### **ÁREA TEMÁTICA No1**

**PRÁCTICO 1 TEORÍA DE LA INFORMACIÓN**

**CANTIDAD DE INFORMACIÓN - ENTROPÍA**

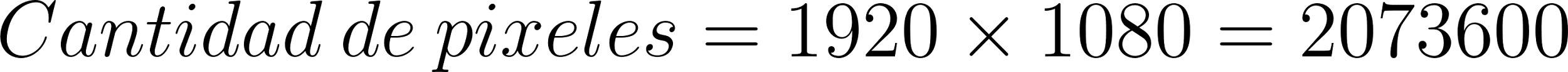
1. Supón una imagen de alta definición dividida en cuadros de 1920 por 1080 píxeles cada uno. Indica la cantidad de información que proporciona un cuadro de imagen si está codificada en 24 bits por píxel. Luego, considera una imagen con una resolución de 3840 x 2160 píxeles, ¿cuánta información se genera si se codifica en HDR (High Dynamic Range) con 10 bits por canal de color?

Pixeles = 1920x1080; por cada cuadro hay 24 niveles de brillo

Calculamos la cantidad de información de un pixel:

[](https://www.codecogs.com/eqnedit.php?latex=%20Informacion%20%5C%3A%20por%20%5C%3A%20pixel%20%3D%20log_2%7B24%7D%20%3D%204.58%20#0)

Ahora multiplicamos la cantidad de información por píxel para el cuadro entero:

[](https://www.codecogs.com/eqnedit.php?latex=%20Cantidad%20%5C%3A%20de%20%5C%3A%20pixeles%20%3D%201920%20%5Ctimes%201080%20%3D%202073600%20#0)

Por lo tanto la cantidad total de bits de información será de:

[](https://www.codecogs.com/eqnedit.php?latex=%20I%20%3D%202073600%20%5Ctimes%204.58%20%3D%209497088%20%5C%3A%20bits%20%5C%3A%20de%20%5C%3A%20informacion%20#0)

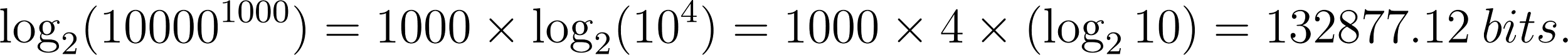
Ahora repetimos el proceso para HDR:

[](https://www.codecogs.com/eqnedit.php?latex=%20Informacion%20%5C%3A%20por%20%5C%3A%20pixel%20%3D%20log_2%7B10%7D%20%3D%203%2C32%20#0)

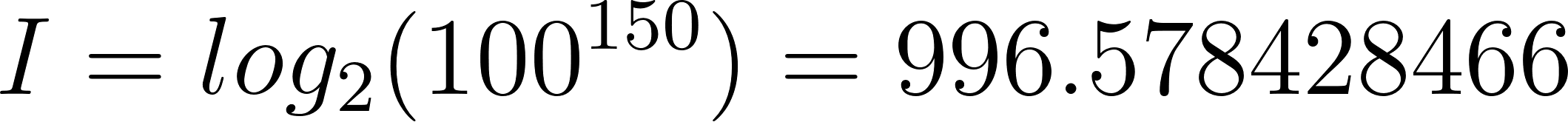
[](https://www.codecogs.com/eqnedit.php?latex=%20Cantidad%20%5C%3A%20de%20%5C%3A%20pixeles%20%3D%203840%20%5Ctimes%202160%20%3D%208294400%20#0)

[](https://www.codecogs.com/eqnedit.php?latex=%20I%20%3D%208294400%20%5Ctimes%20%203.32%20%3D%2027537408%20%5C%3A%20bits%20%5C%3A%20de%20%5C%3A%20informacion%20#0)

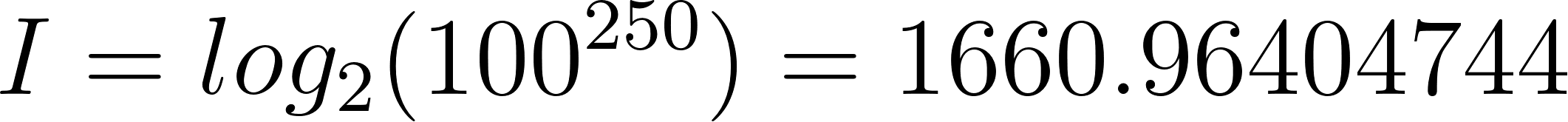
1. ¿Qué información genera un locutor de radio a través de 1000 palabras tomadas al azar de un vocabulario de 10000?¿Qué relación existe entre las dos cantidades anteriores?.

[](https://www.codecogs.com/eqnedit.php?latex=%20%5Clog_2%20(10000%5E%7B1000%7D)%20%3D%201000%20%5Ctimes%20%5Clog_2%20%20(10%5E4)%20%3D%201000%20%5Ctimes%204%20%5Ctimes%20(%5Clog_2%20%2010)%20%3D%20132877.12%20%5C%3A%20bits.#0) La imágen produjo muchos bits más de información que la cantidad de información proporcionada por el locutor

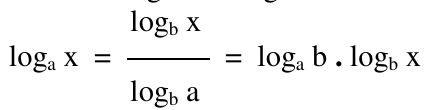
1. ¿Qué cantidad de información genera un mensaje de texto compuesto por 150 caracteres tomados al azar de un alfabeto de 100 símbolos? ¿Cómo cambiaría si el mensaje es de 250 caracteres con el mismo alfabeto?
2. Para 150 caracteres de un vocabulario de 100 la cantidad de información es:

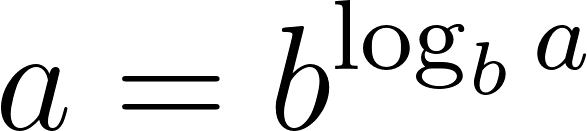
[](https://www.codecogs.com/eqnedit.php?latex=%20I%20%3D%20log_2(100%5E%7B150%7D)%20%3D%20996.578428466%20#0) bits de información

En cambio si se usan 250 caracteres:

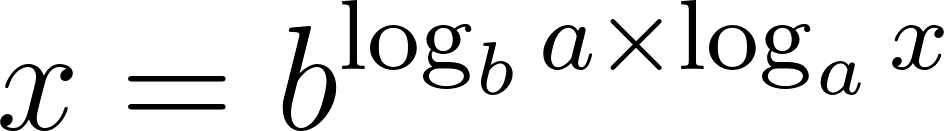
[](https://www.codecogs.com/eqnedit.php?latex=%20I%20%3D%20log_2(100%5E%7B250%7D)%20%3D%201660.96404744%20#0) bits de información

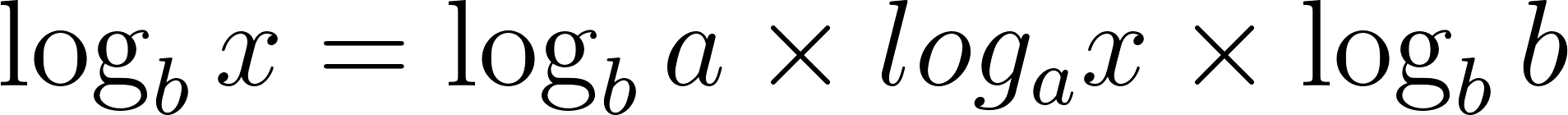
1. Demostrar las siguientes igualdades:

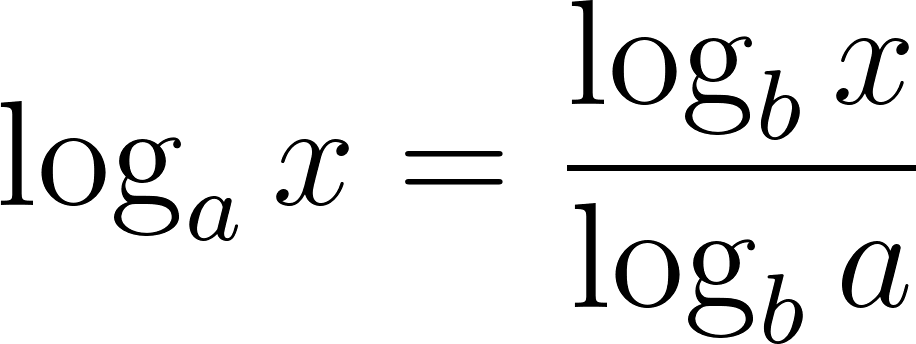


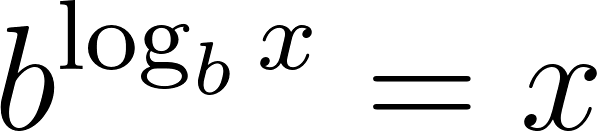
[](https://www.codecogs.com/eqnedit.php?latex=%20a%20%3D%20b%20%5E%20%7B%5Clog_b%20a%7D%20#0)

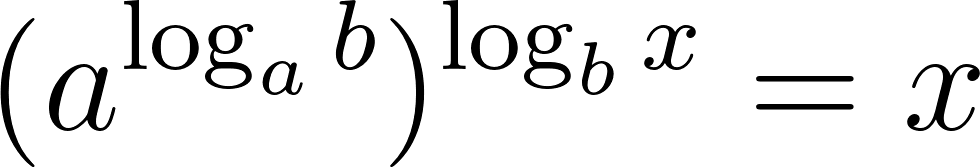
[](http://www.sciweavers.org/tex2img.php?bc=Transparent&fc=Black&im=jpg&fs=100&ff=modern&edit=0&eq=%20a%20%5E%20%20%5Clog_a%20x%20%3D%20(b%20%5E%20%7B%5Clog_b%20a%7D)%20%5E%20%7B%5Clog_a%20x%7D#0)

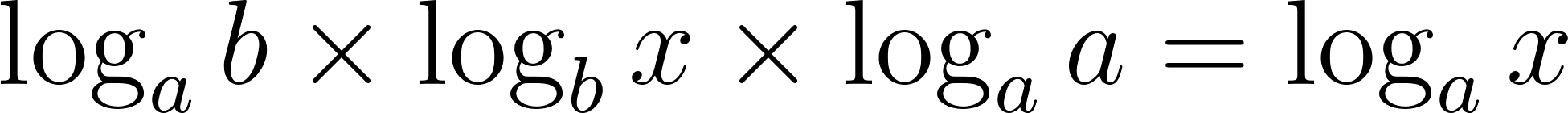
[](https://www.codecogs.com/eqnedit.php?latex=%20x%20%3D%20b%20%5E%20%7B%5Clog_b%20a%20%5Ctimes%20%5Clog_a%20x%7D%20#0)

