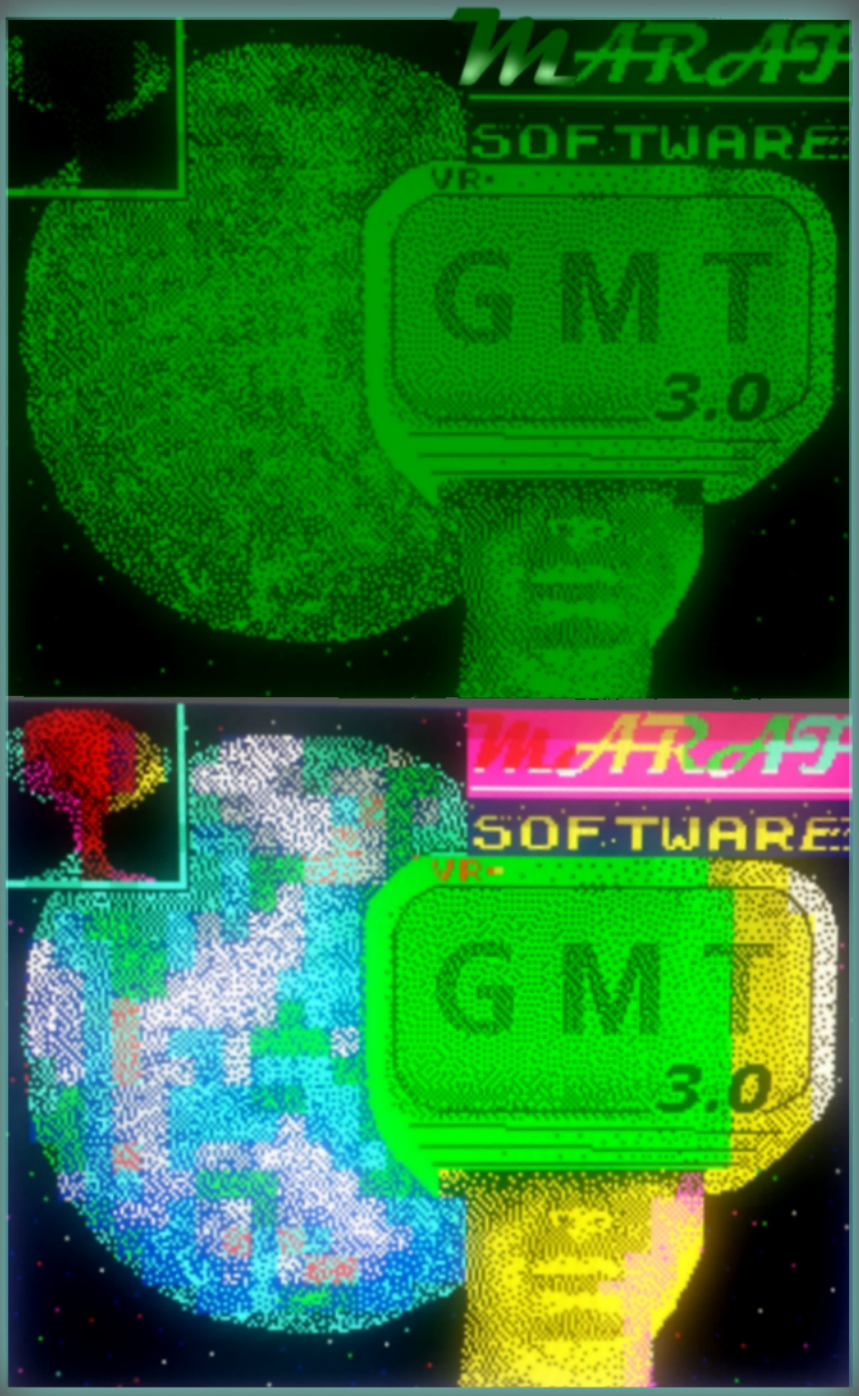
FINDESRETRO



goto NOV:2017

Archive #02

WebZine sobre RetroComputación e Informática Vintage alojado en:

http://calentamientoglobalacelerado.net/findesretro

FINDE's retro

Números disponibles:

ARCHIVE #01 junio 2017 ARCHIVE #02 noviembre 2017

¡¡ Bienvenidos a FINDE's retro !!
Web-magazine digital de tirada completamente irregular y contenido principalmente técnico, nacido con la finalidad de compartir experiencias y conocimientos sobre el RetroComputing y la escena Retro actual

Proyecto iniciado el 21 de mayo de 2017 y alojado en **www.calentamientoglobalacelerado.net**

By Rafael Lomena :: MaRaF SOFT ©© *2017*
eurocamsuite@yahoo.es - eurocamsuite@gmail.com

dowload ARCHIVE#2 PDF

dowload PDF - ARCHIVE #02 para imprimir con fondo blanco

Como podrás comprobar con esta nueva entrega, **FINDESRetro** sigue haciendo gala de un aberrante y tosco aspecto (mira que es feo el jodido) propio de las viejas webs que antaño inundaron la Red de redes, aquello que algunos ya conocen como el internet 1.0 del siglo XX.

En este sentido, no desisto en mi esperanza de que el valor de su contenido pueda de algún modo compensar esta carencia visual dejando a tu sabio juicio el veredicto final y permitiéndome aderezar la lectura del presente número con unas perlas musicales que espero sean de tu agrado:

1. *Perla musical*
2. *Perla musical*
3. *Perla musical*

(ABRIR en nueva pestaña y dejar sonar de fondo)

[ARCHIVE #02]
noviembre 2017

ÍNDICE DE CONTENIDOS:

1> FINDEs RETRO ¡¡SIGUE VIVO!!

2> REFLEXION SOBRE LA RETRO-ESCENA

3> UNA BELLA HISTORIA DE SOFTWARE LEGACY

4> EL PROGRAMA SIMULA Y UN NUEVO RET(R)O COMPUTACIONAL

5> RÁPIDA PUESTA A PUNTO DE UN "VIEJO" PORTATIL QUE SE APAGA

6> LA FUERZA BRUTA EN MILLONES DE OPERACIONES POR SEGUNDO

7> CUELGUES MISTERIOSOS EN EL PUENTE NORTE (NORTHBRIDGE)

8> EVOLUCIÓN DEL CHATARRERO GALÁCTICO, UN JUEGO TIPO

9> CITAS DEL SABIO O LAS PERLAS DE LA SAPIENCIA

Taller de descargas - Link's - Download

1> FINDEs retro ¡ SIGUE VIVO !!

Aunque en algunos momentos podamos tener una impresión diferente, la *retrocomputación* es (y seguirá siendo) un área de interés muy minoritaria y selecta, quizá por esto, tal y como ya imaginaba y respondiendo a la más pura sensatez, era de prever que el feedback a mi webzine personal por parte de posibles lectores y usuarios iba a ser y ha sido prácticamente "nulo" y, dicho esto, paso a explicaros por qué entrecomillo el adjetivo.

Como ya comentaba en mi anterior número, tengo aparcados demasiados trabajos y, lo cierto es que unas veces por pura desmotivación y otras por falta material de tiempo, después de un tiempo de "inactividad" vuelvo a la carga, sujeto a los vaivenes de la inspiración que fluctúan entre el 0 y el 1.

La verdad es que estos meses que han transcurrido desde que publiqué el pasado número han sido bastante intensos en experiencias RETRO, y aunque tenía contenidos interesantes y sentía que no podía dejar de compartir todas estas experiencias en un nuevo ejemplar de FINDE's Retro, faltaba una chispa que me hiciera sentarme delante del ordenador a escribir. Y la chispa llegó, esta vez alentada por un buen amigo, y gracias a unas pocas sesiones de escritura y de oscilante inspiración, llega este nuevo número ARCHIVE #02 repleto de historias trepidantes relacionadas con el mundo de la *retrocomputación* que espero contribuyan a evadirnos un poco del desagradable y angustioso momento que vive el país y el mundo.

Al poco tiempo del lanzamiento del ARCHIVE #01 en mi espacio web personal bajo el dominio calentamientoglobalacelerado.net y en algún que otro espacio de publicación digital como ISSUU, recibí para mi grata sorpresa y ¡¡prácticamente el mismo día!! dos correos electrónicos de unos viejos y entrañables amigos con los que tuve contacto en una etapa ya bastante lejana de mi vida, aproximadamente unos 20 años atrás, cuando vivía en tierras del norte a mil kilómetros de aquí. Ellos no se conocen entre sí aunque coincidieron en tiempo y espacio y mi relación con ambos estuvo marcada siempre por un denominador común que ya pueden imaginar, las computadoras, una pasión innata que tanto yo como ellos llegamos a convertir en un verdadero *lifestyle*. Hoy, 22 años después de que surgiera esa chispa entre nosotros, la todopoderosa Red vuelve a hacer converger nuestras vidas unidas de nuevo por esa pasión digital.

Por ellos más que nada, por mi reencuentro con estos dos grandes y singulares informáticos en estado puro a los que admiro, dos personas de sobrado carisma y grandes cualidades humanas, por Rafa y Miguel,

¡¡Bienvenidos a FINDE's retro ARCHIVE #02!!

2> REFLEXIÓN SOBRE EL HOMEBREW

En realidad, este análisis sobre la escena retro debía ser publicado en una GUÍA RÁPIDA DE DESARROLLO PARA ZX-SPECTRUM que se me ocurrió escribir a raíz del desarrollo del juego EL CHATARRERO GALÁCTICO iniciado el pasado **ARCHIVE #01**, pero finalmente, me extendí en exceso y decidí hacer un CORTA-PEGA para traerlo a esta publicación al entender que era éste el medio más acertado.

Antes de comenzar mis reflexiones acerca de lo que se ha venido a denominar la *escena retro* ó *retroscene* y siempre desde mi humilde opinión y limitado conocimiento de la misma, voy a tratar de diferenciar dos términos que pueden dar lugar a confusión en algunos lectores y que son:

RETROCOMPUTACIÓN (en inglés se usa *retrocomputing*) y **RETROESCENA** (en inglés se usa homebrew), y dado que ambos términos deben encuadrarse en el ámbito de las computadoras clásicas, tampoco estaría mal tratar de definir antes de nada qué es una **computadora clásica**.

Si bien el término *clásico* parece hacer alusión a algo que haya sido de uso cotidiano en tiempos pretéritos dentro de cualquier contexto o ámbito social, en el caso de las computadoras puede perfectamente referirse a cualquier plataforma descatalogada comercialmente y para la que el fabricante dejó de ofrecer hace años soporte técnico. Por tanto, la antigüedad a la que debe responder una computadora clásica es un factor que dejaremos debatir a los círculos más puristas y meticulosos expertos. Aunque en algunos contados casos estas computadoras clásicas podrían incluso seguir siendo perfectamente útiles y productivas para algunas tareas concretas, (sirvan como ejemplo el caso de las ***Pocket Computers*** en tareas de programación y cálculo o incluso el propio *ZX-Spectrum* en el área lúdica), lo cierto es que su uso suele quedar circunscrito a un reducido círculo de usuarios seguidores que por diferentes y a veces misteriosos motivos las han resucitado.

Volviendo ya a los conceptos **RETRO-COMPUTACIÓN** y **RETRO-ESCENA**, vocablos compuestos en ambos casos como habrás observado, debemos saber que si bien ambos términos están relacionados con las computadoras clásicas y se encuentran estrechamente ligados a éstas, la ***retrocomputación*** es para mí un concepto más amplio que abraza todo lo relacionado con la informática desde sus orígenes, incluyendo temas tan diversos como la historia, la preservación, el hardware, documentación, investigación, etc., sobre estas computadoras clásicas, mientras que la ***retroescena*** (*scene* en inglés), viene a referirse en particular al desarrollo de nuevo software para estas plataformas ya en desuso, y éste es precisamente el asunto acerca del cual quería reflexionar con vosotros.

De un tiempo a esta parte, parece que una gran mayoría de seguidores de viejos ordenadores de 8 bits (***ZX-Spectrum***, **AMSTRAD**, **MSX**, **Commodore**, etc.) coinciden en afirmar que el mundo de la retrocomputación y en especial la escena retro de algunos microordenadores, vive en la actualidad un momento álgido, un auge que está llevando el desarrollo de videojuegos para algunas plataformas concretas a ***"niveles que jamás podíamos imaginar hace apenas 10 años"*** (*JuanFra* de *elmundodelspectrum.com* - 2017).

Sin embargo, embargados por el optimismo eufórico y por diversos motivos, esta situación podría llevarnos a equívoco en lo que a la espectativa futura de la retroescena se refiere. Sin ánimos de dramatizar ni erigirme en el rey de los aguafiestas, algunos reconocidos y activos cronistas del **homebrew** en nuestro país ya vaticinan un escenario futuro bien distinto para la escena cuando todos nosotros, aquellos que fuimos testimonios directos de una etapa histórica en el nacimiento de informática doméstica, aquellos que ***"tenemos la conexión vital con la máquina"*** (*Javi Ortiz* de *elmundodelspectrum.com* – 2017), no estemos aquí.

Y es que estos reputados cronistas que viven desde su nacimiento en la primera línea de la *escena* y su divulgación, entienden desde la sensatez que la escena retro al fin y al cabo la conformamos y alimentamos todos los que vivimos aquella historia en primera persona como usuarios, desarrolladores, grafistas, vendedores, jugadores, etc., y por ello, cuando nosotros abandonemos el mundo de los vivos, la retrocomputación de los 8 bits y con ella el fenómeno subyacente de la escena tal y como la conocemos, podría sufrir irremediablemente una muerte súbita, recluyendo a nuestras queridas máquinas a algunos escasos e inertes expositores de algunos pocos y silenciosos museos en cualquier lugar remoto del planeta Tierra.

Es esta una reflexión dura pero objetiva y nadie puede afirmar que la tendencia al alza que hoy vive el *homebrew* no acabará en muy pocas décadas invirtiendo su tendencia a medida que sus protagonistas vayan desapareciedo, hasta convertirse en un triste puñado de enlaces rotos. En este sentido, la reciente pérdida de uno de los mayores artistas gráficos que ilustró las mejores portadas de videojuegos en España para *ZX-Spectrum*, ha vuelto a abrir este debate que suscita la marcha de los protagonistas de una etapa histórica. Su obra es su legado y sin duda gran parte de ella quedará con nosotros, pero también somos conscientes del carácter irreversible de su ausencia, y en especial aquellos que más de cerca le conocieron.

Y llegados a este agrio punto en nuestra reflexión, puede surgir otro planteamiento en el que sin duda muchos ya habrán pensado.

Algunos de vosotros podrá pensar...

... si los jóvenes y personas que vivimos aquel momento mágico del nacimiento de la informática personal y doméstica formamos la retroescena actual, entonces, los niños y jóvenes de hoy día formarán parte de la retroescena del futuro!!!

Expresado más o menos en forma de regla de tres:

Si ...

JÓVENES QUE VIVIERON EL MOMENTO HISTÓRICO

▼

CONFORMAN RETROESCENA ACTUAL

Entonces ...

JÓVENES ACTUALES

▼

CONFORMARÁN RETROESCENA FUTURA **(incógnita X)**

Y es que este sencillo silogismo, plasmado aquí en forma de regla de tres, podría inducirnos a pensar que el futuro de la *escena retro* (incógnita X) está garantizada aunque sufra una cierta evolución lógica. Así, resolviendo la regla de tres aquí planteada e intentando despejar la incógnita X, podemos pensar sin más que los ordenadores actuales de hoy día pasarán a formar parte de la escena retro en un plazo de unos 20-30-40... años, conformando de este modo las que serán las plataformas retro del futuro y con ello, la futura *retroescena*. Sin embargo, lo cierto es que no deberíamos obviar un matiz altamente significativo y es que, en este aparente sencillo razonamiento estamos obviando una variable determinante que creo nos conducirá irremediablemente a una realidad bien distinta.

En mi humilde opinión, para que exista esa conexión mágica y vital antes citada entre usuario y computadora **no es solo importante sino fundamental** que esta última (la máquina) goce de una "personalidad" propia perfectamente definida. ¿Pero qué es eso de la personalidad si estamos hablando de máquinas!!??

Cuando hablo de la personalidad de una máquina concreta estoy expresándome en un sentido metafórico y directamente ligado a las características exclusivas e inherentes a esa máquina que tanto añoramos. Su diseño único, los colores de sus carcasas, sus formas, su tacto, incluso el olor de sus materiales (yo recuerdo el olor de mi Spectrum al sacarlo la primera vez de su embalaje). Y luego, más allá de lo meramente material, está el "alma" de la máquina, grabada en su ROM, su memoria, su lenguaje, el diseño de sus letras en la pantalla, su sonido, la forma de entendernos y comunicarnos con ella, la música celestial de sus cargas, el chasquido de sus teclas, sus virtudes y defectos, su resolución, su velocidad de procesamiento, su color-clash, etc. En resumen, todas las características que la máquina posee y que la hacen única y diferente de todas las demás, quedó grabado a fuego para siempre en alguna región de nuestro cerebro límbico. A todo esto me refiero cuando hablo de "personalidad" propia y es sin duda, al menos a mi juicio, esta identidad propia de cada microordenador de los 80 la que nos cautivó para siempre y plantó la semilla de la retrocomputación y de la escena retro tal y como hoy la conocemos.

Por todo ello, y aunque nos puedan atraer en general las computadoras clásicas, gente como tú y yo solemos sentir ese cariño especial y diferente por un *Spectrum*, un *ZX-81* ó un *Commodore-64* y no por otro ordenador cualquiera, porque todos esos rasgos que he descrito, esa personalidad de la que hablo, hizo que algún circuito neuronal de nuestro cerebro quedara conectado para siempre y hoy, a pesar de haber transcurrido más de 30 años, aún podemos avivar esas conexiones neuronales cada vez que nos sentamos ante la máquina. ¿Cómo si

no íbamos a poder explicar lo que hacemos?

Y toda vez conceptuada la "personalidad" de nuestros microordenadores y al hilo de este asunto, quiero lanzar una pregunta en cuya respuesta seguramente coincidamos, ¿Crees que los ordenadores actuales poseen esta característica peculiar quasi "antropomórfica" a la que he denominado "personalidad" y que nadie parece dudar a la hora de atribuir a nuestras entrañables computadoras de 8 bits? A mi modo de ver, la respuesta a esta cuestión no deja siquiera margen para la duda, rotundamente NO.

Con la imparable evolución tecnológica y la homogeneización del *PC* estábamos asistiendo a una película de fatal desenlace para la retroescena, un film al que podríamos titular: "El ataque de los clones o la despersonalización de las computadoras". Con la llegada de los clónicos a nuestra vida y con ello la era del *PC* compatible, se despojó a los ordenadores de ese alma idealizada que nosotros mismos habíamos otorgado de forma inocente a nuestras máquinas, y a pesar de que los ordenadores de 16 bits (*Sinclair-QL*, *Commodore Amiga*, *Atari-ST*, etc.) lucharon por mantener la identidad propia de la que antaño gozaron las plataformas de 8 bits, la incompatibilidad sucumbió en unos pocos años ante la vertiginosa e imparable irrupción del *PC* que acabó monopolizando todos los sectores de la informática, incluyendo la doméstica, y condenó al destierro a los microordenadores personales hasta que un día, un puñado de personas apasionadas acabaron convirtiéndolas en una nueva forma de vida y diversión, modelando la retroescena tal y como hoy la conocemos.

Y con este panorama y agorero análisis, la pregunta del millón que se nos antoja es... ¿Podría de algún modo perpetuarse en el tiempo la escena retro de forma que trascienda más allá de nuestra generación?

Si queremos buscar una respuesta sensata, es fundamental identificar primero los factores que pueden favorecer la continuidad de la retroescena para así poder potenciarlos. Estamos de acuerdo en que factores básicos como la divulgación, los ferias y otros eventos públicos, talleres, libros temáticos, publicaciones, preservación documental y de hardware, investigación, emulación, etc. contribuyen sin duda a la continuidad del homebrew en forma de pilares básicos, pero tal vez las generaciones futuras necesiten algo más que eso, tal vez necesiten también de herramientas que faciliten su trabajo permitiéndoles ser mucho más productivos en sus desarrollos. Me explico.

Según expertos del *retrocomputing* y el *homebrew*, las motivaciones de los desarrolladores hoy día no son las mismas que hace 20 ó 30 años y los ***"ciclos de desarrollo del software"*** tampoco (*McLeod ZX-UNO – Retro Entre Amigos* Podcast – 2017), y personalmente no puedo estar más de acuerdo con el que algunos conocemos como el mago del *FPGA* y al que debemos nada más y nada menos que el nacimiento del que ya han denominado el auténtico *ZX-Spectrum* del siglo XXI, el **ZX-UNO**.

Si analizamos a la mayor parte de los desarrolladores del *homebrew* actual, observamos en ellos un factor común, y es que la mayoría son programadores aficcionados y/o profesionales que crean videojuegos como diversión o por puro amor al arte.

