Representación interna y aritmética de los números en computadores: Actividades para el laboratorio

Ester M. Garzón, Inmaculada García, José-Jesús Fernández

Dpto. Arquitectura de Computadores y Electrónica
Universidad de Almería. E-04120 Almería
e-mail: {ester,inma,jose}@ace.ual.es.

Resumen

En este trabajo se describe una estrategia para abordar el tema de la representación de los números y la aritmética en los computadores. Este tema está incluido en los planes de estudios de la mayoría de los títulos de ingeniero, especialmente en Ingeniería Informática. Esta iniciativa educativa se basa en un conjunto de ejercicios prácticos, planteados para subrayar las principales características de la representación interna de los números y su aritmética. Los ejercicios se desarrollan con un entorno computacional que constituye una valiosa herramienta para que los estudiantes reflexionen sobre este tema, abordando los aspectos más importantes, eliminando actividades tediosas. Esta propuesta puede incluirse en cursos introductorios relacionados con Estructura de Computadores, Programación, Matemática Discreta o Métodos Numéricos.

1 Introducción

La computación numérica es una herramienta esencial en todos los campos de la Ingeniería y la Ciencia [14]. Ingenieros y científicos hacen un uso intensivo de métodos numéricos y computadores para resolver problemas complejos. Para el desarrollo de la computación numérica, es de enorme importancia entender cómo se representan los números internamente en un computador, cómo se llevan a cabo las operaciones aritméticas y cuáles son sus limitaciones. Algunas de las causas más importantes de fallos de los sistemas están relacionadas directamente con errores de computación numérica [3,4]. Por ejemplo, algunos desastres de la vida real que se atribuyen a errores de de ese tipo son: el fallo del misil Patriot durante la Guerra del Golfo en 1991 [8], la explosión del Ariane 5 lanzado al espacio por la Agencia Espacial Europea en junio de 1996 [1] y otros.

Aunque la computación numérica se basa principalmente en la aritmética en punto flotante, la representación interna de enteros y su aritmética también tienen importancia [10], ya que las operaciones con enteros están incluidas en casi todas las aplicaciones de un computador. La representación interna de enteros más extendida entre los computadores modernos es el complemento a dos. Las principales características de este formato son los rangos de representación y los algoritmos para llevar a cabo las operaciones aritméticas.

Debido a la gran importancia de la computación en punto flotante en el contexto de la computación numérica [7], los científicos e ingenieros deben tener un profundo conocimiento de la aritmética en punto flotante y en particular del formato estándar IEEE 754. Desde el punto de vista de la computación numérica es importante conocer aspectos tales como rango, precisión y propiedades algebraicas relacionadas con cualquier representación finita en punto flotante.

Por otra parte, diversos aspectos del diseño de computadores también requieren un profundo conocimiento de la aritmética en punto flotante: (1) desde el punto de vista de la arquitectura del computador, que aborda el diseño del repertorio de instrucciones, incluyendo aquellas instrucciones que involucran operaciones en punto flotante; (2) desde el punto de vista del compilador y diseño de lenguajes de programación, en el sentido de que la semántica del lenguaje debe ser suficientemente clara para efectuar una adecuada traducción de los programas a lenguaje máquina (3) desde el punto de vista del sistema operativo, en lo que se refiere al tratamiento de excepciones, en el sentido de que los manejadores de excepciones pueden ser definidos por el propio usuario de acuerdo con el problema que se esté tratando.

Las recomendaciones sobre los curricula de los estudios de Informática establecidas por The Joint IEEE Computer Society y ACM Task Force han considerado siempre el tema de la representación interna de los datos como fundamental en el área de Arquitectura de Computadores en la fase introductoria del curriculum; también han considerado este tema relacionado con Lenguajes de Programación y Ciencias de la Computación.

