

# ¿Qué papel tenían los agentes humanos en los inicios de la microcomputación?

La MS51 desde adentro

José G. Morales Pensso y Roberto M. Rivas Jordán

25-03-2022

## Contents

<b>1</b>	<b>Introducción</b>	<b>2</b>
<b>2</b>	<b>Objeto de estudio</b>	<b>3</b>
<b>3</b>	<b>Trabajo de campo</b>	<b>4</b>
3.1	El sistema operativo CP/M . . . . .	4
3.2	Primera visita al Laboratorio de Electrónica . . . . .	7
3.3	Segunda visita . . . . .	9
<b>4</b>	<b>El ensamble Hardware-Software-agente humano</b>	<b>11</b>
4.1	¿Por qué hablamos de <i>ensambles</i> ? . . . . .	11
4.2	Las relaciones entre hardware-software hoy . . . . .	12
4.3	Las relaciones entre hardware y software en el con- texto de la MS51 . . . . .	15
4.4	Operar con una MS51 . . . . .	18
<b>5</b>	<b>Reflexiones finales</b>	<b>20</b>
<b>6</b>	<b>Bibliografía</b>	<b>22</b>

## Abstract

Las personas nos relacionamos de diferentes maneras con las tecnologías computacionales. El estudio de las relaciones entre los componentes de hardware y software de una computadora y los agentes humanos en los inicios de la microcomputación permite elucidar la

emergencia de tres figuras o estereotipos sociales, las cuales se distinguen en virtud del grado de interacción que entablan con dichos componentes: la figura del usuario, la del hacker y la del operador. Esta última fue gradualmente desplazada por la figura del usuario cuando la microcomputación transitaba, durante la década de 1980, un proceso de fuerte expansión. Para analizar el *ensamble* de los componentes de hardware y software de una computadora con los agentes humanos llevamos a cabo un trabajo de campo en el Laboratorio de Electrónica de la FAMAF (UNC). Allí interactuamos con una MS51 construida en el año 1982 por la empresa -fundada y radicada en la ciudad de Córdoba (Argentina)- Micro Sistemas S.A.

## 1 Introducción

En este trabajo<sup>1</sup> analizamos las relaciones que entablan dos conjuntos de componentes constitutivos de una computadora: hardware y software. Las relaciones entre hardware y software no permanecen inertes; por el contrario, exhiben un gran dinamismo puesto que son altamente dependientes de momentos y circunstancias históricas y del rol que juegan los agentes humanos en el marco de dichas relaciones. Consideramos que la reunión de los componentes de hardware y software con los agentes humanos, componen un *ensamble* en el sentido que (Harris 2017) da a este término.

Para llevar a cabo el análisis de este ensamble hemos realizado un estudio de caso. Éste se remonta a estadios tempranos de la cultura digital, cuando la computación transgredía el límite de las universidades y grandes centros de investigación e irrumpía en los hogares de clase media en virtud de la rápida expansión, entre los años 1974 y 1982, de lo que pronto se transformaría en un nuevo paradigma tecnológico, a saber: la microcomputación. Se trata, por tanto, de un estudio de caso situado en un pasado reciente. Atendemos al tipo de relación que entablaban los agentes humanos con la computadora. En ese entonces el término *operador* empezaba a ser sustituido por el término *usuario*. Este último término, a diferencia del primero, supone la figura de un agente humano que adopta una condición de receptor pasivo frente a las

---

<sup>1</sup>Queremos agradecer a Nicolás Wolovick y a Ezequiel Chesini por abrirnos las puertas del Laboratorio de Electrónica de la FAMAF. No sólo eso. En muchos casos nos guiaron, instruyeron y apoyaron entusiastamente para sostener lo que inicialmente fue una vaga idea. También quisiéramos extender el agradecimiento al profesor Andrés Laguens (Instituto de Antropología de Córdoba -CONICET/UNC-, Museo de Antropología -FFyH, UNC), quien nos instruyó en la perspectiva de los denominados Nuevos Materialismos en Arqueología y nos dio el envío necesario para que adoptemos dicha perspectiva en el estudio de las relaciones entre hardware-software y agentes humanos.

tecnologías computacionales.

Para adentrarnos en el período señalado y analizar en su contexto las relaciones entre hardware, software y agentes humanos, realizamos lo que autores como Jussi Parikka y Erkki Huhtamo denominan "una arqueología de medios". (Cf. Huhtamo y Parikka, 2011) Los estudios de arqueología medial se inscriben dentro de los Nuevos Materialismos. Estudian fundamentalmente los medios que han estructurado nuestras sociedades desde de la Revolución Industrial hasta la actual cultura digital cuya irreducible base material ha sido ignorada por algunos enfoques filosóficos.<sup>2</sup> En este marco hemos realizado un trabajo de campo. Éste consistió en interactuar con una computadora MS51 construida en el año 1982, la cual actualmente se encuentra en el Laboratorio de Electrónica de la Facultad de Matemática, Astronomía, Física y Computación (FAMAF) de la Universidad Nacional de Córdoba (UNC). Las MS51 fueron producidas por la empresa Micro Sistemas S.A. Se trata de una empresa nacida en la ciudad de Córdoba (Argentina) a mediados de la década de 1970. Las MS51 utilizaban el sistema operativo (en adelante, SO) CP/M, creado por Gary Kildall y cuya primera versión fue lanzada en el año 1974 para microprocesadores Intel 8080. Una instancia muy significativa previa a nuestro trabajo de campo consistió, entonces, en explorar el SO CP/M.

## 2 Objeto de estudio

La idea de este trabajo surgió cuando supimos de la existencia del SO CP/M. Se trata de un SO para microcomputadoras creado en el año 1974 por Gary Kildall para los que entonces fueron los nuevos microprocesadores Intel 8080. CP/M fue la norma *de hecho* en el ámbito de la microcomputación durante la década de 1970 y parte de la década siguiente. Tras una negociación frustrada de Gary Kildall con IBM, el lugar de centralidad del SO CP/M lo terminaría ocupando MS-DOS, un SO que fue adquirido por la empresa Microsoft y que está basado en CP/M. MS-DOS evolucionaría, en la década de 1990, al SO Windows con interfaz gráfica -o GUI, por las siglas en inglés para Graphical User Interface. Con el SO Windows se aspiraba a que cualquier persona con escasos o nulos conocimientos de computación pudiera utilizar con cierta fluidez una computadora personal.