Y es que, si bien el eterno sueño dorado de ver una obra digital propia publicada en cualquier medio (hoy La Red) y jugada por otros y que en su momento sobrevoló la mente de millones de jóvenes de todo el mundo es ya un motivo de sobrado peso para hacernos pasar al otro lado del *homebrew*, al lado del desarrollador, podemos buscar incluso razones más profundas en los motivos que mueven a alguien al desarrollo de un videojuego. Al fin y al cabo, la creación de un juego es siempre un reto a la inteligencia, y como toda actividad creativa, permite conectar al creador con esa búsqueda de la belleza, entendida ésta última en su concepto más profundo de armonía y perfección.

Sin embargo, debemos de ser sensatos y conscientes de que, tanto el entorno como las circunstancias que pueden empujar a una persona a crear un videojuego para una determinada plataforma clásica, han cambiado. La realidad tecnológica que han vivido los jóvenes de hoy dista bastante de la que nosotros vivimos en nuestro momento y aquel sueño que sobrevoló la mente de tantos chavales en aquellos lejanos ochenta, prácticamente se ha esfumado en las nuevas generaciones. Carezco de datos para hacer una afirmación categórica, pero creo que el número de jóvenes usuarios de videojuegos hoy día uqe intentan o cuando menos sueñan con su propio desarrollo se ha reducido drásticamente, tal vez en parte, por las infinitas posibilidades que hoy ofrece la informática sin escribir una sóla línea de código.

Además, si hablamos ya de computadoras clásicas las motivaciones que pueden empujar a un joven a embarcarse en un desarrollo pueden ser ya meramente testimoniales. Es absurdo pretender que un joven de hoy día se siente delante de un *ZX-Spectrum* para tratar de diseñar un personaje sobre un GDU de 8x8 ó 16x16 píxeles usando código binario y en el que probablemente invertirá horas, los tiempos están cambiando. Puede que como experimento anecdótico resulte hasta gracioso pero su interés por la máquina no pasará de ahí y sus expectativas de desarrollo caerán en picado. Algunos de nosotros lo hicimos en nuestro tiempo simplemente porque no había otro camino para dar rienda suelta a nuestra imaginación y creatividad, hoy, los caminos de la computación son absolutamente infinitos y éste es un hándicap al que debe enfrentarse la *retroescena*.

Parece que en la identificación de factores básicos que pueden contribuir a potenciar el *homebrew* estamos todos casi de acuerdo, ferias, publicaciones, divulgación, etc., pero a tenor de la reflexión habida en el anterior párrafo, llegamos también a la conclusión de la importancia de comprender y atender las necesidades que los futuros desarrolladores puedan demandar si pretendemos que la *retroescena* trascienda en el tiempo.

En este sentido y en mi opinión personal, si dotamos a un joven de las herramientas necesarias que le permitan trabajar y estimular su creatividad y a la vez conseguir resultados gratificantes en tiempos relativamente cortos, esto es, optimizar los ciclos de desarrollo de software de los que hablaba *McLeod - ZX-UNO*, estaremos aumentando las posibilidades de que ese joven sí decida trabajar, al menos experimentalmente, en una plataforma de 8 bits.

Por ello, supongo que si creamos y ponemos en manos de los jóvenes herramientas modernas y productivas para el desarrollo como pueden ser la **CPCtelera** (para plataformas *Amstrad CPC*) impulsada por el profesor de magos *Fran Gallego*, **La Churrera** (plataformas *ZX-Spectrum, NES, Amastrad, Megadrive, MS-DOS, ZX-81...*) de *Mojons Twins*, **Arcade Game Designer** (*ZX-Spectrum* y *Amstrad CPC*) de *Jonathan Caudwell*, el potente compilador cruzado **ZX-BORIEL** (*ZX-Spectrum*) de *Boriel*, **8 BITS DE PODER** (*Amstrad CPC*) de *José Javier García Arnada*, o incluso mis humildes y sencillos desarrollos **ZX-Draw**, **GDUcalc**, tal vez logremos proyectar en algunos de ellos la mágica ilusión de crear un videojuego aunque solo sea por la intensa y nada desdeñable satisfacción personal que ello supone, y mantener así vivas en el tiempo nuestras viejas computadoras a través de nuevos desarrollos realizados por jóvenes que, pese a que no vivieron en primera persona la época dorada de los 8 bits, puedan estimular su creatividad y a la vez comprender y conocer un poco mejor y en primera persona el origen de la mayor industria del mundo.

Sí, vale, es una pretensión demasiado ambiciosa pero honestamente creo que cualquier línea que estimule y apoye el desarrollo *homebrew* es decisivo para contribuir a mantener vivas nuestras máquinas en los jóvenes del mañana, conectando la historia de la informática de la forma más bella y hermosa posible, a través de la creatividad. Es importante comprender que cada programa, cada juego, cada lector y cada nuevo usuario de nuestro microordenador, es una bola de oxígeno para el futuro del *homebrew*. Y aunque es seguro que la supervivencia de la *escena* será díficil y complicada por los motivos expuestos, tal vez con unas herramientas de apoyo potentes y unos "padres" entregados las posibilidades de éxito serán sin duda mayores.

En este sentido, parecen ya estar estar trabajando algunos investigadores e ingenieros de la primera línea en el ámbito docente universitario y parecen estar convencidos de que la ***retrocomputación*** tal vez pueda abrir una línea educativa completamente innovadora para las futuras generaciones de ingenieros y programadores.

Lecturas recomendadas sobre el tema:

¿Puede la retroinformática darnos mejores programadores en el futuro? Este estudio cree que sí

Y llegados a este punto, solo me queda hacer una mención especial homenajeando a los programadores de emuladores y aludiendo para ello a una contundente afirmación del siempre humilde y querido promotor del *homebrew* Javi Ortiz (*elmundodelspectrum.com*) y al que tuve la oportunidad de saludar personalmente en la última edición de **RETROALICANTE-2017**:

"Todo lo que es la escena tal y como la conocemos se lo debemos a los emuladores, ellos nos han traído hasta aquí."

3> UNA BELLA HISTORIA DE SOFTWARE LEGACY

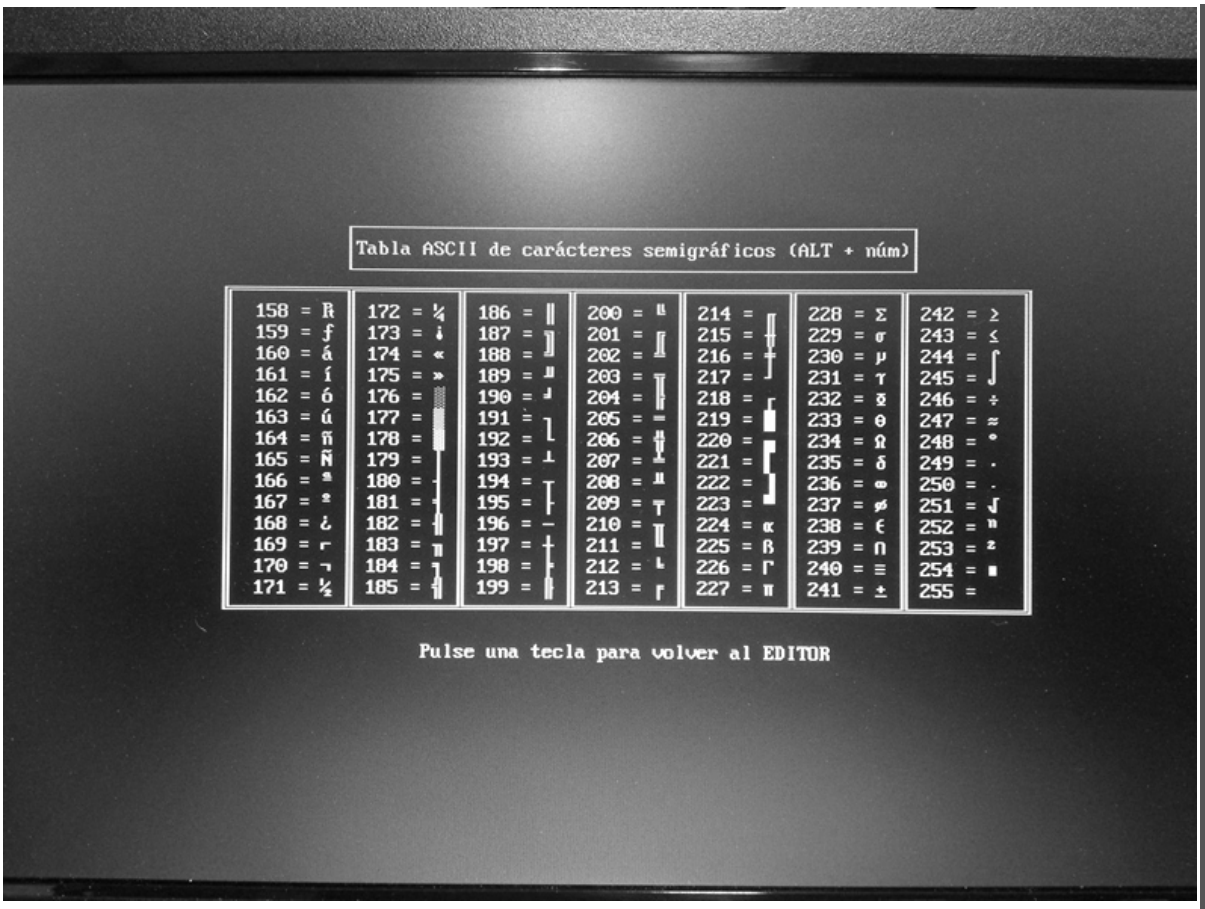
FINDES retro

TEXTCOM ó **TEX2COM** es una vieja utilidad para **MS-DOS** que yo mismo desarrollé y distribuí entre las revistas de la época bajo licencia *Freeware* hace 22 años, cuando la escena corría por el mundillo saltando en disquetes flexibles. Debió ser en aquella época cuando tuvieron lugar mis primeros y únicos contactos con el misterioso e intrigante lenguaje ensamblador de los procesadores de 16 bits *8086/8088*. En este desarrollo conté seguramente con el apoyo de mi viejo y buen amigo Bernabé, el mayor mago binario que jamás he conocido, capaz de programar igualmente en ***C*** una vieja computadora *UNIX* como la bella *CASIO PB-100* que a menudo habitaba el bolsillo de su pantalón de trabajo. Siempre autodidacta. Él me descubrió las pocas cosas que por entonces aprendí de código máquina y también me ayudó a aprobar la tenebrosa asignatura de *Ensamblador/Assembler*. Apenas dos años más tarde de TEXTCOM, Bernabé y yo emprendimos un proyecto de desarrollo de mayor envergadura llamado LEGISLA y que vió la luz en noviembre de 1997, un consultor legal con motor de búsqueda escrito en código máquina y que, según sus más de 600 usuarios en toda España, hizo las delicias de muchos profesionales y opositores. Quien sabe, tal vez en un próximo número de FINDES Retro le dediquemos una sección.

TEXTCOM (también llamado TEX2COM), un sencillo editor de pantallas que desarrollé hace más de 20 años para generar pantallas autovisualizables en un archivo ejecutable de 2000 bytes (2KB) y con extensión .COM. TEXTCOM fue escrito con la versión 4.5 del compilador de Microsoft QuickBASIC y su ejecutable EXE fue comprimido con el compresor de ejecutables PKlite que comprimía archivos EXEs y COMs manteniéndolos ejecutables y descomprimiéndolos en RAM en el momento de su ejecución.

Por aquellos lejanos 90', debía de andar yo de visita vacacional por tierras norteafricanas a juzgar por los datos que se muestran en la pantalla de inicio de la aplicación. Como verás, no aparece ningún email porque sencillamente, no existía internet tal y como lo conocemos y además, el seudónimo *Marien SOFT* con el que firmaba el software demostraba que el pequeño de mis vástagos, también llamado Rafael!, aún no había sido concebido ya que desde su nacimiento conjugué sus nombres resultando en el seudónimo ***MARAF SOFT.***

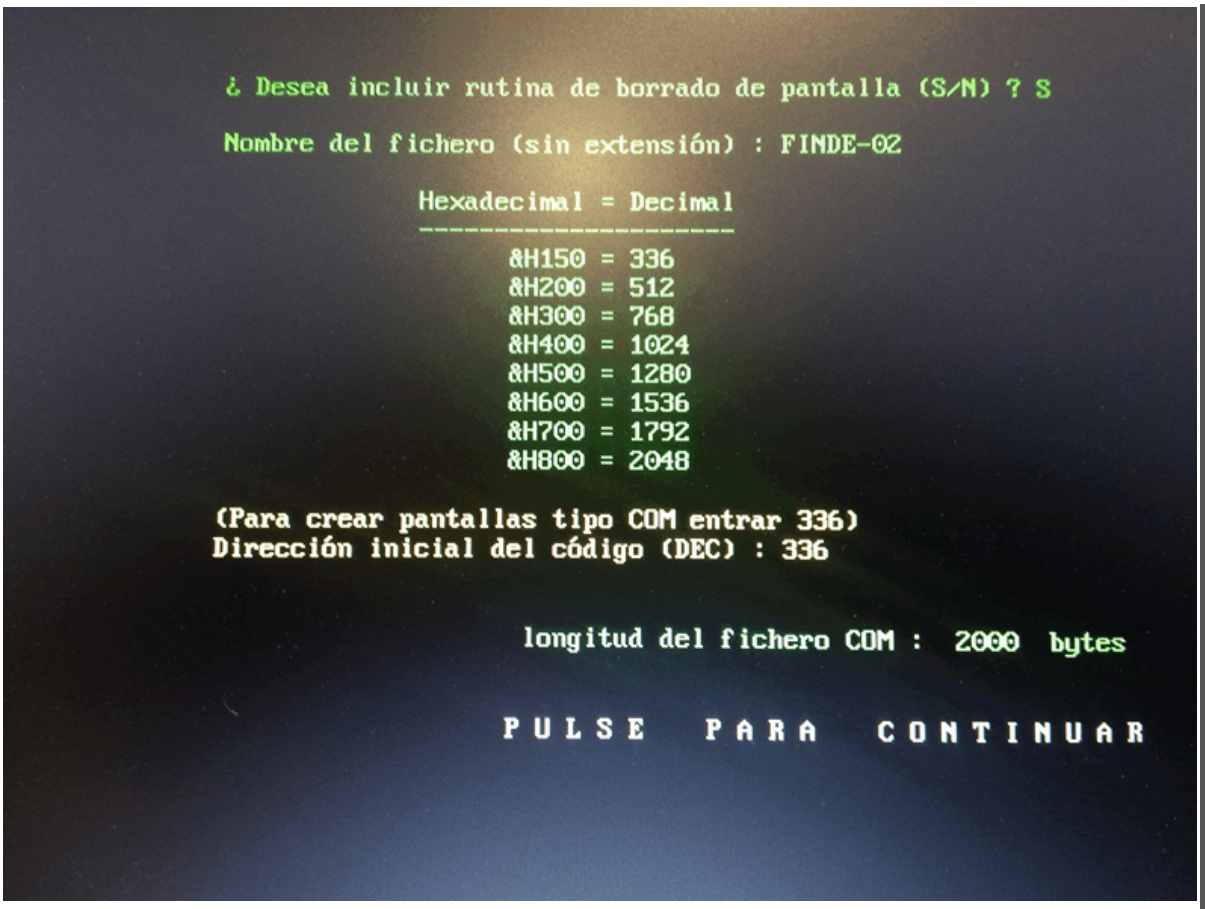
Las comunicaciones eran muy limitadas entonces y los ordenadores personales solo podían conectarse a las *BBS (Buletin Board System)* usando aquellos MODEMs de 1200 ó 2400 baudios por segundo, lo que equivale a velocidades de 1,2 y 2,4 KBytes por segundo. Yo particularmente no fui un usuario activo de las *BBS* por esa época ya que no dispuse de MODEM hasta aproximadamente el año 1996-97, cuando finalmente me conecté a Internet mediante la línea telefónica analógica, pero sí sabía de amigos que se solían conectar a estos servidores (instalados por particulares en su propia casa y utilizando su línea teléfonica) principalmente para la descarga de software.



El sencillo editor de pantalla integrado en TEX2COM nos permite desplazarnos con los cursores del teclado e ir "dibujando" lo que deseamos mostrar dentro de los límites que marca la tabla de caracteres ASCII a la que podemos acceder en todo momento pulsando F1. Una vez finalizada la pantalla procederemos a generar el archivo .ASM con el código ensamblador listo para compilar y generar el archivo ejecutable .COM

Bueno... ¿Y dónde está la "bella historia de software legacy" que encabeza el título de esta sección? Pues imaginen cual fue mi sorpresa al descubrir que Rafa, ese buen amigo y tocayo al que hacía referencia al inicio del presente **ARCHIVE #02**, contactó conmigo buscando alguna copia de **TEXTCOM** en la Red. Y... ¿Por qué podía estar interesado Rafa, hoy un reputado ingeniero informático que trabaja en las trincheras de la informática para una importante entidad financiera, interesado en el vetusto **TEXTCOM**? Debo confesar que me quedé asombrado cuando él mismo me explicó los motivos con la nitidez que le caracteriza:

*"Estoy montando la nueva arquitectura DevOps de la empresa en la que trabajo como ingeniero dentro del área de arquitectura informática. Y aquí es donde entra el **Text2Com**. La empresa tiene un huevo de sistemas "legacy" que se quieren automatizar dentro de un sistema de desarrollo y despliegue continuo de aplicaciones. Eso implica que tenemos que montar la de dios de scripts tanto en sistemas modernos: Docker sobre Linux; como en sistemas legacy: DOS, Windows y mainframes. Total, que el **Text2Com** me puede venir bien para ciertas pantallas de ayuda y menús de los scripts para los operadores."*



TEXTCOM genera a partir de nuestro diseño de pantalla el código en ensamblador y lo escribe en un archivo ASCII de extensión .ASM listo para ser compilado mediante el antiguo comando DEBGUG.

Si bien la rotunda simplicidad de **TEXTCOM** sigue siendo su gran valor, hoy día será necesario utilizar un Sistema Operativo **MS-DOS** ó alguna antigua versión de Microsoft Windows (hasta *Windows XP*) para poder compilar el código generado por **TEXTCOM** mediante el comando **DEBUG** y conseguir nuestras pantallas ejecutables en formato .COM, ya que la últimas versiones de 64 bits del sistema de Microsoft no incorporan dicho comando. Llegados a este punto debemos saber que las versiones Windows de 64 bits presentan mayor incompatibilidad para este tipo de software de 16 bits, ya que estos sistemas de 64 bits requieren que todo los ejecutables estén firmados digitalmente por motivos de seguridad, algo que no ocurre con las aplicaciones compiladas para MS-DOS y/o *Windows* de 16 bits (hasta la versión *3.11*). **Contextualizado en la actual espiral esquizofrénica de consumismo salvaje al que nos someten las grandes corporaciones, el debate Windows 32 bits VS Windows 64 bits no es algo baladí y por ello aprovecho para documentar grosso modo las diferencias que ambos caminos pueden ofrecer al usuario final**.

Este insalvable inconveniente de incompatibilidad ya me obligó hace algunos años a recurrir a la máquina virtual **DOSBox** para poder mantener en vigor mi vieja aplicación de lotería primitiva SIMULA y permitir su ejecución en sistemas Windows de 64 bits (las ventajas e inconvenientes de este método (DOSBOX) las analizaremos en profundidad en nuestra próxima sección titulada ***"EL PROGRAMA SIMULA Y UN NUEVO RET(R)O COMPUTACIONAL")***. En realidad, éste es para mí un argumento suficiente para decantarnos y sopesar la instalación de sistemas de 32 bits en máquinas modernas si, como es mi caso y supongo que el de muchos de vosotros, deseamos seguir corriendo algunas aplicaciones antiguas de 16 bits de forma nativa.

Hoy por hoy, la opción de 32 bits está disponible para cualquier sistema Windows moderno (incluido W10) y puede convertirse en una opción perfectamente válida en una gran cantidad de casos. De hecho, con motivo de la pérdida accidental de las particiones de arranque del disco duro de mi NetBook (2 GB-RAM) con el que trabajo normalmente, probé Windows 10-64bits y después de trabajar con él algunos días (un día entero afinando el rendimiento del sistema para que se mueva con alegría;) me decanté finalmente por la versión de 32 bits. Incluso me atrevo a afirmar que un ordenador portátil (para uso doméstico cotidiano) con 2-4 GB de RAM puede que vaya más ligero con un sistema de 32 bits que con uno de 64 al tener estos últimos requerimientos de RAM más exigentes. Además, no debemos obviar que las aplicaciones compiladas para 64 bits ocupan mayor cantidad de espacio en disco y en memoria con la consecuente reducción de rendimiento general que ello puede conllevar al sistema. Si a esto añadimos su mayor compatibilidad para correr de forma nativa (sin emuladores ni máquinas virtuales) software de 16 bits y que prácticamente el 100% del software actual continúa ofreciéndose en versiones de 32 bits, la decisión está clara y los sistemas de 32 bits se presentan como una opción ganadora, aunque nada parece evitar que este panorama pueda cambiar en los próximos años. En cualquier caso, si deseas conocer a fondo los pormenores de la RAM y los entresijos técnicos de ambas opciones (32 bits VS 64 bits) puedes echarle un ojo a este interesante **blog de José Arrarte** y su artículo ***64GB de RAM en un sistema operativo de 32 bits***.

