Laboratorio de codificación aritmética.

| Andrei Riaño, Andres Linares, Johann Castellanos, Universidad Santo Tomas. |
| --- |

A**bstract -- This is about the process of arithmetic codification, following step by step, it’s important because it is one of the most used coding in multimedia as the Huffman coding, but in this case the binary sequence is longer.**

**Keywords -- Codificación aritmética, números binarios, conversión de números, redondeo.**

I. INTRODUCCIÓN

Para poder mejorar la transmisión de datos, una de las partes fundamentales es su codificación, por lo que se hace necesario reconocer cómo se pueden codificar los datos de manera que sea efectiva y eficiente. Además el uso de herramientas de programación para realizar este tipo de tareas ayuda a desarrollar mejor la lógica y al mismo tiempo nos brinda la posibilidad de conocer de primera mano información sobre la codificación a los datos que ingresamos en el programa, así como su proceso, paso a paso.

En el presente trabajo se realiza un programa en uno de los lenguajes de programación seleccionados, teniendo en cuenta los requerimientos dados por el docente. De acuerdo a esto, los datos de ingreso pueden ser alfanuméricos, con un mensaje de máximo 10 caracteres y un alfabeto de máximo 5, donde se obtiene su valor a codificar, una recta que muestre sus probabilidades, y su resultado en binario.

II. MARCO TEÓRICO

Visto desde el punto de vista de la compresión de datos, hacer uso de la codificación aritmética resulta importante y de gran utilidad para el envío de datos multimedia. Para eso se deben seguir una serie de pasos descritos abajo:

1. Se requiere conocer la probabilidad de cada uno de los símbolos.
2. Las probabilidades son ordenadas en una recta de [0,1).
3. Para iniciar el proceso se debe tomar inicialmente el intervalo, [a-actual,b-actual) = [0,1).
4. Para cada símbolo a codificar se debe utilizar la fórmula:

1. Después de que se codifica cada símbolo, el valor de a-actual y b-actual se actualizan.
2. Finalmente se transforma el valor obtenido por el último a, se convierte en binario.
3. Para realizar la transformación a binario se debe llevar hasta la penúltima posición, donde se debe revisar que los datos sean iguales, para el último dígito se utiliza el redondeo, para de esta forma no utilizar tantos bits.

III. PROCEDIMIENTO

De acuerdo a los procesos que se deben seguir, el programa realiza la toma de datos, muestra la probabilidad de cada carácter por defecto, aunque esto se puede modificar, según se quiera. Para luego mostrar los datos del valor que se debe codificar y su equivalente en binario.

IV. FIGURAS

****

Figura 1. Imagen de inicio.

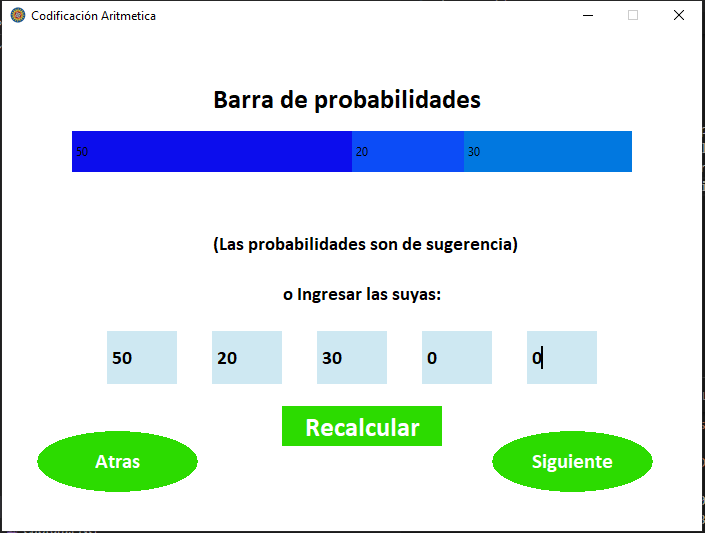


Figura 2. Probabilidades por defecto.



