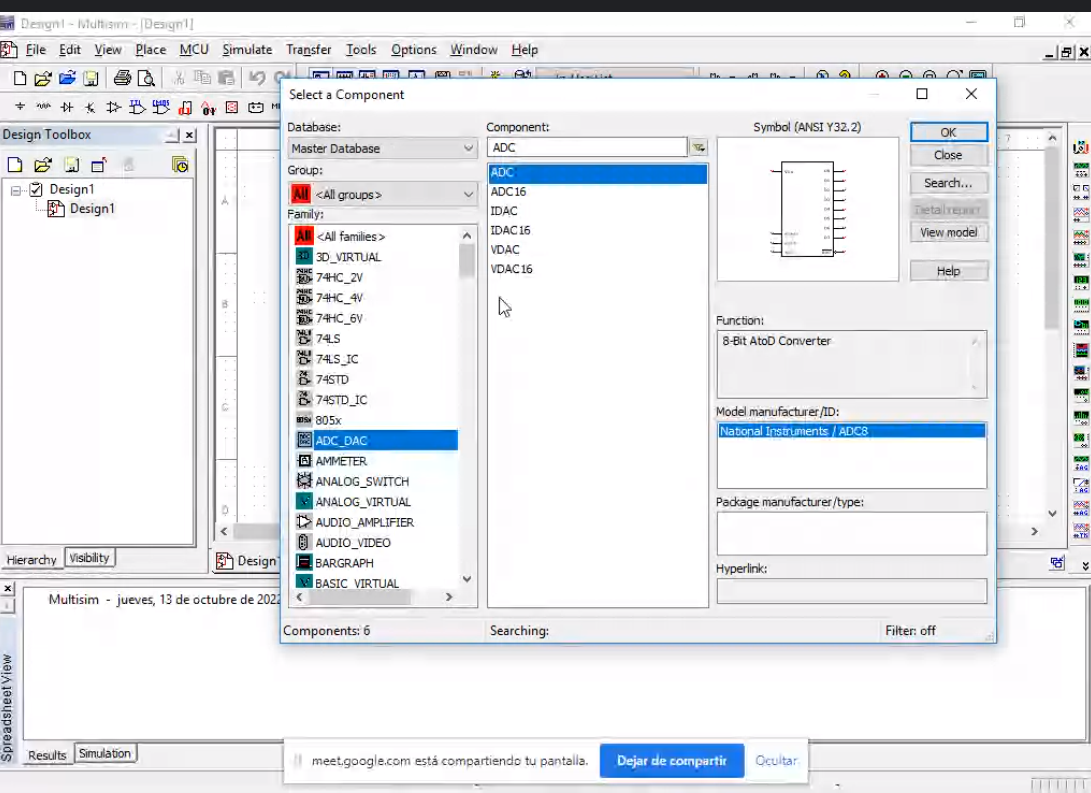
Display

Los cables de los displays se conectan de izquierda a derecha de menor a mayor

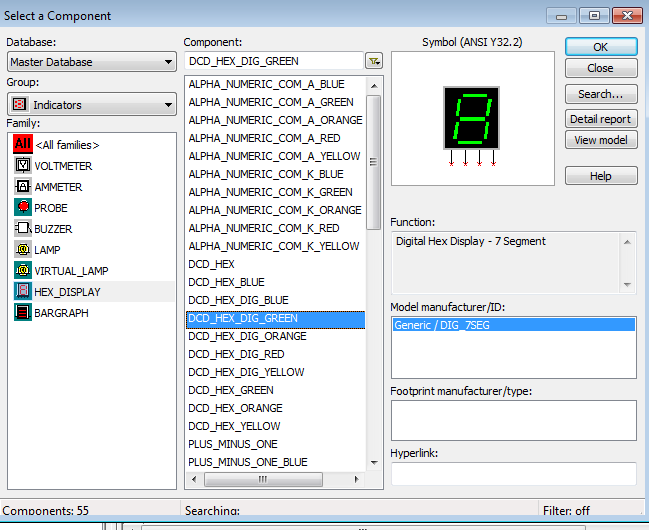


**CONVERTIDOR AC**

****

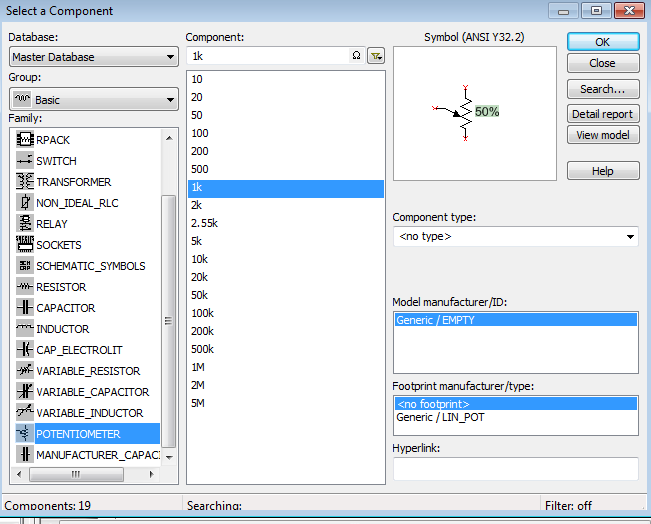
****

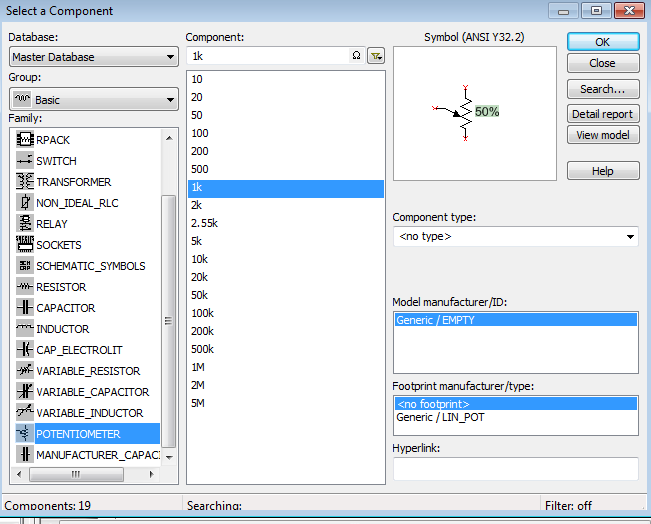