Por todo ello, **TEX2COM** me hizo pensar rápidamente en **DOSBox** para poder correrlo en plataformas Windows de 64 bits, pero lamentablemente **DOSBox** no permite el uso de DEBUG y al intentar ejecutarlo desde la línea de comandos nos devuelve un antipático mensaje *"illegal comand: debug"*. Lo cierto es que, aunque consigamos ejecutar un comando DEBUG de cualquier forma (recuerda que DEBUG es un comando externo contenido en un archivo y no integrado en el núcleo COMMAND.COM) bajo una sesión DOSBOX, acaba dando errores en la compilación, extremo que me confirmó mi buen amigo Rafael en uno de sus mensajes: *"El problema de compilación con DOSBox es que tiene un bug en las llamadas a la interrupción 20"* y que precisamente es utilizada para devolver el control de la ejecución al sistema operativo.

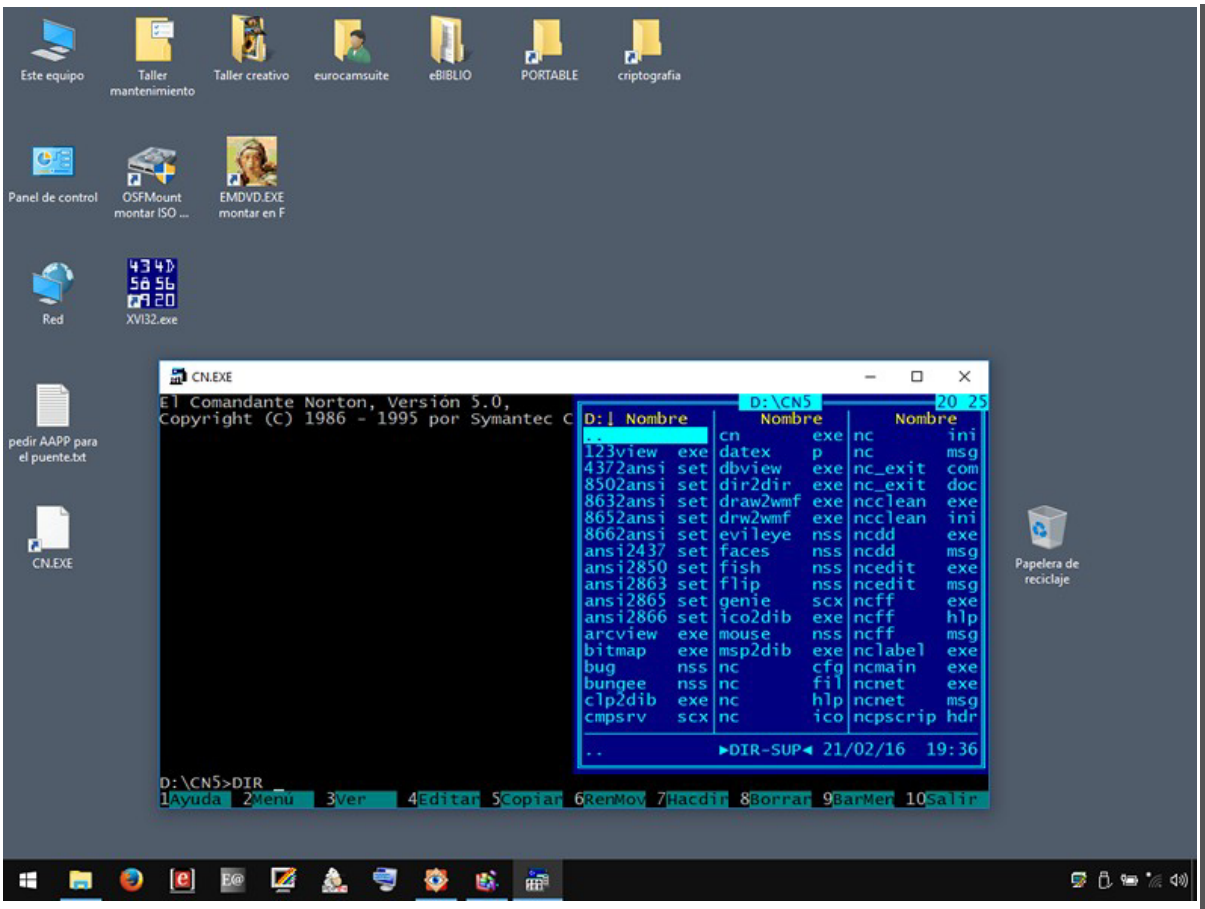
Si nuestro sistema Windows es de 64 bits, podemos llegar a una solución mediante algún sistema gratuito compatible con ***MS-DOS*** como por ejemplo ***FreeDOS***, una distribución gratuita que podemos descargar e instalar en un pendrive de arranque para el uso de este tipo de abandonware (software antiguo actualmente en desuso). Precisamente esta solución mediante Pendrive de arranque es la que elegí para poder correr el programa SIMULA a tope de rendimiento en modernos procesadores de 64 bits y la comentaremos en profundidad en la próxima sección titulada ***"EL PROGRAMA DE LOTERÍA SIMULA Y UN NUEVO RET(R)O COMPUTACIONAL"***.

En el caso de que contemos con un sistema de 32 bits como es mi caso (W10), os cuento como resolví casualmente el problema del compilador **DEBUG** para poder crear las pantallas .COM sin recurrir a métodos drásticos como el arranque desde unidad y cuya solución puede probablemente extrapolarse a otros muchos casos en los que necesitemos correr algún antiguo software específico de gestión, cálculo científico, etc.

Aunque los sistemas Windows de 32 bits pueden ejecutar una gran parte del software de 16 bits y para **MS-DOS** de manera directa, en el caso de **TEX2COM** era doblemente difícil que funcionara ya que, aunque consiguamos que funcione el ejecutable del programa en sí, finalmente no podríamos compilar el código fuente ASM obtenido porque necesitamos ejecutar **DEBUG** y *Windows 10-32bits* ya nos devuelve un error al intentar llamarlo desde el símbolo del sistema. De hecho, la utilidad **TEX2COM** no realiza el proceso de compilado automáticamente, dejando al usuario el control de esta fase mediante la línea de comandos. Pensemos que el programa tiene más de 20 años y está creado para **MS-DOS**, un entorno en el que la línea de comandos era la esencia misma del sistema. Además de esto, como creo recordar que el ejecutable de **TEX2COM** fue comprimido en su día mediante una potente utilidad llamada *PKLite* en un archivo .COM, decidí descomprimirlo pero aún así no conseguí su ejecución en el sistema *W10-32bits*. Por otro lado, a pesar de que **TEX2COM** corre sin problemas con **DOSBox**, la fase de compilado requiere del comando **DEBUG** y ahí ya obtenemos errores pues **DOSBox** no admite de ninguna forma la ejecución del comando **DEBUG**.

De este modo me encontraba en una encrucijada bastante compleja para correr la aplicación con normalidad y compilar las pantallas sin la engorrosa necesidad de reiniciar el ordenador con un PenDrive y un sistema *FreeDOS*. Sin embargo, la fase experimental estaba a punto de dar un giro inesperado hacia el éxito gracias a una veterana aplicación *abandonware*.

Desde hace años, suelo conservar en mis unidades de disco duro un directorio con una versión portable del otrora todopoderoso ***COMANDANTE NORTON*** (abreviado ***CN***). Para los que no lo conocen, el ***CN*** era una herramienta escrita en ensamblador (CPUs 8086/8088) por el mago *Peter Norton* y que en la práctica, constituía un sólido y potente soporte para los usuarios avanzandos de PC, un auténtico *sistema operativo* que otorgaba a **MS-DOS** unas posibilidades de gestión de archivos y unos niveles de productividad prácticamente infinitos. No exagero. Yo aún recuerdo a un viejo amigo al que resultaba imposible de seguir cuando operaba con ***CN*** manejando archivos y directorios a la velocidad de la luz.

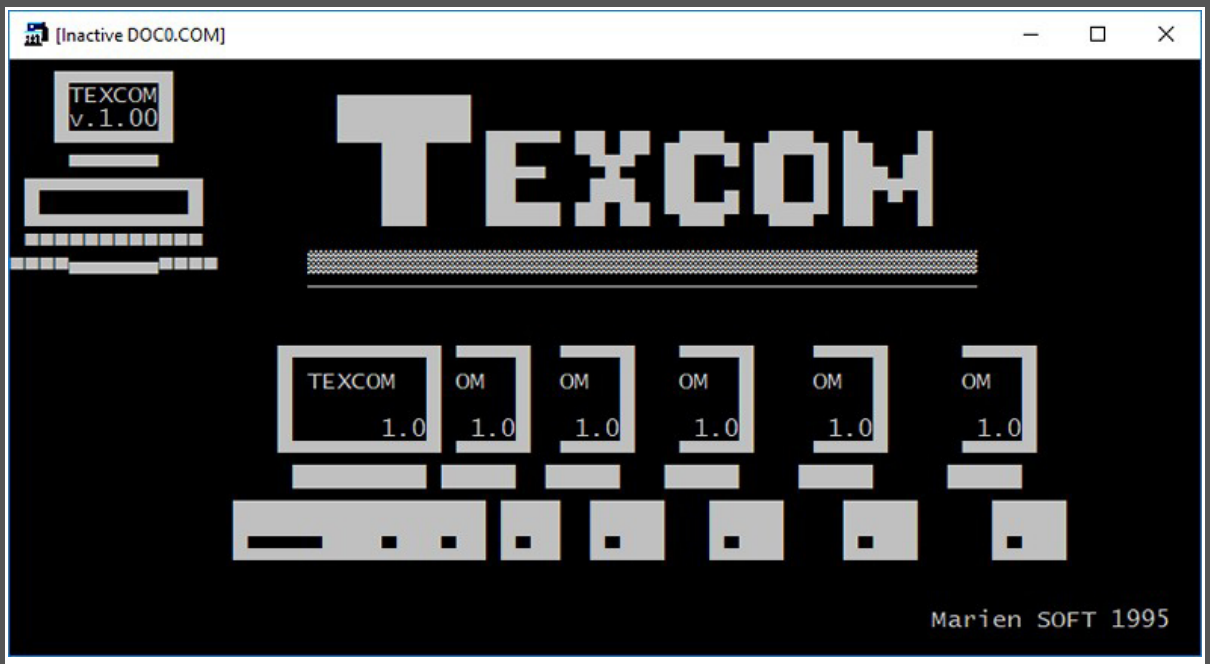


El otrora todopoderoso software de 16 bits ***Comandante Norton (CN)*** corriendo en una ventana "nativa" de Windows 10-32bits. Los programas antiguos de MS-DOS que no hacían uso de los recursos gráficos y que estaban basadas en "modo texto", suelen ofrecer mayores índices de compatibilidad y éxitos a la hora de correr sobre los actuales sistemas operativos de Microsoft.

Impresionado desde siempre con esta maravilla de aplicación que hace 25 años gobernaba una buena parte de los PCs en todo el mundo, me fui directo a la carpeta ***C:\CN5*** para ejecutar este prodigioso programa y funciona perfectamente en *W10-32bits* de forma "nativa" (sin DOSBox ni tan siquiera recurrir a las configuraciones de compatibilidad que ofrece Windows desde las propiedades del ejecutable)

Ahora solo faltaba probar a compilar el fichero .ASM creado por **TEX2COM** (ejecutado éste bajo **DOSBox**) desde la línea de comandos del *Comandante Norton*. En un ejercicio de sinceridad, te diré que no sé ni cómo se me ocurrió probar a compilar el código ASM generado por **TEX2COM** desde ***CN*** y supongo que mi devoción casi religiosa por esta herramienta tuvo algo que ver, pero la cuestión es que FUNCIONA A LA PERFECCIÓN!!

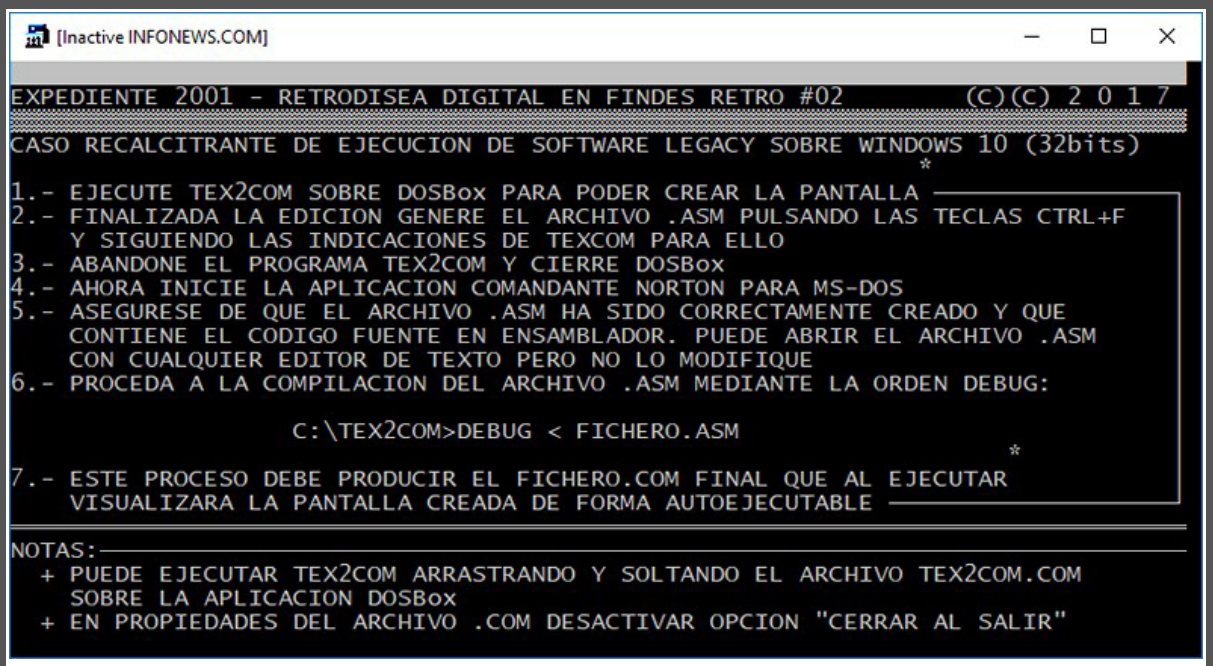
De esta forma, ya podemos correr **TEXTCOM** sobre DOSBox para generar las pantallas y los archivos fuentes en ensamblador (.ASM) para luego pasar a compilarlos en una sesión de *Comandante Norton* usando el comando **DEBUG** que al parecer, va integrado en el propio programa *CN*. No es la panacea pero nos sacará del atolladero.



El seudónimo ***Marien SOFT*** con el que solía firmar el software demostraba que el pequeño de mis vástagos, también llamado Rafael!, aún no

había sido concebido, ya que desde su nacimiento conjugué sus nombres resultando en el seudónimo ***MARAF SOFT*** que aún hoy suelo utilizar. En esta captura se muestra pantalla de ejemplo creada con ***TEXCOM***. Estas pantallas constituyen archivos ejecutables de extensión .COM que pueden abrirse desde *Windows 10-32bits* (y desde cualquier sistema compatible) mostrándose en una nueva ventana, pero debemos considerar que al ejecutar el archivo .COM la ventana se muestra y se cierra rápidamente sin que podamos ver el contenido mostrado, por ello es imprescindible crear un acceso directo al archivo .COM y desactivar en la ventana de ***Propiedades-Programa*** la opción "CERRAR AL SALIR", la cual se encuentra activada por defecto.

Otra vía para alcanzar soluciones en estos **casos recalcitrantes de SOFTWARE LEGACY**, algo más complicada de implementar pero que suele dar buenos resultados para aquellos que nos resistimos a renunciar a ese valioso universo de retroaplicaciones y utilidades o que por necesidades profesionales podemos encontrarnos en un aprieto similar, es hacer uso de un software específico para la creación de máquinas virtuales.



Pantalla aclaratoria de ejemplo creada con ***TEXCOM*** corriendo bajo ***DOSBox*** y compilada posteriormente mediante el comando **DEBUG** integrado en ***Comandante Norton (CN)***. ¡Ojo!! La aplicación CN no debe ser ejecutada en ***DOSBox*** sino de forma directa (funciona en sistemas de 32 bits) ya que de lo contrario la llamada al comando DEBUG no será posible. También recuerda acceder a las propiedades dell archivo .COM y desactivar en la pestaña ***Programa*** la opción "CERRAR AL SALIR", la cual se encuentra activada por defecto, para que la pantalla .COM no se cierre automáticamente y pueda mostrarse de forma permanente.

En internet encontramos diversas opciones pero yo personalmente me decanto por *VMWare* por su sencillez y su total compatibilidad con *W10*. A través de *WMWare* (existe una versión gratuita para uso personal totalmente funcional) y mediante una imagen ISO o el CD de instalación de cualquier sistema, podemos instalar en una máquina virtual prácticamente cualquier sistema operativo. *VMWare* permite luego la ejecución del sistema virtual desde el propio escritorio de *Windows*. Obviamente, los requisitos de memoria suelen ser bastante más exigentes para correr sistemas en máquinas virtuales, ya que el sistema de la máquina virtual y el de la real deberán compartir la RAM disponible.

Comandante Norton (CN) corriendo de forma "nativa" en una ventana de *W10-32bits* compilando directamente desde su línea de comandos mediante el DEBUG integrado en la propia aplicación CN. Al concluir la compilación del archivo ASM (mediante el operador de redireccionamiento <) el propio DEBUG escribe el archivo .COM y devuelve el control al sistema.

Resumiendo un poco nuestro asunto acerca del recalcitrante caso del retrosoftware

TEXT2COM/TEX2COM/TEXTCOM, que para el caso es lo mismo, solo dejar claro que el proceso de creación de pantallas COM puede separarse en dos fases:

1ª FASE) ***En la que se crea la pantalla y el archivo .ASM***
Para ello debemos correr TEXTCOM sobre DOSBox o cualquier sistema compatible)

2ª FASE) ***En la que se compila dicho archivo .ASM (y que contiene el código fuente en ENSAMBLADOR necesario para mostrar la pantalla diseñada en plataformas W10-32bits) generando el archivo ejecutable .COM de idéntico nombre que el .ASM***

En este caso hemos conseguido compilar sin errores haciendo uso de la aplicación abandonware Comandante Norton v5 para DOS y lanzando compilación desde su línea de comandos:

C:\CN5>DEBUG < EJEMPLO.ASM

Al final... ¿Me ha quedado un poco larga la sección?? Bueno, ya para terminar y a propósito de esta historia, no sé si estarán de acuerdo conmigo pero podemos decir que esta vez la magia de La Red fue capaz de convertir un viejo programa perdido en la noche de los tiempos y prácticamente olvidado en un puente para conectar a las personas.

En la próxima sección abordaremos un nuevo y no menos apasionante *ret(r)o* de compatibilidad abandonware digno de estudio y en el que el potencial de procesamiento y la velocidad de cálculo de los microprocesadores marcará sin duda la diferencia.