[](https://www.codecogs.com/eqnedit.php?latex=%20%5Clog_b%20x%20%3D%20%5Clog_b%20a%20%5Ctimes%20log_a%20x%20%5Ctimes%20%5Clog_b%20b%20#0)

[](https://www.codecogs.com/eqnedit.php?latex=%20%5Clog_a%20x%20%3D%20%7B%5Clog_b%20x%20%5Cover%20%5Clog_b%20a%7D%20#0)

[](https://www.codecogs.com/eqnedit.php?latex=%20b%20%5E%20%7B%5Clog_b%20x%7D%20%3D%20x%20#0)

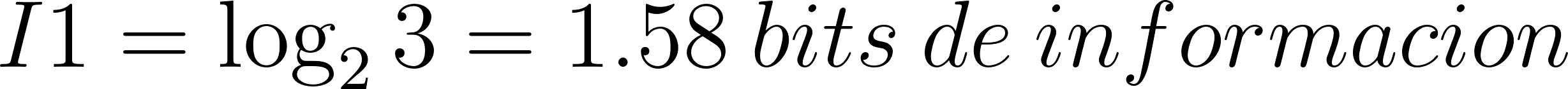
[](https://www.codecogs.com/eqnedit.php?latex=%20(a%20%5E%20%7B%5Clog_a%20b%7D)%20%5E%20%7B%5Clog_b%20x%7D%20%3D%20x%20#0)

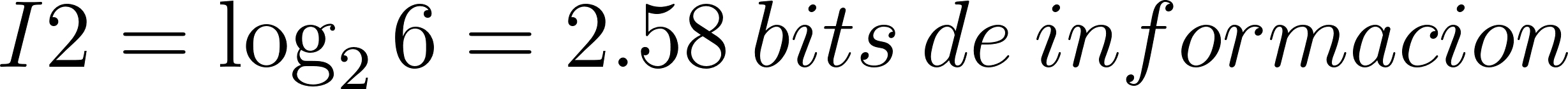
[](https://www.codecogs.com/eqnedit.php?latex=%20%5Clog_a%20b%20%5Ctimes%20%5Clog_b%20x%20%5Ctimes%20%5Clog_a%20a%20%3D%20%5Clog_a%20x%20#0)

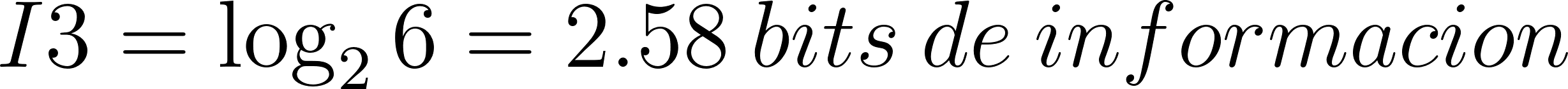
[](https://www.codecogs.com/eqnedit.php?latex=%20%5Clog_a%20x%20%3D%20%5Clog_a%20b%20%5Ctimes%20%5Clog_b%20x#0)

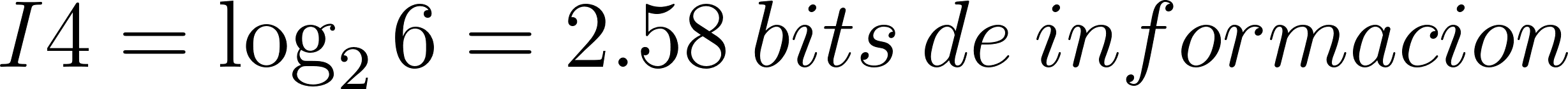
1. Considera una fuente F con cinco símbolos (s1, s2, s3, s4, s5) con las probabilidades p1=1/3, p2=1/6, p3=1/6, p4=1/6, p5=1/6. Calcula la cantidad de información individual y la entropía de esta fuente.

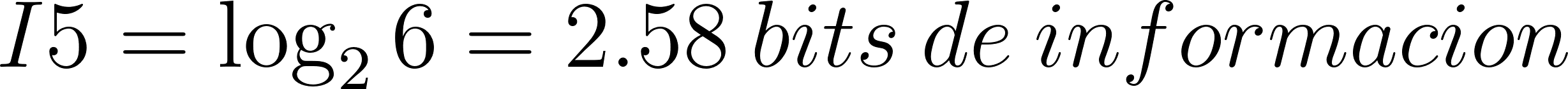
Primero calculamos la cantidad de información de cada símbolo:

[](https://www.codecogs.com/eqnedit.php?latex=%20I1%3D%20%5Clog_2%7B3%7D%20%3D%201.58%20%5C%3A%20bits%20%5C%3A%20de%20%5C%3A%20informacion%20#0)

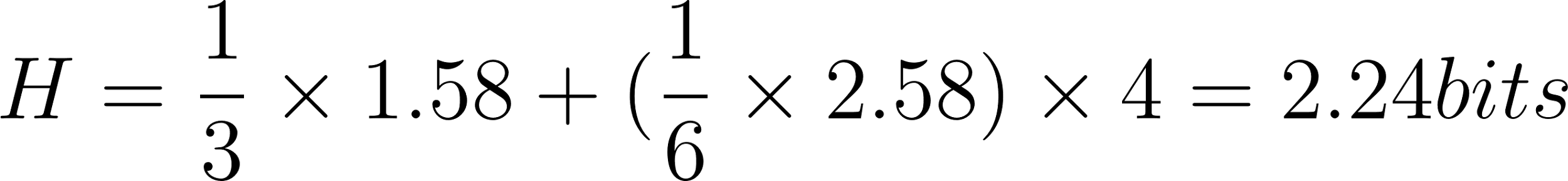
[](https://www.codecogs.com/eqnedit.php?latex=%20I2%3D%20%5Clog_2%7B6%7D%20%3D%202.58%20%5C%3A%20bits%20%5C%3A%20de%20%5C%3A%20informacion%20#0)

[](https://www.codecogs.com/eqnedit.php?latex=%20I3%3D%20%5Clog_2%7B6%7D%20%3D%202.58%20%5C%3A%20bits%20%5C%3A%20de%20%5C%3A%20informacion%20#0)

[](https://www.codecogs.com/eqnedit.php?latex=%20I4%3D%20%5Clog_2%7B6%7D%20%3D%202.58%20%5C%3A%20bits%20%5C%3A%20de%20%5C%3A%20informacion%20#0)

[](https://www.codecogs.com/eqnedit.php?latex=%20I5%3D%20%5Clog_2%7B6%7D%20%3D%202.58%20%5C%3A%20bits%20%5C%3A%20de%20%5C%3A%20informacion%20#0)

Por lo tanto la entropía de la fuente es de:

[](https://www.codecogs.com/eqnedit.php?latex=%20H%20%3D%20%7B1%20%5Cover%203%7D%20%5Ctimes%201.58%20%2B%20(%7B1%20%5Cover%206%7D%20%5Ctimes%202.58)%20%5Ctimes%204%20%3D%202.24%20bits%20#0)

1. Comprobar que al tirar un dado, si las probabilidades individuales son iguales se logra el máximo valor de la información promedio. Luego, compara este resultado con una distribución de probabilidades en el mismo dado donde p1=0.5, p2=0.2, p3=0.1, p4=0.08, p5=0.06, p6=0.04, p7=0.015, p8=0.005.

Si cada cara tiene la misma probabilidad de aparecer la cantidad de información será de:

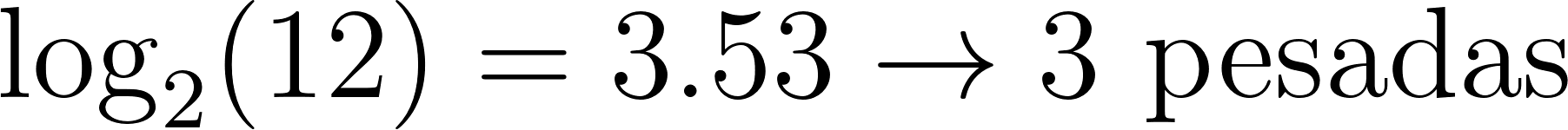
[](https://www.codecogs.com/eqnedit.php?latex=%20I%20%3D%20%5Clog_2%208%20%3D%203%20%5C%3A%20bits%20%5C%3A%20de%20%5C%3A%20informacion%20%5C%3A%20para%20%5C%3A%20cada%20%5C%3A%20cara%20#0)

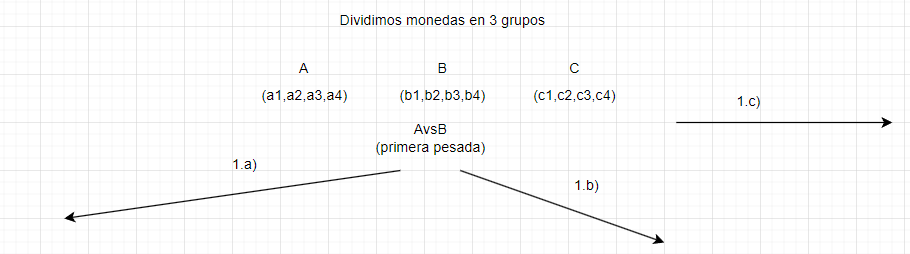
Calculamos la cantidad de información promedio (Entropía):

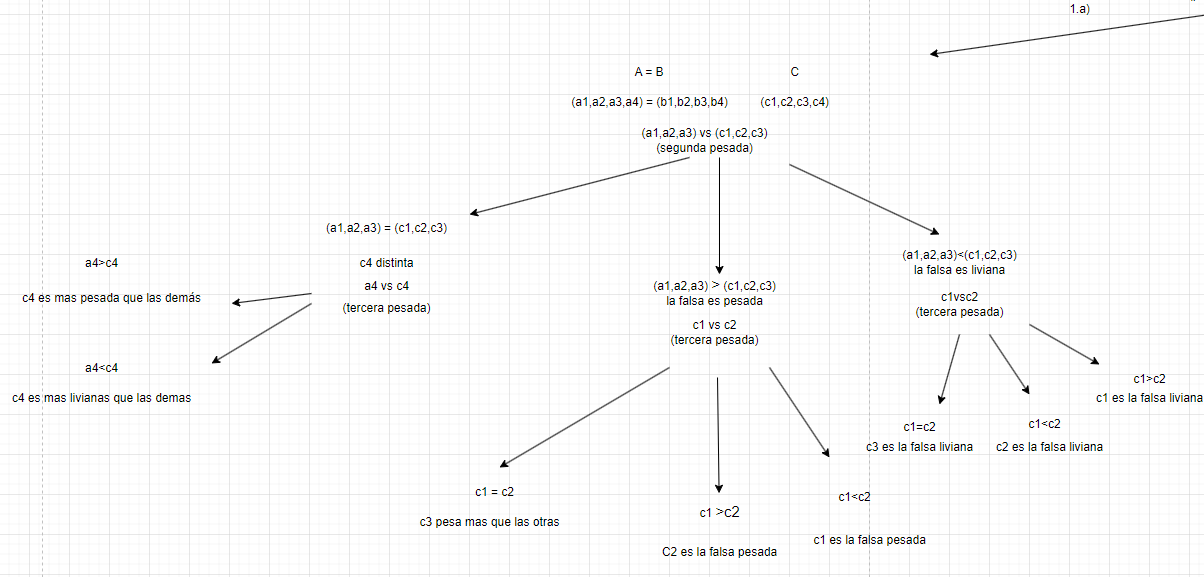
[](http://www.sciweavers.org/tex2img.php?bc=Transparent&fc=Black&im=jpg&fs=100&ff=modern&edit=0&eq=%20H%20%3D%20(%7B1%20%5Cover%208%7D%20%5Ctimes%203)%20%5Ctimes%208%20%3D%203%20%5C%3A%20bits%20%5C%3A%20de%20%5C%3A%20informaci%5C'on%20%5C%3A%20(maximizada)%20%5C%3A%20al%20%5C%3A%20tener%20%5C%3A%20equiprobabilidad%20%5C%3A%20en%20%5C%3A%20todas%20%5C%3A%20las%20%5C%3A%20caras%20%5C%3A%20#0)