Una lista disponible en Wikipedia contiene más de doscientos nombres de

---

<sup>2</sup>La filosofía CyberPunk, desde la década de 1980, ha contribuido fuertemente a invisibilizar los materiales que soportan nuestras experiencias virtuales. (Cf. Parikka 2015, p. 57)

artefactos que utilizaban el SO CP/M, lo que permite constatar el desarrollo notable alcanzado por la microcomputación entre los años 1974-1984 y la centralidad de CP/M. En virtud de una nota de Lucas Viano, publicada en La Voz del Interior el día 12/10/2021, nos enteramos de la existencia de la empresa de informática Micro Sistemas S.A., una empresa nacida en la ciudad de Córdoba (Argentina) que a mediados de la década de 1970 comenzó a fabricar y comercializar computadoras personales. La referida nota de La Voz Interior destaca que las computadoras de Micro Sistemas S.A. utilizaban el SO CP/M. También incluye una entrevista a Nicolás Wolovick, un referente en el campo de la retro-computación que viene trabajando desde hace años en la recuperación y restauración de las computadoras de Micro Sistemas S.A. De inmediato nos contactamos con él para consultarle si ya había arrancado alguna computadora de Micro Sistemas utilizando CP/M. Respondió que en el Laboratorio de Electrónica de la FAMAf se encuentran tres computadoras MS51 que utilizan CP/M. Aclaró luego que todavía no había emprendido la tarea de poner en marcha esos equipos, invitándonos en ese mismo intercambio a hacerlo. El trabajo de campo que realizamos consistió precisamente en lograr poner en marcha, utilizando el SO CP/M, una de las tres computadoras MS51 que actualmente se encuentran en el Laboratorio de Electrónica.

### 3 Trabajo de campo

#### 3.1 El sistema operativo CP/M

Una tarea que realizamos antes de comenzar con nuestro trabajo en el Laboratorio de Electrónica, fue descargar manuales de distintas versiones de CP/M y algunas de las imágenes que están disponibles en línea de dicho SO para diferentes arquitecturas. Seguidamente, utilizando el programa QEMU, ejecutamos con nuestras computadoras las imágenes descargadas de CP/M desde una *máquina virtual*. Esto nos permitió tener un período de experimentación con el SO CP/M previo a nuestra primera visita al Laboratorio de Electrónica.

Otra importante tarea que realizamos consistió en crear disquetes bootables de 5.25 pulgadas de doble densidad con imágenes de CP/M para microprocesadores Zilog 80 (Z80) de 8 bits. Este es el microprocesador que utilizan las MS51. Las tecnologías para el almacenamiento de información evolucionaron lo suficiente durante estas últimas décadas como para volver realmente difícil conseguir un disquete de 5.25 en buen estado. Los disquetes de doble y alta densidad de 5.25 fueron reemplazados, primero,

durante la primera mitad de la década de 1990, por los disquetes de 3.5 pulgadas; poco más tarde, por los CD-ROM y los pendrives; estas últimas tecnologías de almacenamiento, por cierto, todavía circulan. En sitios como Mercado Libre se ofrecen cajas de disquetes en buen estado a valores muy elevados. También se ofrecen disquetes que no funcionan. A éstos por lo general se los emplean para hacer artesanías.

La experiencia con CP/M -ejecutado desde una máquina virtual- en nuestro caso fue amigable. Todos los procesos se ejecutan a través de línea de comandos; y la mayor parte de los comandos se parecen a los de MS-DOS, el SO estándar de las computadoras tipo PC, que a partir de la década de 1990 ingresaron masivamente a nuestro país. Quienes han utilizado MS-DOS, o quienes actualmente utilicen la consola de algún SO de la familia GNU/Linux, seguramente no experimentarán mayores problemas al utilizar CP/M. No obstante, para un *usuario* -término que iremos elucidando en secciones subsiguientes- interactuar con una computadora sin interfaz gráfica puede resultar una experiencia completamente novedosa, por veces frustrante.

Los comandos que empleamos para manipular archivos se denominan, desde entonces a esta parte, *built-in commands*. Éstos se ubican en un sector específico del SO. CP/M está lógicamente dividido en cuatro sectores diferentes. Los *built-in commands* se encuentran en el sector denominado CCP (Console Command Processor). Listamos a continuación los *built-in commands* del SO CP/M:

ERA (borra archivos)

DIR (lista los nombres de los archivos en el directorio)

REN (renombrar archivos)

SAVE (guarda en contenido de la memoria en un archivo)

TYPE (imprime en pantalla el contenido de un archivo)

USER (permite trabajar con archivos, en forma separada, en el mismo directorio).

El comando para copiar archivos en CP/M presenta una particularidad. El nombre del comando es PIP (Peripheral Interchange Program) y el programa se encuentra situado en un sector lógico del SO llamado TPA (Transient Program Area), de manera que PIP es un programa alojado en un sector lógico diferente al de los *built-in commands*. En MS-DOS el comando

para copiar archivos sí forma parte de los *built-in command*. Por otra parte, tanto el nombre del comando como su sintaxis, a diferencia de lo que sucede con el comando PIP de CP/M, resultan fácilmente accesibles a la cognición humana. Veamos este punto con más detalle.

En MS-DOS el comando para copiar archivos se denomina `COPY`. En los SO basados en UNIX -como es el caso de los SO de GNU/Linux- el comando para copiar archivos se denomina `CP`. En ambos casos, los nombres del comando remiten a la acción de copiar. Asimismo, las sintaxis de los comandos `COPY` y `CP` ubican a la izquierda el nombre del archivo o conjunto de archivos que se quiere copiar, y a la derecha el/los nombre/s del/los archivo/s destino. Por el contrario, para realizar la acción de copiar uno o más archivos con el comando PIP, se debe indicar primeramente el destino y luego el o los archivos que queremos copiar interponiendo el signo "=" . Notemos que a una persona que domina algún lenguaje de programación la sintaxis del comando PIP le resultará familiar y tan accesible como la del comando `COPY` o `CP`.

En el mismo sector lógico de PIP se encuentra el editor de textos ED, desarrollado en 1969 para el SO UNIX.<sup>3</sup> Nuestra experiencia cotidiana con editores de texto se reduce a tres acciones:

1. abrir el editor (NotePad en el caso de Windows, mcedit, Kate, xed, etc. en el caso de GNU/Linux);
2. tipear el texto que deseamos escribir;
3. guardar el contenido en un archivo.