A pesar de su gran importancia y del hecho de ser un aspecto fundamental en el curriculum de Ingeniería Informática, la representación en punto flotante es poco conocida por los estudiantes y sólo es bien entendida por expertos. Ha habido diversas iniciativas en los años noventa [12,2,13] y otras más recientes [6,9] para hacer más asequible este tema a los estudiantes. Algunas de ellas [2,6,9] se centran directamente en el estándar IEEE de punto flotante, explicando la propia representación y los aspectos relacionados con ésta. Otras [12,13] hacen uso de una representación intermedia en punto flotante para ilustrar los principales conceptos. Los trabajos [12] y [9] también incluyen ejercicios para clarificar los conceptos de precisión y rango presentes en cualquier representación finita en punto flotante. Finalmente, la referencia [9] ofrece una serie de programas para mostrar claramente las propiedades y limitaciones del estándar IEEE, desde la perspectiva de la computación numérica.

En este trabajo, describimos nuestra propuesta para la enseñanza de la representación interna de los números y la aritmética en los computadores, tanto enteros como reales, con el objetivo de ser incluida en asignaturas de los primeros cursos de Ingeniería Informática. Con el fin de facilitar la ilustración de los principales conceptos relacionados con la representación interna de los números enteros y reales, esta propuesta hace uso de (1) un conjunto de ejercicios prácticos; (2) un entorno computacional constituido por una serie de programas auxiliares y (3) un formato en punto flotante reducido que sirve para introducir el formato IEEE.

2 Representación interna de enteros. Aritmética entera

La representación más extendida para los números enteros es el complemento a dos [10]. El rango de la representación de enteros en complemento a dos, con cadenas de p bits, es $[-2^{p-1}, 2^{p-1}-1]$. Con este código se tiene una única representación para el cero por lo que se facilita su detección. Por otro lado, esta representación es especialmente adecuada desde el punto de vista del hardware porque no requiere circuitos adicionales para la operación de la resta.

Con el objetivo de ilustrar los aspectos relacionados con esta representación, se ha diseñado un conjunto de ejercicios prácticos, basados en un entorno computacional que se presenta mas adelante. Estos ejercicios se pueden clasificar en dos grupos relacionados con la representación y con la aritmética.

2.1 Nuestra propuesta para enseñar representación de números enteros

Los ejercicios incluidos en este grupo proponen al alumno: (1) convertir de código decimal a complemento a dos, diversos enteros empleando distintos valores de longitud de palabra (p); (2) determinar el número mínimo de bits que se necesita para representar un número dado; (3) determinar el rango de la representación en complemento a dos para diferentes longitudes de palabra.

Un ejemplo de ejercicio incluido en este grupo es:

(a) Determinar la representación en complemento a 2 de los enteros siguientes: $-15, 127, 128, -300$*, con una longitud de palabra* $p = 8$.

(b) Con la misma longitud de palabra, efectuar la conversión inversa aplicando como entrada los resultados del apartado anterior.

(c) En principio los resultados obtenidos en (b) deberían ser iguales a las entradas de (a). Analizar por qué en algunos casos eso no ha ocurrido.

2.2 Nuestra propuesta para enseñar aritmética entera

Los ejercicios que se clasifican en este grupo incluyen el desarrollo de las operaciones aritméticas usando diferentes longitudes de palabra. El software que se ha diseñado para estos ejercicios ilustran los algoritmos que se implementan en las unidades aritméticas de enteros (suma/resta, algoritmo de Booth, división con y sin restauración), ofreciendo el detalle del desarrollo de dichos algoritmos y teniendo en cuenta la situación excepcional overflow.

En el desarrollo de este tipo de ejercicios se sugiere a los alumnos que efectúen las diferentes operaciones aritméticas con diversos valores de longitud de palabra, haciendo uso del entorno computacional.

3 Representación y aritmética en punto flotante

El estándar IEEE ha sido ampliamente adoptado para la representación interna de los reales. Se utiliza prácticamente en todos los procesadores y coprocesadores numéricos de hoy en día para llevar a cabo los cálculos en punto flotante [2,5]. Esta sección comienza revisando los aspectos más destacables de este estándar y a continuación se centra en la descripción de nuestra propuesta para facilitar la enseñanza de estos conceptos.

3.1 El estándar IEEE 754 para la representación binaria en punto flotante

El estándar IEEE define un formato de simple precisión, de forma que el número real r expresado en notación exponencial como

$$r = (-1)^s \times (1.m) \times 2^{E-S}$$

se representa por una cadena de 32 bits que se organizan como sigue: un bit para el signo (s), $n_e = 8$ bits para el exponente sesgado $E = e + S$ (denotando e el exponente y $S = 2^{n_e - 1} - 1$ el sesgo) y $n_m = 23$ bits para la mantisa. La base implícita que se considera en esta representación es 2. El formato de doble precisión dispone de 64 bits, de los cuales 11 están destinados al exponente, 52 a la mantisa y uno al signo.