4> EL PROGRAMA DE LOTERÍA SIMULA Y UN NUEVO RET(R)O COMPUTACIONAL

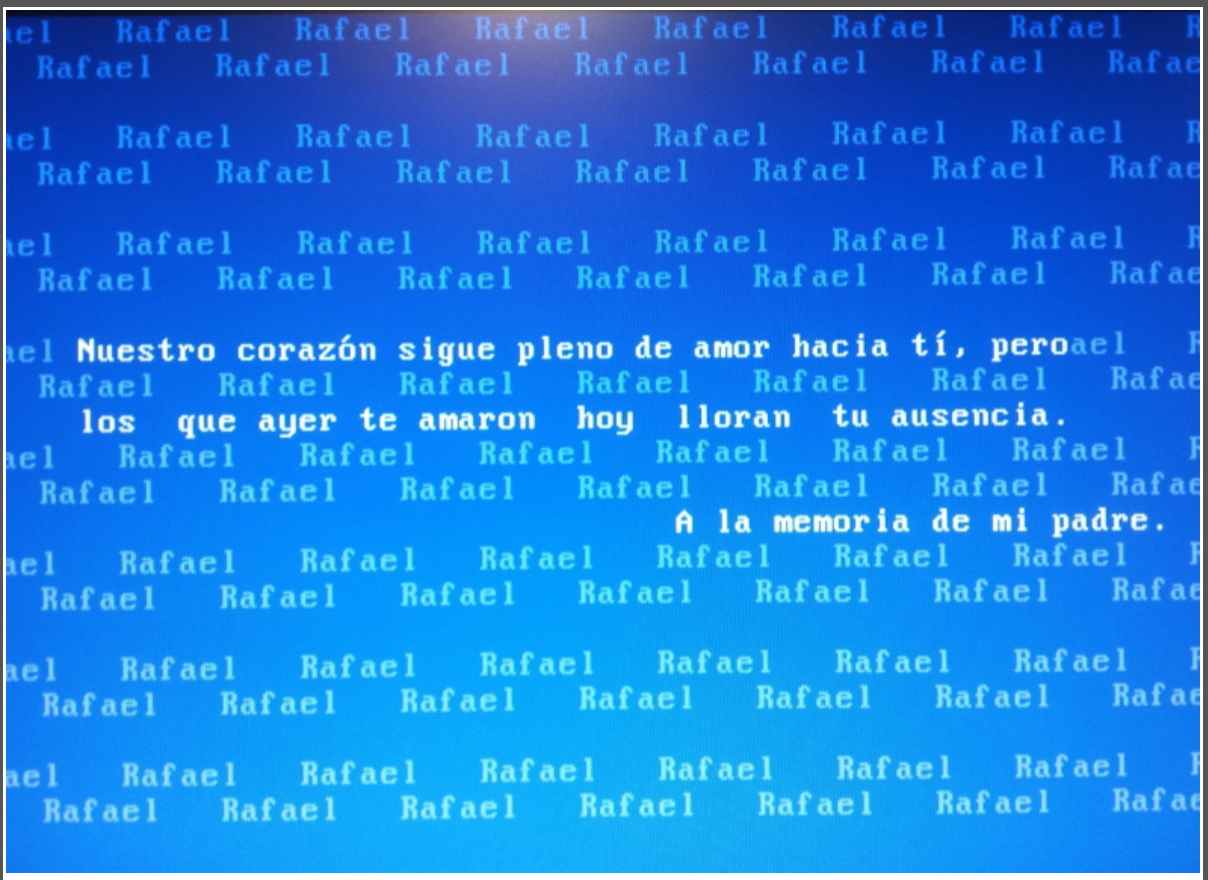
Nota sobre el artículo: LOS TIEMPOS DE CÁLCULO OFRECIDOS EN ESTE ARTÍCULO HAN SIDO OBTENIDOS A PARTIR DEL CÁLCULO DE REFERENCIA, TOMADO COMO TAL LA SIMULACIÓN DE UN MILLÓN (1.000.000) DE SORTEOS MEDIANTE EL MÓDULO **PROBABILIDAD DE GRUPOS** Y ENTRANDO LOS VALORES NUMÉRICOS: **6** y **49**

Simultaneándose con esta primera aventura de TEX2COM, otra curiosa y no menos trepidante historia estaba a punto de conectar dos vidas que se encontraban a más de mil kilómetros. Esta vez, **SIMULA**, otro viejo programa *Freeware* que desarrollé inicialmente para mi reluciente *Sinclair ZX-Spectrum*!! y posteriormente adaptado a máquinas compatibles sobre MS-DOS, iba a convertirse en protagonista absoluto de este relato.

La última versión del programa freeware **SIMULA** de Lotería Primitiva y Bonoloto 6/49 compilado en 16 bits, puede aún ejecutarse directamente sobre plataformas *Windows 10-32bits* sin necesidad del uso de máquinas virtuales ya que las versiones de Windows 32 bits permiten la ejecución directa de software de 16 bits en una "especie" de máquina virtual integrada en el propio sistema, a diferencia de los sistemas operativos de 64bits en los que forzosamente debemos recurrir a otras vías alternativas como DOSBox ó USB-boot (llaves USB de arranque) y que veremos en esta sección.

SIMULA no es para mi un programa más. Es un desarrollo propio bastante más antiguo que TEXTCOM cuya creación se pierde realmente en la noche de los tiempos y por el que guardo un especial cariño. No en vano, sus primeros esbozos fueron escritos por un inquieto adolescente sobre un flamante *gomas* de 48K allá por el año 1984, justamente cuando solía arder la calle al sol del poniente y cuando en el insituto estudiábamos algoritmos para conseguir la generación sin repetición de números pseudoaleatorios.

Pero si hay alguien que ha marcado realmente la esencia y personalidad propia de este programa, ese ha sido sin duda su veradero autor intelectual, mi padre (fallecido en 1992), y todas las adaptaciones y actualizaciones posteriores que tuvieron lugar hasta la pérdida accidental del código fuente en 1997, han continuado la línea inicial del concepto que él supo transmitirme y del cual se impregnó SIMULA.



Dedicatoria oculta en el software SIMULA al verdadero autor intelectual de esta aplicación.

SIMULA es hoy un superviviente del siglo XX, una suerte de género ***SurviveSoftware*** (ahora que están muy de moda los anglicismos) que ha resistido al paso de las décadas y el cambio de siglo en un panorama evolutivo, el

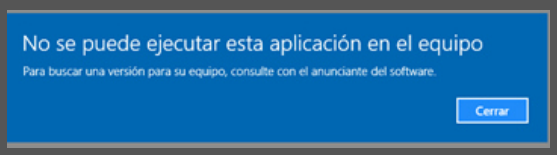
del software, cuya velocidad de cambio suele enterrar aplicaciones en el olvido en un plazo medio de 24 meses. Cuando hablo de SIMULA como un software *superviviente* ó *survivesoftware* del siglo pasado, no me refiero únicamente a que pueda ejecutarse en máquinas actuales como gran cantidad de software retro, sino a que aún despierta el interés de numerosos usuarios y continúa ofreciendo al investigador y/o apasionado de los juegos de azar las mismas herramientas incluso con mayores ventajas que en sus orígenes derivadas de la significativa potencia bruta de cálculo del hardware actual. Y de eso precisamente vamos a tratar en esta sección.

Cada cierto tiempo, algún usuario de SIMULA de cualquier ricón del mundo suele contactarme para consultarme algo sobre el uso del programa o sencillamente para felicitarme por el mismo. En ese sentido, hace ya años, a través de la web oficial de SIMULA que yo mismo mantengo, recibí varios mensajes de usuarios a los que el programa ya no les funcionaba sobre plataformas de Microsoft Windows de 64bits. Esto me sorprendió porque yo utilizaba (y utilizo normalmente) sistemas de 32bits y, hasta entonces SIMULA, aún tratándose de software de 16 bits, había corrido de forma impecable en los sistemas de Microsoft Windows XP. Lo cierto es que tras algunos experimentos descubrí ciertamente la ineludible incompatibilidad de las aplicaciones de 16bits con los últimos sistemas operativos de 64bits.

A raíz de ello, intenté buscar alguna solución al problema para poder mantener viva esta reliquia del cálculo probabilístico en los juegos de azar y cuya eficacia había sido manifestada en numerosas ocasiones por usuarios de todo el mundo, los cuales me contactaron a través de correo electrónico para realizarme consultas de todo tipo o simplemente para felicitarme por mi programa.

De este modo, lo primero que se me ocurrió fue servirme de la máquina virtual gratuita **DOSBox** para crear una distribución compatible con los sistemas Windows de 64bits y así ha estado funcionando hasta prácticamente hoy día. La solución eran bien sencilla ya que tan solo consistía en lanzar la aplicación (contenida en los ejecutables SIMULA.EXE, S01.EXE y s02.EXE) desde una sesión **DOSBox**, todo ello de forma totalmente transparente al usuario y sin necesidad de abir la máquina virtual ni pelear con la línea de comandos, es decir, directamente desde Windows.

Hasta aquí todo correcto y nada nuevo bajo el sol, sin embargo, hace un par de meses más o menos, un veterano usuario de SIMULA que me confesó utilizar mi programa desde hacía más de 20 años, contactó conmigo para plantearme un "pequeño" inconveniente al trabajar con SIMULA en su flamante equipo de sobremesa dotado de un microprocesador i7 y Windows 7-64bits. Transcribo aquí parte de su mensaje:

> *Hola Rafael. Hace ya muchos años que tuve conocimiento de tu programa y lo utilicé asíduamente. Por cosas de la vida, la loto pasó a un segundo plano y, lógicamente, perdí contacto con todo lo relacionado con ella. También los cambios de ordenador supusieron el fin de Simula. He intentado volver a tú página para descargarlo nuevamente y nunca he podido dar con la misma. Y hace una par de días, ¡aleluya!, buscando otra cosa me aparece tu web con "gutemberg30". Al parecerme interesante y los vagos recuerdos que tu nombre me inspiraba, pasé a la lectura obligatoria y dí con aquel famoso soft que tan bueno me pareció y me sigue pareciendo.*
>
> *Mi sistema es el siguiente: intel core i7 2700K @ 3.5 Ghz - 8 GB RAM - Windows 10 de 64bits*
>
> *Vamos por parte. Yo ya tengo 70 años y supongo comprendes que en esto de la loto tengo "el culo pelado", como se dice vulgarmente. Actualmente estoy con unas combinaciones que en el tiempo me suelen dar un rendimiento entre un 300 y 500%. Lógicamente no todas las veces, pero sí a medio plazo. Todo es hallar el equilibrio entre el gasto y el ingreso. ¿A qué es debido esto? Pues no lo sé; estoy igual que tú cuando te haces las mismas o parecidas preguntas al referirte a "EL GCP (Proyecto de la Conciencia Global) Y LOS GNA's" y sus causas y efectos, pero haberlas haylas. Contra toda lógica del azar, existen combinaciones mejores y otras peores. Pero entendiendo que los "0" y "1" dan casi siempre una línea plana, ¿cuál es la razón del cambio? Estoy de acuerdo con tus ideas sobre las claves de La Biblia. Creo que es debido también al más puro azar y solo es necesario un buen programa informático que lo detecte. En estos párrafos que llevo, ¿cuántas variaciones, permutaciones y combinaciones distintas habrá? ¿Y cuáles y cuántas serían las sorpresas? Apasionante y difícil el mundo del azar. Al intentar abrir la aplicación Simula de 88 KB; primi.exe y bono.exe; me sale la ventana siguiente:*
>
> 
>
>
> *Si intento abrir los archivos por lotes **BONOXP.BAT** ó **PRIMIXP.BAT** se me abre el programa del siguiente tenor:*

*Y puedo hacer lo que dicen los puntos, eso sí, sin darlos con el puntero del ratón, pues entonces no puedo salir de ahí. Tampoco me simula **1.000.000 de sorteos** como dices en el manual en 36 segundos. Me tarda bastantes minutos. Y eso, como ya te especifiqué en el anterior mail, que no es lento mi PC.*

El mensaje de error se obtiene al intentar ejecutar directamente la aplicación SIMULA contenida en los archivos .EXEs. Como decía antes, las aplicaciones de 16bits no pueden ejecutarse directamente en los sistemas operativos Windows de 64bits ya que no están firmadas digitalmente y Microsoft suprimió definitivamente la posibilidad de correr software de 16bits.

En el segundo caso en el que ya consigues arrancar la aplicación , al ejecutar los archivos .BATs ó archivos de procesos por lote, el sistema abre primero una máquina virtual DOSBox y luego lanza el ejecutable **SIMULA.EXE** desde la máquina virtual, con lo cual el programa SIMULA funciona de forma correcta pero con una limitación de velocidad que por supuesto no pasa desapercibida a Joaquín. Por otro lado, el problema del uso del ratón no es solucionable y aunque solo las últimas actualizaciones incluían la posibilidad de manejar el menú mediante ratón, el manejo del programa sigue siendo posible y necesario mediante el teclado.

Este pequeño inconveniente acerca del rendimiento al que hacía ilusión mi hoy amigo Joaquín García, no era una cuestión baladí. En sus primeros orígenes SIMULA corría sobre un microordenador de 8 bits y no fueron pocas las tardes enteras en las que padre e hijo quedábamos ensimismados frente al televisor observando con entusiasmo los contadores de aciertos. Lamentablemente no conservo ninguna copia de SIMULA para Spectrum, pero es bastante probable que un *ZX-Spectrum* inviertiera varias horas en efectuar 1000 (MIL) simulaciones y escrutar los resultados de las mismas. Esta limitación impuesta por el hardware, fue algo que quedo grabada a fuego en mi mente para siempre, y por ello, lo primero que hice cuando pude adquirir un PC con procesador de 32 bits (486) fue ponerme manos a la obra para desarrollar una nueva versión de SIMULA para PCs compatibles. Corría una fría navidad de 1994 por tierras del norte de España. Es importante destacar que el desarrollo de SIMULA en plataformas PC no solo se benefició del nuevo hardware de 32 bits, sino que además permitió compilar el código que antes era solo interpretado por el lento microprocesador ***Z-80***. Ni que decir hay que el software conoció una nueva dimensión en este panorama marcado por la carrera tecnológica de la velocidad.

Después, pasados muchos años desde aquellos lejanos 90' y tal y como he comentado antes, llegaron los sistemas Windows de 64 bits totalmente incompatibles ya con el software de 16 bits y ello me obligó a recurrir a remedios drásticos para conseguir mantener vivo mi programa. Así fue como adapté SIMULA para poder continuar ejecutándolo en sistemas de 64 bits. En la captura anterior verás que se muestra el menú principal del programa SIMULA en una ventana **DOSBox**, y en en el título de ésta se muestra un parámetro de **DOSBox** que es precisamente el número de ciclos (3000) al que se está ejecutando la máquina virtual o emulación. Sin embargo, **DOSBOx** nos permite ajustar este parámetro hasta 20 mil ciclos y conseguir una emulación más rápida pero aún así, continúa siendo excesivamente lento, tal y como pude constatar en mis pruebas y en mi respuesta a Joaquín:

REPUESTA 1.-******************** *En el aspecto técnico, debo decirte que SIMULA es un programa que desarrollé hace más de 20 años y fue compilado en un sistema operativo MS-DOS.*

Un día, debido a un accidente tal y como narro en las instrucciones de uso del propio SIMULA, perdí todo el código fuente y solo pude salvar y conservar el código ejecutable del mismo (archivos EXEs). Estos archivos EXEs que te comento funcionaban sin demasiados problemas (y además con un rendimiento altísimo) en máquinas con CPUs 486 y/o PENTIUMs incluso sobre sistemas Windows 3.1, 95, NT, 2000 y XP, hasta que llegaron los sistemas de 64 bits.

La cuestión es que para poder preservar el programa y seguir usándolo en sistemas operativos de 64 bits, la única forma que encontré fue utilizando una máquina virtual, esto es en realidad una aplicación (capa) que se ejecuta entre el sistema operativo y la propia aplicación SIMULA. Por ello, si tu sistema Windows es de 64 bits no podrás correr directamente los ejecutables de SIMULA, porque te dará error. En tu caso (cualquier sistema 64 bits), debes descargar la versión PORTABLE 3.17 (que acabo de publicar en la

web del programa) y tras descomprimir el archivo ZIP, ejecutar los archivos: PRIMI64.BAT ó BONO64.BAT , dependiendo de si quieres calcular rentabilidades/amortizaciones para el juego de la primitiva o el bonoloto.

REPUESTA 2.- ******************** *Al ejecutar una aplicación sobre una máquina virtual y no de forma nativa, el rendimiento o velocidad de cálculo de SIMULA está supeditado a la máquina virtual y ello ralentiza de forma tremenda el rendimiento. Para que nos entendamos,* ***si ejecutamos SIMULA de forma nativa en un ordenador más o menos actual con un sistema operativo Windows de 32bits, la simulación y el escrutinio de un MILLON de sorteos puede que no supere los 5-10 segundos (para procesadores de gama baja).***

Sin embargo, en la versión actual que acabo de publicar, he ajustado la máquina virtual al máximo rendimiento posible (20 mil ciclos) y ahora emplea ***15 segundos*** *en el desarrollo de* ***100 MIL SORTEOS****, en lugar de los 47 que empleaba en la versión anterior, por lo que espero que pueda serte algo más útil para tus experimentos. Aún así, sigo investigando la forma de multiplicar esta velocidad.*

Además de esta mejora, no olvides también que puedes aprovechar la capacidad multitarea de Windows lanzando varias ventanas con distintos experimentos de forma simultánea sin que el rendimiento/velocidad de cada una de estas ventanas se vea prácticamente afectado.

A pesar de mejorar notablemente la velocidad de cálculo mediante el ajuste de la máquina virtual **DOSBox**, ajustando la configuración a 20 mil ciclos y teniendo en cuenta que en un i7 ya se puede conseguir simular UN MILLÓN en unos **150 segundos** con este sistema, lo cierto es que tanto Joaquín como yo quedamos instatisfechos y por ello decidí seguir experimentando con el asunto del rendimiento.

El resultado del rendimiento obtenido mediante **DOSBox** no era de recibo. Tal y como reza en el propio menú principal de SIMULA, un ordenador de hace más de 20 años dotado de un procesador *PENTIUM* de primera generación a 100 Mhz de frecuencia, era capaz de simular y escrutar UN MILLÓN de sorteos en un tiempo medio de **36 segundos**. Por esto, era triste que corriendo SIMULA en un Intel-Core i7 a 3.5 Ghz sobre **DOSBox** a la máxima frecuencia de ciclos posible QUE YO CREÍA (20 mil), se invierta del orden de los **150 segundos** en realizar UN MILLÓN de simulaciones. De hecho, aún puedo recordar que al migrar LA IDEA del programa SIMULA al primer PC-Compatible que tuve en 1993, un 486 DX2 a 66 Mhz de intel, las primeras versiones de SIMULA, aún compiladas con compiladores no demasiado rápidos, ya simulaban y escrutaban UN MILLON de sorteos en apenas **dos minutos**, eso sí, corriendo en **MS-DOS**.

Por otro lado, al ejecutar SIMULA sobre *Windows XP-32bits* en mi ordenador de sobremesa, un modesto Pentium 4 a 2.5 Ghz, ya se pueden obtener cronos ciertamente fulgurantes de unos **10 segundos** incluso corriendo el programa en una ventana de Windows. El reto era claro, teníamos que intentar "cortar los manguitos de freno" al sistema operativo para que dejara a la CPU i7 desatar su verdadera potencia y a ello me puse. Más bien, puentear el propio sistema operativo Windows.

Tras estas pruebas, al profundizar debidamente en la documentación de las últimas versiones de **DOSBox**, observé el archivo **dosbox.conf** que contiene información acerca de los parámetros de configuración de la máquina virtual **DOSBox** y que nos permite en la sección relativa a la CPU un ajuste de velocidad que puede superar a los 20 mil ciclos, los cuales yo imaginaba erróneamente como techo del rendimiento:

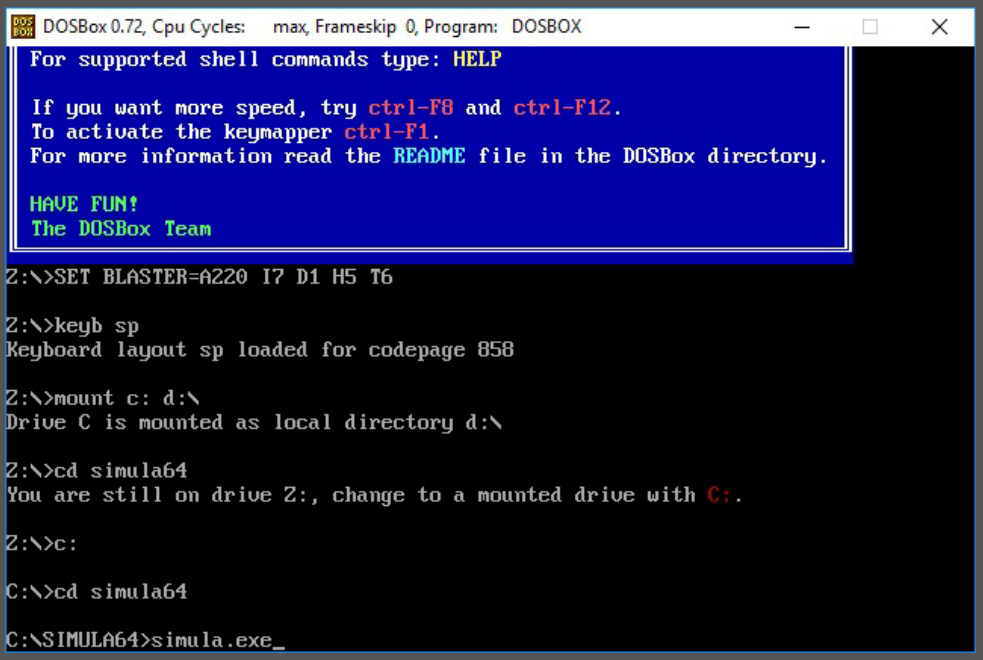
```
[cpu]
# core: CPU Core used in emulation. auto will switch to dynamic if
available and appropriate.
# Possible values: auto, dynamic, normal, simple.
# cputype: CPU Type used in emulation. auto is the fastest choice.
# Possible values: auto, 386, 386_slow, 486_slow, pentium_slow,
386_prefetch.
# cycles: Amount of instructions DOSBox tries to emulate each
millisecond.
# Setting this value too high results in sound dropouts and lags.
# Cycles can be set in 3 ways:
# 'auto' tries to guess what a game needs.
# It usually works, but can fail for certain games.
# 'fixed #number' will set a fixed amount of cycles. This is what
you usually need if 'auto' fails.
# (Example: fixed 4000).
# 'max' will allocate as much cycles as your computer is able to
handle.
# Possible values: auto, fixed, max.
# cycleup: Amount of cycles to decrease/increase with keycombo.
(CTRL-F11/CTRL-F12)
# cycledown: Setting it lower than 100 will be a percentage.
```

```
core=auto
cputype=auto
cycles=max
cycleup=10
cycledown=20
```

De este modo, al probar SIMULA en un *intel-Core i7-4720 HQ* de 64bits a 2.6 Ghz con la configuración MAX, el programa SIMULA consigue ya un crono nada desechable de **15 segundos** en la simulación y escrutinio de UN MILLON de sorteos. Pero, ¿Eso es todo? ¿Era esto lo máximo que podíamos exprimir nuestra máquina? Evidentemente no, y era sencillo de llegar a esa conclusión. Coincidirán ustedes conmigo en que si un procesador PENTIUM-1 a 100 Mhz podía realizar UN MILLON de simulaciones en apenas **36 segundos**, parece no responder a la lógica ni a la proporcionalidad que un procesador actual con una frecuencia de trabajo 30 veces superior y una arquitectura infinitamente más optimizada, apenas fuera el doble de rápido que un intel PENTIUM de primera generación comercializado hace más de 20 años.