Ahora bien, solamente daremos el primero de estos tres pasos si queremos trabajar con el editor de texto ED. Así, luego de ejecutar el comando ED para abrir el editor -paso 1-, veremos que el programa queda a la espera de que introduzcamos algún comando que permita realizar alguna acción, por caso, la de insertar texto -para lo cual se emplea el comando *i*. La tercera edición del manual de CP/M 2.2, publicado en septiembre de 1983, dedica largas páginas a la descripción de los comandos de ED. Una de las mayores dificultades que enfrentamos al trabajar con ED es que, con excepción de la línea sobre la que estamos escribiendo, el texto no se va imprimiendo sobre la pantalla. Por supuesto que existen comandos para guardar el contenido y para volver sobre ese contenido a fin de insertar modificaciones, reemplazos, etc. Esta dificultad que experimentamos al utilizar ED guarda

---

<sup>3</sup>UNIX es el SO sobre el cual están basados los SO GNU/Linux.

relación directa con las limitaciones de la memoria disponible en el hardware de la época. En la descripción de ED que ofrece el manual de CP/M 2.2, encontramos varias secciones dedicadas al uso que hace de la memoria este editor de texto. Por ejemplo, una de las secciones se denomina *Memory Buffer Organization*; otra *Memory Buffer Operation*. También se exhiben diversos gráficos que ilustran la relación entre los dispositivos de entrada y salida (I/O), el sector de memoria utilizado y el sistema de archivos. Este último punto también presenta una particularidad. Para optimizar la memoria disponible, cuando editamos un texto con ED se generan dos archivos: un archivo denominado "fuente", y otro denominado "temporario".<sup>4</sup> ED opera sobre el archivo fuente y va pasando el texto a un sector de la memoria (buffer memory) donde puede verse y modificarse. El archivo temporario es, entonces, el archivo sobre el que vamos editando texto.

Finalmente, destacamos que CP/M 2.2, al igual que la primera versión de MS-DOS, no posee un sistema de archivos jerárquico que permita anidar archivos -en una estructura de directorios y subdirectorios. Más que a una imposibilidad técnica, la ausencia de un sistema de directorios probablemente se debió a que éste no era realmente necesario: el volumen de archivos que se manejaba en ese tiempo era todavía muy bajo. No obstante, CP/M 2.2 dispone del comando USER -el último *built-in command* de la lista que presentamos arriba. El usuario por defecto, una vez cargado en SO, es el usuario cero o USER 0. Este se ubica en el directorio principal -lo cual podemos constatar si ejecutamos el comando DIR. Para utilizar un área lógica diferente en el mismo directorio, disponemos de un número de 16 usuarios -USER 0-15. Esto es vagamente equivalente a contar con un directorio con dieciséis carpetas para organizar los archivos. Por ejemplo, si ejecutamos el comando USER 1, estaremos en el mismo directorio pero en un sector lógico diferente al del USER 0.

### 3.2 Primera visita al Laboratorio de Electrónica

Realizamos dos visitas al Laboratorio de Electrónica de la FAMAFA, localizado en Ciudad Universitaria en el mismo edificio de la facultad. Debido a las restricciones de acceso por la pandemia, sumado a que no somos docentes o estudiantes de la facultad, gestionamos el acceso enviando una solicitud al vice rector. Obtuvimos permiso para ingresar durante el mes de noviembre.

La primera visita fue el día 09/11/2021. Ingresamos al edificio alrededor

---

<sup>4</sup>Un archivo de CP/M se divide en *nombre* y *extensión*. Ésta última posee tres caracteres. MS-DOS adoptará esta convención. ED asigna arbitrariamente la extensión \$\$\$ al archivo temporario.

de las 10 am. En el laboratorio no se encontraba Nicolás Wolovick. A través de él establecimos contacto, previo a nuestro arribo al Laboratorio, con Ezequiel Chesini, quien nos recibió esa mañana. Sin el acompañamiento de Ezequiel poco habríamos avanzado. Cuenta con profundos conocimientos sobre computación, en particular, sobre programación de bajo nivel. Realiza, junto con Nicolás, actividades de retro-computación, de manera que estaba completamente informado del tipo de actividad que nos propusimos realizar con Roberto: bootear una MS51 con una imagen del SO CP/M 2.2 empleando un disquete de doble densidad de 5.25 pulgadas. Para esta primera visita Roberto no había generado el disquete. Consideraba que no tenía ningún sentido hacerlo si no conocíamos los aspectos técnicos de la disquetera que tenía la MS51 con la que íbamos a trabajar.

En el laboratorio hay tres ejemplares de MS51. Cada una de ellas posee dos disqueteras. La doble disquetera podía resultar ventajoso en algún sentido que desconocemos; quizás facilitaba la copia de información de un disquete a otro. Vale aclarar que los ejemplares de MS51 del laboratorio no poseen disco duro.<sup>5</sup> Uno de los ejemplares tiene disqueteras SA400, las disqueteras de las dos restantes son SA455. Está disponible en línea los documentos en PDF con los manuales de estas disqueteras.

Ezequiel nos guió hasta el pequeño cuarto donde se encontraban las MS51. Trasladamos los tres artefactos a una habitación más grande para poder trabajar cómodos. Empezamos retirándoles la coraza. Con Roberto advertimos que los tres artefactos no eran idénticos. Más allá del aspecto exterior, hubo un aspecto técnico que las diferenciaba de manera significativa y gracias al cual pudimos formular un criterio para decidir con cuál de los tres ejemplares empezar a trabajar. Sólo una de ellas, nos hizo notar Ezequiel, tenía una especie de "placa madre". Las otras dos no nucleaban en una sola placa todos los componentes esenciales de una computadora como el microprocesador, la memoria RAM y memoria ROM. Ezequiel comentó que las computadoras de Micro Sistemas S.A. se construían de acuerdo con estándares no muy estrictos. El proceso de fabricación era cuasi-artesanal, y la empresa disponía de escasos recursos para responder a la fuerte demanda de un mercado que las utilizaba como herramienta para la especulación financiera -práctica común hacia fines de la década de 1970 e inicios de la siguiente en Argentina. Seleccionamos para trabajar el ejemplar de MS51 que, por el hecho de tener una placa madre, guardaba mayor relación con la tecnología actual.

---

<sup>5</sup>Desconocemos si alguna computadora de Micro Sistemas S.A. llegó a incorporar discos duros.

Trasladamos la computadora seleccionada a otra habitación en la que se encontraba un soplete. Luego de hacer una limpieza cuidadosa, procurando retirar el polvo que pudiera estar depositado en el interior del artefacto, probamos, ahí mismo, encenderla. Encendía pero la pantalla permanecía apagada. Revisamos que no hubiera algo mal conectado. En el ínterin, buscamos los teclados. Ezequiel encontró tres. Nos generaron un gran desconcierto. Tenían un cable con una entrada macho y, adjunto, una ficha de entrada de vídeo. Por otra parte, no encontrábamos en la MS51 una entrada hembra para conectar el teclado. Pasó un buen rato hasta que Ezequiel recordó que la entrada del teclado se encuentra en la base de la MS51, y que allí mismo también está el conector para la ficha de vídeo que sale del teclado y que permite que la pantalla encienda. Conectamos y prendimos la computadora. La pantalla se encendió. Exhibía solamente un cursor color fósforo blanco. No respondía a acciones de entrada -"input"- desde el teclado.