**DISPLAY**



**resistencia, potenciometro**

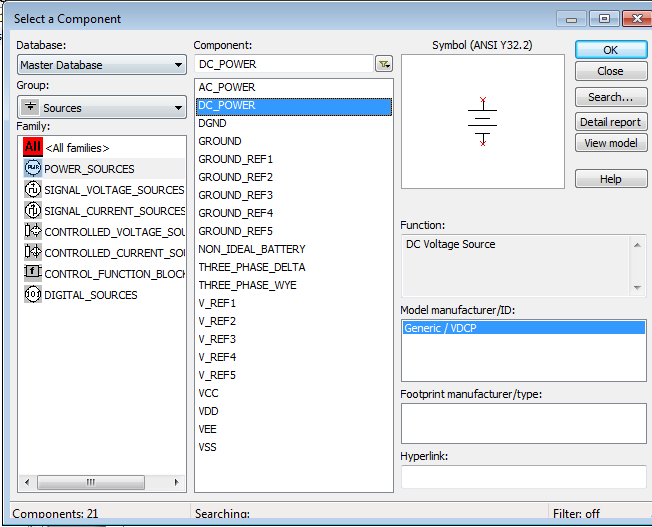






**FUENTES ALIMENTACIÓN**

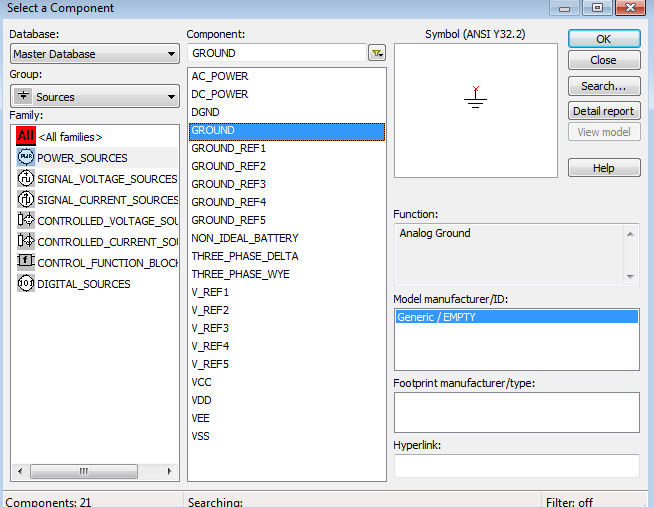
****

****

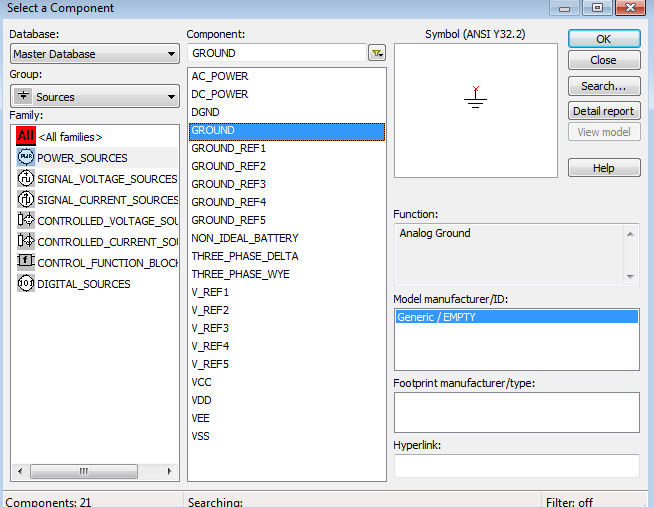
***DOBLE CLICK EN COMPONENTE PARA CAMBIAR VOLTAJE***