Era evidente que una buena parte de ciclos de procesador seguían escapando a nuestro control y ahora había que buscar por dónde se perdían para intentar llevar la velocidad de cálculo al límite del procesador. Si un viejo procesador con más de 20 años había sido capaz de conseguir un tiempo de 36 segundos corriendo sobre MS-DOS, por tanto, la solución debía estar en el sistema operativo.

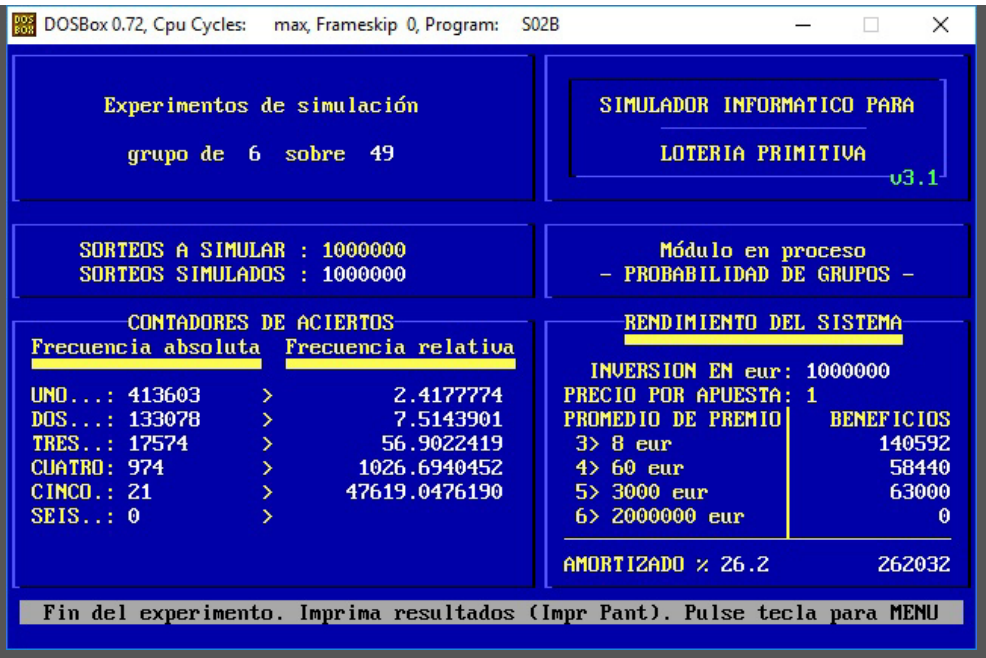
(Bueno, ahora que parece que toda la familia tiene algo que hacer retomo mi trabajo en esta tranquila tarde otoñal de domingo ;))) Tomo mi PEN-Drive, enciendo el ordenador de mi hijo y arranco *W10-64 bits* en su equipo con procesador AMD-Athlon II X2-250 a 3.00 Ghz, ... pincho en un USB frontal el Pendrive y abro el archivo (que lees) **index2.html** con un *Dreamwaver 8* de 2005 (lo cierto es que a mi me parece demasiado moderno jejejejeje!! y aún recuerdo cuando escribía mis primeras páginas web con el bloc de notas jejejeje!!!). Ya estoy trabajando.... Abro una sesión de DOSBox directamente desde el Pendrive (DOSBox es totalmente portable si adjuntas al ejecutable DOSBOX.EXE los archivos SDL.DLL y SDL_NET.DLL :



... preparo una sesión para probar SIMULA a la MAXima velocidad posible que **DOSBox** nos ofrece y voy corriendo a por mi viejo cronómetro para comparar el rendimiento del ordenador portátil de mi hija con procesador *intel-Core i7-4720 HQ* de 64bits a 2.6 Ghz contra el ordenador de sobremesa de mi hijo dotado de un procesador *AMD-Athlon II X2-250 a 3.00 Ghz*, algo más antiguo...

Resultado: ***40 segundos*** invertidos en el cálculo de referencia, es decir, UN MILLON de sorteos simulados y escrutados. No olvidemos que esta misma prueba es la que había arrojado una cifra de ***15 segundos*** en el ordenador de mi hija, un portátil GAMER marca ASUS dotado de procesador *intel-Core i7-4720 HQ de 64bits a 2.6 Ghz.* La diferencia entre ambas plataformas es significativa teniendo en cuenta que ejecutamos SIMULA (software de 16 bits) en una máquina virtual DOSBox (de 32 bits) corriendo en ambas máquinas sobre *Windows 10-64bits y 8GB de RAM*. En estas circunstancias, el *iNTEL i7* saca pecho y aplasta con cómoda ventaja al "modesto" *AMD-Athlon II X2-250* aún corriendo a menor frecuencia (2.60 Ghz del *i7* frente a los 3.00 Ghz del *AMD Athlon II X2*)

¿Qué lectura podemos extraer de esta diferencia? Pues que no todos son los Hertzios de frecuencia de procesador son iguales y que las arquitecturas más modernas y complejas pueden obtener mejores resultados ejecutando sistemas operativos de 64 bits.



Sin embargo, la última palabra en cuanto al rendimiento extremo no estaba ni mucho menos dicha. Ahora había llegado el momento de liberar al equipo de esa pesada capa que es *Windows 10-64bits* y exprimir cada ciclo de procesador al límite. Para ello, necesitamos cargar un sistema operativo que apenas ocupe unos pocos KBs en memoria y lo que es más importante, que no entretenga a la CPU con cientos de servicios y procesos residentes, necesitamos *FreeDOS*.

FreeDOS es una distribución de **LiNuX** totalmente gratuita y compatible con *MS-DOS* que puedes instalar en un viejo Pendrive (de 1 GB sobra) para arrancar en cualquier ordenador. Para conseguir esto puedes servirte de algunas utilidades que sirven para crear Pendrives de arranque cargando en ellos un sistema operativo.

Tras algunos experimentos fallidos con algunas utilidades destinadas a la creación de USB de arranque, incluyendo bloqueos varios y reventón total del sistema operativo de mi netbook, finalmente me decanté por UNitbootin para Windows por su enorme sencillez y por su garantía de resultados. Para crear el Pendrive de arraque deberás tener conexión a internet ya que la propia utilidad nos ofrece la posibilidad de conectarse y descargar el sistema operativo completo en el Pendrive que le indicamos. Entre los sistemas operativos se encuentran un sinfín de distribuciones de *LiNux* además de otros interesantes sistemas completos. También nos ofrece la posibilidad de instalar el sistema desde un fichero de imagen tipo ISO.

Tras realizar los experimentos oportunos y preparar algún viejo Pendrive que andaba perdido por casa, el sistema *FreeDOS* ya estaba a punto. Una vez que tengamos el USB listo, tenemos básicamente dos opciones para poder arrancar el sistema desde el USB, para lo cual es imprescindible que el arranque desde USB debe esté soportado por la BIOS de nuestra computadora. De no ser así, deberíamos intentar actualizar la **BIOS** para acceder a esta posibilidad, pero normalmente, todos los ordenadores de con menos de 10-12 años ofrecen esta función.

> **La primera opción** es la más sencilla y podemos adoptarla si la **BIOS** de nuestra placa base lo permite. Para ello es fundamental averiguar la tecla de FUNCIÓN que debemos presionar durante el inicio del ordenador para que se nos muestre en pantalla el menú que permite arrancar el sistema desde USB, CD-Rom o cualquier otro dispositivo que la **BIOS** detecte. En el caso del ordenador de mi hijo, la tecla de marras es la ***F8*** (pero puede ser ***F9***, ***F12***, etc.. en función de la placa) y debemos ser rápidos y pulsarla justo al mostrarse el logo de ASUS en la pantalla. Mejor dejarla pulsada desde que encendemos el equipo.
>
> **La segunda opción** es quizá algo más engorrosa pero puede que sea la única posible y pasa por acceder a la **BIOS** (ya sabes, ese programa que el fabricante graba en la memoria ROM de la placa y que configura todo el hardware de nuestro sistema. El acceso a la **BIOS** suele ser mediante las teclas de borrado ***SUPR*** ó la de fundión ***F2***, y debemos pulsarla justo en el momento de encender el ordenador y antes de que se cargue el sistema operativo. Una vez abierta la **BIOS**, para que sea posible arracar desde USB debemos establecer al dispositovo USB como de máxima prioridad en el arranque, justo por encima del disco duro y cualquier otro dispositivo que podamos tener (CD-ROM, LAN, etc..). Esta opción se encuentra normalmente en la sección BOOT. Eso sí, en la **BIOS** toca siempre lo mínimo imprescindible ya que puedes desconfigurar algo y hacer que el sistema se vuelva inestable o simplemete deje de funcionar. Así que ya sabes, entrar, modificar lo justo, guardar y salir.

Y concluida ya la fase de "taller" y con nuestro flamate Pendrive de arraque con *FreeDOS* pinchado en un puerto USB, cruzamos los dedos y nos disponemos a la prueba definitiva, puentear a Windows para conseguier alcanzar la potencia extrema. ¿Lo conseguiremos? Júzgalo tú mismo.

TIEMPO INVERTIDO por el equipo con procesador *AMD-AThlon FX2 64bits @ 3.00 Ghz* en simular UN MILLÓN de sorteos: **1,3 segundos**

Video de SIMULA procesando **DIEZ MILLONES DE SORTEOS** en ***13 segundos a razón de UN MILLÓN por cada 1.3 sg***

Ahora era el momento de dar una vuelta más de tuerca subiendo el *FrontBUS* del procesador desde los 200 hasta los 240 Mhz para conseguir una frecuencia final efectiva de 3.60 Ghz (240 x 15 = 3600). Esto es lo que siempre se ha conocido como overclocking y en los procesadores actuales suele ser bastante fiable si el sistema de refrigeración no compromete a la estabilidad del sistema, además, tampoco debemos andar trasteando los jumpers de la placa base sino que podemos configurarlo cómodamente desde el programa de la ***BIOS***. Por otro lado, los procesadores modernos, tanto de *iNTEL (SpeedStep)* como *AMD (Power Now!/Cool'n'Quiet/Optimized Power Management)*, incorporan una sofisticada tecnología que reduce de forma automática la frecuencia del procesador en función de la demanda, reduciendo drásticamente el calor disipado y la energía consumida por éste. Podríamos decir que este sistema nos permite incrementar el rendimiento de un procesador actual entre un 10 y 15% con ciertas garantías (y sin acabar retorciéndonos bajo la mesa para reajustar los jumpers de la placa base!!)

TIEMPO INVERTIDO por el equipo con el mismo procesador *overclockeado* *AMD-AThlon FX2 64bits @ 3.60 Ghz* en simular UN MILLÓN de sorteos: **1,2 segundos**

Video de SIMULA procesando **DIEZ MILLONES DE SORTEOS** en ***12 segundos a razón de UN MILLON por cada 1.2 sg***

Para nuestra sorpresa, el "modesto" *AMD-AThlon FX2 64bits* (incluso sin overclock;) acaba tumbando por goleada al novísimo *iNTEL i7 @ 2.6 Ghz* del portátil ASUS GAMER que ha marcado un crono de **2.7 segundos** en desarrollar UN MILLON de sorteos. ¿De veras quieres saber a qué se puede deber esta diferencia de rendimiento? Pues aquí la respuesta del sabio a esta sorpresa:

> *"Me resulta interesante que el software de **SIMULA** vaya mejor en un AMD64 que en un i7, pero si lo piensas, puede tener sentido. La causa sería la **BIOS**. Cuando IBM sacó sus primeros PCs tuvo una idea grandiosa: en vez de tener que generar drivers específicos vía software para cada componente (algo que ocurre con los sistemas operativos modernos) lo que hizo fue generar una capa de bajo nivel en el **firmware** de la placa base (las famosas interrupciones de la **BIOS** como la 20h, 21h, etc.) y ya se encargaba el fabricante de turno de adaptar su componente hardware para que funcionara con una interfaz de llamadas a la BIOS. Con esto se consiguió que el **MS-DOS** no llevara ningún driver en 1981. De hecho, si te das cuenta, en los 90 solo metíamos drivers al **MS-DOS** de hardware que no exitía en el 81 (tarjetas de sonido, tarjetas de red, ...) pero nunca teníamos que meter controladores de disco, disketes, etc.*
>
> *Linux y Windows abandonaron el uso lmitado de la BIOS y comenzaron a interactuar con el hardware ellos mismos. La limitación es que la **BIOS** tiene una interfaz genérica y por lo tanto trata a todo el hardware por igual, no aprovechando características específicas. Me explico: si tu tienes un disco duro con una caché de varios niveles o un disco híbrido, si accedes a este hardware vía **BIOS** te va a obligar a interactuar con el disco de la misma manera que interactúas con un disco duro sencillo y simple. Si accedes a él vía driver, puedes aprovechar todas sus características. Más allá del acceso al hardware, la **BIOS** también se encarga de arrancar el sistema operativo, y aquí es donde vienen los problemas más gordos.*
>
> *La **BIOS** es un software de 16bits porque los procesadores arrancan por temas de compatibilidad en **"modo real"**. Sin entrar en muchos detalles (tampoco sé si los conoces y te estoy aquí dando la chapa) el **"real mode"** es la configuración para que el procesador funcione a 16bits (a parte de otras cosas). En un sistema moderno, como el AMD64 de Rafa, ocurre la siguiente magia:*
>
> - *El procesador entra en **Real Mode** al arrancar.*
> - *Ejecuta la BIOS (que es de 16 bits).*
> - *La BIOS busca el sistema operativo y lo carga en memoria RAM*
> - *El procesador se pasa a "modo protegido"*
> - *Se le da la ejecución al SO*
>
> *Realmente el proceso es un pelín más complejo debido a algunos bugs que arrastramos en el hardware desde la época del 286, pero este sería el resumen. Activar el modo protegido de un procesador tiene varias implicaciones: por un lado das el salto a que el procesador ya pueda funcionar en 32bits, por otro activas todos los mecanismos del procesador para poder ejecutar un sistema multitarea, pero tiene un efecto colateral: impide que puedas usar las interrupciones de la **BIOS**. Por eso los sistemas multitarea como Windows o Linux no pueden funcionar con la **BIOS** y deben tirar de drivers para cada componente hardware. Al **MS-DOS** esto le da igual, es monotarea y no necesita el modo protegido. Un apunte: si el sistema operativo es de 64bits, el procesador tiene que entrar en un modo más: el **"Long mode"**, pero para la explicación no nos afecta.*

*Bajo todas las premisas anteriores tenemos un problema muy jodido: el componente que arranca el SO es un programa de 16bits (pufff, me doy cuenta de que escribo muchas imprecisiones por evitar enrollarme mucho, no tomes todo esto que te cuento como un dogma de fe, sólo pinceladas para tener una idea del tema). Por precisar un poco: a efectos prácticos, por magias del **"real mode"**, es como si fuera un programa de **20bits**. Y aquí viene el lío: **2^20 = 1MB**, por lo tanto tienes un megabyte de RAM para gestionar el arranque de un sistema operativo. En la época del **MS-DOS** ibas sobrado (recuerda los **640K**), pero en los últimos 15 años esto ha sido un dolor de cabeza para los desarrolladores de SS.OO. De ahí que Linux necesite un **"Loader"** y rollos raros así. Hay sistemas que cargan un **minikernel** que luego gestiona la carga del **kernel** completo. Bueno, soluciones al problema hay muchas, pero... es un problema. Este es uno de los más gordos, pero hay más, como el tamaño máximo de un disco de arranque y cosas así, tampoco es plan de sacar todos los trapos sucios de la gloriosa **BIOS** jajaja. Los fabricantes de hardware se pusieron hace unos años de acuerdo y decidieron sacar una **BIOS** de **32bits**, sin las limitaciones que tiene la vieja **BIOS**. Es lo que se conoce ahora como **UEFI**, aunque en la mayoría de las placas base se sigue llamando al **UEFI** como **BIOS** para evitar líos mentales al usuario y la gente se cree que sigue teniendo una **BIOS** estándar en su PC.*

*Entre las muchas diferencias entre **UEFI** y **BIOS**, la que nos interesa es que **UEFI** no tiene el sistema de interrupciones implementado, ya que los sistemas modernos de 32 y 64 bits no lo necesitan. Sin embargo esto impediría poder ejecutar sistemas operativos actuales como **FreeDOS**, que al igual que el viejo **MS-DOS**, funcionan a base de llamadas a la **BIOS**. Para solucionarlo, el **UEFI** te mete una capa de emulación de la **BIOS** (lo que en algunas placas llaman **BIOS Legacy**). Una capa extra = menos rendimiento.*

*Por otro lado los procesadores más modernos van dejando de lado y descuidando el set de instrucciones de 16bits, lo que implica que tienen un peor rendimiento ejecutando programas de 16bits que si ejecutaran los mismos programas en 32 ó 64bits. El motivo es que algunos simulan el **"real mode"** en lugar de ejecutarlo realmente. No tener algo optimizado a tan bajo nivel empeora el rendimiento de manera notable a alto nivel."*

5> PUESTA A PUNTO DE UN "VIEJO" PORTATIL

Hace aproximadamente 10 años, en 2007, adquirí para mi hija un equipo portátil MSI de bajo coste a un precio bien atractivo para el momento. Sus características eran muy básicas, procesador *AMD SEMPRON de 32 bits* (equivalente al *intel Celeron* de la época), 512 MB de RAM (DDR2 @ 667 Mhz), HD de 80 GB e interfaz SATA y una generosa pantalla de 15'6 pulgadas de 1280 x 800 píxeles de resolución gestionada por una modesta gráfica *GeForce 6100*.