Recordamos luego un vídeo que había publicado Nicolás en *Twitter* donde se mostraba un mensaje de error impreso en la pantalla de una MS51. Le consultamos por mensaje si se trataba de una de las MS51 del laboratorio, y si era así, qué había hecho para que apareciera ese mensaje de error en la pantalla -lo que parecía ser un avance respecto de la situación nuestra tras el encendido. Nicolás nos respondió que era una de las MS51 del laboratorio, pero que no recordaba qué había hecho para llegar a que la pantalla imprimiera ese mensaje: sólo recordaba haber encendido la computadora. Repetimos los mismos pasos con las otras dos MS51: limpiamos, conectamos un teclado y encendimos. El resultado era el mismo. Probamos intercambiar teclados sin obtener un resultado diferente. El vídeo posteado por Nicolás no era del todo legible; sin embargo, llegamos a advertir que había un disquete puesto. Se trataba de un detalle clave que Nicolás no recordaba.

Con Roberto solicitamos permiso para extraer una de las disqueteras de la MS51 con la que empezamos a trabajar. La idea era analizar en casa cómo lee la disquetera de la MS51 para, seguidamente, lograr generar un disquete con una imagen de CP/M 2.2 ajustada a ese modelo.

### 3.3 Segunda visita

Roberto había trabajado en su casa con la disquetera de la MS51. Para ello, tras una limpieza profunda -en particular del cabezal de lectura-, la conectó a una vieja PC que tiene en su casa. Luego de varios intentos y ajustes, comprobó que la disquetera funcionaba sin problemas. Generó varios disquetes con diferentes imágenes de CP/M que se pueden descargar de un sitio web.

Para ello utilizó dos programas de MS-DOS: Omnidisk y IMD. Tenía una profunda desconfianza respecto de lograr, utilizando esas imágenes de CP/M, que la MS51 arrancara. Su escepticismo tenía un fundamento muy razonable: se trataba de imágenes de CP/M para computadoras que seguramente tenían características muy diferentes a las de la MS51. En los inicios de la microcomputación había una gran variedad de computadoras personales y cada una de ellas tenía sus especificidades. Estas dificultades causadas por la falta de homologación de los artefactos desaparecerían cuando las -por entonces nuevas- computadoras IBM-PC monopolizaron el mercado de la microcomputación.

La segunda visita, entonces, tenía como objetivo, una vez reinstalada la disquetera, arrancar la MS51 con al menos una de las imágenes de CP/M que Roberto había generado en los disquetes de 5.25. El día 18/11/2021 concretamos esta segunda visita al Laboratorio. Ingresamos alrededor de las 11 am. Esta vez introdujimos los disquetes antes de pulsar el botón de encendido, esperando que la computadora se iniciara con CP/M. Una vez más, sólo aparecía la pantalla con un cursor blanco. Probamos todos los disquetes con las imágenes de CP/M, y el resultado era el mismo: la MS51 no reconocía el SO CP/M. Permanecimos un largo rato tipificando el sonido que producía la disquetera -en sus frustrados intentos por leer los disquetes-, procurando identificar algún patrón. Reinaba el desánimo y el desconcierto.

Para trabajar con la MS51, esta vez apoyamos la computadora en una pequeña mesa que estaba en la entrada de la misma habitación donde estaban guardadas las tres MS51. Esa habitación es como un depósito de artefactos computacionales del pasado. Empezamos a buscar disquetes. Existía la posibilidad de que aquel disquete que había colocado Nicolás -que arrojaba un mensaje de error- estuviera en algún lugar del laboratorio. Pasado un buen rato, encontramos una caja de disquetes con un rótulo que decía "Disquette. Carlos Alberto Rizzi. San Francisco". Abrimos la caja. Uno de los disquetes decía "MS51". Probamos arrancar la computadora con ese disquete y apareció por fin el mensaje de error que le había aparecido a Nicolás. A continuación fuimos introduciendo los otros disquetes que se encontraban en la caja. Para nuestra sorpresa, uno de ellos contenía una imagen de CP/M para la MS51 con la que finalmente pudimos arrancar el equipo. La computadora reconocía la unidad de floppy drive -para lo que CP/M emplea el símbolo  $A>$ -. Luego probamos ejecutar algunos comandos de CP/M básicos, por caso, el comando DIR, y éste imprimió en pantalla -como era de esperar- la lista de archivos contenidos en el directorio principal.

## 4 El ensamble Hardware-Software-agente humano

### 4.1 ¿Por qué hablamos de *ensambles*?

Como andamiaje teórico para este trabajo adoptamos una perspectiva relacional. Ésta, en el marco establecido por el denominado Nuevo Materialismo, puede ser caracterizada a partir del siguiente supuesto ontológico: entre los entes no existen jerarquías, i. e., todos los entes son ontológicamente simétricos. Este supuesto tiene consecuencias epistemológicas importantes, entre otras cosas, porque disuelve la dualidad ontológica -fuertemente arraigada en la cultura de occidente a partir de los aportes del filósofo René Descartes- entre mente y cuerpo, y con ello, entre sujeto y objeto. (Cf. *Princ. A/T VIII*, 48 y 54)

En el marco cartesiano, conocer significa formarnos una representación de la cosa u objeto de estudio. Las cosas son reemplazadas por las representaciones que nos formamos de ellas, y éstas son construidas por el sujeto en virtud de un conjunto finito de ideas que anteceden y prefiguran la experiencia. (Cf. Robles- Benitez 1994, pp. 111-132) Ahora bien, ¿qué significa conocer una vez que hemos retirado de nuestros esquemas conceptuales la noción de un sujeto cognocente y la representación como instrumento cognitivo? Al disolverse la dualidad sujeto-objeto, y con ella, la representación como medio para relacionarnos con los entes, nos quedamos -por así decir- frente a las cosas mismas. Pero al no haber jerarquía alguna entre ellas, el objeto de estudio se desplaza de las cosas a las relaciones que se establecen entre ellas. Aquí surgen algunos interrogantes: ¿anteceden las cosas a las relaciones o las relaciones son primarias? (cf. Fowler y Harris 2015); ¿cómo algo deviene objeto de estudio una vez declarada la no jerarquía ontológica de las cosas y la ausencia de principios ordenadores antepuestos por una mente? La perspectiva relacional sostiene que la realidad puede ser entendida en términos de ensambles o composiciones de cosas. Los ensambles se caracterizan entonces como reuniones o composiciones de cosas que "actúan sobre, pero no definen totalmente, sus partes constituyentes". (Harris 2017, p. 4) De esto parece seguirse que los ensambles no poseen, adoptando la terminología de la metafísica clásica, una esencia sino que están supeditados al devenir histórico.

