

2º curso / 2º  
cuatr.

Grado Ing.  
Inform.

Doble Grado  
Ing. Inform. y  
Mat.

# Arquitectura de Computadores (AC)

## Cuaderno de prácticas.

### Bloque Práctico 0. Entorno de programación

Estudiante (nombre y apellidos): José Manuel Pérez Lendínez

Grupo de prácticas:A2

Fecha de entrega:

Fecha evaluación en clase:

1. Incorpore volcados de pantalla que muestren lo que devuelve `lscpu` en atcgriid y en su PC.

#### CAPTURAS:

```
[jose manuel perez lendinez A2estudiante18@atcgriid:~] 2018-02-24 Saturday
$cat STDIN.o61034
Architecture:          x86_64
CPU op-mode(s):        32-bit, 64-bit
Byte Order:            Little Endian
CPU(s):                24
On-line CPU(s) list:   0-23
Thread(s) per core:    2
Core(s) per socket:    6
Socket(s):             2
NUMA node(s):          2
Vendor ID:             GenuineIntel
CPU family:            6
Model:                 44
Model name:            Intel(R) Xeon(R) CPU           E5645  @ 2.40GHz
Stepping:              2
CPU MHz:               2681.440
CPU max MHz:           2401.0000
CPU min MHz:           1600.0000
BogoMIPS:              4800.14
Virtualization:        VT-x
L1d cache:             32K
L1i cache:             32K
L2 cache:              256K
L3 cache:              12288K
NUMA node0 CPU(s):    0-5,12-17
NUMA node1 CPU(s):    6-11,18-23
Flags:                 fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clflush dts
acpi mmx fxsr sse sse2 ss ht tm pbe syscall nx pdpe1gb rdtscp lm constant_tsc arch_perfmon pebs bts rep
_good nopl xtopology nonstop_tsc cpuid aperfmperf pni dtes64 monitor ds_cpl vmx smx est tm2 ssse3 cx16 x
tpr pdcm pcid dca sse4_1 sse4_2 popcnt lahf_lm epb pti retpoline tpr_shadow vnmi flexpriority ept vpid d
therm ida arat
```

jmplz14@DESKTOP-KGK4SOA: /mnt/d/AC/practica0

```
[jose manuel perez lendinez jmplz14@DESKTOP-KGK4SOA:/mnt/d/AC/practica0] 2018-02-24 Saturday
$lscpu
Architecture:          x86_64
CPU op-mode(s):        32-bit, 64-bit
Byte Order:            Little Endian
CPU(s):                4
On-line CPU(s) list:   0-3
Thread(s) per core:    1
Core(s) per socket:    4
Socket(s):             1
Vendor ID:             GenuineIntel
CPU family:            6
Model:                 94
Model name:            Intel(R) Core(TM) i5-6300HQ CPU @ 2.30GHz
Stepping:              3
CPU MHz:               2301.000
CPU max MHz:           2301.0000
BogoMIPS:              4602.00
Virtualization:        VT-x
Hypervisor vendor:     virtual
Virtualization type:    full
Flags:                 fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clflush d
ts acpi mmx fxsr sse sse2 ss ht tm pbe syscall nx pdpe1gb rdtscp lm pni pclmulqdq dtes64 monitor ds_cpl
vmx est tm2 ssse3 fma cx16 xtpr pdcm pcid sse4_1 sse4_2 x2apic movbe popcnt tsc_deadline_timer aes xsave
osxsave avx f16c rdrand
```

Conteste a las siguientes preguntas:

- a. ¿Cuántos cores físicos y cuántos cores lógicos tiene atcgrid de prácticas o su PC?

**RESPUESTA:**

Mi pc tiene 4 cores físicos y los mismos cores lógicos al tener un único hilo por core.

Atcgrid tiene 12 físicos por nodo y son 3 nodos, el total de nodos físicos es 36 y lógicos son un total de 72

- b. ¿Cuántos cores físicos y cuántos cores lógicos tiene un nodo de atcgrid?

**RESPUESTA:**

Físicos tiene 6 por procesador, al tener 2 procesadores tiene 12 físicos y lógicos 24.