Figura 3. Resultados de la codificación.

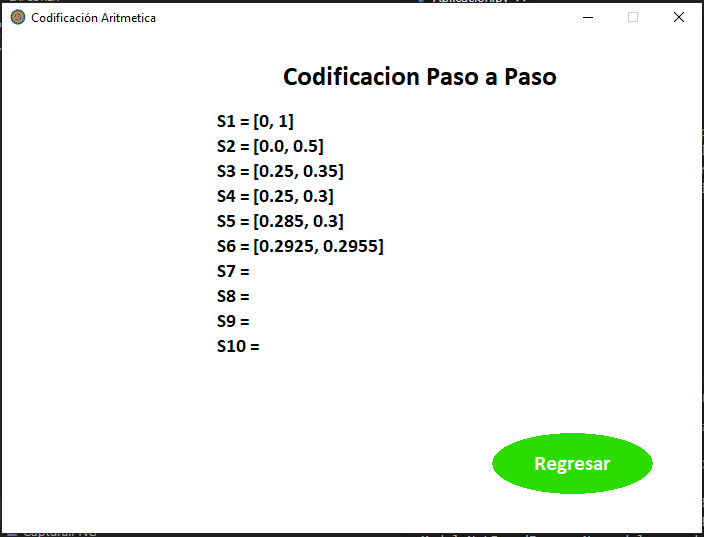


Figura 4. Muestra de los resultados de codificación paso a paso.

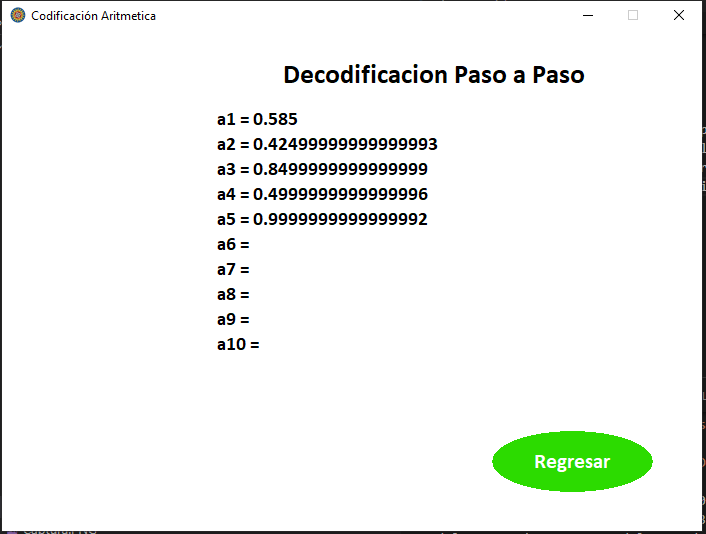


Figura 5. Decodificación paso a paso.

V. CONCLUSIONES

Antes de iniciar la codificación, se deben aclarar exactamente los objetivos que se pretenden lograr con el programa, como los datos específicos que se le deben proporcionar al usuario.

Durante el proceso de codificación y desarrollo se debe mantener un orden, además una claridad en las ideas, en lo que hace cada función, que parámetros reciben y que valores retornan.

Al igual que en otros lenguajes de programación hay que profundizar en el lenguaje y aprovechar las herramientas que nos proporciona, no hay que reinventar la rueda.

Siempre se debe tener un punto de referencia por medio del cual se puedan verificar resultados y concretar que el programa va por buen camino, además de tener la teoría muy clara antes de plasmarla en codificación o en otros ámbitos.

Es recomendable realizar los procesos de forma análitica, para así percibir los posibles errores al momento de realizar el programa.

VI. REFERENCIAS

[1] Apuntes de clase, “Sistemas de Telecomunicaciones II”,(2020) Codificación aritmética.