Un punto planteado en (Harris 2017) que resulta relevante para nuestro trabajo es que la teoría de los ensambles permite trabajar con diferentes escalas en arqueología. Por ejemplo, cuando nuestro objeto de estudio es una computadora construida a principios de la década de 1980, ¿debemos priorizar las relaciones de los componentes al interior del artefacto, la relación

de estos componentes con el SO, las condiciones locales de producción de computadoras de ese tiempo, o el estado del arte de las tecnologías computacionales en el marco de una emergente economía de libre mercado?

De acuerdo con Harris, la teoría de ensamblajes puede ofrecer una aproximación multiescalar. Más aún, los propios ensamblajes, así como los procesos y subprocesos que los conforman, son multiescalares puesto que se componen de otros ensamblajes:

Así que el lago y sus habitantes mencionados anteriormente se formaron a través de un proceso que se puede dividir en subprocesos que podemos analizar como geológicos y biológicos, y la comunidad (más que humana) se formó a través de una gama de desarrollos geológicos / biológicos / históricos (DeLanda 1997; 2002; 2006; 2016). Por lo tanto, en lugar de argumentar que debemos entender que los procesos históricos operan de manera diferente en varias escalas, como ocurre con el perspectivismo temporal (Bailey 2007, 200), podemos usar la misma aproximación ontológica para comprender cómo surgieron las entidades en diferentes escalas en el pasado. (Harris 2017, p. 5)

En base al pasaje citado, decimos que un ensamblaje se compone de otros ensamblajes; y que las diferentes escalas guardan entre sí estrechos vínculos. Cada escala contiene todas las escalas; sin embargo, no existe prioridad de una escala sobre otra sino que éstas se pueden diferenciar en términos de *intensidad*: la escala sobre la que elegimos trabajar, afirma Harris (citando a Protevi 2009), representa el punto más alto de intensidad.

A partir de este marco conceptual, examinemos el conjunto de relaciones que entablan los componentes de hardware y software de una computadora con los agentes humanos en la actualidad.

## 4.2 Las relaciones entre hardware-software hoy

Como describimos en la sección 2, el punto de partida de nuestro trabajo consistió en experimentar con el SO CP/M. Las funciones de un SO son múltiples. Una caracterización general es la que podemos encontrar en Wikipedia, donde se define un SO como "(...) el conjunto de programas de un sistema informático que gestiona los recursos de hardware y provee servicios a los programas de aplicación de software". Esta caracterización, cabe notar, no problematiza la compleja -y a veces poco nítida- distinción entre hardware y software. No obstante, para los fines de este trabajo, diremos por el momento

que un SO, por tratarse de un ecosistema de programas, es un conjunto de líneas de código escrito en uno o más lenguajes de programación. En nuestras experiencias cotidianas guarda un tipo particular de relación con las aplicaciones que empleamos todos los días con variados fines, ya sea desde nuestro celular o desde una computadora personal. Las aplicaciones, por cierto, son también programas computacionales, por lo que -al igual que a un SO- podemos caracterizarlas como un conjunto de líneas de código. Esta relación entre SO y aplicaciones la concebimos, aunque equívocamente, de manera análoga a como concebimos -también equívocamente- la relación entre espacio y figura. A un SO operativo se lo suele pensar como una condición de posibilidad para que las aplicaciones puedan ser ejecutadas de la misma manera que solemos asumir que la experiencia que tenemos con las cosas es posible porque éstas se acomodan de algún modo en el espacio. Ahora bien, cuando una aplicación deja de funcionar, o no lo hace correctamente, un *usuario* -i. e., un agente humano que es un receptor pasivo de las tecnologías computacionales- no suele cuestionarse la relación entre SO y aplicaciones. El SO, al igual que el espacio, se le presenta como un ser etéreo, ubicuo e intangible.

Esta relación que un usuario establece entre SO y aplicaciones cambia una vez que reemplazamos el SO. El grueso de los usuarios, para el caso de las computadoras personales, utiliza Windows. Si efectuamos un reemplazo de este SO por alguna de las decenas de distribuciones de SO de código abierto de la familia GNU/Linux, la relación de un usuario con la computadora se trastoca en forma notable. El usuario empieza a cuestionarse aquello que se presentaba como dado. Antes, SO y Windows eran términos sinónimos; luego del reemplazo, frente a un problema, el usuario se plantea preguntas propias de un usuario más avanzado; éstas apuntan a cuestionar la relación entre SO y aplicaciones. Retomando la imagen anterior, el usuario empieza a discriminar figura de espacio, contenido de continente.

Ahora bien, hasta aquí hemos acotado la discusión a los componentes de software de una computadora<sup>6</sup> y al tipo de relación que un usuario entabla entre dos grandes niveles de líneas de código: las del SO y las de las aplicaciones. Un componente, o conjunto de componentes, que aún no hemos discutido es el hardware de una computadora. ¿De qué manera se ensambla el hardware con las distintas capas de software? ¿Qué tipo de relación entabla un usuario con el hardware en la actualidad?

---

<sup>6</sup>Hasta el momento hemos distinguido sólo dos niveles o capas de software: la del del SO y el de las aplicaciones. Pero dentro de un mismo SO se pueden distinguir varias capas de software. En todo caso, sería más acertado partir de la distinción entre software de bajo y software de alto nivel.

Para un usuario, el hardware significa la puerta de acceso a un universo inagotable de experiencias virtuales. Sin embargo, la base material que soporta tales experiencias sólo en una mínima parte se explica a partir de los componentes de los dispositivos que utilizamos a diario. Un usuario difícilmente se plantea interrogantes acerca del soporte material que hace posible el universo virtual en el que estamos inmersos. Plantear una pregunta de ese tipo nos obliga a que amplíemos la escala de análisis. El ensamble hardware-software deberá incluir, entre otras cosas, al pesado cableado transocénico que nos mantiene conectados y a colosales granjas de computadoras ubicadas en las zonas más frías del planeta -o donde el suministro de carbón permita el abastecimiento energético a muy bajo costo-, granjas de computadoras que nos permiten almacenar buena parte de nuestra información personal. La base de nuestras experiencias virtuales depende entonces de la materialidad del hardware; y éste, de extraordinarias cantidades de minerales y metales -tungsteno, silicio, platino, antimonio, germanio, etc. Así, como ha mostrado Jussi Parikka, el subsuelo de nuestras experiencias virtuales se remonta a los estratos geológicos del planeta Tierra:

La información se alimenta del medioambiente tanto a través de la geología como de la demanda de energía. (Jussi Parikka 2015, p. 56)

El no reconocimiento del hardware, base material irreducible de nuestras experiencias virtuales, produce un efecto o estado de *ilusión de inmaterialidad*. Es esta misma ilusión de inmaterialidad la que hace que nuestra interacción con la computadora sea cotidiana y amable. La figura del usuario es una figura construida en virtud de un proceso de programación, a lo largo de décadas, de capas y más capas de software que sin pausa fue avanzando hacia la constitución de un lenguaje afín al tipo de cognición humana, dejando al mismo tiempo atrás el lenguaje asequible a una máquina. Del sistema de numeración binario empleado por una máquina Turing, al símbolo, y de éste, al icono. De la línea de comandos, pasando por el mouse y la interfaz gráfica, a las pantallas sensitivas. La abstracción alcanzada ha invisibilizado la compleja relación que existe actualmente entre hardware y software. Ha facilitado el acceso a las tecnologías computacionales, vale decir, al grueso de la población mundial poniendo a un *clic* de distancia un número cada vez mayor de aplicaciones. Dado que el universo virtual en el que estamos inmersos posee, bien que en el subsuelo, una fuerte base material, preguntamos: ¿qué precio hay que pagar por la abstracción en computación? Dejamos para el cierre alguna respuesta posible para esta pregunta. Pasemos ahora

al estudio del ensamble hardware-software-agente humano en los inicios de la microcomputación.

### 4.3 Las relaciones entre hardware y software en el contexto de la MS51

¿De qué manera se ensamblan los componentes de hardware y software en una computadora como la MS51? ¿Guarda dicho ensamble algún punto de contacto con las relaciones que se establecen entre hardware y software en la actualidad? ¿Qué papel tenían los agentes humanos en el conjunto de relaciones entre hardware y software en los inicios de la microcomputación?

CP/M es un SO sin interfaz gráfica. En la historia de la computación, la interfaz gráfica llegaría algunos años más tarde. Esto significa que con CP/M había que interactuar a través del uso de línea de comandos. Dijimos que los denominados *built-in commands* son aquellos que están ubicados en una de las divisiones lógicas del SO CP/M y que sirven -en términos generales- para la manipulación de archivos. Los nombres dados a los *built-in commands* evocan las acciones que estos comandos pueden realizar. Por ejemplo, el comando para eliminar archivos ERA evoca el vocablo ERASURE, que al español traducimos como "borrar". Esto los volvía más accesibles a los agentes humanos. Un comando muy importante en relación con el manejo de archivos es el comando PIP. Se trata de un programa más complejo. Quizás por eso haya sido separado de los *built in commands* y ubicado en un sector lógico diferente del SO. Tanto el nombre del comando como su sintaxis pueden resultar poco transparentes para un usuario. Consultando el archivo de la prestigiosa revista BYTES, encontramos un interesante pasaje sobre el comando PIP.

En la nota principal del Vol. 7, Nro. 1, de la revista Bytes, publicada en el mes de enero de 1982, se presenta una reseña exhaustiva de la *IBM Personal Computer*. Fue la primera computadora personal de la empresa IBM, lanzada al mercado en agosto de 1981. La PC-IBM impactaría de manera decisiva en el mercado de la microcomputación. Tras una negociación frustrada de Gary Kildall con IBM, DOS -que luego se llamaría MS-DOS- se transformaría en el SO de las PC-IBM. Con el acuerdo entre IBM y Microsoft empezaron los tiempos de los pactos de confidencialidad y del software de código propietario.<sup>7</sup> La reseña de la revista Bytes compara el SO DOS con

---

<sup>7</sup>CP/M estuvo a una firma de ser el sistema operativo de las PC-IBM. Finalmente el lugar de CP/M lo ocuparía MS-DOS, el sistema operativo sobre el que más tarde se montaría Windows. A principios de 1980 IBM decidió pisar fuerte en la industria de la microcomputación. Aún no habían desarrollado su propio SO, de manera que decidieron

CP/M. Indica una ventaja de aquél sobre este último. Según la revista, el comando `COPY` de DOS para copiar archivos está "mejor redactado" que el comando `PIP` de CP/M:

"In addition, the commands are better worded than in CP/M .

For example, the cryptic

```
PIP B :NEWFILE1 = A:MYFILE1
```

of CP/M is replaced by

```
COPY A:MYFILE1 B:NEWFILE1
```

which copies MYFILE1 from drive A to drive B, where it will be named NEWFILE1".

El criterio para establecer la ventaja de un comando por sobre el otro es la facilidad de uso. Adoptando un anglicismo típico de la jerga computacional, el comando `COPY` de DOS es un comando *user friendly*, esto es, un comando amigable para un usuario. Este criterio empezaba a adquirir mayor fuerza a medida que el mercado de la microcomputación se expandía. Así, en una emisión de marzo de 1984 del programa televisivo *The Computer Chronicles*, Stewart Cheifet -conductor del programa- entrevistó a Gary Kildall -quien a su vez co-conducía el programa- con motivo del reciente lanzamiento al mercado de una versión *multi-tarea* de CP/M-86. Esta versión de CP/M permitía ejecutar varios programas al mismo tiempo en diferentes consolas, posibilidad técnica que en otros sistemas operativos llegaría varios años más tarde, promediando la década de 1980. Stewart Cheifet le solicitó a Gary Kildall que le explicara a la audiencia qué es un SO. La respuesta puede

---

iniciar negociaciones con Gary Kildall para adquirir CP/M. Gary Kildall junto con Tom Rolander, su mano derecha desde la creación de *Digital Research*, viajaron en avioneta para reunirse con directivos de IBM. En una entrevista que le hicieron en el año 1995, Tom Rolander -recordando aquella reunión- definió los términos de IBM como un "acuerdo unidireccional y confidencial". Tras la negativa de *Digital Research* de cerrar un acuerdo en esos términos, un joven llamado Bill Gates decidió comprar un sistema operativo basado en CP/M e iniciar negociaciones con IBM. Así nació MS-DOS, el sistema operativo con el que muchos de nosotros crecimos. *Digital Research* inició acciones legales contra IBM por infringir derechos de autor. Ante esto, IBM retrocedió en las negociaciones con Bill Gates. Finalmente llegaron a un acuerdo. La nueva y poderosa computadora IBM podría bootearse con CP/M o con MS-DOS. La idea era dejar que el cliente decida. La sorpresa llegó cuando Gary y su equipo advirtieron que la nueva computadora IBM había sido lanzada al mercado ofreciendo MS-DOS a 40u\$d, mientras que aquellos que optaran por CP/M debían pagar seis veces ese valor, esto es, 240u\$d. La inesperada política de comercialización de IBM marcó el inicio del ocaso de CP/M. Muy pronto IBM y Microsoft ocuparían un lugar dominante en el mercado.

