1. **CÓDIGOS DE REPRESENTACIÓN NUMÉRICA Y NO NUMÉRICA**
   1. **CÓDIGOS DE REPRESENTACIÓN NUMÉRICA**

Los códigos de representación numérica son esquemas que representan datos cuantitativos (enteros o reales) en formatos binarios optimizados para operaciones eficientes. Un ejemplo destacado es la representación signed – digit, en la cual cada dígito puede ser positivo o negativo. Esto permite realizar operaciones, como la suma; en el tiempo constante sin propagación de acarreo, lo cual es crucial en procesadores de alto rendimiento y hardware especializado [1].

A continuación, se verán tipos de códigos de representación numérica:

* + 1. **Representación en aritmética paralela (Signed-Digit)**

La representación signet-digit (dígito con signo) es una técnica moderna donde cada posición puede tomar valores positivos o negativos, eliminando así la necesidad de propagar el acarreo entre dígitos. Esto permite ejecutar sumar y restas en tiempo constante, una característica clave en procesadores paralelos y hardware especializado. Su aplicación en sistemas embebidos y de procesamiento de señales es fundamental para optimizar el uso del hardware [2].

* + 1. **Convenciones de representación numérica (Overflow)**

Las convenciones de representación numérica definen como se gestionan los errores cuando los valores calculados exceden el rango permitido por el sistema. Esto se conoce como overflow, y su detección y manejo son fundamentales para preservar la integridad de los resultados. En arquitecturas modernas, estas convenciones se implementan desde el diseño del hardware o mediante lenguajes formales que detectan y validan limites numéricos, evitando fallos en tiempo de ejecución[3].

* + 1. **La codificación universal de números reales**

La codificación universal de números reales consiste en representar valores reales como secuencias binarias finitas o infinitas mediante técnicas como bisección de intervalos. Esta estrategia permite representar valores continuos con un alto grado de precisión; superando las limitaciones de formatos tradicionales. Es especialmente útil en aplicaciones cientifícas y criptográficas donde se requiere trabajar con precisión arbitraria y sin errores de redondeo acumulativo[4].

* 1. **CÓDIGOS DE REPRESENTACIÓN NO NUMÉRICA**

Los códigos de representación no numérica engloban formatos para codificar elementos como texto, símbolos y caracteres Unicode en sistemas informáticos. Un ejemplo reciente es la validación eficiente de texto CESU-8 mediante instrucciones SIMB. Esta técnica permite procesar y validar bloques de texto multilingüe de forma paralela, garantizando precisión y compatibilidad en aplicaciones modernas [1], [5].

A continuación, se verán tipos de códigos de representación no numérica:

* + 1. **Extensión ASCII para texto**

El código ASCII, aunque ampliamente adoptado, fue originalmente limitado a 128 caracteres. Ante la necesidad de incluir alfabetos no latinos, símbolos matemáticos y caracteres gráficos, se desarrollaron múltiples extensiones compatibles con ASCII. Una de las más destacadas fue diseñada para permitir el uso de símbolos no numéricos en ambientes interactivos, permitiendo que los sistemas pudieran operar sin errores al alternar entre diferentes conjuntos de caracteres [6].

Este tipo de codificación se volvió vital para sistemas operativos, terminales, procesadores de texto y navegadores web, especialmente en un mundo cada vez más globalizado y multilingüe. La compatibilidad hacia atrás con ASCII aseguraba que los documentos antiguos pudieran seguir siendo interpretados, mientras que la extensión proporcionaba mayor expresividad [6].

* + 1. **IPC: Information Processing Code**

El IPC fue diseñado como una solución avanzada para el procesamiento de datos no numéricos, especialmente texto. Se trata de un código de 8 bits, ampliable a 9 bits, que distribuye sus valores entre caracteres alfabéticos, de control y otros símbolos necesarios en el procesamiento de documentos, edición textual y recuperación de información. Una de sus características más destacadas es que conserva compatibilidad parcial con ASCII, pero con una extensión que facilita la interacción entre sistemas con diferentes alfabetos y funciones especiales [7].

Esta codificación fue fundamental en los primeros sistemas interactivos y sigue teniendo relevancia histórica, al haber sentado las bases de posteriores estándares de codificación multilingüe. Gracias a su diseño modular, permitió expandir el conjunto de caracteres sin alterar la lógica central de los sistemas que lo empleaban [7].

* Implementar un ejercicio práctico de conversión de texto a binario y viceversa