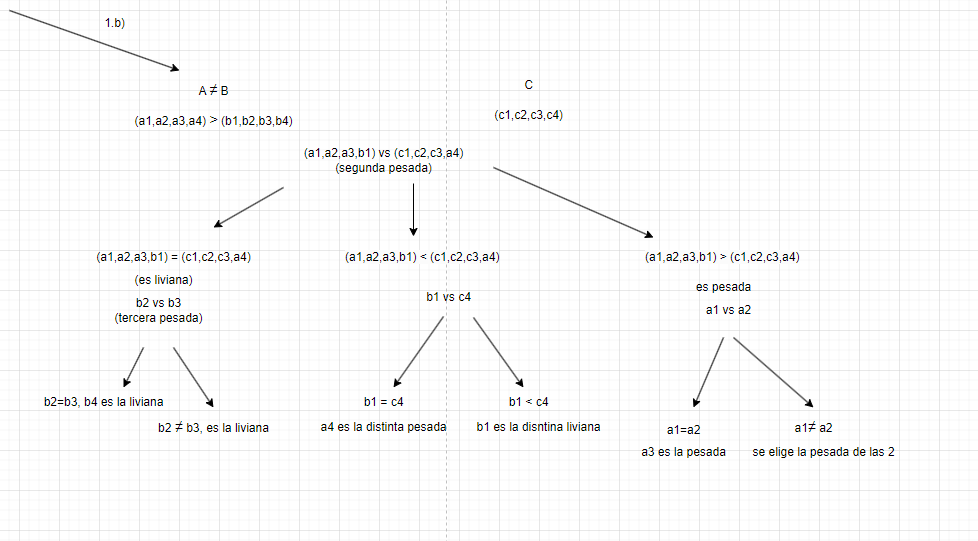
1. Sean 12 monedas una de las cuales tiene peso diferente, indicar cuántas pesadas son necesarias para encontrarla, especifique la metodología de peso

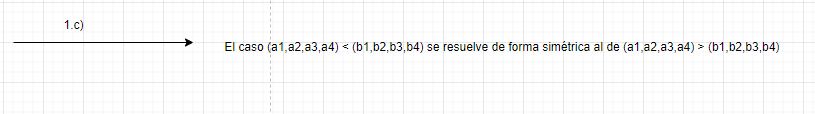
Para obtener la cantidad de pesadas necesarias hacemos:

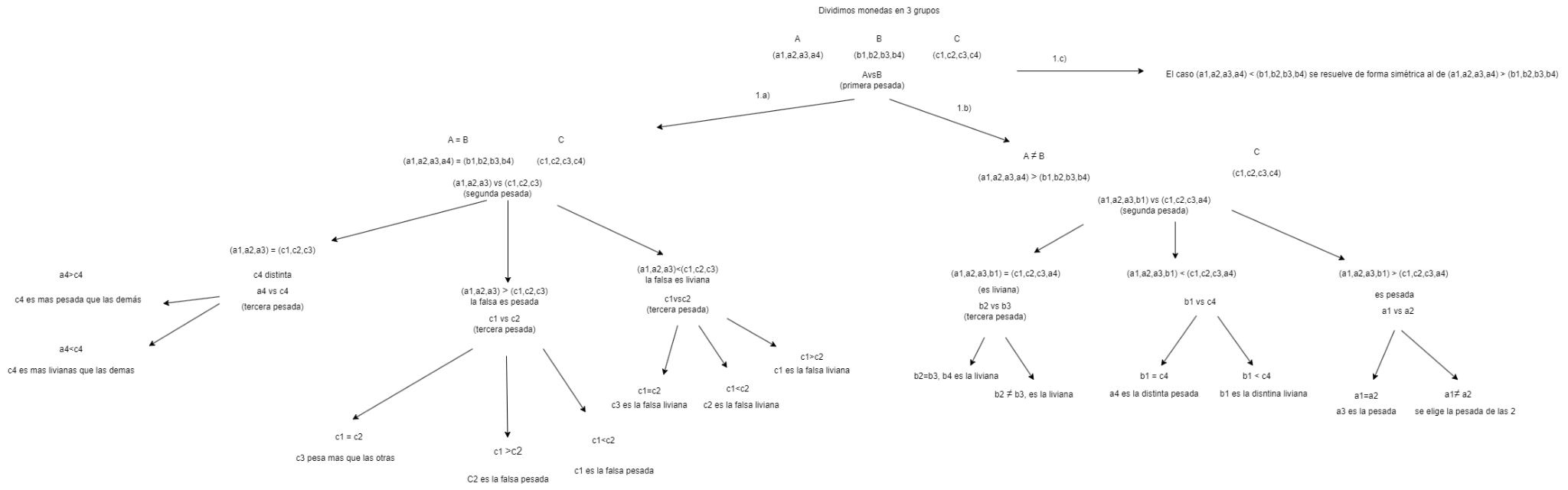
[](https://www.codecogs.com/eqnedit.php?latex=%20%5Clog_2(12)%20%3D%203.53%20%5Crightarrow%203%20%5Ctext%7B%20pesadas%7D#0)









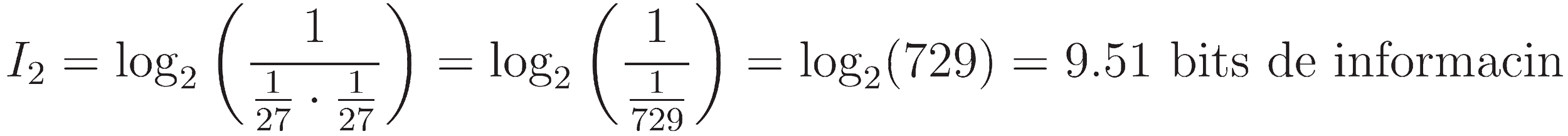


1. ¿Cuál es el contenido de información de una letra del abecedario de 27 letras? ¿Cuál es el de las letras tomadas de a dos y de a tres?, suponiendo a priori, que no existe preferencia por ninguna letra.

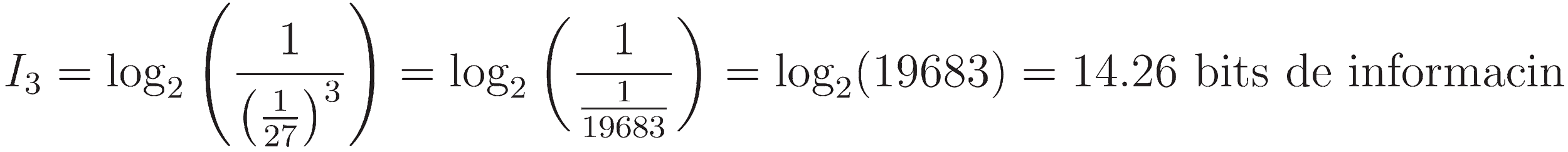
Para una letra la cantidad de información será de:

[](http://www.sciweavers.org/tex2img.php?bc=Transparent&fc=Black&im=jpg&fs=100&ff=modern&edit=0&eq=I_1%20%3D%20%5Clog_2(27)%20%3D%204.75%20%5Ctext%7B%20bits%20de%20informaci%C3%B3n%7D#0)

Si tomamos las letras de 2 en 2:

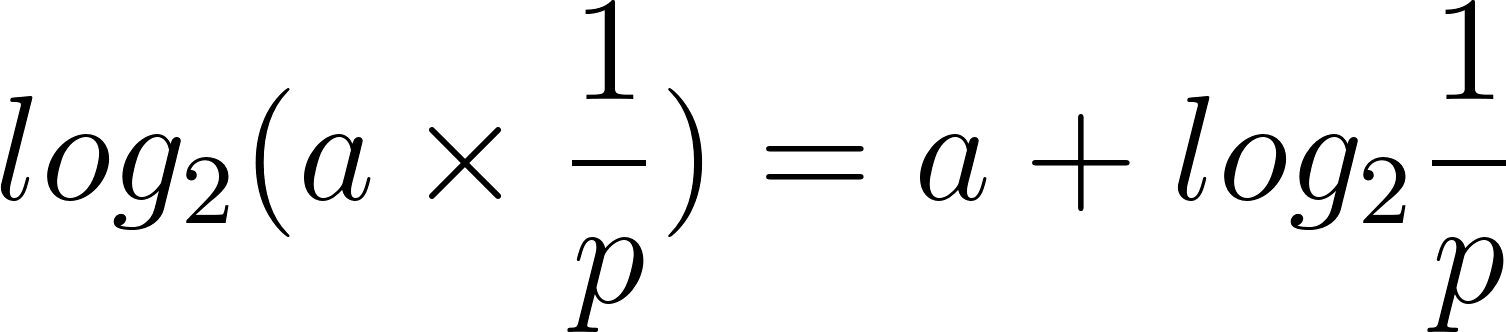
[](http://www.sciweavers.org/tex2img.php?bc=Transparent&fc=Black&im=jpg&fs=100&ff=modern&edit=0&eq=I_2%20%3D%20%5Clog_2%5Cleft(%5Cfrac%7B1%7D%7B%5Cfrac%7B1%7D%7B27%7D%20%5Ccdot%20%5Cfrac%7B1%7D%7B27%7D%7D%5Cright)%20%3D%20%5Clog_2%5Cleft(%5Cfrac%7B1%7D%7B%5Cfrac%7B1%7D%7B729%7D%7D%5Cright)%20%3D%20%5Clog_2(729)%20%3D%209.51%20%5Ctext%7B%20bits%20de%20informaci%C3%B3n%7D#0)