Existe una gran variedad de conceptos relacionados con la representación en punto flotante que cualquier Ingeniero Informático debe conocer:

- El número de bits del exponente y mantisa definen el rango y la precisión de la representación.

- La distancia entre dos datos en punto flotante consecutivos varía a lo largo de la recta real. A medida que se consideran valores mayores, esa distancia es mayor y viceversa.

- Existe distancia relativamente significativa entre el cero y el menor real distinto de cero. No obstante, el uso de números denormalizados permite reducir esta distancia.

- Se define la precisión de la máquina, ϵ_{mach}, como la distancia entre 1.0 y el menor número mayor que 1.0 representable por el formato. Este parámetro se puede tomar como una medida de la precisión de las operaciones en punto flotante.

- El cero se representa por una cadena de bits de valor cero.

- Los errores de redondeo, truncamiento y cancelación están presentes en la aritmética en punto flotante.

- Se asocian códigos especiales (Infinito, Not-a-Number) a situaciones excepcionales.

- Las rutinas que tratan las excepciones están bajo el control de usuario/programador.

- La representación en punto flotante no cumple exactamente las reglas fundamentales del álgebra convencional. Propiedades como asociativa o distributiva no se verifican exactamente.

3.2 Nuestra propuesta para enseñar aritmética en punto flotante

Por sencillez, en el desarrollo de los ejercicios prácticos se va a utilizar una versión reducida del formato IEEE 754 de representación de números reales. En términos generales, nuestro formato sigue todas las convenciones del estándar IEEE 754, pero emplea sólo 12 bits, de los cuales $n_e = 5$ bits están destinados al exponente, $n_m = 6$ bits a la mantisa y uno al signo. El empleo de este formato de representación tan particular, nos va a permitir ver más fácilmente todos los conceptos y problemas de aproximación, redondeos y precisión asociados a la representación de los números reales. La Tabla 1 resume las principales características de este formato reducido.

El uso de esta versión reducida del formato estándar IEEE 754 trata de alcanzar los siguientes objetivos: (1) identificar fácilmente la relación entre el número real y su representación en punto flotante así como los efectos de redondeo en la conversión; (2) ilustrar claramente la relación de compromiso rango/precisión; (3) desarrollar los algoritmos de las operaciones aritméticas en punto flotante; (4) amplificar las anomalías propias de la computación en punto flotante, de forma que el uso de este formato reducido permite identificarlas más fácilmente; (5) permitir cálculos con lápiz y papel poco tediosos, si son necesarios. En resumen el formato que proponemos es bastante adecuado para ilustrar los aspectos más relevantes de la representación en punto flotante.

Tabla1. Reprentación en punto flotante reducida

CARACTERÍSTICAS MÁS IMPORTANTES DE LA REPRESENTACIÓN

Longitud representación:	12 bits: signo: 1 bit exponente: n_e =5 bits mantisa: n_m =6 bits
Sesgo del exponente:	$2^{n_e-1} - 1 = 15 = 01111_2$
Rango del exponente:	$[-14, 15]$
Precisión de la máquina:	$\epsilon_{\text{mach}} = 2^{-n_m} = 2^{-6}$
redondeos empleados:	*Trunc., más cercano*
Números denormalizados:	Considerados
Excepciones consider.:	*Overflow, Underflow, Invalid, División por cero*

VALORES ESPECIALES POSITIVOS MÁS IMPORTANTES

Valor especial	Representación	Valor
Mayor normal.:	0 11110 111111	$+1.984375 \times 2^{15}$
Menor normal.:	0 00001 000000	$+1.000000 \times 2^{-14}$
Mayor denormal.:	0 00000 111111	$+0.984375 \times 2^{-14}$
Menor denormal.:	0 00000 000001	$+0.015625 \times 2^{-14}$
+ cero	0 00000 000000	$+0.0$
+ Infinito:	0 11111 000000	---
NaN:	0 11111 111111	---