2. En el Listado 1 se puede ver un código fuente C que calcula la suma de dos vectores y en el Listado 2 una versión con C++:

```
v3 = v1 + v2; v3(i) = v1(i) + v2(i), i=0,...N-1
```

Los códigos utilizan directivas del compilador para fijar el tipo de variable de los vectores (v1, v2 y v3). En los comentarios que hay al principio de los códigos se indica cómo hay que compilarlos. Los vectores pueden ser:

- Variables locales: descomentando en el código `#define VECTOR_LOCAL` y comentando `#define VECTOR_GLOBAL` y `#define VECTOR_DYNAMIC`
- Variables globales: descomentando `#define VECTOR_GLOBAL` y comentando `#define VECTOR_LOCAL` y `#define VECTOR_DYNAMIC`
- Variables dinámicas: descomentando `#define VECTOR_DYNAMIC` y comentando `#define VECTOR_LOCAL` y `#define VECTOR_GLOBAL`. Si se usan los códigos tal y como están en Listado 1 y Listado 2, sin hacer ningún cambio, los vectores (v1, v2 y v3) serán variables dinámicas.

Por tanto, se debe definir sólo una de las siguientes constantes: `VECTOR_LOCAL`, `VECTOR_GLOBAL` o `VECTOR_DYNAMIC`.

- a. En los dos códigos (Listado 1 y Listado 2) se utiliza la función `clock_gettime()` para obtener el tiempo de ejecución del trozo de código que calcula la suma de vectores. En el código se imprime la variable `ncgt`, ¿qué contiene esta variable? ¿qué información devuelve exactamente la función `clock_gettime()`? ¿en qué estructura de datos devuelve `clock_gettime()` la información (indicar el tipo de estructura de datos y describir la estructura de datos)?

**RESPUESTA:**

`clock_gettime` devuelve la información en un struct `timespec` con la siguiente estructura.

```
struct timespec {
    time_t  tv_sec;    /* segundos */
    long    tv_nsec;   /* nanosegundos */
};
```

`Clock_gettime` rellena los campos del struct con los tiempos del reloj que se le pasa como primer parámetro en el momento en el que se llama a la función.

`Ncgt` es un double en el que almacenaremos el tiempo en segundos en la parte entera y la parte decimal los nanosegundos convertidos en segundos. La parte de los segundos se calcula restando a la segunda medición de tiempo en segundos el tiempo en segundos de la primera medición y la parte de

nanosegundos se hace el mismo calculo pero con el campo `ty_nsec` que contiene los nanosegundos, después se transforman los nanosegundos en segundos y se suman.

- b. Escribir en el cuaderno de prácticas las diferencias que hay entre el código fuente C y el código fuente C++ para la suma de vectores.

**RESPUESTA:**

Descripción diferencia	En C	En C++
Reserva y liberación de memoria.	Se utiliza <code>malloc</code> y <code>free</code> para reservar y liberar respectivamente.	Se utiliza <code>new</code> y <code>delete</code> para reservar y liberar respectivamente.
Mostrar resultados por pantalla	Se usa <code>printf</code> .	Se usa <code>cout</code> .
Cabeceras	Son sin <code>.h</code> al final	Son con <code>.h</code> al final
Espacios de nombres	No se utiliza	Utilizamos <code>std</code> .

3. Generar el ejecutable del código fuente C del Listado 1 para vectores locales (para ello antes de compilar debe descomentar la definición de `VECTOR_LOCAL` y comentar las definiciones de `VECTOR_GLOBAL` y `VECTOR_DYNAMIC`). Incorporar volcados de pantalla que demuestren la ejecución correcta en `atcgrid` o en su PC.

**RESPUESTA:**

```
[jose manuel perez lendinez A2estudiante18@atcgrid:~] 2018-02-26 Monday
$echo './SumaVectores 14' | qsub -q ac
61607.atcgrid
[jose manuel perez lendinez A2estudiante18@atcgrid:~] 2018-02-26 Monday
$ls
helloomp.eje STDIN.e61606 STDIN.o61034 STDIN.o61607
STDIN.e61034 STDIN.e61607 STDIN.o61606 SumaVectores
[jose manuel perez lendinez A2estudiante18@atcgrid:~] 2018-02-26 Monday
$cat STDIN.o61607
Tiempo(seg.):0.000000268          Tamaño Vectores:14          V1[0]+V2[0]=V3[0](1.400000+1.400000=2.800000) V
1[13]+V2[13]=V3[13](2.700000+0.100000=2.800000)
[jose manuel perez lendinez A2estudiante18@atcgrid:~] 2018-02-26 Monday
$cat STDIN.e61607
[jose manuel perez lendinez A2estudiante18@atcgrid:~] 2018-02-26 Monday
$
```