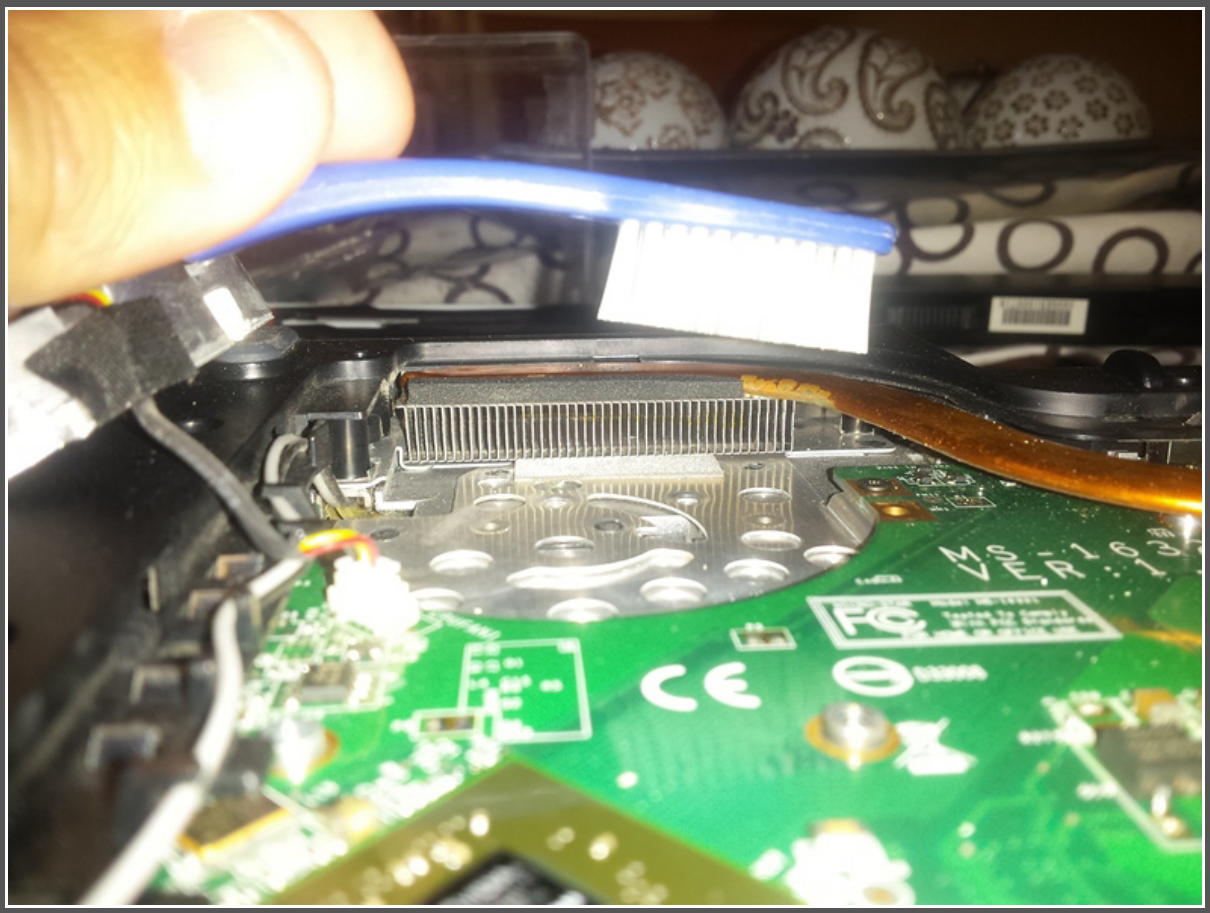
Este equipo costó 300 euros y en su momento podemos decir que fue una inversión redonda. Su reducido precio venía dado en parte al carecer de Sistema Operativo, ya que venía solo con una versión de *FreeDOS* instalada. Yo le instalé *Windows 2000* y *Windows XP* en la misma partición, ya que de esta forma podías compartir desde ambos sistemas aplicaciones como (Officce, Visual Basic, etc.) en la misma carpeta **ARCHIVOS DE PROGRAMAS** con el consiguiente ahorro espacio. Además, con la instalación DUAL siempre te garantizas el poder trabajar si se daña alguno de los sistemas por cualquier motivo (Virus, etc). En cuanto pude, amplié su RAM hasta 1,5 GB y el equipo siempre fue lo suficiente ágil en tareas de ofimática y trabajos escolares, sobre todo moviendo *Windows 2000*, una evolución de Windows NT bastante robusta con la que conviví bastante tiempo y cuyos requisitos de hardware eran bastante menos exigentes que *Windows XP*.

Lo cierto es que este equipo, hoy casi desterrado, sigue andando por casa y no son pocas las veces que lo pongo en marcha para hacer cualquie cosa. Finalmete conseguí ampliarle la RAM hasta los 3 GB y aunque ya eliminé el sistema Windows 2000, la velocidad de trabajo con Windows XP es realmente buena y su pantalla sigue siendo más generosa que la de mi machacado NETBook SAMSUNG con solo 10 pulgadas. Si embargo, de un tiempo a esta parte el equipo se apagaba al transcurrir un rato, síntoma muy típico de los portátiles cuando tienen algunos años. De manera que me dispuse a examinarlo para intentar llegar a un diagnóstico y a una solución.

En realidad, no es que sea yo ningún manita ni mucho menos, es que la experiencia con ordenadores durante casi 40 años me han revelado "misterios" que a veces pueden escapar al común de los mortales. En este caso, se trata de un misterioso secreto celosamente oculto entre plásticos y carcasas, consistente en una amalgama amorfa de ácaros, polvo y pelusas que acaban asfixiando a la máquina.

Véase la imagen antes:

Y después de un buen repaso de cepillo y aspirador...



La gran parte de las placas base de portátiles, suelen incorporar un sistema de seguridad que apaga el equipo en el caso de que se alcance una temperatura elevada en la **CPU**. Este límite es a veces ajustable desde la **BIOS** y garantiza que nuestro procesador no acabe tostándose.

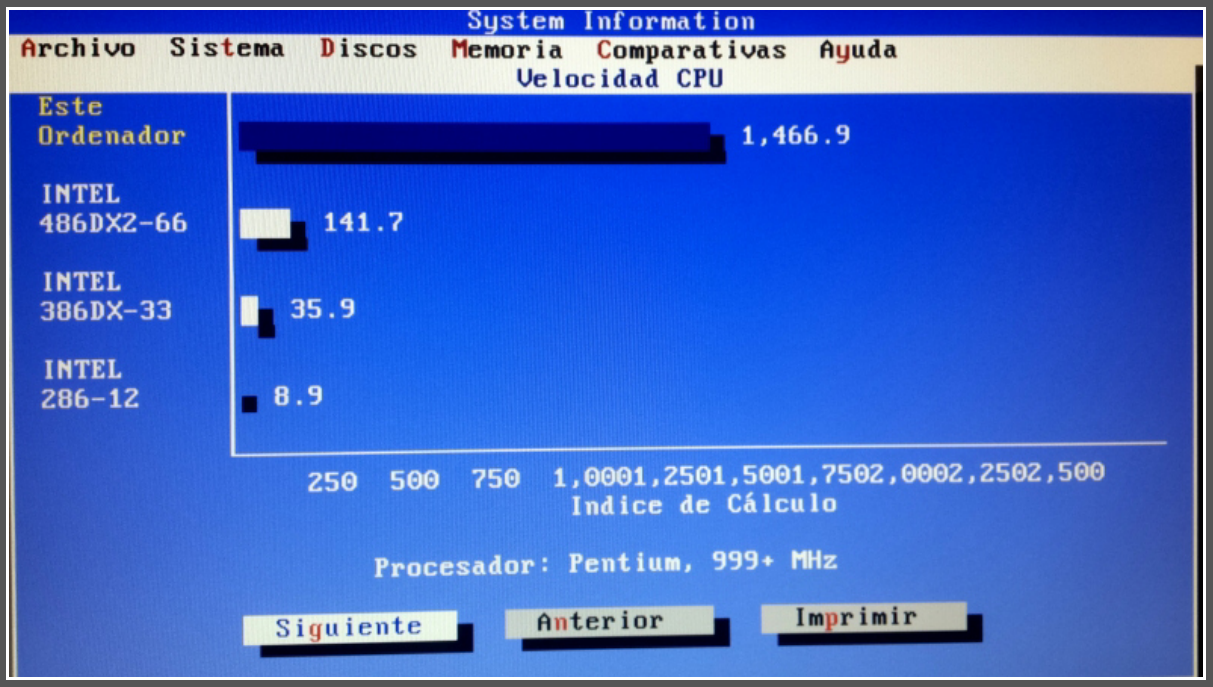
Debo decir que al monitorizar la temperatura de la placa y la CPU tras la exhaustiva limpieza (fundamental insistir en disipadores, rejillas de salida y también en las aspas del ventilador), el equipo ha reducido

notablemente la temperatura de trabajo y funciona ya correctamente en determiandas tareas que no requieren un uso intensivo de procesamiento (ofimáticas), sin embargo, al trabajar con aplicaciones exigentes y pesadas (juegos, video, navegación...) sigue alcanzando el techo de temperatura permitida y continúa apagándose, por lo que podemos decir que solo hemos resuelto el problema a medias, y la intuición me dice que la silicona térmica que se utiliza para montar el disipador de la CPU ha podido dejar de cumplir con su misión, algo lógico si consideramos la edad del equipo superior a los 10 años.

Pero este asunto lo trataremos en un próximo número (cuando disponga de silicona de plata o térmica para sustituirla) y prometo daros fiel cuenta de ello.

6> LA FUERZA BRUTA EN MILLONES DE OPERACIONES POR SECUNDO

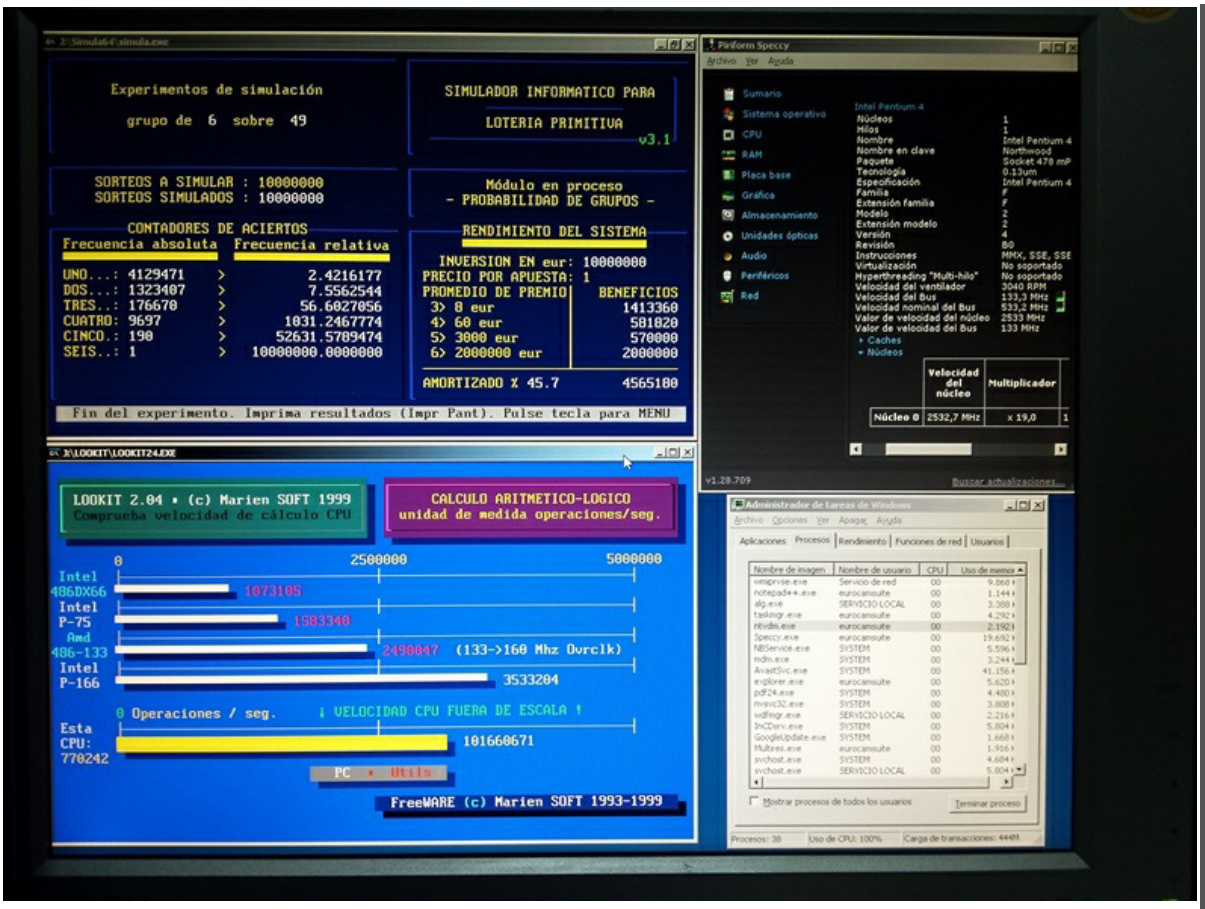
Al arrancar el sistema *FreeDOS* desde USB y al hilo de los numerosos experimentos realizados con SIMULA en la búsqueda del MÁXIMO rendimiento y los soberbios resultados obtenidos, volvieron a revolotear en mi estomágo las mariposas que despertaron mi ancestral interés por el tema y me resulto imposible resistirme a la tentación de continuar sometiendo a los procesadores a otras pruebas de rendimiento distintas. El resultado de los experimentos aún nos deparaba alguna inesperada sorpresa.



Las NORTON UTILITIES 7 para MS-DOS detectan al procesador *AMD-Athlon II X2-250 a 3.00 Ghz* como un procesador intel PENTIUM a más de 999 Mhz.

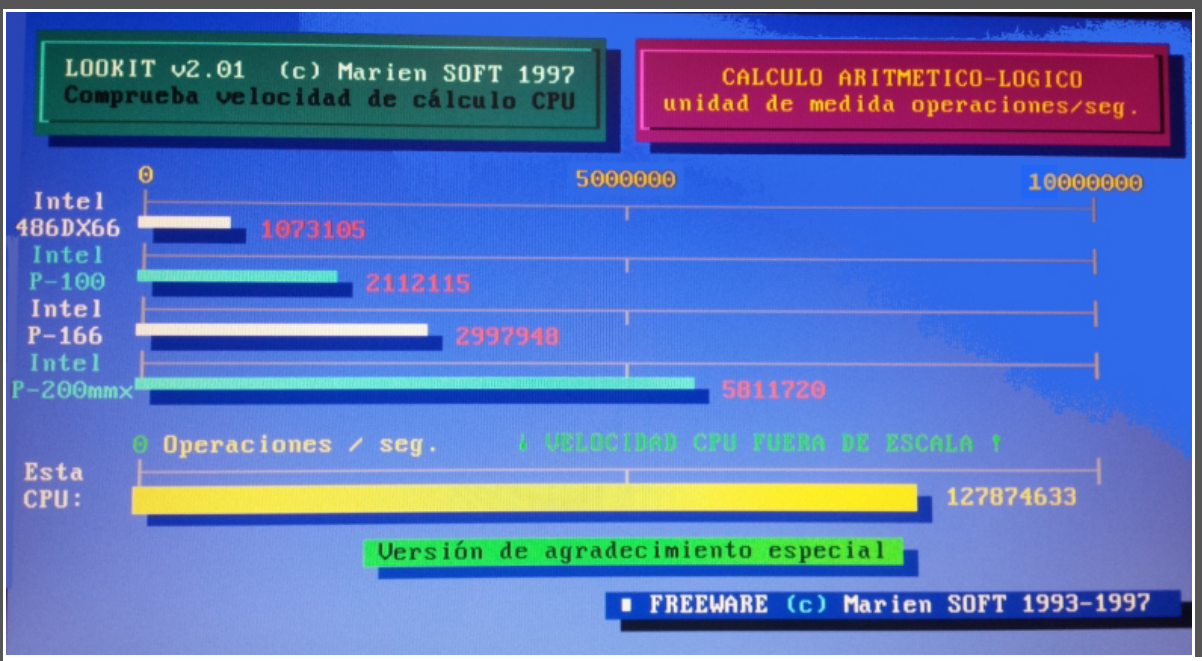
Si en la sección anterior dedicada al programa de lotería SIMULA tomabamos como referencia de cálculo para medir la velocidad del procesaror el tiempo invertido en la simulación de UN MILLÓN de sorteos (desde la opción PROBABILIDAD DE GRUPOS de esta aplicación), ahora me centré en una utilidad dedicada específicamente a la medición del rendimiento que yo mismo desarrollé y que gozó de cierta popularidad entre las revistas técnicas del momento (1996-97): **LOOKIT**. ¿Los resultados? Sencillamente IMPRESIONANTES.

En sistemas Windows XP-32bits podemos disfrutar de una compatibilidad total y una ejecución intachable de prácticamente cualquier software de 16 bits creado para *MS-DOS*. Por ello, es este un sistema altamente recomendable para usuarios que no quieren renunciar al amplio repositorio de sofware de 16 bits hoy disponible.



En la imagen se muestran las enormes posibildades y el alto rendimiento que pueden ofrecer procesadores ya desfasados como el PENTIUM 4 Nortwood (un único núcleo) a 2,53 Ghz corriendo sobre un *Windows XP-32bits*. Veáse el resultado que se muestra en la utilidad LOOKIT 2.04, con una cifra que supera los **100 millones de operaciones por segundo**.

En las pruebas de referencia realizadas en mi veterano *iNTEL PENTIUM IV @ 2.53 Ghz* sobre Windows XP-32bits, el tiempo invertirdo por **SIMULA** en simular UN MILLÓN de sorteos se ha situado entre los **5 y los 10 segundos**, en función del nivel de prioridad que asignemos al proceso (este parámetro puede ajustarse desde el propio ADMINISTRADOR DE TAREAS DE WINDOWS). Nada desdeñable si consideramos que un flamante procesador *intel i7 @ 2.6 Ghz* ha invertido **2.7 segundos** en realizar el mismo número de simulaciones corriendo *FreeDOS*.

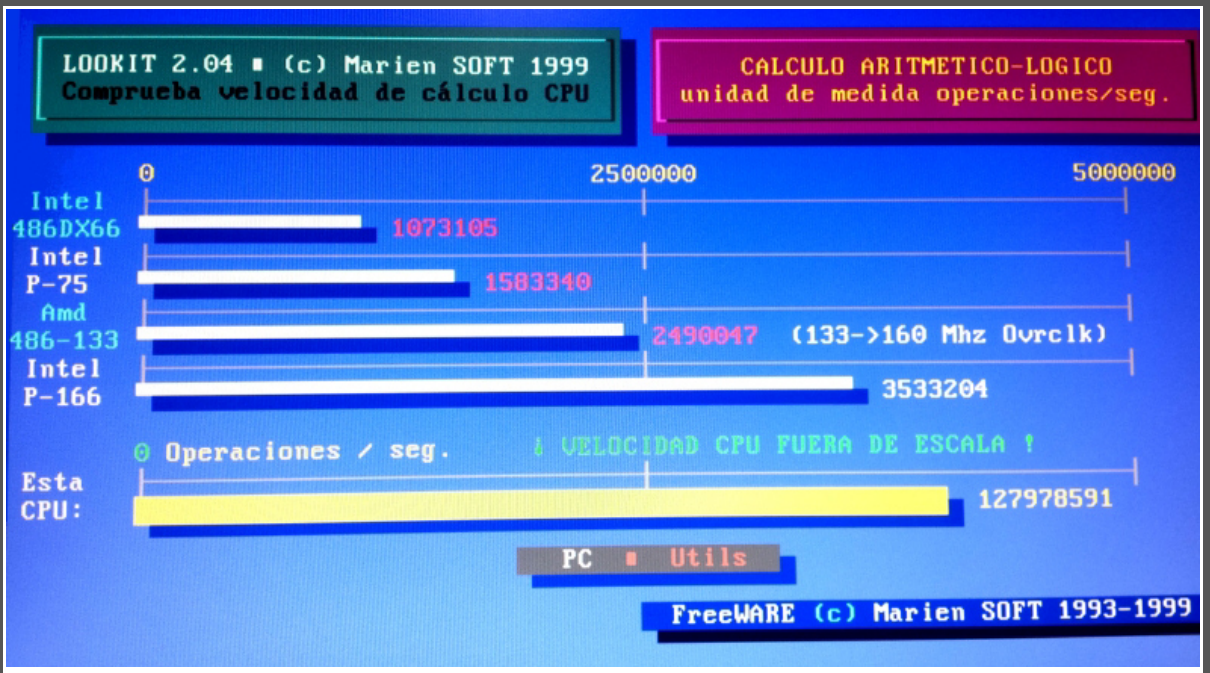


La prueba aquí mostrada es la realizada a un procesador *AMD-Athlon II X2-250 a 3.00 Ghz* y que logra un total de operaciones aritméticas de suma de más de 127 MILLONES por segundo.

La utilidad **LOOKIT** fue un desarrollo sobre *MS-DOS* bastante espartano y simplista que servía para medir la velocidad de cálculo de los procesadores mediante comparativa con otros procesadores que pasaban por mis manos y que yo mismo testeaba.

Aunque no conservo ninguna copia de la versión 1.0, recuerdo vagamente que las primeras versiones de **LOOKIT** fueron compiladas en ***TURBO BASIC*** de Borland, un compilador de BASIC que generaba un código ejecutable bastante más rápido que el compilador de Microsoft *QuickBASIC*. Debo decir que el código fuente de

LOOKIT era de lo más sencillo y se apoyaba básicamente en un bucle **DO... WHILE** que se dedicaba a incrementar un acumulador numérico y que comprobaba a cada segundo el valor alcanzado por dicho acumulador.