Si tomamos las letras de 3 en 3:

[](http://www.sciweavers.org/tex2img.php?bc=Transparent&fc=Black&im=jpg&fs=100&ff=modern&edit=0&eq=I_3%20%3D%20%5Clog_2%5Cleft(%5Cfrac%7B1%7D%7B%5Cleft(%5Cfrac%7B1%7D%7B27%7D%5Cright)%5E3%7D%5Cright)%20%3D%20%5Clog_2%5Cleft(%5Cfrac%7B1%7D%7B%5Cfrac%7B1%7D%7B19683%7D%7D%5Cright)%20%3D%20%5Clog_2(19683)%20%3D%2014.26%20%5Ctext%7B%20bits%20de%20informaci%C3%B3n%7D#0)

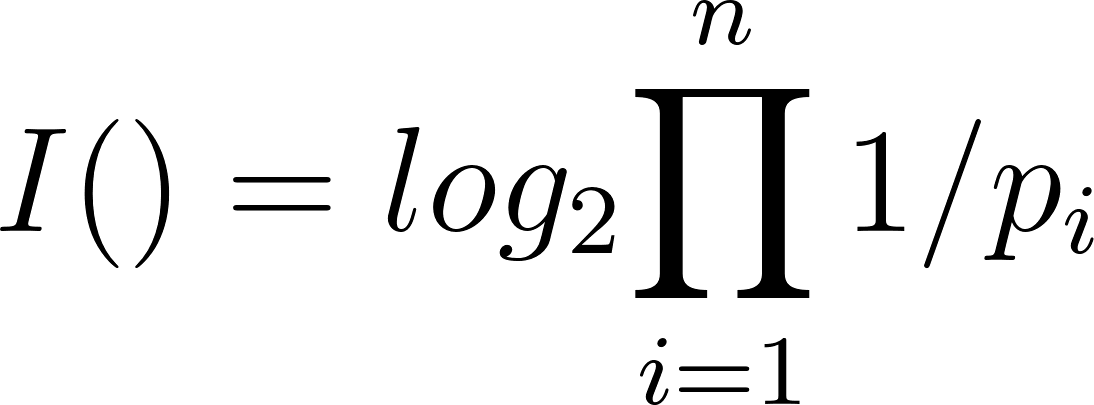
1. ¿Cómo serán las cifras anteriores respecto de la forma castellana de escritura cotidiana? Si a su criterio existiera diferencia ¿a qué se debería?

En el lenguaje castellano estos valores variarían drásticamente, ya que en el lenguaje español las letras no aparecen con equiprobabilidad, y al tomarlas en patrones de 2 o 3 encontraríamos constantes como por ejemplo la “u” seguida siempre de una “q”.

1. Explica las propiedades de la cantidad de información. Da un ejemplo para cada una de estas propiedades.
2. Es siempre positiva.  
   Ej. La cantidad de información transmitida por una imagen de 1920x1080 con 24 niveles de brillo es de 9497088 bits de información (valor positivo).
3. Es inversamente proporcional a la probabilidad de ocurrencia del evento.  
   [](https://www.codecogs.com/eqnedit.php?latex=log_2(%7Ba%20%5Ctimes%20%7B1%20%5Cover%20p%7D%7D)%20%3D%20a%20%2B%20log_2%7B1%20%5Cover%20p%7D#0)
4. Aumenta con la cantidad posible de eventos o mensajes.  
   Para 150 caracteres de un vocabulario de 100 la cantidad de información es:

[](http://www.sciweavers.org/tex2img.php?bc=Transparent&fc=Black&im=jpg&fs=100&ff=modern&edit=0&eq=I%20%3D%20%5Clog_2(100%5E%7B150%7D)%20%3D%20996.578428466%20%5Ctext%7B%20bits%20de%20informaci%C3%B3n%7D#0)

En cambio si se usan 250 caracteres:  
[](http://www.sciweavers.org/tex2img.php?bc=Transparent&fc=Black&im=jpg&fs=100&ff=modern&edit=0&eq=I%20%3D%20%5Clog_2(100%5E%7B250%7D)%20%3D%201660.96404744%20%5Ctext%7B%20bits%20de%20informaci%C3%B3n%7D#0)

1. La cantidad de información de eventos o mensajes independientes A, B, …, C; es aditiva, I(A, B, …, C) = I(A) + I(B) + … + I(C).  
   [](https://www.codecogs.com/eqnedit.php?latex=I%20()%20%3D%20log_2%7B%5Cprod_%7Bi%3D1%7D%5En%7B1%2Fp_i%7D%7D#0)
2. La cantidad de información promedio (al que se denomina Entropía) se maximiza ante sucesos equiprobables.

Ejemplo: al tener un dado con 8 caras todos con equiprobabilidad (pi = ⅛) tenemos que:

[](http://www.sciweavers.org/tex2img.php?bc=Transparent&fc=Black&im=jpg&fs=100&ff=modern&edit=0&eq=I%20%3D%20%5Clog_2(8)%20%3D%203%20%5Ctext%7B%20bits%20de%20informaci%C3%B3n%20para%20cada%20cara%7D#0)

Al calcular la cantidad de información promedio (Entropía) tenemos:

[](http://www.sciweavers.org/tex2img.php?bc=Transparent&fc=Black&im=jpg&fs=100&ff=modern&edit=0&eq=H%20%3D%20%5Cleft(%20%5Cfrac%7B1%7D%7B8%7D%20%5Ctimes%203%20%5Cright)%20%5Ctimes%208%20%3D%203%20%5Ctext%7B%20bits%20de%20informaci%C3%B3n%20(maximizada)%20al%20tener%20equiprobabilidad%7D#0)

1. Dada una variedad V = 1000 sucesos encontrar la base óptima de representación que hace mínima la siguiente relación número más corto, es decir que minimice I . b (cantidad de información por base).

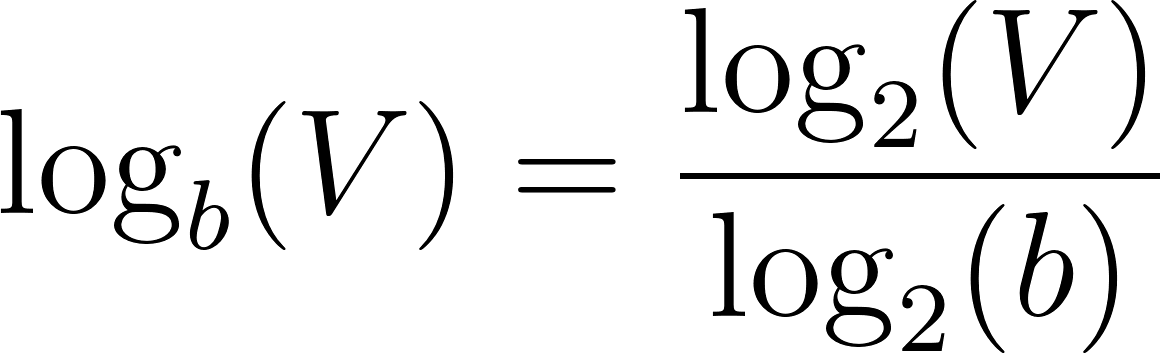
Cantidad de información I en bits para representar cada suceso en una base b

[](https://www.codecogs.com/eqnedit.php?latex=%20I%20%3D%20%5Clog_b(V)%20#0)

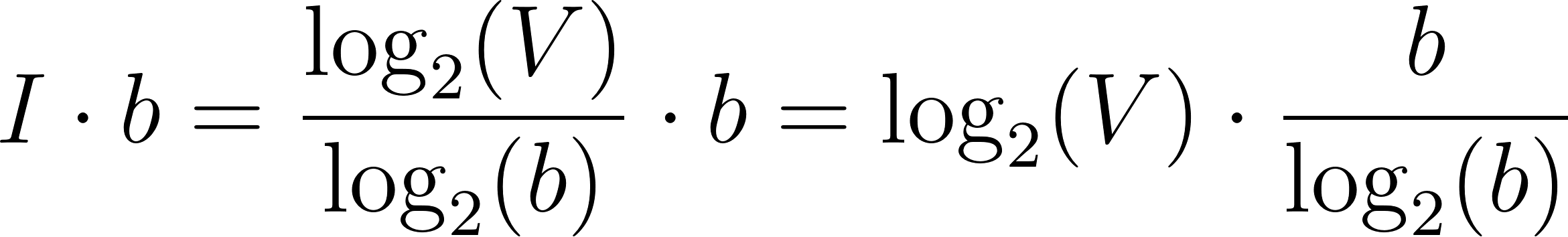
Producto a minimizar

[](https://www.codecogs.com/eqnedit.php?latex=%20I%20%5Ccdot%20b%20%3D%20%5Clog_b(V)%20%5Ccdot%20b%20#0)

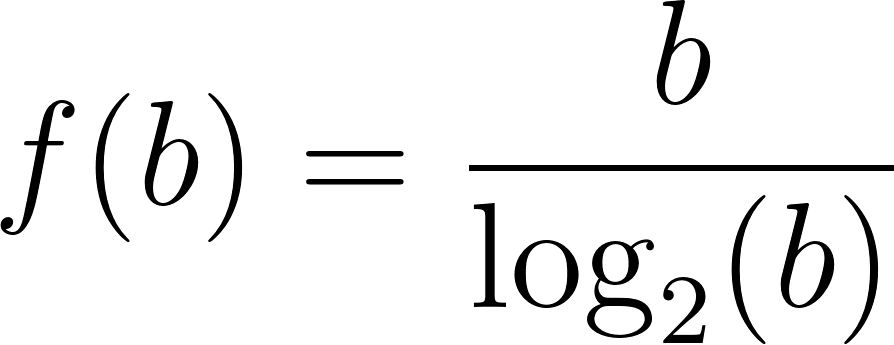
Cambio de base

[](https://www.codecogs.com/eqnedit.php?latex=%20%5Clog_b(V)%20%3D%20%5Cfrac%7B%5Clog_2(V)%7D%7B%5Clog_2(b)%7D%20#0)

