Documentación del Proyecto I : EIF207 - Estructuras de datos

David Guevara Sánchez y Mario Arguello Borge

Universidad Nacional

II Ciclo 2019

Documentación del Proyecto I : EIF207 - Estructuras de datos

Índice

Introducción

El propósito de esta estructura de datos llamada **Integer** es almacenar un número de un largo inespecificado de dígitos y poder realizar operaciones aritméticas básicas con el mismo.

Estructura

A como se puede ver en la Figura 1, la clase **Integer** es en esencia una lista enlazada de **NodeInteger**, cada uno de los cuales gasta 128 bytes de memoria. Esto en un sistema de x86_64 equivale a 32 **BasicInteger**, cada uno de estos capaz de almacenar 9 dígitos. Esto implica que un solo **NodeInteger** por su cuenta es capaz de almacenar $9 * 32$ dígitos, lo cual equivale a 288 dígitos.

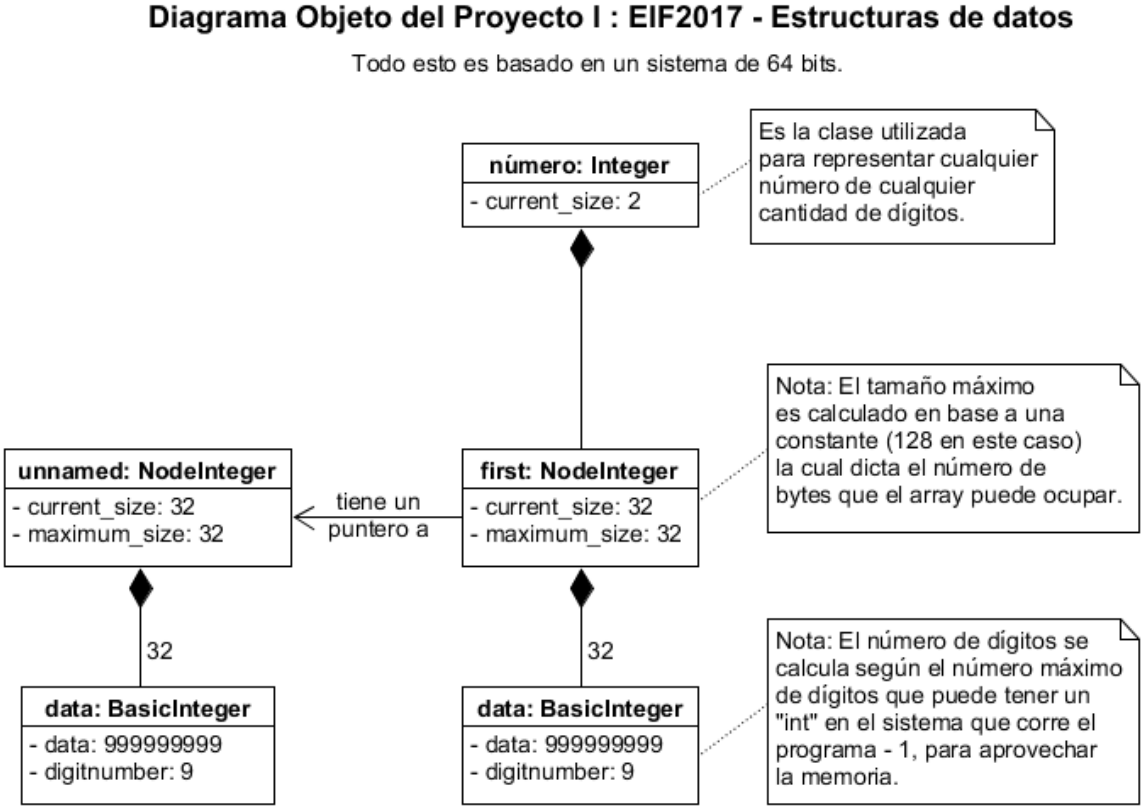

Figura 1. Diagrama objeto de **Integer**

Desperdicio

Debido a que cada **NodeInteger** abarca un total de 128 bytes en esta implementación el desperdicio de memoria en caso de que un **NodeInteger** contenga un 0 se mantiene constante.

Tomando b como el tamaño de bytes de cada celda, b equivale a 4 bytes; y tomando n como el número de celdas de cada **NodeInteger**, este equivale a $\frac{128}{b}$; esto implica que la fórmula $d(b, n)$ equivale a la constante 128 más el número de bytes que un puntero ocupa, el cual dependiendo de la arquitectura es 4 u 8.

En el mejor de los casos, cuando cada uno de los **NodeInteger** está completamente lleno, el desperdicio es 4 y 8 por cada **NodeInteger** que el **Integer** posee. Y en el peor de los casos el desperdicio es $(128\ + 4$ u $8) * IntegerSize$, dependiendo de la arquitectura.

Operaciones aritméticas

En lo que corresponde a las operaciones aritméticas, se le delegó a las unidades más básicas, **BasicInteger** y **NodeInteger**, las operaciones básicas de la suma y la resta, debido a la facilidad con la que estas operaciones se pueden implementar de manera modular. Para las operaciones más complejas de la multiplicación y la división, estas se implementaron en la clase **Integer**.