4. Ejecutar en atcgrid el código generado en el apartado anterior usando el script del Listado 3. Generar el ejecutable usando la opción de optimización `-O2` tal y como se indica en el comentario que hay al principio del programa. Ejecutar el código también en su PC para los mismos tamaños. ¿Se obtiene error para alguno de los tamaños? En caso afirmativo, ¿a qué se debe este error? (Incorporar volcados de pantalla)

**RESPUESTA:** Si da error tanto en el servidor como en mi equipo. El error se debe a que se sobrepasa el tamaño máximo de la pila.

```
[jose manuel perez lendinez A2estudiante18@atcgrid:~] 2018-02-26 Monday
$cat STDIN.e61692
./SumaVectores.sh: line 20: 21445 Segmentation fault      (core dumped) $PBS_O_WORKDIR/SumaVectoresC $N
./SumaVectores.sh: line 20: 21448 Segmentation fault      (core dumped) $PBS_O_WORKDIR/SumaVectoresC $N
./SumaVectores.sh: line 20: 21451 Segmentation fault      (core dumped) $PBS_O_WORKDIR/SumaVectoresC $N
./SumaVectores.sh: line 20: 21455 Segmentation fault      (core dumped) $PBS_O_WORKDIR/SumaVectoresC $N
./SumaVectores.sh: line 20: 21459 Segmentation fault      (core dumped) $PBS_O_WORKDIR/SumaVectoresC $N
./SumaVectores.sh: line 20: 21462 Segmentation fault      (core dumped) $PBS_O_WORKDIR/SumaVectoresC $N
./SumaVectores.sh: line 20: 21469 Segmentation fault      (core dumped) $PBS_O_WORKDIR/SumaVectoresC $N
./SumaVectores.sh: line 20: 21473 Segmentation fault      (core dumped) $PBS_O_WORKDIR/SumaVectoresC $N
[jose manuel perez lendinez A2estudiante18@atcgrid:~] 2018-02-26 Monday
$
```

5.

```
[jose manuel perez lendinez jmplz14@DESKTOP-KGK4S0A:/mnt/d/AC/practica0] 2018-02-27 Tuesday
$./SumaVectoresMiEquipo.sh
Tiempo(seg.):0.000548000      Tamaño Vectores:65536      V1[0]+V2[0]=V3[0](6553.600000+6553.600000=13107.200000) V1[6
5535]+V2[65535]=V3[65535](13107.100000+0.100000=13107.200000)
Tiempo(seg.):0.001044500      Tamaño Vectores:131072      V1[0]+V2[0]=V3[0](13107.200000+13107.200000=26214.400000) V1
[131071]+V2[131071]=V3[131071](26214.300000+0.100000=26214.400000)
Tiempo(seg.):0.002410200      Tamaño Vectores:262144      V1[0]+V2[0]=V3[0](26214.400000+26214.400000=52428.800000) V1
[262143]+V2[262143]=V3[262143](52428.700000+0.100000=52428.800000)
./SumaVectoresMiEquipo.sh: line 4: 412 Segmentation fault      (core dumped) ./SumaVectoresC $N
./SumaVectoresMiEquipo.sh: line 4: 413 Segmentation fault      (core dumped) ./SumaVectoresC $N
./SumaVectoresMiEquipo.sh: line 4: 414 Segmentation fault      (core dumped) ./SumaVectoresC $N
./SumaVectoresMiEquipo.sh: line 4: 415 Segmentation fault      (core dumped) ./SumaVectoresC $N
./SumaVectoresMiEquipo.sh: line 4: 416 Segmentation fault      (core dumped) ./SumaVectoresC $N
./SumaVectoresMiEquipo.sh: line 4: 417 Segmentation fault      (core dumped) ./SumaVectoresC $N
./SumaVectoresMiEquipo.sh: line 4: 418 Segmentation fault      (core dumped) ./SumaVectoresC $N
./SumaVectoresMiEquipo.sh: line 4: 419 Segmentation fault      (core dumped) ./SumaVectoresC $N
```