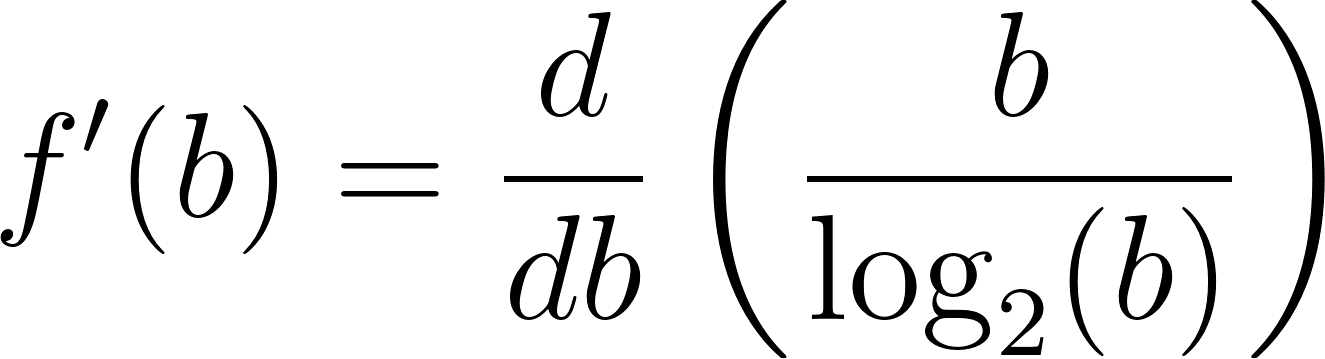
Producto con cambio de base

[](https://www.codecogs.com/eqnedit.php?latex=%20I%20%5Ccdot%20b%20%3D%20%5Cfrac%7B%5Clog_2(V)%7D%7B%5Clog_2(b)%7D%20%5Ccdot%20b%20%3D%20%5Clog_2(V)%20%5Ccdot%20%5Cfrac%7Bb%7D%7B%5Clog_2(b)%7D%20#0)

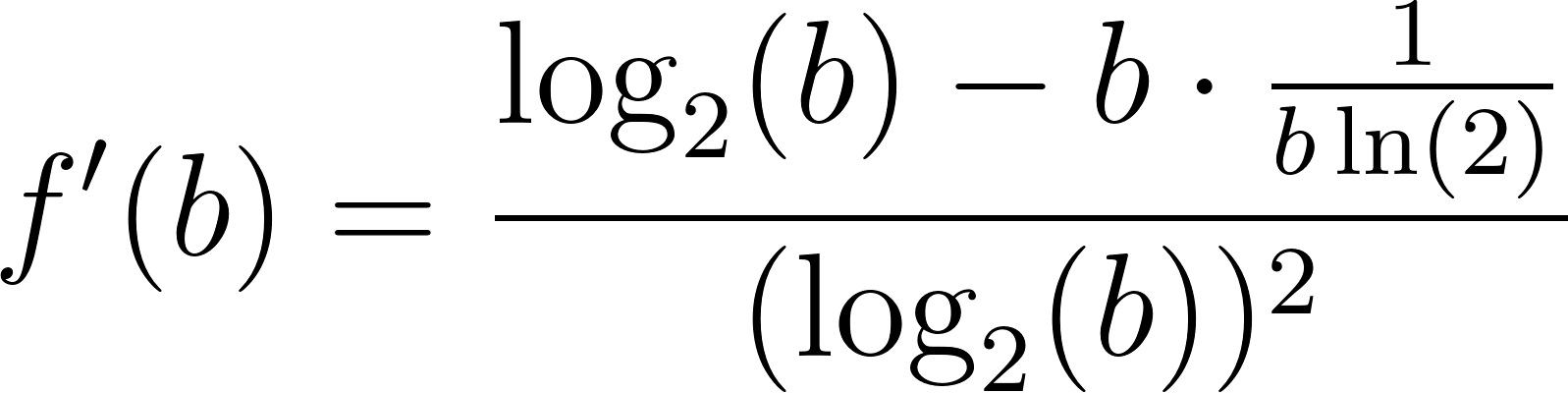
% Función a minimizar

[](https://www.codecogs.com/eqnedit.php?latex=%20f(b)%20%3D%20%5Cfrac%7Bb%7D%7B%5Clog_2(b)%7D%20#0)

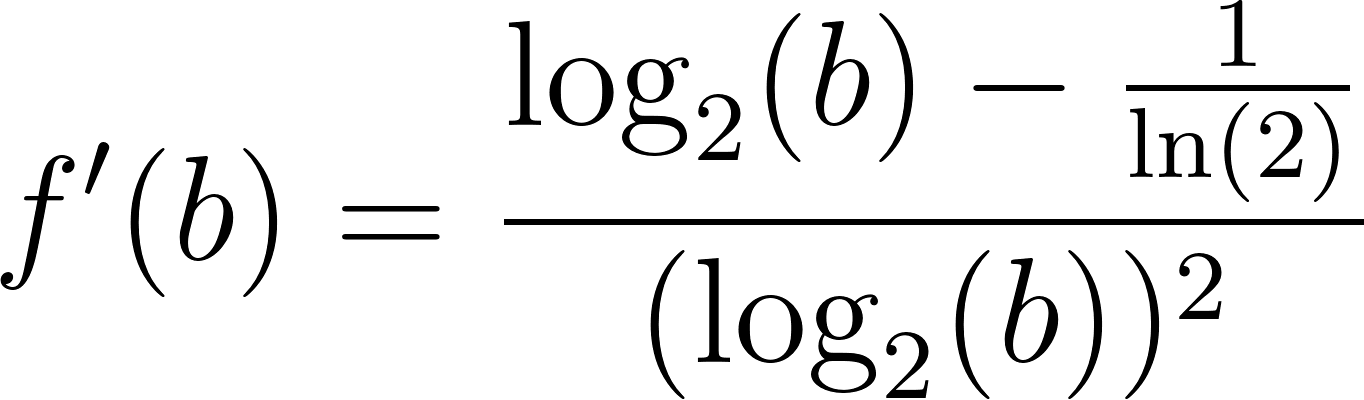
Derivada de f(b) con respecto a b

[](https://www.codecogs.com/eqnedit.php?latex=%20f'(b)%20%3D%20%5Cfrac%7Bd%7D%7Bdb%7D%5Cleft(%5Cfrac%7Bb%7D%7B%5Clog_2(b)%7D%5Cright)%20#0)

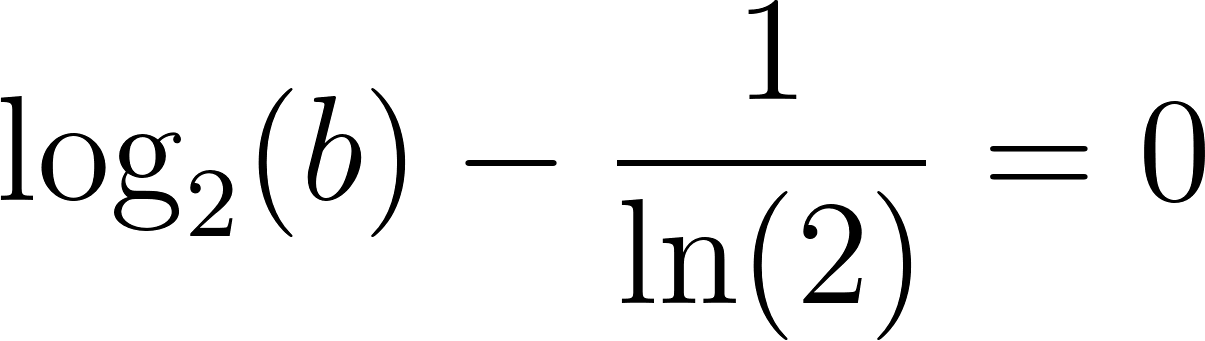
Derivada aplicada

[](https://www.codecogs.com/eqnedit.php?latex=%20f'(b)%20%3D%20%5Cfrac%7B%5Clog_2(b)%20-%20b%20%5Ccdot%20%5Cfrac%7B1%7D%7Bb%20%5Cln(2)%7D%7D%7B(%5Clog_2(b))%5E2%7D%20#0)

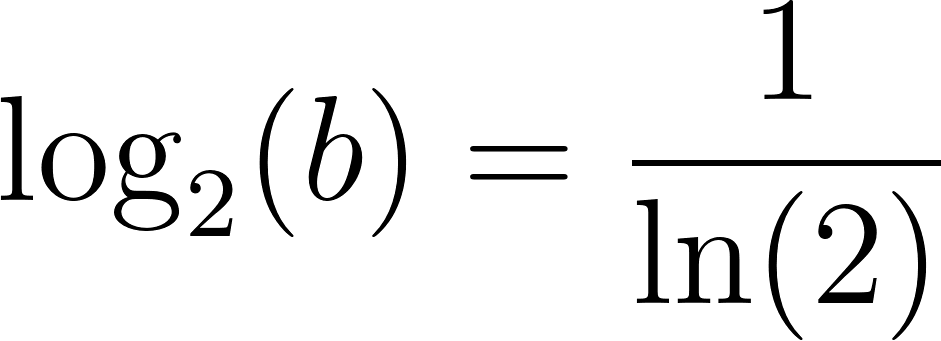
Simplificación de la derivada

[](https://www.codecogs.com/eqnedit.php?latex=%20f'(b)%20%3D%20%5Cfrac%7B%5Clog_2(b)%20-%20%5Cfrac%7B1%7D%7B%5Cln(2)%7D%7D%7B(%5Clog_2(b))%5E2%7D%20#0)

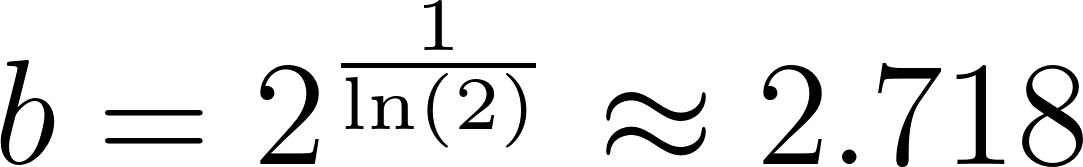
Igualación de la derivada a cero para encontrar el valor óptimo de b

[](https://www.codecogs.com/eqnedit.php?latex=%20%5Clog_2(b)%20-%20%5Cfrac%7B1%7D%7B%5Cln(2)%7D%20%3D%200%20#0)

Solución para b

[](https://www.codecogs.com/eqnedit.php?latex=%20%5Clog_2(b)%20%3D%20%5Cfrac%7B1%7D%7B%5Cln(2)%7D%20#0)

Valor óptimo de b

[](https://www.codecogs.com/eqnedit.php?latex=b%20%3D%202%5E%7B%5Cfrac%7B1%7D%7B%5Cln(2)%7D%7D%20%5Capprox%202.718#0)

