

Códigos de representación numérica y no numérica.

Según Dhanasekharan Natarajan (2020) la representación de la información en las computadoras es una de las partes más fundamentales de las ciencias de la computación. Todo dato procesado por las computadoras, sea este un texto, números, o imágenes, debe ser transformado a un formato que pueda ser almacenado y manipulado. [1]

Representación no numérica.

El código ASCII (American Standard Code for Information Interchange) es un sistema de codificación el cual se basa en asignar un valor numérico a caracteres. Por ejemplo, la letra “A” esta asignada al 65 en sistema decimal, que a su vez está representado como 01000001 en sistema binario. Este código es importante para que los caracteres alfabéticos, números y símbolos sean almacenados y procesados por la computadora. El código ASCII original utiliza 7 bits para almacenar hasta 128 caracteres, pero existen otras versiones como ASCII extendido y Unicode que nos permiten almacenar miles de símbolos. [2]

Representación numérica.

El código BCD (Binary-Coded Decimal) funciona de una manera muy diferente en este cada dígito en sistema decimal se representa por separado en sistema binario. Por ejemplo, el número 25 se representa en BCD como 0010 0101, como se puede observar cada uno de los dígitos se codifica de manera independiente. A pesar de que este método de codificación no es tan eficiente como la representación del sistema binario original, BCD se utiliza en otros sistemas donde la conversión directa entre el sistema decimal y el sistema binario debe ser más rápida y fácil, como en calculadoras y en algunos sistemas financieros. [3]

Códigos de detección y corrección de errores.

Al momento de transmitir, convertir y almacenar datos digitales, pueden existir ciertos errores debido a interferencias, ruido o fallos de hardware. Para solucionar estos errores se crearon códigos de detección y corrección de errores como los de paridad, Hamming, Reed-Solomon, entre otros. En los códigos de paridad se suele añadir un bit adicional que indica si el número total de bits “1” es par o impar. A diferencia de los códigos de Hamming los cuales detectan estos errores y al mismo tiempo son capaces de corregirlos. [4]

Según Dimitri Mortelmans (2024) una de las tendencias más actuales es la codificación es la clasificación automática de errores para introducir el código que más encaje según el tipo de información dada. Esto nos ayuda a optimizar el uso de la redundancia y mejorar la eficiencia en los sistemas de transmisión de datos. [5]

Bibliografías

- [1] D. Natarajan, *Fundamentals of Digital Electronics*, vol. 623. Cham: Springer International Publishing, 2020. doi: 10.1007/978-3-030-36196-9.
- [2] D. Sannella, M. Fourman, H. Peng, and P. Wadler, “Data Representation,” 2021, pp. 189–204. doi: 10.1007/978-3-030-76908-6_20.

- [3] H. Mora, M. T. Signes-Pont, F. A. P. López, J. Mora-Pascual, and J. M. G. Chamizo, “Advancements in number representation for high-precision computing,” *J Supercomput*, vol. 80, no. 7, pp. 9742–9761, May 2024, doi: 10.1007/s11227-023-05814-y.
- [4] R. Z. Cabada, H. M. C. López, and H. J. Escalante, “Methods for Data Representation,” in *Multimodal Affective Computing*, Cham: Springer International Publishing, 2023, pp. 55–65. doi: 10.1007/978-3-031-32542-7_5.
- [5] D. Mortelmans, “Coding with Classifications,” 2025, pp. 89–113. doi: 10.1007/978-3-031-66014-6_9.