Por otra parte, se ha diseñado un conjunto de ejercicios prácticos que tienen como objetivo ilustrar tales aspectos. Estos ejercicios se resuelven haciendo uso de un entorno computacional que se convierte en una valiosa herramienta, desde el punto de vista pedagógico. Este conjunto de ejercicios se puede clasificar en diferentes categorías:

1. Conversión de decimal a punto flotante, analizando los efectos de los distintos modos de redondeo.

2. El siguiente grupo de ejercicios, trata de facilitar el aprendizaje de conceptos tales como precisión, rango y compromiso rango/precisión. Incluye las actividades siguientes:

– Determinación de la representación de diferentes números reales usando diversos formatos, es decir variando el número de bits dedicados al exponente y mantisa.
– Determinación de la distancia entre dos números en punto flotante consecutivos en algunos intervalos, para diversos formatos de representación.

– Determinación de los rangos de representación de varios formatos en punto flotante, incluyendo los sub-rangos de números denormalizados y normalizados.
– Cálculo de los números mínimos de bits para exponente y mantisa sujetos a ciertas restricciones, por ejemplo, ser capaz de distinguir la representación de dos números muy próximos.

Como ejemplo, la Figura 1 resume el proceso para determinar el numero mínimo de bits en exponente y mantisa, para distinguir los reales 15.9 y 15.925 y a la vez representar el número 100000.

Determinación del número mínimo de bits en la mantisa para distinguir 15.9 y 15.925

	15.9	15.925
Formato Float	Representación	Representación
$n_e = 5$, $n_m = 6$	010010 111111	010010 111111
$n_e = 5$, $n_m = 7$	010010 1111110	010010 1111110
$n_e = 5$, $n_m = 8$	010010 11111100	010010 11111101
$n_e = 5$, $n_m = 8$	15.9→15.875	15.925→15.90625

Determinación de el número mínimo de bits para que además 100000. representable

	Número Real: 100000.	
Formato Float	Representación	Valor
$n_e = 5$, $n_m = 8$	0 11111 00000000	+Infinito
$n_e = 6$, $n_m = 8$	0 101111 10000110	99840.0

Figura 1. Proceso para determinar el formato en punto flotante más corto en el que los números 15.9 y 15.925 se distinguen y el real 100000.0 es representable. Solución obtenida: $n_e = 6$, $n_m = 8$

3. El siguiente grupo de ejercicios tiene como objetivo que los estudiantes analicen los procedimientos que se llevan a cabo para efectuar operaciones aritméticas con datos en punto flotante. Los ejercicios desarrollan sumas, restas, productos y cocientes con diferentes operandos, haciendo uso de distintos modos de redondeo y midiendo el error relativo de los resultados. Los operandos que se proponen se han escogido de forma que se presenten diversas situaciones. Por ejemplo, sumas de parejas de reales de magnitud muy distinta (en nuestro formato reducido $1000.0 + 2.5 \rightarrow 1000.0$).

Por otra parte, los programas incluidos en el entorno generan, como salida, una explicación detallada del proceso que se lleva a cabo para efectuar la operación aritmética en cuestión. De esta forma los estudiantes pueden distinguir fácilmente los distintas acciones asociadas a dichas operaciones.

4. Otra categoría de ejercicios analiza las distintas anomalías algebraicas que tienen su origen en la naturaleza finita de la representación en punto flotante. Nos parece importante que los alumnos reflexionen sobre la aritmética en punto flotante del computador y también sobre las limitaciones de ésta. Los ejercicios de este grupo incluyen:

– Análisis del error de cancelación que se produce en la sustracción de dos operandos de magnitud muy próxima.
– Ejercicios que muestran el efecto de acumulación de errores de redondeo. Por ejemplo, se multiplica un real por un entero mediante la suma sucesiva del primero, aplicando distintos modos de redondeo. Los estudiantes analizan la evolución del error durante el proceso de la suma. La Figura 2 muestra la salida que ofrece el entorno cuando se pide efectuar el producto 1.999×10.