resultar aun hoy muy esclarecedora. A su lado se encontraba Tony Fanning, un alto personal de ingeniería de Hewlett-Packard, a quien, tras la respuesta de Kildall, le consultaron qué esperaba él, como usuario, de un sistema operativo. Fanning respondió:

"[B]ut really when I'm using it, I want the operating system to be transparent to me. I don't want to see it. I just want to use an application (...) if I'm trying to do is something like, for example, word processing, I'd like the operating system just disappear and not bother me". (The Computer Chronicles, Programa 5, año 1984, min. 10:00-10:26.<sup>8</sup>)

Las palabras de Fanning caracterizan con notable precisión la figura del usuario. Indicaban la necesidad de entablar una relación entre el agente humano y el software exclusivamente en el nivel de las aplicaciones. Poniendo la imagen en negativo: indicaban la necesidad de velar la compleja trama de variados niveles de software y componentes de hardware de una computadora. El concepto de computación empezaba a cambiar. Pronto la computadora dejaría de ser vista como un artefacto intrínsecamente interesante para transformarse en un medio para realizar determinadas tareas. La figura del usuario se empezaba a proyectar en relación con las aplicaciones, y éstas, con un campo laboral que iba a sufrir importantes transformaciones a medida que muchas actividades, otrora realizadas por agentes humanos, empezaban a ser reemplazadas por aplicaciones. En otras palabras, los dichos de Fanning estaban exigiendo la creación de un SO a la medida de un usuario. Pero este usuario debía ser construido. Para ello, había que dejar atrás una figura todavía muy vigente en ese tiempo, a saber, la figura del *operador*. A esta figura la podemos identificar en el propio manual de CP/M, y en incontables pasajes de la revista Bytes de ese tiempo. El término operador se empleaba para referir a un agente humano que era capaz de utilizar un SO como CP/M.

Para un operador estaba diseñado un programa como el editor ED -que analizamos más arriba. El empleo de ED requería el uso de comandos, así como el reconocimiento del sistema de archivos y de algunos componentes del hardware de la computadora como la memoria RAM. Una manera de pensar las relaciones entre hardware-software-agente humano es involucrando esta

---

<sup>8</sup>"En verdad cuando voy a usar un sistema operativo quiero que me resulte transparente, no quiero poder verlo, tan sólo quiero poder usar una aplicación (...) si lo que estoy tratando de hacer es, por ejemplo, procesar un texto, me gustaría que el sistema operativo desaparezca y no me moleste". (La traducción es nuestra)

nueva figura, la del operador. Preguntamos: ¿de qué manera se relacionan los componentes del ensamble hardware y software con un agente humano *qua* operador?

#### 4.4 Operar con una MS51

Las computadoras MS51 de Micro Sistemas S.A. utilizan el SO CP/M 2.2.<sup>9</sup> Desconocemos qué modificaciones hicieron los ingenieros informáticos de la empresa para adaptar una versión de CP/M a los componentes de una MS51. Es probable que hayan modificado el código de la BIOS del SO. Se trata de tareas de programación de muy bajo nivel, es decir, muy próximas al lenguaje de máquinas. Un aporte destacable que realizamos en el marco de esta investigación, fue digitalizar la imagen de CP/M del disquete que permitió arrancar la MS51, tarea que -por la dificultad técnica que supone- realizó Roberto.<sup>10</sup> Utilizó, una vez más, los programas Omnidisk y IMD. A través de éstos se puede "escanear" el disquete para averiguar su formato (no-estándar), y con ello volcar todo el contenido en un archivo "imagen".

De acuerdo con la escasa bibliografía disponible sobre las computadoras de Micro Sistemas S.A., sabemos que la empresa impulsó el desarrollo, y en algunos casos la creación, de importantes empresas en la provincia de Córdoba. Podemos citar el caso de la cementera Corcemar, empresa que empleaba una MS-101 para controlar sus balanzas. También se utilizaron computadoras de Micro Sistemas para realizar operaciones de apuestas en el Jockey Club y otras operaciones del Banco de Córdoba. Además demandaron sus servicios La Voz del Interior y las Fuerzas Armadas -para realizar *juegos de guerra*.<sup>11</sup> Dado el estado del arte de la microcomputación en ese tiempo -entre 1976 y 1984-, un *usuario* de esas computadoras seguramente tenía un grado de interacción muy bajo con la computadora. Nicolás Wolovick, quien aún hoy está en contacto con los creadores de Micro Sistemas S.A. y con ex empleados de la empresa, nos explicaba -en una entrevista que le realizamos el día 23/02/2022- que una persona que empleaba una MS en Corcemar o en el Banco de Córdoba, seguramente se limitaba a realizar acciones básicas directamente relacionadas con la aplicación instalada por la empresa o entidad bancaria. Usar una computadora en ese contexto era,

---

<sup>9</sup>Aquí surgen algunas inquietudes. Es altamente probable que la MS51 utilice la versión 2.2 de CP/M, pero es un dato a confirmar en futuras investigaciones.

<sup>10</sup>El trabajo de investigación impulsado por Nicolás Wolovick sobre las computadoras de Micro Sistemas S.A. está disponible en <https://micro-sistemas.github.io/>.

<sup>11</sup>Esta información fue tomada de la ya referida nota de Lucas Viano, publicada en La Voz del Interior el día 12/10/2021.

*mutatis mutandis*, similar a usar un cajero automático para retirar dinero. Un cajero automático contiene sólo una aplicación, la cual se presenta ante nosotros ofreciendo un menú con una serie de opciones que podemos aceptar o rechazar. Un usuario, en los tiempos de las MS, tenía un radio de interacción muy pequeño con la computadora.

La figura del operador era una figura intermedia entre la pasividad del usuario y la figura del hacker. El estado actual del desarrollo de las tecnologías computacionales ha vuelto casi imposible que una sola persona conozca todos los componentes de una computadora y los intersticios de un sistema. Esto sí era posible en el período que estamos considerando. Un hacker poseía entonces un vasto conocimiento de casi todos los componentes de una computadora y la capacidad de controlar y programar los sistemas. Mas también, un hacker era aquella persona que facilitaba y promovía el libre acceso a la información, abriendo al público el código fuente de los programas, una práctica bastante usual hasta ese momento. De hecho, el SO CP/M era un conjunto de programas de código abierto. Esto permitió que pueda adaptarse a los componentes de una MS51.

