



**Universidad Nacional Autónoma de México**



**Facultad de Ingeniería**

**Sistemas Operativos**

**Reseña Fascículo 21**

**Integrantes del equipo:**

- **Garcia Riba Emilio**

**Semestre 2026-1**

La revista The Home Computer Course o MiComputer era una revista dedicada a todo lo que competía a la computación en los 80 's. Aun considerando que la información ya no está al día y las limitaciones de la época opino que vale la pena leer una de sus revistas. Sin contar el valor histórico (el uso del *he* sobre el *they* para hablar de un tercero neutro me saltó el ojo) creo que son informativas de la época, en el tomo 24 vi pestaños o inicios de cosas que hoy en día tenemos. Como las LLM's, No Code apps, música digital, gráficos, laptops, etc. En esta reseña hablaré de dos de los artículos: una reseña de una computadora y la corrección de textos con ayuda de las computadoras.

### Osborne-1

Este artículo habla de la computadora Osborne-1. En términos generales fue una computadora portátil exitosa gracias a su hardware y software que siguió vendiendo hasta después de que la compañía que la creó entró en bancarrota. El costo-beneficio de la máquina era muy bueno. Hablando ya en específicos, el hardware (ordenado por lo más impactante en mi opinión) se puede caracterizar por:

- Su portabilidad: Considero que es su mayor ventaja competitiva en el momento. Para su época era una maquina portatil, pesando apenas 10.5 kg. A excepción de una batería, haciendo que se tuviera que conectar a un enchufe, tenía todo lo necesario para funcionar.
- Calidad: Con sus specs se defiende para la época. Ahora cuando vemos su precio, esto es muy buenos specs.

En el software está más interesante para lo relacionado a la materia:

- Software de la época: Viene incluido una paquetería de aplicaciones para uso de oficinistas y gracias a su OS una gran gama de programas que se pueden correr en él.
- OS/Monitor: El OS (por parte del programa de control/controlador) es independiente a su hardware, no en el sentido que corra sin él sino en que es agnóstico. Esto hace que no importa el hardware que tenga la computadora el OS podrá correr en este. Con esta abstracción se da el mismo beneficio hacia las aplicaciones. En el sentido económico deja que el mercado de aplicaciones disponibles para la 'laptop' sea mayor, haciendo un ecosistema más rico, más barato y de mejor calidad. En términos del desarrollo a nosotros nos permite no preocuparnos (por la parte de hacerlo para él, no los recursos) del hardware, más recursos se podrán utilizar para la aplicación para si no la adaptación hacia el hardware. Por último esto ayuda el hardware ya que puede ser cambiado para mejoras o actualizaciones sin preocupación del software.

Este mismo programa de control/controlador también hace rutinas normales hoy en día, en específico el manejo de archivos. Para la época, asumo que era nuevo/innovador ya que por su mención de esto.

Con todo esto fue exitosa la computadora, siendo la primera laptop exitosa. Hoy hay miles de laptops en el mercado con mejores recursos, software, hardware, etc. pero no creo que el costo-beneficio de la Osborne-1 sea algo que tal vez le siga ganando las laptops actuales.

## **Magic Spell**

En el artículo habla sobre la edición de cadenas basado y la corrección de textos con ayuda de un software. Por lo que va sobre la corrección, la forma más sencilla es corregir los errores de palabras escritas mal. Entonces que el software checador pueda detectar que 'halcolol' está mal escrito, avisar y poder corregir a 'alcohol' con aprobación del usuario. Es un problema de memoria (tener las palabras escritas correctamente) y recursos (poder buscar la palabra candidata) para el uso de un diccionario. La parte más difícil es el significado. Todo texto da un significado que no una computadora no podrá entender (hasta el día de hoy en mi opinión) entonces no puede corregir el significado si está mal o tratar de traducirlo. Decir 'nada de hamburguesa' para 'nothing burger' esta mal, sería mejor 'nadota', una computadora se le complica hacer eso.