Generar los ejecutables del código fuente C para vectores globales y para dinámicos. Genere el ejecutable usando `-O2`. Ejecutar los dos códigos en atcgrid usando un script como el del Listado 3 (hay que poner en el script el nombre de los ficheros ejecutables generados en este ejercicio) para el mismo rango de tamaños utilizado en el ejercicio anterior. Ejecutar también los códigos en su PC. ¿Se obtiene error usando vectores globales o dinámicos? ¿A qué cree que es debido? (Incorporar volcados de pantalla)

**RESPUESTA:** En estos casos no se tiene un error, esto se debe a que ya no tenemos la limitación de espacio que teníamos en el ejercicio anterior por culpa de la pila.

## Globales

```
[jose manuel perez lendinez jmplz14@DESKTOP-KGK4S0A:/mnt/d/AC/practica0] 2018-02-27 Tuesday
$./SumaVectoresMiEquipo.sh
Tiempo(seg.):0.000226700      Tamaño Vectores:65536      V1[0]+V2[0]=V3[0](6553.600000+6553.600000=13107.200000) V1[6
5535]+V2[65535]=V3[65535](13107.100000+0.100000=13107.200000)
Tiempo(seg.):0.000442700      Tamaño Vectores:131072      V1[0]+V2[0]=V3[0](13107.200000+13107.200000=26214.400000) V1
[131071]+V2[131071]=V3[131071](26214.300000+0.100000=26214.400000)
Tiempo(seg.):0.001223100      Tamaño Vectores:262144      V1[0]+V2[0]=V3[0](26214.400000+26214.400000=52428.800000) V1
[262143]+V2[262143]=V3[262143](52428.700000+0.100000=52428.800000)
Tiempo(seg.):0.001843100      Tamaño Vectores:524288      V1[0]+V2[0]=V3[0](52428.800000+52428.800000=104857.600000) V
1[524287]+V2[524287]=V3[524287](104857.500000+0.100000=104857.600000)
Tiempo(seg.):0.003568000      Tamaño Vectores:1048576      V1[0]+V2[0]=V3[0](104857.600000+104857.600000=209715
.200000) V1[1048575]+V2[1048575]=V3[1048575](209715.100000+0.100000=209715.200000)
Tiempo(seg.):0.007302700      Tamaño Vectores:2097152      V1[0]+V2[0]=V3[0](209715.200000+209715.200000=419430
.400000) V1[2097151]+V2[2097151]=V3[2097151](419430.300000+0.100000=419430.400000)
Tiempo(seg.):0.014440900      Tamaño Vectores:4194304      V1[0]+V2[0]=V3[0](419430.400000+419430.400000=838860
.800000) V1[4194303]+V2[4194303]=V3[4194303](838860.700000+0.100000=838860.800000)
Tiempo(seg.):0.028962200      Tamaño Vectores:8388608      V1[0]+V2[0]=V3[0](838860.800000+838860.800000=167772
1.600000) V1[8388607]+V2[8388607]=V3[8388607](1677721.500000+0.100000=1677721.600000)
Tiempo(seg.):0.057910700      Tamaño Vectores:16777216      V1[0]+V2[0]=V3[0](1677721.600000+1677721.600000=3355
443.200000) V1[16777215]+V2[16777215]=V3[16777215](3355443.100000+0.100000=3355443.200000)
Tiempo(seg.):0.126359900      Tamaño Vectores:33554432      V1[0]+V2[0]=V3[0](3355443.200000+3355443.200000=6710
886.400000) V1[33554431]+V2[33554431]=V3[33554431](6710886.300000+0.100000=6710886.400000)
Tiempo(seg.):0.123557700      Tamaño Vectores:33554432      V1[0]+V2[0]=V3[0](3355443.200000+3355443.200000=6710
886.400000) V1[33554431]+V2[33554431]=V3[33554431](6710886.300000+0.100000=6710886.400000)
```