MULTIPLICACIÓN 1.999×10

TRUNCAMIENTO — REDONDEO MÁS CERCANO

```
Iter    Representación      Value      Representación    Valor
-------------------------------------  -----------------------------
1 ->   0 01111 111111 ->   1.99900     0 01111 111111 -> 1.99900
2 ->   0 10000 111111 ->   3.96875     0 10000 111111 -> 3.96875
3 ->   0 10001 011111 ->   5.93750     0 10001 011111 -> 5.93750
4 ->   0 10001 111110 ->   7.87500     0 10001 111111 -> 7.93750
5 ->   0 10010 001110 ->   9.75000     0 10010 001111 -> 9.87500
6 ->   0 10010 011101 ->   11.62500    0 10010 011111 -> 11.87500
7 ->   0 10010 101100 ->   13.50000    0 10010 101111 -> 13.87500
8 ->   0 10010 111011 ->   15.37500    0 10010 111111 -> 15.87500
9 ->   0 10011 000101 ->   17.25000    0 10011 000111 -> 17.75000
10->   0 10011 001100 ->   19.00000    0 10011 001111 -> 19.75000
```

Figura 2. Producto 1.999×10 mediante sumas sucesivas, haciendo uso del formato en punto flotante reducido, con dos modos de redondeo.

– Ejercicios que demuestran que algunas reglas fundamentales del álgebra convencional no se verifican en computación en punto flotante: (1) adición y producto en punto flotante no son exactamente asociativas; (2) la propiedad distributiva del producto respecto de la suma no se verifica exactamente; (3) la propiedad de cancelación no siempre es válida, es decir si los números positivos en punto flotante A, B, C verifican $A+B = A+C$, no siempre se verifica $B = C$; (4) el producto de un número en punto flotante por su inverso no siempre es igual a 1; (5) es casi siempre erróneo preguntar si dos números en punto flotante son iguales.

Un ejemplo muy interesante relacionado con las anomalías algebraicas en punto flotante está relacionado con la computación de la Serie Armónica:

$$H_N = 1 + 1/2 + 1/3 + \ldots + 1/N.$$

Hacemos uso de esta serie para mostrar que el orden distinto de sucesivas sumas produce resultado distinto, es decir, no se verifica la propiedad asociativa para la suma. Se propone calcular la serie en el orden que aparece en la definición de la serie y, por otra parte, calcular ésta en orden inverso y comparar los resultados. La Figura 3 muestra la salida que genera el entorno computacional, cuando se desarrollan estos ejercicios.

Por otra parte, está demostrado matemáticamente que esta serie diverge; no obstante, en aritmética en punto flotante no diverge debido a los errores de redondeo. Se han propuesto también ejercicios relativos a esta cuestión.

COMPUTACIÓN DE LA SERIE ARMÓNICA

$\sum_{n=1}^{10} \frac{1}{n}$ $\qquad$ $\sum_{n=10}^{1} \frac{1}{n}$

Index	Number	Sum 1..N	Number	Sum N..1
1	1/1 = 1.00000	1.000000	1/10 = 0.10000	0.100000
2	1/2 = 0.50000	1.500000	1/9 = 0.11111	0.210938
3	1/3 = 0.33333	1.828125	1/8 = 0.12500	0.335938
4	1/4 = 0.25000	2.062500	1/7 = 0.14286	0.476562
5	1/5 = 0.20000	2.250000	1/6 = 0.16667	0.640625
6	1/6 = 0.16667	2.406250	1/5 = 0.20000	0.843750
7	1/7 = 0.14286	2.562500	1/4 = 0.25000	1.093750
8	1/8 = 0.12500	2.687500	1/3 = 0.33333	1.421875
9	1/9 = 0.11111	2.812500	1/2 = 0.50000	1.921875
10	1/10 = 0.10000	2.906250	1/1 = 1.00000	2.937500

Figura 3. Computación de la serie armónica con $N = 10$, haciendo uso del formato en punto flotante reducido y distintos ordenes de los sumandos

5. Finalmente, los ejercicios relacionados con las excepciones constituyen otro grupo de actividades. Puesto que el estándar IEEE 754 permite que las rutinas que tratan las excepciones estén bajo el control del usuario/programador, es extremadamente importante que el alumno esté familiarizado con las situaciones que producen excepciones, cómo tratarlas y qué valores en punto flotante generan excepciones. Los ejercicios que están dentro de esta categoría están relacionados con las siguientes cuestiones:

– Algunas operaciones aritméticas con datos numéricos en punto flotante pueden generar excepciones, debido a que el resultado sea no representable. Por ejemplo, el producto de dos números de muy pequeña magnitud puede generar una excepción de tipo *Underflow*, o la suma de dos números de gran magnitud puede generar una excepción de tipo *Overflow*, etc. Dos ejemplos concretos en el formato en punto flotante reducido producen excepciones: $0.0009 * 0.001$ genera *Underflow* y $1875 * 35$ genera *Overflow*.
– Dado un número en punto flotante A, determinar el resultado de las operaciones en punto flotante $A + \infty$, $A * \infty$, $A/0$, A/∞.
– determinar el resultado de las operaciones en punto flotante $\infty + 0$, $\infty + \infty$, $\infty - \infty$, $\infty * 0$, $\infty * \infty$, $\infty * (-\infty)$, $0/0$, ∞/∞, $\infty/0$.

La Figura 4 muestra diferentes situaciones excepcionales, las operaciones con las que se asocian y el valor en punto flotante que generan. Esta correspondencia debe verificarse por todo formato en punto flotante que se rija por el estándar IEEE.

Cabe destacar, para concluir esta sección, que los ejercicios prácticos descritos se desarrollan con un soporte computacional que se describe en la siguiente sección. Los programas que forman parte de este entorno están diseñados para generar las respuestas y justificaciones de los ejercicios que acaban de ser descritos.

Singular operation	Exception	Returned value
$A - \infty$	Overflow	-Infinito
$\infty + \infty$	Overflow	+Infinito
$\infty - \infty$	Invalid	NaN
$A * \infty$	Overflow	±Infinito
$\infty * 0$	Invalid	NaN
$\infty * (-\infty)$	Overflow	-Infinito
$A/0$	Division-by-zero	±Infinito
A/∞	Underflow	±Cero
$0/0$	Invalid	NaN
∞/∞	Invalid	NaN
$-\infty/0$	Overflow	-Infinito

Figura 4. Diversas situaciones que producen excepciones en nuestro formato. El símbolo ± significa que el signo del resultado depende del signo del operando.

4 El entorno computacional auxiliar

Hemos desarrollado un entorno computacional auxiliar para que los estudiantes desarrollen actividades prácticas sobre la representación y la aritmética de los computadores, facilitándoles la comprensión de los principales aspectos relacionados con este tema. El entorno está constituido por un conjunto de programas escritos en lenguaje de programación C estándar. Con el objetivo de construir una herramienta independiente de la plataforma, en la medida de lo posible, los programas internamente aplican técnicas relacionadas con aritmética de múltiple precisión, para generar las representaciones y las operaciones aritméticas en punto flotante. Hemos comprobado los programas en diferentes plataformas (PC, SGI, Sun, Alpha) y diferentes sistemas operativos (Windows, Linux, IRIX, Solaris) obteniendo resultados coherentes. Los programas pueden ser ejecutados en consola, especificando en la línea de comandos los operandos de entrada y las opciones. Para los programas relacionados con la representación y aritmética en punto flotante, las opciones incluyen la posibilidad de especificar el número de bits para el exponente y para la mantisa; la opción por defecto está asociada con el formato reducido que hemos descrito en la sección anterior ($n_e = 5$, $n_m = 6$).

También hemos desarrollado un entorno más amigable y basado en web para los programas. Con este entorno los alumnos no necesitan trabajar directamente en una ventana consola y la línea de comandos, sino que pueden acceder al entorno a través de su navegador y, de forma transparente, ejecutar los programas en el servidor web.

El entorno web está basado en formularios HTML (HyperText Markup Language) y tecnología CGI (Common Gateway Interface) [11]. Los estudiantes hacen uso de su navegador para: (1) seleccionar el programa que debe ejecutarse, (2) especificar los operandos y las opciones y (3) enviar los datos al servidor. La tecnología CGI se encarga de obtener los datos enviados por el cliente web, ejecutar el programa en el servidor con los parámetros de entrada especificados y, finalmente, enviar los resultados al cliente.