Ya en lo apropiado para OS quiero hablar de los diccionarios. Expandir en sus complicaciones y las soluciones usadas en la revista:

- Memoria: Hay [25k - 50k] incluidas en la memoria.
  - Optimización: Sólo se usan los caracteres ASCII usados para escribir palabras y minúsculas. Con esto cada caracter necesita 5 bits de info a diferencia de 8 que seria con todo los caracteres.
  - Tokenizing: Ciertas combinaciones de caracteres naturalmente aparecerán más que otros. Aprovechando esto podemos guardarlos en un espacio de memoria compartido, e.i: , [auto], [auto]movil, [auto]matizar, etc. podemos tokenizar auto a un espacio de memoria y un pointer haciendo que se convierta en: [\*], [\*]movil, [\*]matizar. Salvamos espacio al no tener que repetirnos.
- Recursos:
  - Búsqueda: Comparar cada palabra con nuestro diccionario sería muy costoso, aunque estén escritas bien. En tiempo real todavía está lejos de ser una realidad entonces se hace con el documento completo. Luego en un escrito muchas palabras se repiten (hasta ahora van 378 palabras únicas en este documento con 848 palabras en total, un  $378/848 = 43.63\%$  de unicidad) entonces solo checamos las palabras una vez no cada vez que aparecen. Reducimos la cantidad de veces que tenemos que revisar un error común a una vez.

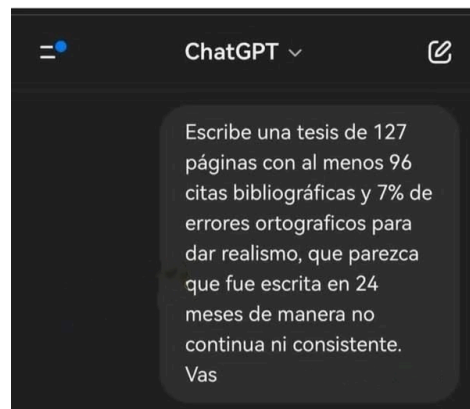
Cada vez que se detecta un error se hace lo siguiente. Estarán en un estado de indeterminado para el programa, podrán estar bien, mal o no ser una palabra (e.i.

hcnjlisa). Se deja en manos del usuario. Puede agregarlo al diccionario como una palabra nueva, ignorarlo o corregirlo. Es una pregunta primero de contexto (bien/mal) y después de memoria (la palabra volverá a aparecer haciendo que valga la pena agregarla al diccionario o ignorarla). En el software que uso para tomar notas, obsidian con la extensión 'LanguageTool Integration', te da las misma opciones aunque ignorar viene implícito al no decidir

well. Ah yes, what every writer would adore:  
'COMMAND > GENERATE ARTICLE, LENGTH 1200  
WORDS, BEGIN'

1

Imagen 1: Algo que el autor quisiera que fuese posible.



2

Imagen 2: Algo que una persona actual quisiese que fuese posible,  
Nada nuevo bajo el sol.

## Conclusion

Con las reseñas y comentarios hechos en este documento podemos leer el valor histórico, entender de dónde vienen y cómo se hicieron ciertas cosas. Para la materia de OS hubo varios temas de donde analizar. Me enfoqué en dos artículos hablando del OS en específico, la memoria, etc. pero hay más. Además está bien hecha la revista, sus escritos y diseño, haciendo placentera leerla. En general la recomiendo.

## Fuentes:

- Orbis Publishing. (1983). *The Home Computer Course* (21) [Magazine].
- Coinbench. (2025, May 13). *ChatGPT hazme una tesis con errores para que parezca real y con 96 citas. Gracias bro.* Memedroid.  
<https://es.memedroid.com/memes/detail/4514261/ChatGPT-hazme-una-tesis-con-errores-para-que-parezca-real-y-con-96-citas-Gracias-bro>

<sup>1</sup> (Orbis Publishing, 1983, p. 405)

<sup>2</sup> Coinbench. (2025, May 13). *ChatGPT hazme una tesis con errores para que parezca real y con 96 citas. Gracias bro.* Memedroid.  
<https://es.memedroid.com/memes/detail/4514261/ChatGPT-hazme-una-tesis-con-errores-para-que-parezca-real-y-con-96-citas-Gracias-bro>