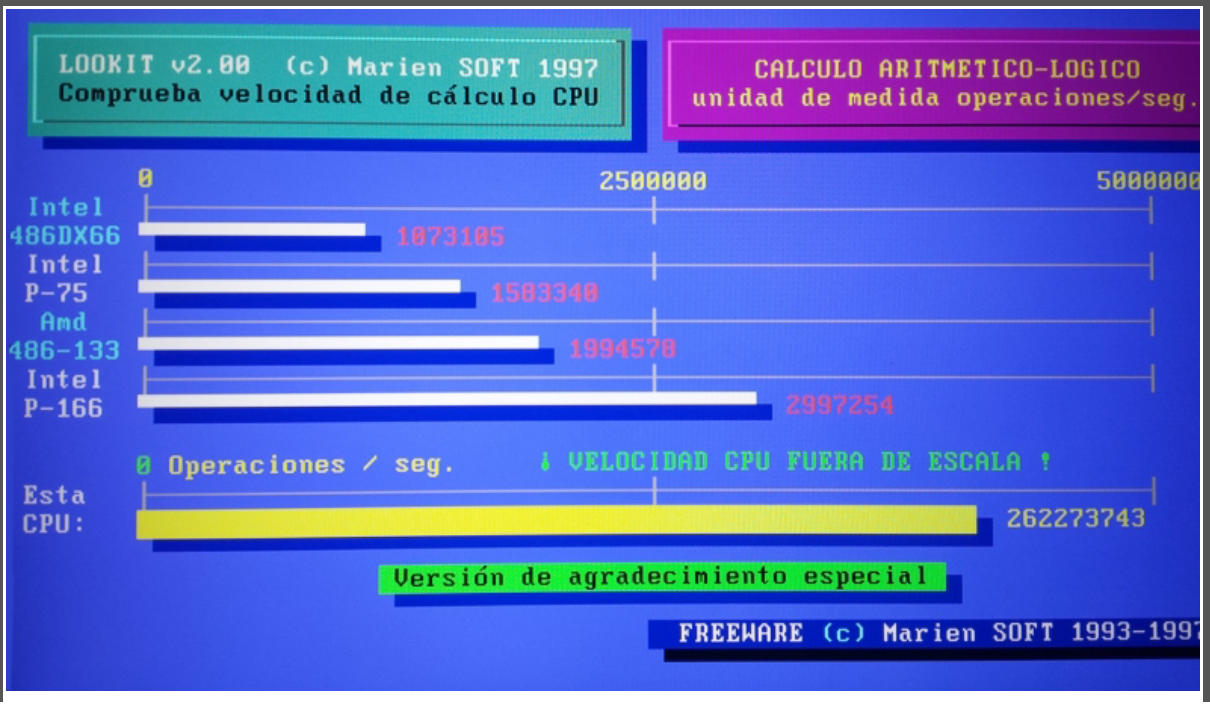


Ya puestos, ¿Cómo no probar las diferentes versiones de esta entrañable y minúscula aplicación de apenas 30 KB de tamaño? En la versión de la imagen, a la que numeré 2.04, se puede apreciar el "descomunal" rendimiento del procesador ***AMD-5x86-133*Mhz** *overclockeado* a 160Mhz, un procesador que paso por mis manos y que ostenta el récord de ser el procesador con arquitectura 486 más rápido de la historia. En la serigrafía del chip y en las referencias comerciales se podía ver la leyenda 5x86, pero esto no era más que una denominación puramente comercial, en realidad se trataba de un 486, eso sí, rápido como el rayo y casi capaz de duplicar a un señor intel Pentium a 75 Mhz en algunos procesos.

Las últimas revisiones de la utilidad **LOOKIT** fueron compiladas usando ya el supercompilador ***PowerBASIC 3.2*** de *Robert Zale*, probablemente uno de los más rápidos compiladores del mundo para lenguaje BASIC (de la era 16 bits) y que también sirvió para crear el código ejecutable de la últimas versiones de **SIMULA**.

> **NOTA:** Quiero aprovechar para informar a los interesados en conocer esta herramienta que mi viejo amigo Miguel del que hablaba al inicio de este FINDE's Retro #2 y gran conocedor de este entorno de desarrollo, me comentó que en breve podrían liberar los derechos de la versión **3.5** de ***PowerBASIC***, una grandísima noticia para todos aquellos que un día pudimos disfrutar de esta auténtica maravilla de la era **MS-DOS**.

Y ahora la sorpresa. Cuando ya parecía que el *iNTEL i7* era incapaz de desenvolverse al mismo ritmo que el *AMD Athlon x2 de 64 bits* (mucho más antiguo) ejecutando viejo software de 16 bits, un aplastante resultado de **262 millones de operaciones por segundo** en **LOOKIT 2.0** por parte del procesador de INTEL acaban finalmente fulminando a su competidor con un registro de rendimiento que supera en **260** veces a mi querido y "poderoso" *intel 486 DX-2 @ 66 Mhz* de 1993.



En contra de todo pronóstico, con la utilidad LOOKIT 2.0 corriendo sobre *FreeDOS*, el moderno procesador ***iNTEL i7 @ 2.6 Ghz*** vuelve a sacar pecho aplastando literalmente al ***AMD Athlon II X2 de 64 bits*** en idénticas condiciones. Nada más y nada menos que 262 millones de operaciones por segundo.

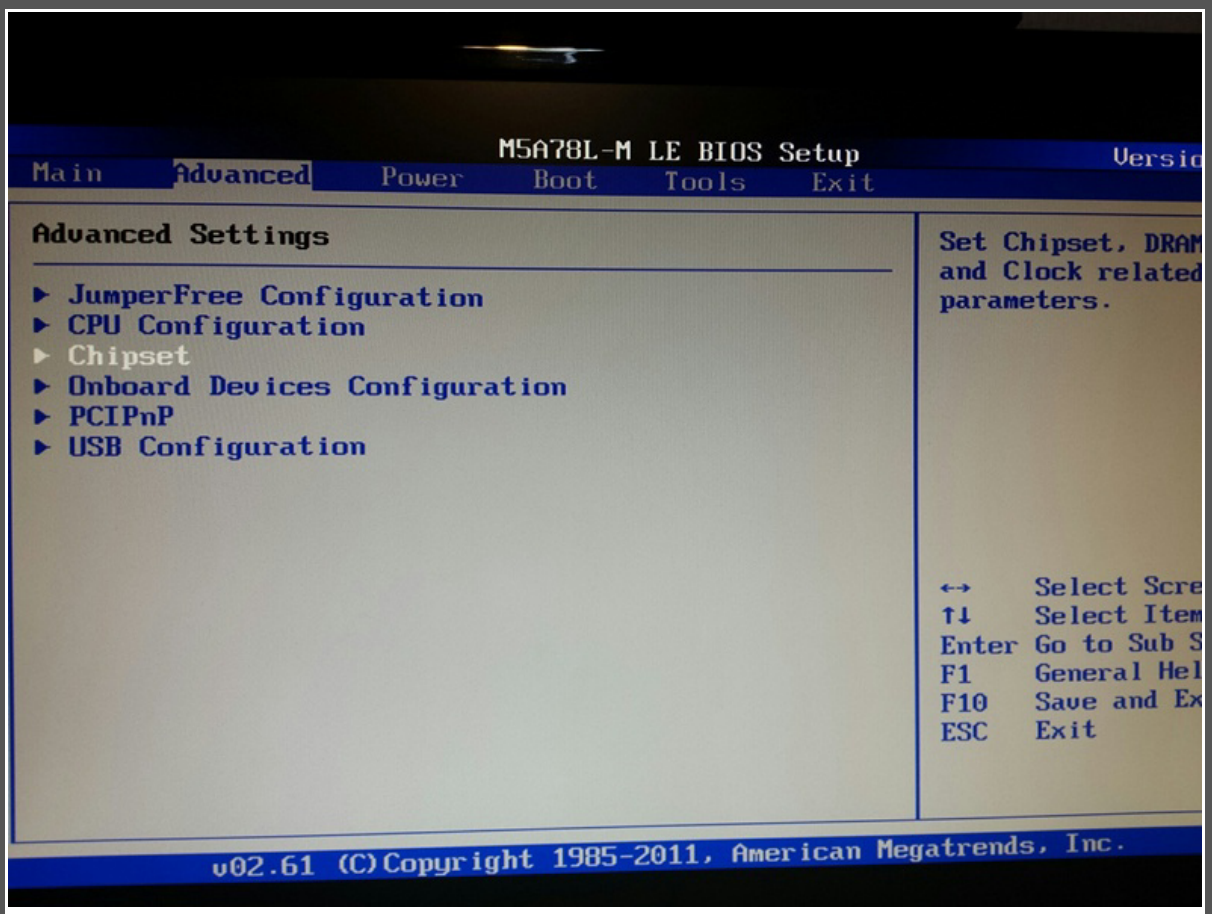
7> CUELGUES MISTERIOSOS EN EL PUENTE NORTE (NORTHBRIDGE)

Hace algunos meses, y dado que conseguí una licencia original de **Windows 7** (al comprar mi NETBook) que me permitió actualizarme a *Windows 10* sin coste alguno, decidí actualizar también el equipo de mi hijo, un *AMD ATHLON II x2 64bits* con 8 GB de RAM y una aceleradora gráfica *GEFORCE GT440 de 2 MB*, ya que la versión de **Windows 7** con la que me dieron el equipo no era original y como llevaba años instalada estaba dando ya algunos problemas. En resumen, que decidí actualizar el sistema operativo de este equipo e instalarle *Windows 10-64bits* desde cero.

Hasta ahí todo correcto, pero al cabo de poco tiempo comienzan a producirse una serie de bloqueos en el ordenador completamente aleatorios. Estos cuelgues, consistían en la congelación de la imagen y el bloqueo completo e irrecurable de todo el sistema, casi me cuesta la vida.

Tras desmontar todo el equipo y revisar escrupulosamente su refrigeración, comencé barajando posibles microcortes en la red eléctrica o caídas de tensión, alguna falla en la fuente de alimentación, la propia CPU, la memoria RAM, el disco duro y como no, la tarjeta gráfica. Realicé pruebas de todo tipo y, cuando parecía haber solucionado el problema, el muy condenado volvía a congelar la pantalla y a caerse. Además, me encontraba sin la posibilidad de probar los dispositivos por separado al no disponer de respuestos para ello. De manera que a nivel de hardware solo pude descartar la *RAM* al tener el sistema dos tarjetas de memoria de 4GB cada una y alternarlas para las pruebas.

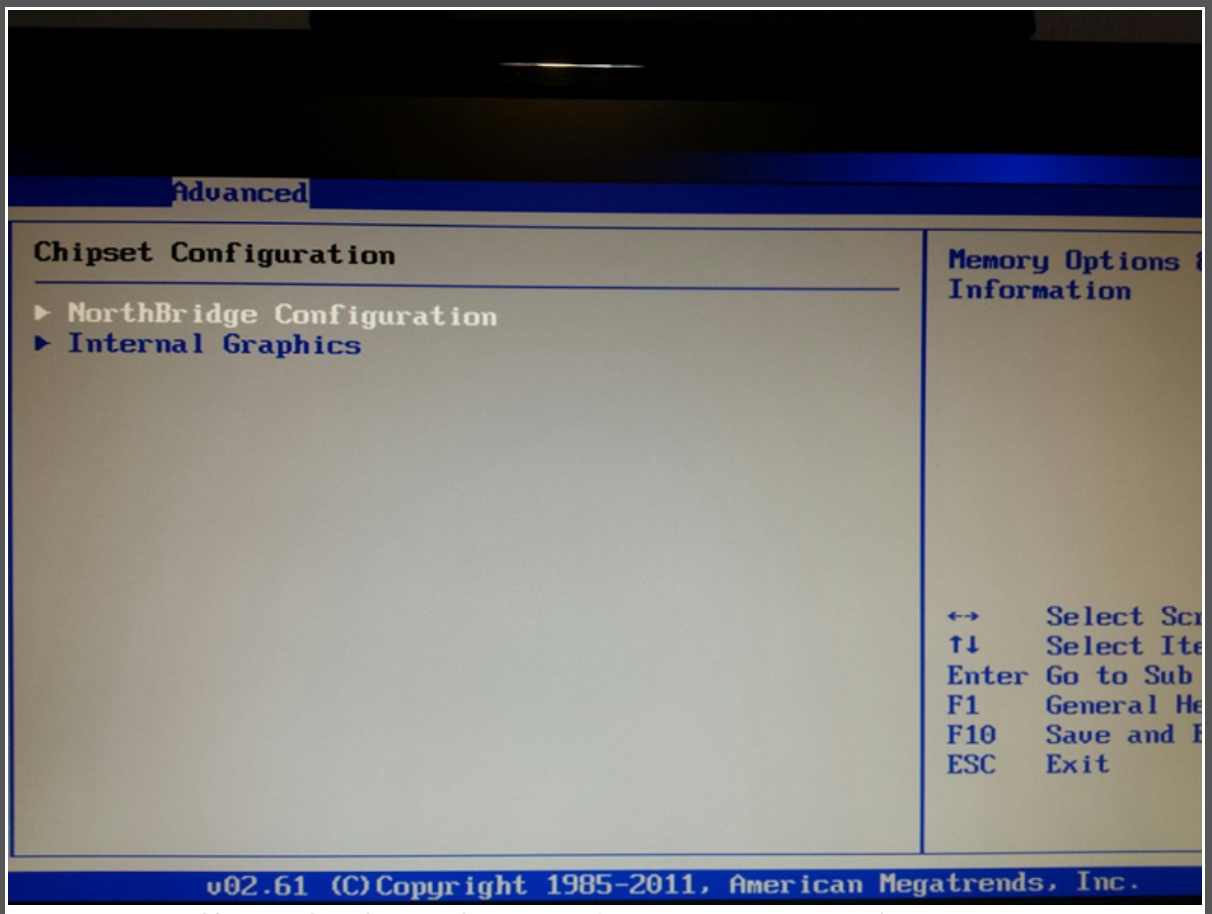
Lo peor de todo, es que los bloqueos eran muy aleatorios, igual aparecían al rato de iniciar el equipo como podían tardar horas e incluso días en manifestarse, pero llegaban.



En la BIOS no solo podemos realizar ajustes para mejorar el rendimiento de nuestro equipo, sino también controlar parámetros que aumenten la estabilidad del sistema evitando cuelgues fortuitos en nuestro ordenador.

En mis desesperados intentos por conseguir resolver los bloqueos misteriosos, recurrí sin éxito incluso al *DOWNCLOCKING* a través de la **BIOS**, es decir, reducir el rendimiento de la **CPU** y otros componentes del sistema.

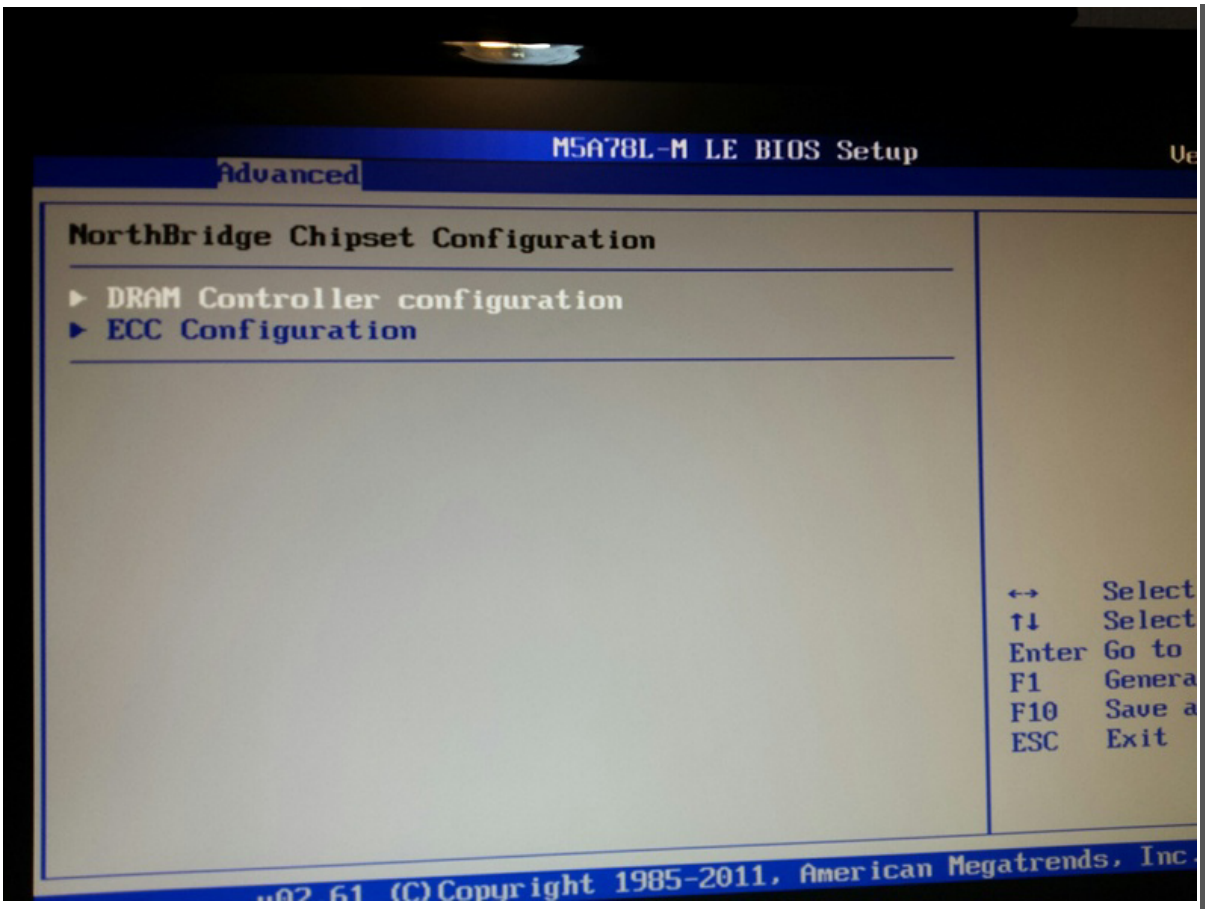
También sospeché y ataqué a gran parte del software instalado, sobre todo al antivirus que acabé desinstalando finalmente sin conseguir solucionar nada. Estaba frustrado y lo que era peor, barajando ya un desembolso de cuantía indefinida e inminente. Llevaba días con el problema sin conseguir nada y el ordenador es una herramienta básica de estudio de la que no puedes privar a un joven que cursa bachillerato de ciencias. Pero la última prueba, estaba a punto de dejar caer la manzana desde lo alto del árbol.



El chipset se divide en SOUTHBRIDGE y NORTHBRIDGE, siendo este último el responsable de las transfeerncias de alta velocidad entre el procesador, la RAM y el sistema gráfico.

Hablando con un colega del trabajo, éste se mostró convencido en apuntar a la gráfica (tarjeta) como la posible causante de los cuelgues misteriosos que a punto estaban de arrastrarme a la locura, y charlando, ambos caímos en la cuenta del sistema gráfico integrado en la placa. Hace años que la mayor parte de las placas base suelen integrar un sistema gráfico (bastante lento normalmente;) y que comparte parte de la memoria *RAM* como memoria de video.

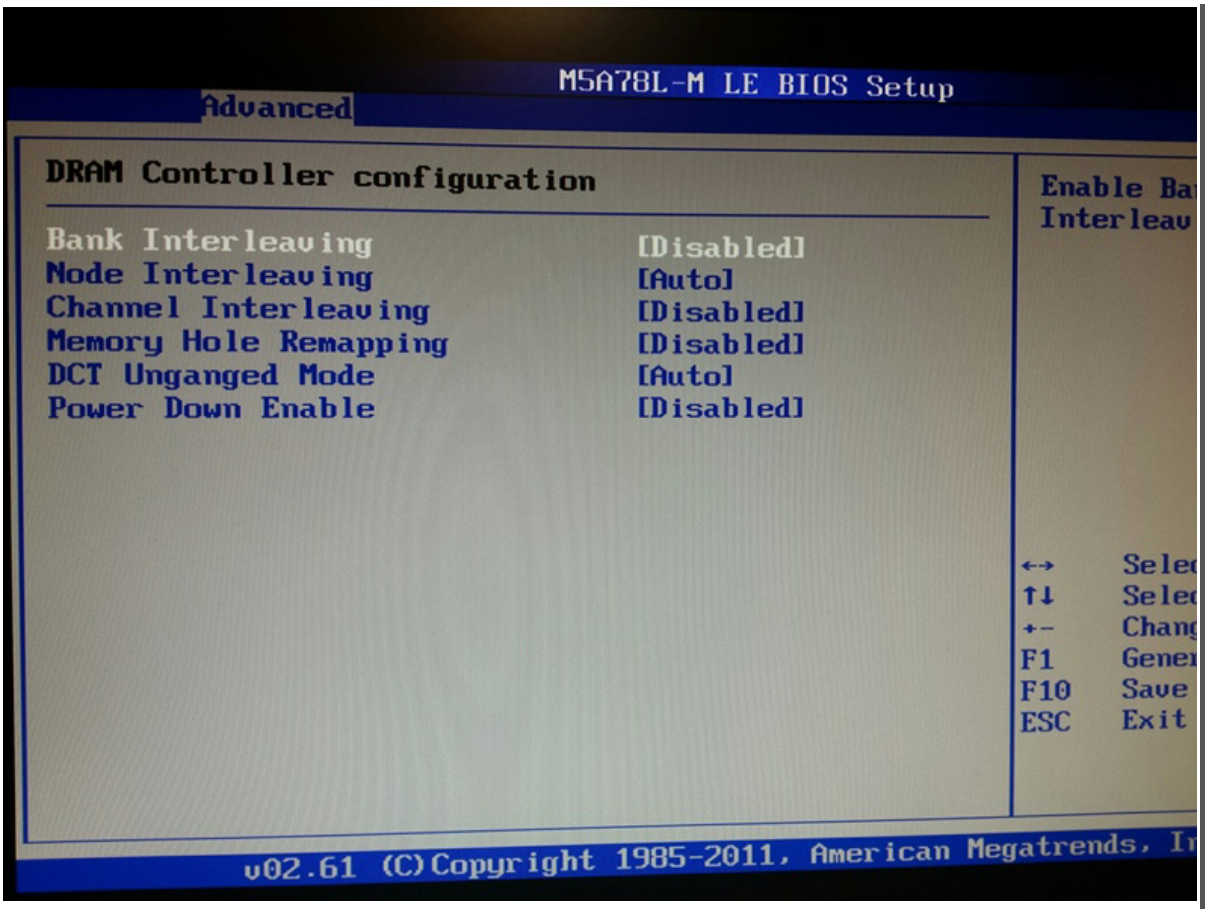
Con relativa esperanza, llegué a casa con la idea de desmontar la tarjeta gráfica *(GEFORCE GT440)* y arrancar el ordenador con la gráfica integrada en la placa base. Total, solo hay que desmontar la tarjeta gráfica del SLOT donde está insertada y cambiar el cable de señal de video al puerto (VGA/DVI/HDMI) integrado en la placa base. Hecho esto, arranco el sistema Windows 10 y tras algunos ajustes, se instalan los drivers y ya estaba listo para funcionar. Eso sí, con un rendimiento gráfico bastante pobre sobre todo para el uso de juegos.