Nuestro entorno, que está disponible en la URL `http://www.ace.ual.es/RAC/`, está estructurado en dos bloques principales. Por un lado, todos los programas relacionados con la representación y aritmética de los enteros y por otro, los relativos a representación y aritmética en punto flotante. Además, existe un manual explicativo que contiene un resumen de los conceptos teóricos y los enunciados de todos los ejercicios, que están relacionados con el amplio espectro de conceptos asociados a la representación y aritmética de los números en un computador, como se ha descrito en este artículo.

Desde el punto de vista pedagógico, este entorno constituye una herramienta auxiliar ideal para los estudiantes. El entorno y el conjunto de actividades diseñadas facilitan enormemente, y desde una perspectiva eminentemente práctica, la compresión de todos los conceptos relacionados con la representación de números de longitud finita y su aritmética en computadores.

5 Conclusiones

En este artículo hemos descrito una estrategia educativa para desarrollar el tema de representación y aritmética de los números en un computador, tanto enteros como reales. Con el objetivo de preparar a los estudiantes de ingenierías informáticas, para que puedan abordar problemas computacionales de la vida real, se desarrollan un conjunto de actividades relacionadas con los conceptos de computación numérica. Nuestra estrategia también está dirigida a ofrecer a los estudiantes conceptos elementales que forman parte de cursos sobre arquitectura de computadores, lenguajes de programación y métodos numéricos.

Nuestra propuesta consiste, principalmente, en un conjunto de ejercicios prácticos cuidadosamente diseñados para destacar los aspectos más importantes relacionados con la representación finita de los números. Además, ofrecemos un entorno computacional para que los estudiantes desarrollen dichos ejercicios. Después de trabajar siguiendo esta estrategia educativa durante varios cursos, en el desarrollo de la asignatura Estructura de Computadores de Ingeniería Técnica Informática de Sistemas, podemos decir que nuestra iniciativa facilita el aprendizaje de dicho tema.

Referencias

[1] Ariane 501 Inquiry Board. "Ariane 5. Flight 501 failure." *Inquiry Board Report, 1996*, [Online] Available: http://www.esa.int/export/esaCP//Pr_33_1996_p_EN.html.

[2] D. Goldberg. What every computer scientist should know about floating-point arithmetic. *ACM Computing Surveys*, 23(1):5–48, 1991.

[3] L. Hatton and A. Roberts. How accurate is scientific software? *IEEE Trans. Software Eng.*, 20(10):785–797, 1994.

[4] N.J. Higham. *Accuracy and Stability of Numerical Algorithms.* SIAM, 1996.

[5] W. Kahan. IEEE standard 754 for binary floating-point arithmetic. World-Wide Web document, 1996. [Online] Available: http://www.cs.berkeley.edu/∼wkahan//ieee754status/ieee754.ps

[6] W. Kahan. Ruminations on the design of floating-point arithmetic. World-Wide Web document, 2000. [Online] Available: http://www.cs.nyu.edu/cs/faculty//overton/book/docs/

[7] D.E. Knuth. *The art of computer programming, third edition.*, volume 2, Seminumerical Algorithms. Addison-Wesley, 1998.

[8] General Accounting Office (Information Management and Technology Division). "Patriot Missile Software Problem," General Accounting Office Report, 1992 [Online] Available: http://www.fas.org/spp/starwars/gao//im92026.htm

[9] M.L. Overton. *Numerical Computing and the IEEE Floating Point Standard.* SIAM, 2001.

[10] D.A. Patterson and J.L. Hennessy. *Computer Organization and Design. The Hardware/Software Interface.* Morgan Kaufmann Pub., 1998.

[11] G. Birznieks S. Guelich, S. Gundavaram. *CGI Programming on the World Wide Web.* O'Reilly, 2000.

[12] T.J. Scott. Mathematics and computer science at odds over real numbers. *ACM SIGCSE Bulletin (ACM Special Interest Group on Computer Science Education)*, 23(1):130–139, 1991.

[13] C.W. Steidley. Floating point arithmetic basic exercises in mathematical reasoning for computer science majors. *Computers in Education Journal*, 2(4):1–6, 1992.

[14] C.W. Ueberhuber. *Numerical Computation. Methods, Software, and Analysis.* Springer-Verlag, 1997.