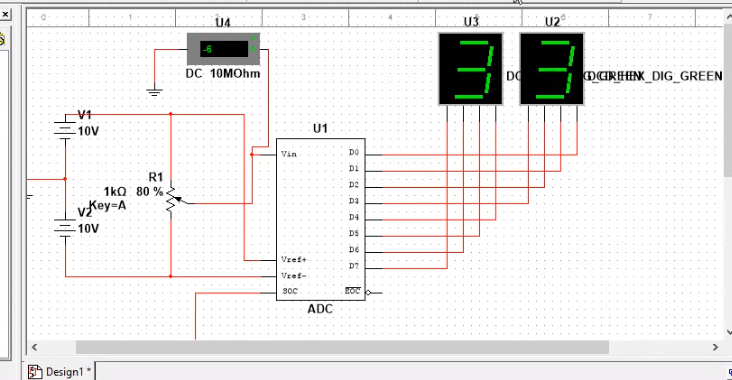
***TOMA A TIERRA***

****



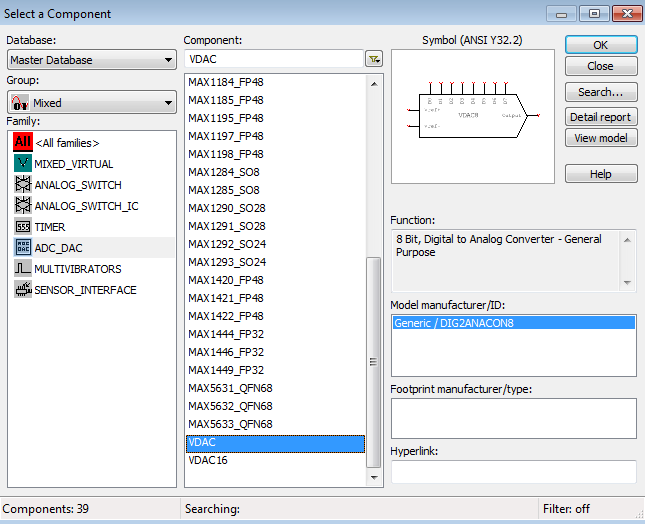
**RELOJ**

****

****

**Vin/0.078=Valor decimal=3A**

**COSAS DE LA PRACTICA RARA**

****

** OSCILOSCOPIO (num4 de la derecha)**

**multimetro (num1 de la derecha)**

# **HUFFMAN**

Algoritmos:

* RLE PATATA BARATA
* LZ77 P->1
* LZ78 A->6
* LZW T->3

B->1

R->1

A = 0

T = 10

P = 110

B= 1110

R = 1111

Tasa de compresión = Bits comprimidos del binario / Bits totales sin comprimir

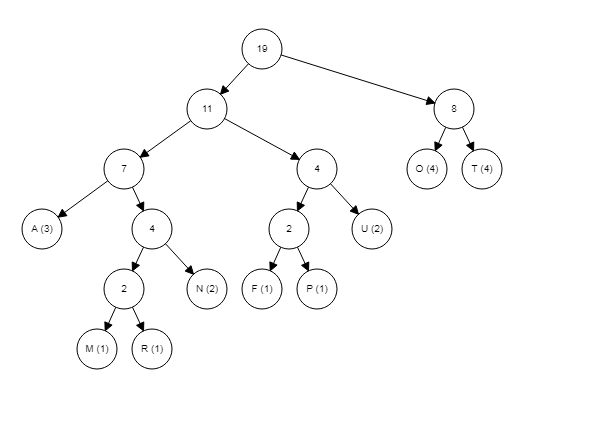
1100 **(P)** 0 **(A)** 10 **(T)** 0 **(A)** 10 **(T)** 0 **(A)** 1110 **(B)** 0 **(A)** 1111 **(R)** 0 **(A)** 10 **(T)** 0 **(A)**

Ejemplo:

[**Huffman Coding Visualization**](https://cmps-people.ok.ubc.ca/ylucet/DS/Huffman.html)-> página Gucci