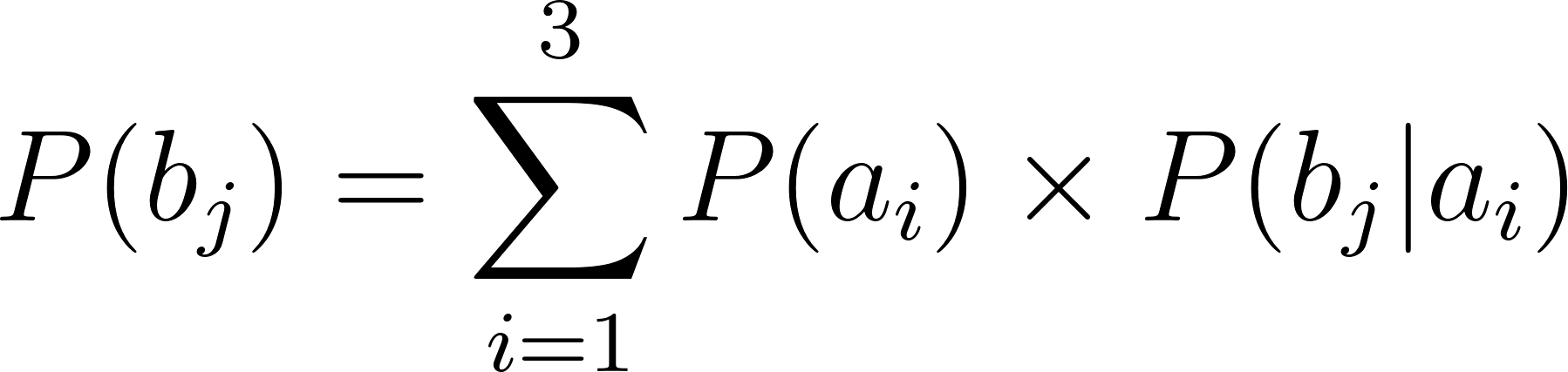
**PRÁCTICO 2 CANAL DE INFORMACIÓN**

1. Sea el siguiente canal:

|  | b1 | b2 | b3 |
| --- | --- | --- | --- |
| a1 | 0,7 | 0,2 | 0,1 |
| a2 | 0,3 | 0,4 | 0,3 |
| a3 | 0,5 | 0,3 | 0,2 |

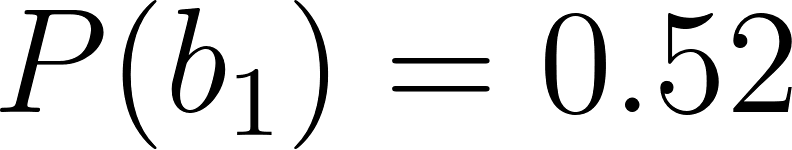
Calcular los valores de p(ai/bj) y las probabilidades de salida para el caso particular de p(a1)=0.4, p(a2)=0.3, p(a3)=0.3

Calculamos las probabilidades de salida de cada bj

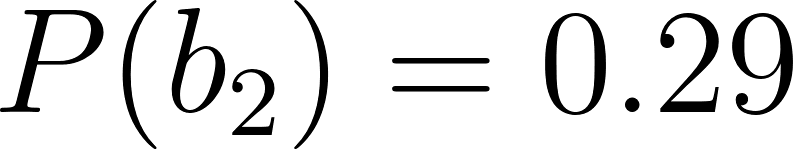
[](https://www.codecogs.com/eqnedit.php?latex=%20P(b_j)%20%3D%20%5Csum_%7Bi%3D1%7D%20%5E%203%20%7BP(a_i)%20%5Ctimes%20P(b_j%20%5Cvert%20a_i)%7D%20#0)

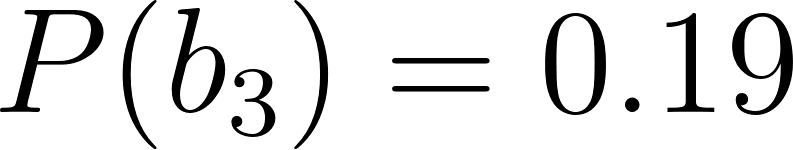
Por ejemplo, para b1:

[](https://www.codecogs.com/eqnedit.php?latex=%20P(b_1)%20%3D%200.4%20%5Ctimes%200.7%20%2B%200.3%20%5Ctimes%200.3%20%2B%200.3%20%5Ctimes%200.5%20#0)

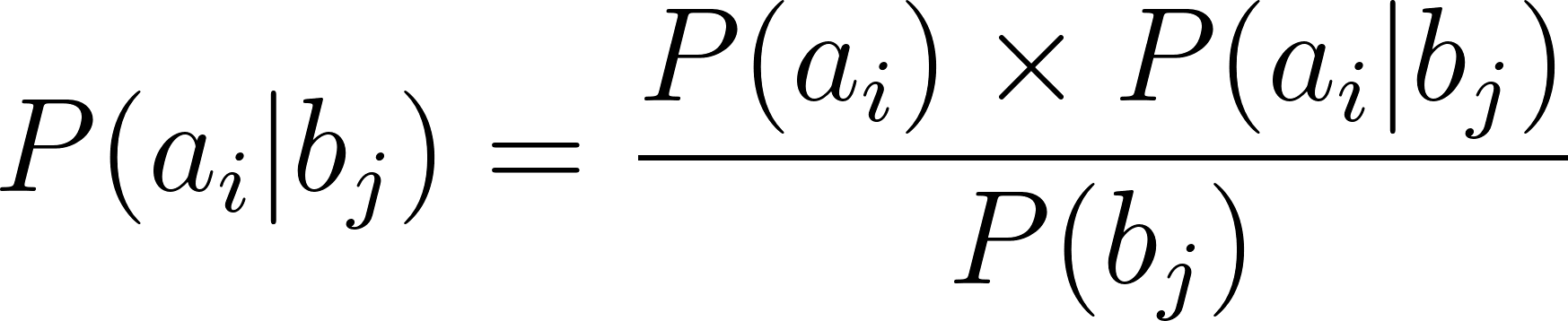
[](https://www.codecogs.com/eqnedit.php?latex=%20P(b_1)%20%3D%200.52%20#0)

Repitiendo para b2 y b3, tenemos:

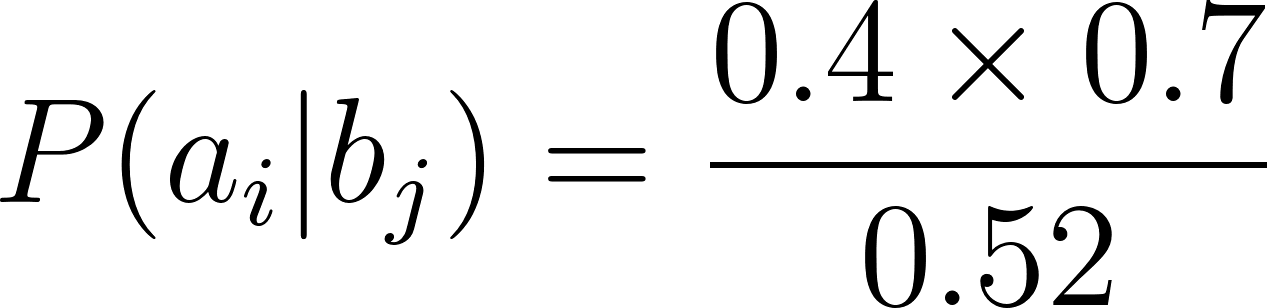
[](https://www.codecogs.com/eqnedit.php?latex=%20P(b_2)%20%3D%200.29%20#0)

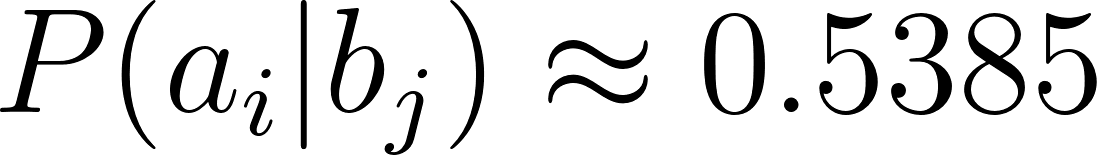
[](https://www.codecogs.com/eqnedit.php?latex=%20P(b_3)%20%3D%200.19%20#0)

Luego, para obtener las probabilidades condicionales P(ai | bj) aplicamos

[](https://www.codecogs.com/eqnedit.php?latex=%20P(a_i%20%5Cvert%20b_j)%20%3D%20%7BP(a_i)%20%5Ctimes%20P(a_i%20%5Cvert%20b_j)%20%20%5Cover%20P(b_j)%7D%20#0)

Por ejemplo, para P(a1 | b1).

[](https://www.codecogs.com/eqnedit.php?latex=%20P(a_i%20%5Cvert%20b_j)%20%3D%20%7B0.4%20%5Ctimes%200.7%20%5Cover%200.52%7D%20#0)

[](https://www.codecogs.com/eqnedit.php?latex=%20P(a_i%20%5Cvert%20b_j)%20%5Capprox%200.5385%20#0).

Repitiendo para todos los bj y ai tenemos

|  | a1 | a2 | a3 |
| --- | --- | --- | --- |
| b1 | 0.5385 | 0.1730 | 0.2885 |
| b2 | 0.2759 | 0.4137 | 0.3103 |
| b3 | 0.2105 | 0.4737 | 0.3158 |

1. Considera un Canal Binario Asimétrico con las siguientes probabilidades:

* p(a1)=1/3
* p(b1/a1)=5/6
* p(b1/a2)=2/6

a) Calcula las probabilidades condicionales hacia atrás y las probabilidades conjuntas.

b) Obtén la entropía del emisor, y las entropías condicionales de la fuente para

cualquier símbolo de salida.

c) Calcula la información mutua y la capacidad del canal.

a)

Tabla de datos iniciales:

|  | b1 | b2 |
| --- | --- | --- |
| a1 | 5/6 | 1/6 |
| a2 | 2/6 | 4/6 |

P(a1) = ⅓ , P(a2) = ⅔

Tabla condicionales hacia atras:

|  | a1 | a2 |
| --- | --- | --- |
| b1 | 5/9 | 4/9 |
| b2 | 1/9 | 8/9 |

P(b1) = P(b1/a1)\*P(a1) + P(b1/a2)\*P(a2) = 5/6 \* 1/3 + 2/6 \* 2/3 = 1/2

P(b2) = P(b2/a1)\*P(a1) + P(b2/a2)\*P(a2) = 1/6 \* 1/3 + 4/6 \* 2/3 = 1/2

Luego:

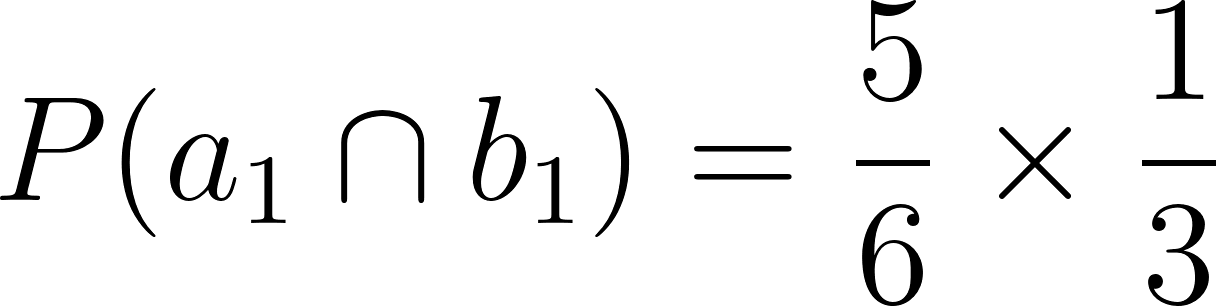
P(a1/b1)= (P(b1/a1)\*P(a1))/P(b1) = (5/6 \* 1/3)/(1/2) = 5/9

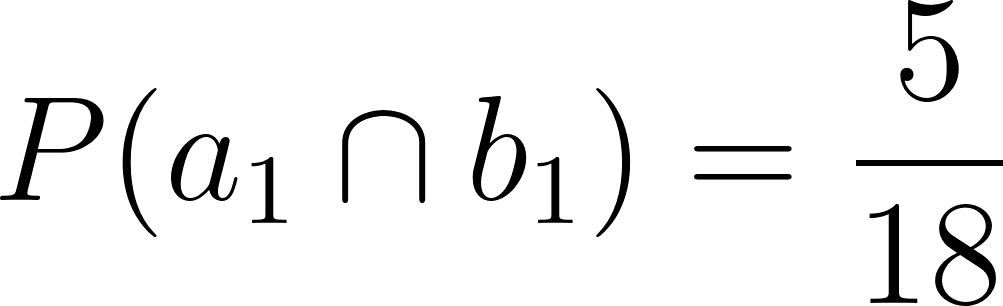
De la misma forma resolvemos los demás

Luego calculamos las probabilidades conjuntas:

[](https://www.codecogs.com/eqnedit.php?latex=%20P(a_i%20%5Ccap%20b_j)%20%3D%20P(b_j%20%5Cvert%20a_i)%20%5Ctimes%20P(a_i)%20#0)

Por ejemplo:

[](https://www.codecogs.com/eqnedit.php?latex=%20P(a_1%20%5Ccap%20b_1)%20%3D%20%7B5%20%5Cover%206%7D%20%5Ctimes%20%7B1%20%5Cover%203%7D%20%20#0)

[](https://www.codecogs.com/eqnedit.php?latex=%20P(a_1%20%5Ccap%20b_1)%20%3D%20%7B5%20%5Cover%2018%7D%20%20#0)

Obteniendo para todas, queda:

|  | b1 | b2 |
| --- | --- | --- |
| a1 | 5/18 | 1/18 |
| a2 | 2/9 | 4/9 |

b) Entropía del emisor:

H(A) = 1/3\*log2(3)+2/3\*log2(3/2) = 0.9182958341

Entropías condiciones de la fuente para cualquier símbolo de salida:

H(A/B) =

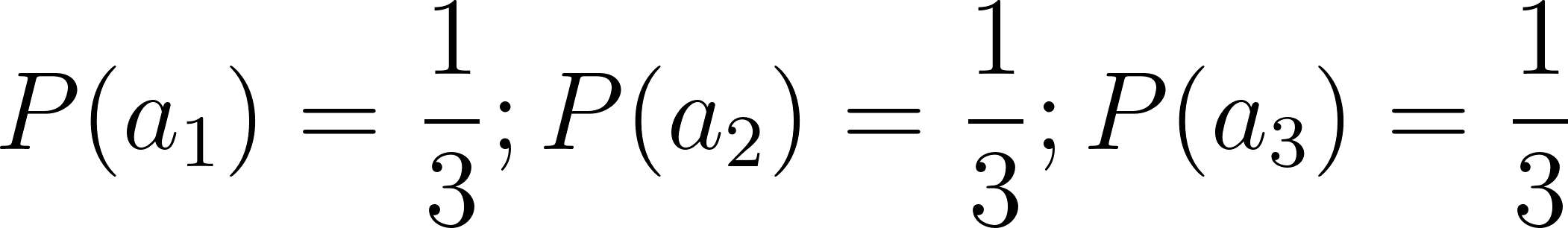
0.7471671973

c)

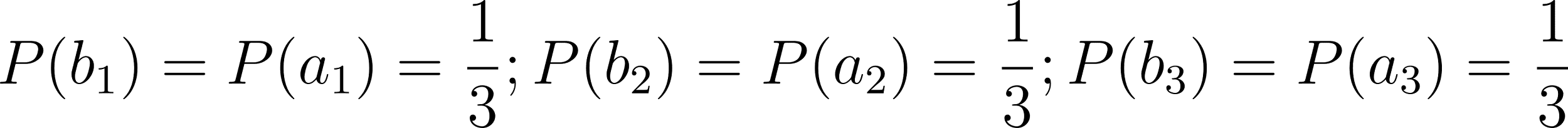
I(A,B) = 0.1711286368

1. Considera un canal determinista con tres símbolos de entrada (a1,a2,a3) y tres símbolos de salida (b1,b2,b3), donde cada símbolo de entrada se corresponde exclusivamente con un símbolo de salida. Define las probabilidades para cada símbolo de entrada y calcula la información mutua.

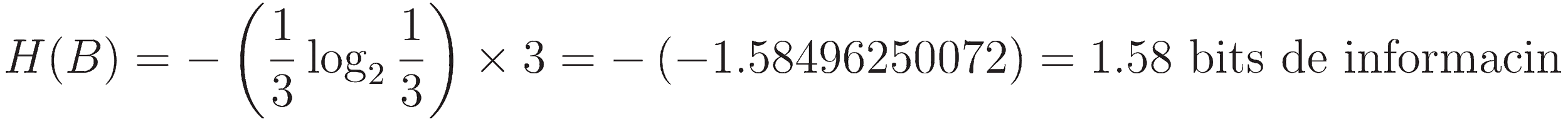
Consideremos una equiprobabilidad, donde la probabilidad de cada entrada es ⅓ dado que hay 3 entradas

[](https://www.codecogs.com/eqnedit.php?latex=P(a_1)%20%3D%20%5Cfrac%7B1%7D%7B3%7D%3B%20%5C%5C%20P(a_2)%20%3D%20%5Cfrac%7B1%7D%7B3%7D%3B%20%5C%5C%20P(a_3)%20%3D%20%5Cfrac%7B1%7D%7B3%7D#0)

Para las salidas, dado que tenemos un canal determinista, la probabilidad de la salida será igual al de su entrada correspondiente, teniendo:

[](https://www.codecogs.com/eqnedit.php?latex=P(b_1)%20%3D%20P(a_1)%20%3D%20%5Cfrac%7B1%7D%7B3%7D%3B%20%5C%5CP(b_2)%20%3D%20P(a_2)%20%3D%20%5Cfrac%7B1%7D%7B3%7D%3B%20%5C%5CP(b_3)%20%3D%20P(a_3)%20%3D%20%5Cfrac%7B1%7D%7B3%7D#0)

Calculamos la entropía

[](http://www.sciweavers.org/tex2img.php?bc=Transparent&fc=Black&im=jpg&fs=100&ff=modern&edit=0&eq=H(B)%20%3D%20-%20%5Cleft(%20%5Cfrac%7B1%7D%7B3%7D%20%5Clog_2%20%5Cfrac%7B1%7D%7B3%7D%20%5Cright)%20%5Ctimes%203%20%3D%20-%20%5Cleft(%20-1.58496250072%20%5Cright)%20%3D%201.58%20%5Ctext%7B%20bits%20de%20informaci%C3%B3n%7D#0)

En este caso la entropía condicional será H(B/A) = 0 ya que estamos ante un canal determinista, por lo que a cada entrada se le mapea una salida, por ende al aplicar la fórmula de la información mutua resulta:

[](http://www.sciweavers.org/tex2img.php?bc=Transparent&fc=Black&im=jpg&fs=100&ff=modern&edit=0&eq=I(A%2C%20B)%20%3D%20H(A)%20-%20H(A%20%5Cmid%20B)%20%3D%20H(B)%20-%20H(B%20%5Cmid%20A)%20%3D%201.58%20-%200%20%3D%201.58%20%5Ctext%7B%20bits%20de%20informaci%C3%B3n%7D#0)

Observamos que la información mutua obtenida es igual a la entropía de B

1. Dada la siguiente matriz de canal, obtén las respectivas codificaciones de la fuente.

p(a)=1/4, p(b)=1/8, p(c)=5/8

|  | a | b | c |
| --- | --- | --- | --- |
| a | 1/4 | 2/4 | 1/4 |
| b | 1/5 | 3/5 | 1/5 |
| c | 1/3 | 1/3 | 1/3 |

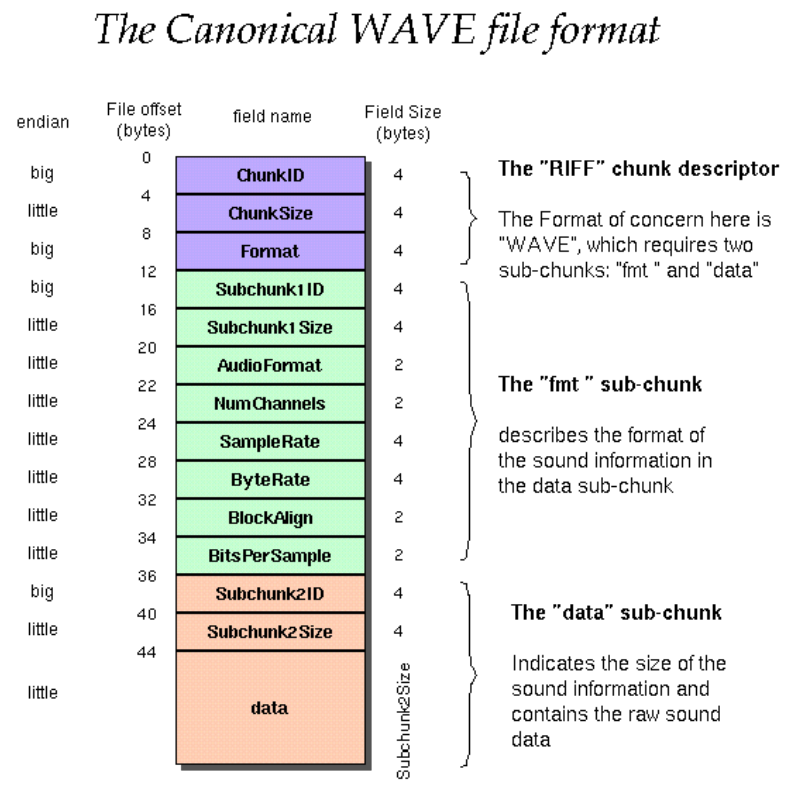
**PRÁCTICO DE MÁQUINA 1**

1. Escribir un programa en lenguaje a elección, que permita:
   1. Ingresar el nombre de un archivo con extensión .wav
   2. Valide que el archivo sea con el formato adecuado.
   3. Muestre los datos de la cabecera del archivo .wav

### ***Formato Wav.***

El formato Wave es un subconjunto de la especificación RIFF de Microsoft para formatos multimedia. Los formatos RIFF se caracterizan por una cabecera más una secuencia de bloques de datos.