Los ajustes finos desde la BIOS en el chipset no están precisamente muy documentados en el manual de la placa, y el ensayo prueba/error puede que acabe siendo nuestro mejor guía.

La prueba estaba en marcha. El sistema ya estaba funcionando con la gráfica integrada y yo aproveché para limpiar a fondo los disipadores de la aceleradora y engrasar los ventiladores. Ahora solo faltaba esperar a que se produjera un nuevo bloqueo y al cabo de un par de días, mientras mi hijo Rafa se encontraba trabajando, el equipo se congela y se cae, esta vez mostrando una imagen corrompida. Esto me produjo ya un fuerte "chispazo" en la cabeza.

Puede que continuara bloqueando, pero acababa de descartar otro componente como el causante de la falla: la tarjeta aceleradora gráfica, y en ese preciso momento mi foco de atención cambió hacia el auténtico corazón de la placa, el circuito integrado más importante del chipset, el ***NORTHBRIDGE*** ó ***PUENTE NORTE*** en español, por el lugar que ocupa en la parte superior de las placas de formato ATX.



Esta configuración del controlador de la memoria DRAM es la responsable de que el sistema lleve meses sin producir bloqueos.

La función principal del ***NORTHBRIDGE*** ó ***PUENTE NORTE*** es controlar el funcionamiento del bus del procesador, la memoria, el puerto AGP y el ***SOUTHBRIDGE***, haciendo de "puente" entre la placa madre y los citados componentes así como gestionando las transferencias entre ellos.

Finalmente, tras algunas pruebas y ajustes en el menú de la BIOS destinado a dicho elemento, conseguí superar por fin los **CUELGUES MISTERIOSOS EN EL PUENTE NORTE**.

Lo más curioso de toda esta historia, es que en mis muchos ajustes y pruebas (incluso había recurrido al *downclocking*) en busca de la solución al problema de los cuelgues y pese a cargar (en varias ocasiones) la configuración por defecto (DEFAULT) de todos los parámetros de la **BIOS,** ello no consiguó resolver los bloqueos.

8> EVOLUCIÓN DEL CHATARRERO GALÁCTICO

SINOPSIS:

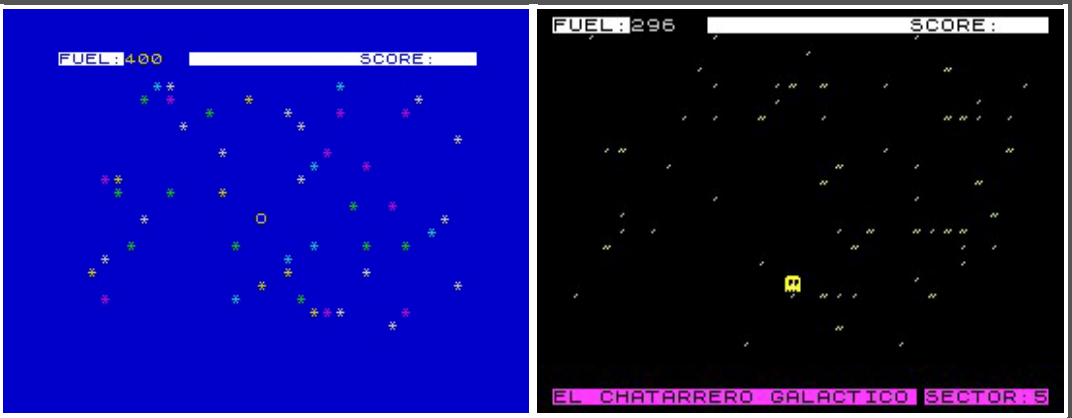
Debes guiar al antagonista de pacman, actualmente en delicada situación de desempleo, en el primer día de su nuevo trabajo.

Puede que recoger toda la basura espacial de cada sector de la galaxia no sea tan trepidante como otras aventuras del pasado, pero de ello depende tu futro y el de toda la RetroGalaxia ...

¡¡ SECTOR 4 INCLUYE RECOMPENSA !!

En el primer número de FINDE's Retro, invadido por la ilusión y la euforia de programar directamente sobre mi

reluciente ZX-Spectrum+3, decidí escribir el código que sirviera de base a un sencillo juego arcade. Este juego fue bautizado con el nombre del CHATARRERO GALÁCTICO y cuenta con el fantasma del clásico PACMAN como protagonista de una trepidante aventura de reciclaje espacial.



Capturas de pantallas del juego **EL CHATARRERO GALÁCTICO v0.1B** y 0.22B respectivamente.

En el primer FINDE's Retro prometí convertir a este juego cutre y simplón propio de revistas ochenteras en algo medio serio, y eso es lo que he intentado a lo largo de las sucesivas actualizaciones del mismo que, aún en fase beta, me han permitido llegar a una versión 0.37b plenamente jugable y acabada.

Video de partida ***al CHATARRERO GALÁCTICO v0.37B (nivel 4)***

Video de partida ***al CHATARRERO GALÁCTICO v0.37B (nivel 2 ¡¡SUPERADO!!)***

El juego EL CHATARRERO GALÁCTICO ejecutándose en un móvil con sistema Android con emulador.

El CHATARRERO GALÁCTICO está diseñado en su totalidad en lenguaje BASIC (posteriormente compilado a código máquina con MCODER 3), a excepción de una miúscula rutina de un par de líneas que modifican el juego de caracteres por defecto del ZX-Spectrum por otro tipo de letra bastante chula.

Las técnicas empleadas en el desarrollo de este juego tipo han sido implementadas en ZX-BASIC original, y son las más básicas que pueden emplearse en un juego, tales como:

- Desplazamiento en el eje X e Y del protagonista sin borrado de las estrellas de fondo
- Colisión y detección de objetos en pantalla
- Control y refresco de barra progresiva de FUEL y SCORE en tiempo real
- Interpretación de melodía continua con detección de teclas
- Diseño y uso sprites gráficos (8x8) definidos por el usuario (GDU's) dinámicos con las herramientas ZX-DRAW y/o GDUCalc
- Técnicas de optimización de código BASIC
- Compilación de código BASIC y traducción a código máquina

Finalmente, todo el código es compilado con MCODER3 para imprimir más agilidad y velocidad al juego, lo cual genera un bloque BYTEs en código máquina.

Si quieres el código BASIC puedes descargarlo más abajo impreso en formato BMP o solicitarlo a eurocamsuite@yahoo.es

Si bien se trata de un juego de apenas 10 KB de código, la mecánica del juego puede resultar incluso adictiva.

Como imagino que a algunos os gustará echar un vistazo al código fuente al menos por curiosidad, aquí tenéis el listado completo en un archivo de imagen BMP (impreso desde una ZX-Printer virtual), aunque también aprovecho para deciros que si sois capaces de **superar el nivel 4**, me comprometo a concluir y enviaros una **guía rápida de desarrollo para ZX-Spectrum** (que está casi lista;) explicando con todo detalle el código del juego y otras técnicas que pueden serviros para adentraros en la RETROESCENA! de esta milagrosa y entrañable computadora.

```
    1 REM **** EL CHATARRERO GALA
CTICO ****
    2 PAPER 0: BORDER 0: BRIGHT 0
: CLS : RANDOMIZE : GO SUB 1000:
 REM carga SPRITEs: A=▟ B=▛ C=▜
D=▙ E=▚ F=▞ G=▜ H=◢ P=▒ Q=▞ R=▒
S=¡
    3 GO TO 2000: REM FONTs- solo
 en modo 48K
    4 CLS : GO SUB 900: GO SUB 13
05: GO SUB 1400: REM PARA REINIC
IAR JUEGO GOTO 4
    5 PRINT #1; INK T+2; BRIGHT 0
; INVERSE 1;"EL CHATARRERO GALAC
TICO ";  BRIGHT 1;"SECTOR:";t
    6 INK 7: BRIGHT 1: FOR n=1 TO
 45: PRINT AT INT (RND*19)+1,INT
 (RND*29)+1;"´": NEXT n
    7 LET r=0: LET cb=t*5: BRIGHT
 0: REM contador de basura; nume
ro de basuras=nivel t*10
    8 FOR n=1 TO cb: LET x=RND*29
+1: LET y=RND*19+1: IF ATTR (y,x
```

'EL CHATARRERO GALÁCTICO' v0.37B - by F.NDE's retro ©© 2017

- Listado completo en archivo de imagen **BMP**
- VERSION 0.37B compilada en formato **.SNA** para correr en cualquier emulador

- **FORO** abierto en speccy.org para testar dificultad del Juego EL CHATARRERO GALACTICO

- VERSION EJECUTABLE DE EL CHATARRERO para correr directamente sobre **WINDOWS 32/64 bits**

[Ejecutar el archivo CHATARRERO.BAT]

TECLAS PARA MOVER AL CHATARRERO:

6 <
7 >
8 V
9 ^

¡¡Y recuerda que al superar el **Nivel 4** puedes solicitarme la **GUÍA RÁPIDA DE DESARROLLO**!!

9> CITAS DEL SABIO ... ó LAS PERLAS DE LA SAPIENCIA

A raíz de mi intercambio de mensajes con Rafa Fdez., todo un ingeniero informático en la primera línea del entramado corporativo y amante del retro donde los haya, descubrí que sus trepidantes historias estaban cargadas de vida y eran auténticas aventuras digitales que cualquier amante de la retrocomputación sabrá valorar sin duda. Por ello y en lo sucesivo, contando con la supuesta supervivencia de este humilde webzine digital, voy a publicar en esta sección (ya fija) una serie de citas anecdócticas vividas por mi buen amigo y colaborador y que no pueden quedar sin compartir.

En cada una de ellas descubrirás auténticos microrelatos escritos en primera persona con una narrativa fresca y sin tapujos (COPY-PASTE en estado puro) esperando que a los lectores les deje tan buen sabor de boca como a mi y sepan extraer la enorme sapiencia que de ellas se desprende.

FINDE'S retro
CITA DEL SABIO

*"Un tema interesante de la retrocomputación es su utilidad en nuestros días. El año pasado estuve en el **FOSDEM**, una convención de desarrolladores de software libre que se hace en la universidad de Bruselas. Una de las charlas a las que asistí fue la de "**Necrocomputación**", que es una variante de la **retrocomputación**. Se trata de coger un software de hoy en día e intentar ejecutarlo en una máquina antigua. El ejemplo que pusieron fue el intento de ejecutar en un VAX de finales de los 70 una versión moderna de la base de datos **PostgreSQL** (es la base de datos que usa yahoo, por ejemplo). Obviamente no arrancaba, así que tuvieron que modificar el código de **PostgreSQL** para optimizarlo y que corriera en esa máquina. El resultado es que las modificaciones se pudieron incorporar al código oficial de PostgreeSQL de tal manera que incrementó su rendimiento al ejecutarse en sistemas actuales. El tema es que una serie de variables estaban declaradas como "float", pero las máquinas VAX no manejaban números en coma flotante (¡no saben operar con decimales!). Descubrieron que esas variables se podían declarar como "integers" sin problema. Y eso es mágico, porque un ordenador tarda muchísimo menos en operar con integers que con floats. Así que la retrocomputación no es sólo un tema para nostálgicos, hoy en día tiene una utilidad muy interesante."*

FINDE'S retro
CITA DEL SABIO

*Es curioso lo del lenguaje **BASIC**, está muy denostado, sobre todo a raíz de un artículo que publicó Dijktra titulado "GOTO Statement Considered Harmful". Luego han salido mucho defensores, pero no tenían tanto renombre y no han tenido tanta repercusión como el primero. De hecho hay un ensayo buenísimo titulado "'GOTO Considered Harmful' Considered Harmful" que sale en defensa de **BASIC**. Yo aprendí a programar yo solo, usando la ayuda del QBasic que venía con el MS-DOS 6.20. Sólo con leer la ayuda y ver los ejemplos que traía me bastó para empezar a hacer cosas muy chulas ¿Cómo puede ser tan malo ese lenguaje? Si intentas hacer lo mismo con el lenguaje C no lo consigues en 10 años.*

FINDE'S retro
CITA DEL SABIO

Seguro que revienta todo!! Los sistemas estás cogidos con pinzas. La crisis ha hecho mucho daño al sector. No se busca hacer las cosas, se busca que sean muy baratas. Lo más baratas posibles y luego ya ir tirando como se pueda. Al final pasan cosas absurdas. En Correos y Telégrafos teníamos un software que centralizaba el envío de archivos entre todos los servidores y a los clientes. Por ahí pasaban muchas cosas: peticiones de envío, los justificantes que la gente firma y luego se escanean, los albaranes para las entregas, las facturas a los clientes,... Pues el sistema recibía los archivos con una cabecera y en función de la cabecera los mandaba a un sitio o a otro. De repente empiezan a fallar los envíos de algunas facturas sin motivo aparente, como las del Banco Santander. El motivo: la mierda del motor de clasificación que tenía dentro procesaba el nombre de Santander de la siguiente manera: ***SANT AND ER****. Los nombres no podían contener ni* ***AND*** *ni* ***OR*** *ni* ***XOR*** *ni* ***NOT****. ¡¡Soberana chapuza!! Y los campeones tardaron meses en arreglarlo. Total, que me saqué la BBDD de enrutamiento a un* ***.TXT*** *plano, hice un shell script de 20 líneas y con eso me enrutaba los archivos. ¡¡Iba más rápido que el original!! De hecho cuando arreglaron el problema, conservamos el script, porque cuando recibíamos muchos ficheros a la vez, el sistema se atascaba. Lo parábamos, arrancábamos el script y deshacíamos el atasco. Un software de más de 50 mil pavos (por cierto, dinero público) se podía sustituir con un script de 20 lineas.*

FINDE'S retro
CITA DEL SABIO

Esta semana he probado el Visual Studio 2017 y me puse a mirar qué novedades tenía. Visualmente muy bonito, al estilo de Windows 10, con colores planos y así. El tema funcional ya es otra cosa. El programa lo hice en C#, un lenguaje que hacía años que no tocaba y así lo podía refrescar. Bueno, pues para familiarizarme con el entorno creé un nuevo proyecto y le dí a compilar directamente. Así que allí me apareció la típica ventana de Windows vacía. El ejecutable pesa unos pocos kilobytes. Ahora bien, una aplicación vacía de VStudio 2017 ocupa en memoria la friolera de 14MB. Casi me caigo de culo. Eso sí, no tiene nada que ver el entorno de desarrollo con los antiguos VisualBasic y VisualC. Puedes ejecutar test de rendimiento y estrés sobre la aplicación para ver donde tienes los cuellos de botella en tu código, herramientas de análisis de calidad y temas así. Todo muy chulo, pero no me puedo creer que una aplicación vacía, para dibujar una ventana necesite 14MB de RAM. Lo cual me confirma ***la teoría de que Microsoft no usa Visual Studio para desarrollar sus propios programas****. Por ejemplo, si ejecutas la aplicación "winver" que es poco más que una ventana con etiquetas y un botón para cerrarla, ocupa un mega y medio (que por cierto, ya le vale también, que eso ocupaba el Flight Simulator 4 y hacía muchas más cosas;).*

Es que tengo grabadas a fuego unas palabras de mi profesor de programación de primero: hacemos ordenadores más rápidos para poder ejecutar más cosas en paralelo, no para que podamos programar peor. Si lo piensas, ¿Qué diferencia hay entre una ventana de Windows 3.11 y una de Windows 10 ó WordPerfect 6 con un Word 2016?. En mi opinión no justifica tanto consumo de RAM.

Y hasta aquí este ARCHIVE #02. Ha sido un placer compartir contigo todo este FINDEs RETRO y, si el ánimo nos ayuda y la inspiración nos alcanza, te espero en el próximo :

ARCHIVE #03 de FINDE's retro

eurocamsuite@yahoo.es

Agradeciendo mucho que me hagas llegar tu opinión acerca de este magazine ya que será el único feedback que tenga con los lectores;)

... TALLER DE LINK'S / DOWNLOAD / TÉLÉCHARGER

FINDE'S retro

LINK's a recursos comentados en este magazine y otros temas relacionados

Enlaces de interés en este número ...

- VERSION EJECUTABLE de EL CHATARRERO GALÁCTICO para correr directamente sobre **WINDOWS 32/64 bits.**
Mediante DOSBOX y el emulador de ZX-Spectrum BACTERIA para MS-DOS, he preparado una versión de nuestro juego ELCHATARRERO listo para correr sobre sistemas WINDOWS de 32 y 64 bits. El juego se ejecuta a pantalla completa, pero puedes alternar al modo ventana mediante la combinación de telcasl ALT + ENTER.
Para lanzar el juego una vez descomprimido el .ZIP solo debes ejecutar el archivo **CHATARRERO.BAT**

- Utilidad Windows freeware para crear memorias USB de arranque con cualquier sistema operativo:
UNETBOOTin

- Utilidad freeware **LOOKIT** para MS-DOS/FreeDOS.
Se incluye DOSBOX en el mismo directorio para poder ejecutar LOOKIT desde Windows con solo arrastrar el archivo ejecutable LOOKIT y soltar sobre el ejecutable DOSBOX.
DOSBOX se ejecuta en modo ventana pero puedes alternar al modo pantalla completa mediante la combinación de teclas ALT + ENTER.
LOOKIT.zip

- Impresionante proyecto que permite emular los juegos originales de ZX-Spectrum a 256 colores.
Para ejecutar desde Windows lance el archivo BAT
SPECTRUM-256.BAT
El emulador SPEC256 está compilado para MS-DOS, por lo que se incluye DOSBOX en el mismo directorio para poder ejecutar desde Windows. DOSBOX se ejecuta en modo ventana pero puedes alternar al modo pantalla completa mediante la combinación de teclas ALT + ENTER.
spec256.zip
(pulsa F7 desde el emulador para cargar ROMs en 256 colores)

Algunos proyectos propios en desarrollo ...

- Utilidad cruzada de dibujo para ZX-Spectrum (**website de ZX-Draw**)

- Utilidad cruzada de diseño y gestión de GDU's ZX-Spectrum (**website de GDUcalc**)

- PROGRAMA DE VARIACIONES SIN REPETICIÓN (completo) (generador de todas las combinaciones de la lotería primitiva)

- Web del programa de azar SIMULA (completo)
(software de simulación aleatoria para lotería primitiva)

- Web del proyecto GUTENBERG 3.0
(iniciativa propia para la preservación documental de la bibliografía técnica de retrocomputing)

- Portal de Inteligencia Artificial
(website sobre esta temática)

- Colección ALBACO (en curso...)
(Algoritmos Básicos Computacionales)

Números disponibles de FINDE's retro:

ARCHIVE #01 (junio 2016)

ARCHIVE #02 (noviembre 2017)

Proyecto iniciado el 21 de mayo de 2017 y alojado en **www.calentamientoglobalacelerado.net**

By Rafael Lomena :: MaRaF SOFT ©© 2017
eurocamsuite@yahoo.es - eurocamsuite@gmail.com