**FORTUNATOMATOUNPATO**

**19 CARACTERES == 19\*8= 152 BITS**

****

**M-> 00100 0100 10 00101 11 011 0011 000 11 10 00100 000 11 10 011 0011 0101 000 11 10**

**R-> 00101**

**F-> 0100 Tasa de compresión = 57 / 152 = 0.375**

**P-> 0101**

**N-> 0011**

**U-> 011**

**A-> 000**

**O-> 10**

**T-> 11**

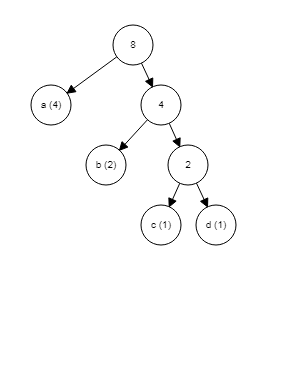
**frase= aabacdab = 0 0 10 0 110 111 0 10 (es sin espacios pero para verlo mejor)**

**a=4=> 0**

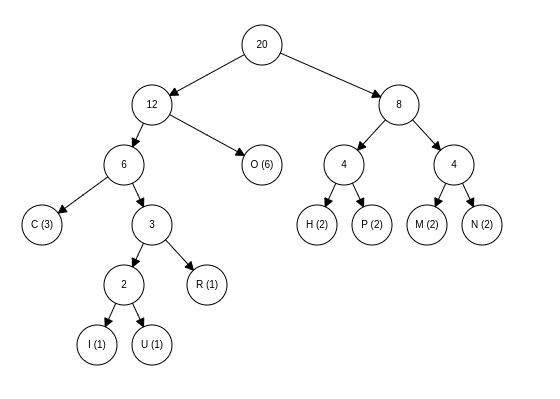
**b=2=> 10**

**c=1=> 110**

**d=1=>111**

****

**PINOCHOCOMPROUNMOCHO->101 00100 111 01 000 01 000 01 101 0011 01 00101 111 110 01 000 01**



**I->00100**

**U->00101**

**R->0011**

**N->111**

**M->110**

**P->101**

**H->100**

**C->000**

**O->01**

**T=60/160**

# **LZ77**

**ABRACADABRA**

TUPLAS

**TC= 66/ 88 (ASCII)**

con la última transformación: 3+3+5=11\*6 (nº de veces que has realizado el procedimiento)=66

-7-> necesitamos 3 bits

-4-> necesitamos 3 bits

-#-> necesitamos 5 bits

**(0,0,a)**

**-cuantas letras por delante están las letras**

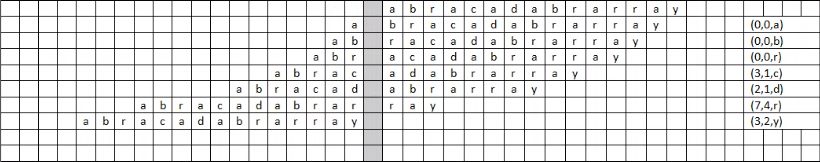
**-nº de caracteres repetidos: que aparecen en ese encuentro**

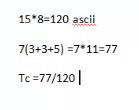
EJ1:

**ABABBABCABABBABBCA**

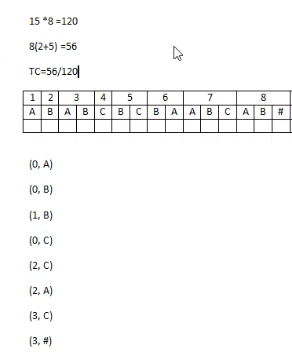
EJ2:

**(0,0,M) (0,0,A) (2,2,A) (3,3,A) (4,2,A) (4,2,E) (0,0,L) (5,1,#)**

****

****

# **LZ78**



**15 caracteres de la cadena dada \* 8 bits(no cambia) = 1207 tuplas(2 bits nº máximo de los paréntesis \* 5 bits del alfabeto) = 56**

**TC = 56 /120**

# **RLE**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

**72 bits (9 filas \* 8 columnas) \* 24 bits de cada color (no cambia) = 1728 bits**

|  |  |  |  |  |  |  |  |  | 3B 5R | 53\* |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  | 1B 5V 2B | 78 |
|  |  |  |  |  |  |  |  |  | 2B 3Az  3B | 78 |
|  |  |  |  |  |  |  |  |  | 2B 3Az 3B | 78 |
|  |  |  |  |  |  |  |  |  | 2B 3A 3B | 78 |
|  |  |  |  |  |  |  |  |  | 2B 3A 3B | 78 |
|  |  |  |  |  |  |  |  |  | 2B 3A 3B | 78 |
|  |  |  |  |  |  |  |  |  | 2B 3A 3B | 78 |
|  |  |  |  |  |  |  |  |  | 1B 6A 1B | 77 |

**B→Blanco**

**R→Rojo**

**V→Verde**

**Az→Azul**

**\*53= 2 bits ( el 3 se representa en binario con 2 bits) + 24 bits + 3bits (el 5 se representa con 3 bits) + 24 bits)**

**53+78+78+78+78+78+78+78+77=676 TC=676/1728**