En este sentido el formato Wav divide el archivo en dos bloques, un primer bloque que representa el formato de los datos así como su muestra, y un segundo bloque con los datos propiamente dichos.



| OFFSET | SIZE | NAME | DESCRIPTION |
| --- | --- | --- | --- |
| The canonical WAVE format starts with the RIFF header: | | | |
| 0 | 4 | ChunkID | Contains the letters "RIFF" in ASCII form  (0x52494646 big-endian form). |
| 4 | 4 | ChunkSize | 36 + SubChunk2Size, or more precisely:  4 + (8 + SubChunk1Size) + (8 + SubChunk2Size)  This is the size of the rest of the chunk  following this number. This is the size of the entire file in bytes minus 8 bytes for the two fields not included in this count:  ChunkID and ChunkSize. |
| 8 | 4 | Format | Contains the letters "WAVE" (0x57415645 big-endian form) |
| The "WAVE" format consists of two subchunks: "fmt " and "data":  The "fmt " subchunk describes the sound data's format: | | | |
| 12 | 4 | Subchunk1ID | Contains the letters "fmt " (0x666d7420 big-endian form) |
| 16 | 4 | Subchunk1Size | 16 for PCM. This is the size of the rest of the Subchunk which follows this number |
| 20 | 2 | AudioFormat | PCM = 1 (i.e. Linear quantization)  Values other than 1 indicate some form of compression |
| 22 | 2 | NumChannels | Mono = 1, Stereo = 2, etc |
| 24 | 4 | SampleRate | 8000, 44100, etc |
| 28 | 4 | ByteRate | == SampleRate \* NumChannels \* BitsPerSample/8 |
| 32 | 2 | BlockAlign | == NumChannels \* BitsPerSample/8  The number of bytes for one sample including  all channels. I wonder what happens when  this number isn't an integer? |
| 34 | 2 | BitsPerSample | 8 bits = 8, 16 bits = 16, etc |
| 2 | ExtraParamSize | if PCM, then doesn't exist |
| x | ExtraParams | space for extra parameters |
| The "data" subchunk contains the size of the data and the actual sound: | | | |
| 36 | 4 | Subchunk2ID | Contains the letters "data" (0x64617461 big-endian form) |
| 40 | 4 | Subchunk2Size | == NumSamples \* NumChannels \* BitsPerSample/8  This is the number of bytes in the data.  You can also think of this as the size of the read of the subchunk following this number. |
| 44 | \* | Data | The actual sound data |

1. Escribir un programa en lenguaje a elección, que permita:
   1. Ingresar el nombre de un archivo con extensión .bmp
   2. Valide que el archivo sea con el formato adecuado.
   3. Muestre los datos de la cabecera del archivo .bmp

Un archivo gráfico en formato BMP (bitmap) tiene la estructura que se ilustra en la siguiente figura:

| Cabecera (14 bytes) |
| --- |
| Propiedades de la imagen (40 bytes) |
| Paleta de color (opcional – su tamaño puede variar) |
| Datos de la imagen |

En la cabecera se incluye la siguiente información:

| Campo | bytes | Descripción |
| --- | --- | --- |
| Signature | 2 | Siempre es “BM” |
| FileSize | 4 | Tamaño del fichero |
| Reserved | 4 | No se usa |
| DataOffset | 4 | Posición relativa del comienzo de los datos de imagen |
| En cuanto a la zona de propiedades de la imagen, podemos encontrar la siguiente información: | | |
| Size | 4 | Tamaño de esta sección (siempre es 40 bytes) |
| Width | 4 | Anchura de la imagen en pixels |
| Height | 4 | Altura de la imagen en pixels |
| Planes | 2 | Número de planos (siempre es 1) |
| BitCount | 2 | Bits por píxel (1, 4, 8, 16, 24) |
| Compression | 4 | Tipo de compresión empleado (0, 1, 2) |
| ImageSize | 4 | Tamaño de la imagen comprimida (=0 si no se comprime) |
| XPixelsPerM | 4 | Resolución horizontal: píxeles por metro |
| YPixelsPerM | 4 | Resolución vertical: píxeles por metro |
| ColorsUsed | 4 | Número de colores usados, si hay paleta. |
| ColorsImportant | 4 | Número de colores importantes (=0 si son todos) |

1. Desarrollar una aplicación de software que calcule la entropía y redundancia, de una fuente con símbolos vistos en forma independiente y dependiente en O(1). Realizar comparaciones para diferentes archivos (\*.txt, \*.exe, \*.zip etc.)
2. Desarrollar una aplicación de software que calcule la Capacidad de Canal de un canal R-ario Uniforme y No-Uniforme. El soft debe aceptar como entrada el valor de R que identifica al canal (R= 2 Binario, R=3 Ternario hasta R=4) los valores de probabilidades condicionales que representan la matriz del canal entregando como salida el valor de las probabilidades independientes de cada uno de los símbolos de entrada que maximiza la información mutua, esto es, lograr la capacidad de canal.
3. Desarrollar una aplicación de software que almacene en un archivo los siguientes datos de personas: Apellido y nombre, dirección, dni y 8 campos bivaluados, del tipo S/N o True/False, por ejemplo: estudios primarios (S/N), estudios secundarios (S/N), estudios universitarios (S/N), tiene vivienda propia (S/N), obra social (S/N), trabaja (S/N), etc. El programa desarrollado deberá permitir:

a) almacenar y recuperar los datos en un archivo con campos de longitud fija (archivo “fijos.dat”).

b) almacenar y recuperar los datos en un archivo de longitud variable (archivo “variable.dat”).

Ingresar los mismos datos en ambos archivos para unas 20 (veinte) personas, comparar el tamaño de ambos archivos.

1. Genere un informe que marque las diferencias esenciales (tipo de documento, si es un formato comprimido o no, tamaños, calidades, etc.) entre los siguientes documentos de texto \*.doc, \*.docx, \*.djvu, \*.pdf, \*.epub. Encuentre una aplicación que a un archivo de texto en cualquier formato pueda pasarlo a la mayoría de los restantes.
2. Desarrollar un programa para comparar dos cadenas, no en forma tradicional (carácter por carácter), sino que implemente un algoritmo, propuesto por Ud., que determine el parecido, por ejemplo de cadenas como: Juan Perez y Jaun Perez, Horacio López y Oracio López, cadenas que si se tratan en comparando carácter por carácter, son muy poco parecidas o incluso no se parecen en nada.
3. Implementar un programa que valide un CUIT/CUIL ingresado por teclado.
4. En la actualidad, muchos de los procesos que se ejecutan en una computadora requiere obtener o enviar información a otros procesos que se localizan en una computadora diferente, o en la misma. Para lograr esta comunicación se utilizan los protocolos de comunicación TCP y UDP. Una implementación de comunicación entre procesos, se puede realizar utilizando sockets.

Los sockets son una forma de comunicación entre procesos que se encuentran en diferentes máquinas de una red, los sockets proporcionan un punto de comunicación por el cual se puede enviar o recibir información entre procesos.

Los sockets tienen un ciclo de vida dependiendo si son sockets de servidor, que esperan a un cliente para establecer una comunicación, o socket cliente que busca a un socket de servidor para establecer la comunicación.

Haciendo uso de sockets, implemente un servidor y un cliente (a modo de ejemplo, se proveen un servidor y un cliente en lenguaje Python), que permita desde el cliente, enviar un archivo comprimido utilizando el alfabeto ABCDEFGH, y en el servidor, descomprimir el archivo.

Servidor.py

| import socket def conexiones(socket\_cliente):  peticion = socket\_cliente.recv(1024)  print ("[\*] Mensaje recibido: %s" % peticion)  respuesta = commands.getoutput(peticion)  socket\_cliente.send(respuesta)  socket\_cliente.close() ip = "0.0.0.0" puerto = 5555 max\_conexiones = 5 servidor = socket.socket(socket.AF\_INET, socket.SOCK\_STREAM) servidor.bind((ip, puerto)) servidor.listen(max\_conexiones) print ("[\*] Esperando conexiones en %s:%d" % (ip, puerto)) cliente, direccion = servidor.accept() while True:  print ("[\*] Conexion establecida con %s:%d" % (direccion[0] , direccion[1]))  #Recibimos el mensaje, con el metodo recv recibimos datos y como parametro  #la cantidad de bytes para recibir  recibido = cliente.recv(1024)  #Si el mensaje recibido es la palabra close se cierra la aplicacion  if recibido == "close":  break  #Si se reciben datos nos muestra la IP y el mensaje recibido  print (str(direccion[0]) + " dice: ", recibido)  #Devolvemos el mensaje al cliente  cliente.send(recibido) print ("Adios.") #Cerramos la instancia del socket cliente y servidor cliente.close() servidor.close() |
| --- |

Cliente.py

| import socket #Servidor y Cliente se ejecutan en la misma máquina, conocida como localhost #su dirección es 127.0.0.1 servidor = "127.0.0.1" puerto = 5555 cliente = socket.socket(socket.AF\_INET, socket.SOCK\_STREAM) cliente.connect((servidor, puerto)) Mensaje = "hola mundo" cliente.send(Mensaje.encode('utf-8')); respuesta = cliente.recv(4096) print ("[\*] Respuesta recibida: "+str(respuesta)) Mensaje = "Mande frutas" cliente.send(Mensaje.encode('utf-8')) respuesta = cliente.recv(4096) print ("[\*] Respuesta recibida: "+str(respuesta)) cliente.close() |
| --- |

**Documentación a Entregar:**

Se entrega el programa en funcionamiento, y el código fuente debidamente comentado. Pueden usar Colab o Jupyter notebooks para el desarrollo y luego compartir el archivo formato ipynb.

Fecha de entrega: 01 de Septiembre 00hs

Mail: teoriadelainformacion.fcefn@gmail.com