## Dinamicos

```

[jose manuel perez lendinez jmp1z14@DESKTOP-KGK450A:/mnt/d/AC/practica0] 2018-02-27 Tuesday
$ ./SumaVectoresMiEquipo.sh
Tiempo(seg.):0.000621800      Tamaño Vectores:65536   V1[0]+V2[0]=V3[0](6553.600000+6553.600000=13107.200000) V1[6
5535]+V2[65535]=V3[65535](13107.100000+0.100000=13107.200000)
Tiempo(seg.):0.001068500      Tamaño Vectores:131072  V1[0]+V2[0]=V3[0](13107.200000+13107.200000=26214.400000) V1
[131071]+V2[131071]=V3[131071](26214.300000+0.100000=26214.400000)
Tiempo(seg.):0.002129300      Tamaño Vectores:262144  V1[0]+V2[0]=V3[0](26214.400000+26214.400000=52428.800000) V1
[262143]+V2[262143]=V3[262143](52428.700000+0.100000=52428.800000)
Tiempo(seg.):0.004869300      Tamaño Vectores:524288  V1[0]+V2[0]=V3[0](52428.800000+52428.800000=104857.600000) V
1[524287]+V2[524287]=V3[524287](104857.500000+0.100000=104857.600000)
Tiempo(seg.):0.009066200      Tamaño Vectores:1048576  V1[0]+V2[0]=V3[0](104857.600000+104857.600000=209715
.200000) V1[1048575]+V2[1048575]=V3[1048575](209715.100000+0.100000=209715.200000)
Tiempo(seg.):0.018104800      Tamaño Vectores:2097152  V1[0]+V2[0]=V3[0](209715.200000+209715.200000=419430
.400000) V1[2097151]+V2[2097151]=V3[2097151](419430.300000+0.100000=419430.400000)
Tiempo(seg.):0.036686600      Tamaño Vectores:4194304  V1[0]+V2[0]=V3[0](419430.400000+419430.400000=838860
.800000) V1[4194303]+V2[4194303]=V3[4194303](838860.700000+0.100000=838860.800000)
Tiempo(seg.):0.072184700      Tamaño Vectores:8388608  V1[0]+V2[0]=V3[0](838860.800000+838860.800000=167772
1.600000) V1[8388607]+V2[8388607]=V3[8388607](1677721.500000+0.100000=1677721.600000)
Tiempo(seg.):0.149195500      Tamaño Vectores:16777216  V1[0]+V2[0]=V3[0](1677721.600000+1677721.600000=3355
443.200000) V1[16777215]+V2[16777215]=V3[16777215](3355443.100000+0.100000=3355443.200000)
Tiempo(seg.):0.276211800      Tamaño Vectores:33554432  V1[0]+V2[0]=V3[0](3355443.200000+3355443.200000=6710
886.400000) V1[33554431]+V2[33554431]=V3[33554431](6710886.300000+0.100000=6710886.400000)
Tiempo(seg.):0.571806200      Tamaño Vectores:67108864  V1[0]+V2[0]=V3[0](6710886.400000+6710886.400000=1342
1772.800000) V1[67108863]+V2[67108863]=V3[67108863](13421772.700000+0.100000=13421772.800000)

```

6. Rellenar una tabla como la Tabla 1 para atcgrid y otra para su PC con los tiempos de ejecución obtenidos en los ejercicios anteriores para el trozo de código que realiza la suma de vectores. En la columna “Bytes de un vector” hay que poner el total de bytes reservado para un vector. Ayudándose de una hoja de cálculo represente en una misma gráfica los tiempos de ejecución obtenidos en atcgrid y en su PC para vectores locales, globales y dinámicos (eje y) en función del tamaño en bytes de un vector (los valores de la segunda columna de la tabla, que están en escala logarítmica, deben estar en el eje x). Utilice escala logarítmica en el eje de ordenadas (eje y). ¿Hay diferencias en los tiempos de ejecución?

**RESPUESTA:**

**Tabla 1 . MI PC**

Nº de Componentes	Bytes de un vector	Tiempo para vect. locales	Tiempo para vect. globales	Tiempo para vect. dinámicos
65536	524288	0.000548000	0.000226700	0.000621800
131072	1048576	0.001044500	0.000442700	0.001068500
262144	2097152	0.002410200	0.001223100	0.002129300
524288	4194304		0.001843100	0.004869300
1048576	8388608		0.003568000	0.009066200
2097152	16777216		0.007302700	0.018104800
4194304	33554432		0.014440900	0.036686600
8388608	67108864		0.028962200	0.072184700
16777216	134217728		0.057910700	0.149195500
33554432	268435456		0.126335900	0.276211800
67108864	536870912		0.123557700	0.571806200

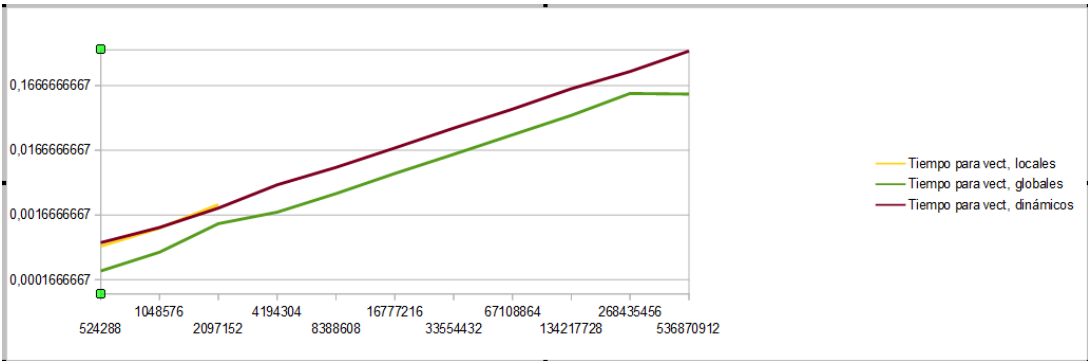
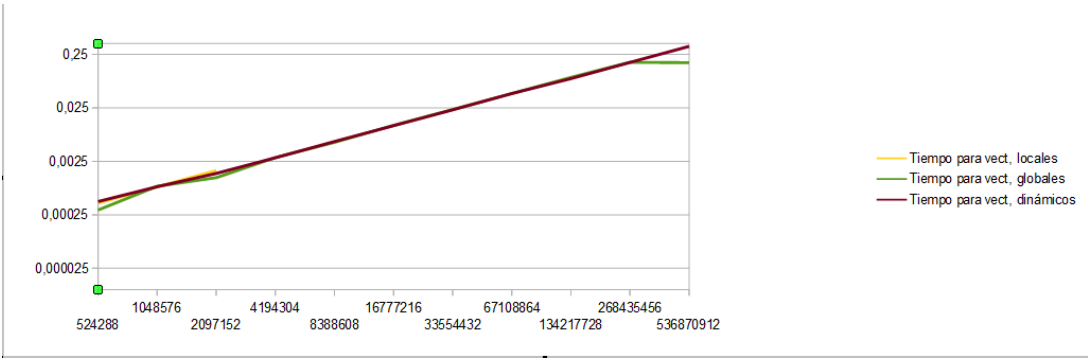


Tabla 2 . atcgrid

Nº de Componentes	Bytes de un vector	Tiempo para vect. locales	Tiempo para vect. globales	Tiempo para vect. dinámicos
65536	524288	0.000416521	0.000306587	0.000442900
131072	1048576	0.000835595	0.000855131	0.000838640
262144	2097152	0.001683818	0.001238026	0.001490114
524288	4194304		0.002945830	0.002918290
1048576	8388608		0.005667646	0.005881687
2097152	16777216		0.011804727	0.011648575
4194304	33554432		0.023303975	0.023007856
8388608	67108864		0.046385275	0.046476797
16777216	134217728		0.093262656	0.088249269
33554432	268435456		0.178874508	0.177521487
67108864	536870912		0.175875957	0.355099992



En atcgrid están muy igualados los tiempos en cambio en mi pc sale mas rápidos los vectores globales aunque imagino que seria debido a que a la hora de ejecutarlos el procesador tendría mas recursos y estaría menos ocupado que en los demás tipos.

7. Modificar el código fuente C para que el límite de los vectores cuando se declaran como variables globales sea igual al máximo número que se puede almacenar en la variable N ( $MAX=2^{32}-1$ ). Generar el ejecutable usando variables globales. ¿Qué ocurre? ¿A qué es debido? Razone además por qué el máximo número que se puede almacenar en N es  $2^{32}-1$ .

**RESPUESTA:**El código ya realiza esto.

$2^{32}-1$  es el mayor número entero que se puede almacenar ya que un entero ocupa 4B. Si no se daría desbordamiento.