La figura del operador era la de un agente con las habilidades técnicas suficientes, entre otras cosas, para utilizar los comandos de un SO. Reconocía el sistema de archivos, realizaba acciones como renombrar, borrar y copiar archivos, instalaba nuevos programas, etc. Se puede distinguir al operador del usuario porque, a diferencia del primero, un operador realizaba acciones que modificaban algunos aspectos de una computadora. Interactuaba al punto de advertir la complejidad de una computadora. Esta capacidad de controlar los aspectos básicos de un SO le permitía a un operador realizar las tareas que en la actualidad puede realizar un usuario tipo. Al igual que el hacker, parte del proceso de aprendizaje en un operador se producía a través de la generación de hipótesis y de instancias de ensayo y error típicas de la dinámica del juego.<sup>12</sup> La computadora despertaba su curiosidad. Un operador, no obstante, no creaba nuevos sistemas ni poseía conocimientos sobre los fundamentos teóricos de la computación.

La figura del operador guardaba relación con el estado del arte de la microcomputación de finales de la década de 1970, principios de la década siguiente. Las condiciones materiales para el ejercicio de la computación dotaban de sentido a una tal figura. *Operar* con una MS51 requería cierta *expertise* y mucha paciencia. Más aún, operar con una MS51 suponía experimentar un sentimiento de goce frente a los obstáculos que a menudo se

---

<sup>12</sup>Una discusión sobre el rol del *juego* en relación con la figura del hacker la podemos encontrar en el siguiente ensayo publicado en la revista POGO: El manual del buen hacker.

presentaban para la realización de acciones que hoy consideraríamos básicas. El ensamble hardware-software-operador, de acuerdo con lo dicho, exhibe relaciones equilibradas, simétricas. Los vínculos entre estos tres conjuntos de componentes son vínculos vivaces y explícitos. La ilusión de inmaterialidad que experimenta un usuario del siglo XXI, todavía quedaba bastante lejos.

La figura del operador se fue diluyendo a medida que el grado de abstracción de los programas computacionales crecía. El ideal al que se apuntaba estaba perfectamente expresado por las palabras de Tony Fanning citadas anteriormente: el SO debía ser completamente transparente. Es decir, debía alcanzar un grado de abstracción tal respecto del lenguaje de las máquinas hasta lograr que un usuario pudiera realizar sin problemas el tipo de acciones propias de un operador. Este ideal recién se alcanzaría en su totalidad con la llegada de los celulares inteligentes. El SO Windows fue, sin duda, un importante paso en aras de este ideal.

## 5 Reflexiones finales

Los agentes humanos se han relacionado de diferentes modos con las computadoras desde los inicios de la microcomputación a esta parte. El estudio de las relaciones, y las transformaciones en esas relaciones en el tiempo, entre los componentes de hardware y software de una computadora permite elucidar la emergencia de tres *figuras* o estereotipos de agentes humanos. La figura del hacker, del usuario y del operador. A estas tres figuras las podemos pensar en el marco de la teoría de ensamblajes. El ensamble hardware-software-agente humano presenta un gran dinamismo. El conjunto de mutuas relaciones entre los componentes del ensamble, en diferentes circunstancias y puntos temporales, componen específicas configuraciones dotando de sentido a cada una de estas tres figuras.

Los agentes humanos, en su relación con los componentes de hardware y software, desempeñan diferentes papeles. Estos papeles se distinguen de acuerdo con el grado de interacción que entablan con dichos componentes. La figura del usuario, en los tiempos de una MS51 y en la actualidad, no ha variado. Hoy como entonces, el usuario adopta una actitud de receptor pasivo frente a las tecnologías computacionales. Por supuesto que el margen de acción de un usuario frente a una MS51 era mínimo, lo cual se explica si atendemos a qué tipo de relaciones se establecían entre hardware y software en una MS51.

Una figura, hoy perdida, que tenía un papel más activo frente a una com-

putadora como la MS51, era la figura del operador. Ésta tuvo sentido en el marco de una particular trama de relaciones entre hardware y software, trama en la que dichas relaciones se acotaban a la propia computadora. Actualmente, analizar las relaciones entre hardware y software nos demanda ampliar la escala de análisis. Esto se debe a que la base material que soporta la variedad de aplicaciones que utilizamos minuto a minuto excede por mucho a los materiales del dispositivo que empleamos para ejecutarlas. Así, fue necesario la invisibilización del hardware, y capas de abstracción de software, para desprender casi por completo el lenguaje con el que interactuamos con una computadora del lenguaje de las máquinas, abriendo con ello el paso a una superpoblación de usuarios.

Finalmente, la pérdida de una figura como la del operador iba también a dar paso a una fuerte polarización entre la figura del usuario y la del hacker. El precio que debíamos pagar por la abstracción -en el desarrollo de la microcomputación- era el de la pérdida de control sobre la computadora. El acuerdo entre IBM y Microsoft en 1981 marcaría el inicio del ocaso del SO CP/M; mas también, el inicio del ocaso del libre acceso al código fuente de un programa computacional. La pérdida de control sobre la computadora no se explica sólo a partir de la condición pasiva de un usuario. Las políticas de restricción de acceso al código fuente de un programa fue un paso decisivo para que el control se depositara en unas pocas manos. La imposibilidad de acceder al código fuente de un programa socavó una serie de libertades, como la de distribuir copias fieles de los programas, y la de realizar mejoras sobre ellos o modificaciones para adaptarlos a nuestros deseos y necesidades. La expansión de la figura del usuario fue de la mano de un proceso de cajaneización de las tecnologías computacionales. El ascenso de la figura del hacker a partir de 1984 -en el marco del proyecto GNU iniciado por Richard Stallman- se produjo justamente como una contraofensiva a las políticas que restringían de acceso al código fuente de un programa.

Cerramos con un interrogante: ¿cómo se presentaría la figura, hoy perdida, del operador en el marco del actual conjunto de relaciones entre los componentes de software y hardware? ¿Qué acciones podría realizar hoy un agente que en los inicios de la microcomputación era capaz de instalar y ejecutar diversas aplicaciones, y dominar el sistema de archivos? Quizás la tan mentada alfabetización tecnológica requiera la recuperación de esta figura que la breve historia de la microcomputación ha dejado atrás hace ya cuarenta años.

## 6 Bibliografía

E. Chesini, G. del Dago, N. Wolovick, "MS101, La Maquinita de Bazán", SHIALC 2018.

*CP/M - Features and Facilities* (1983). DIGITAL RESEARCH TM, CP/M Operating System Manual.

Fowler, Chris y Harris, Oliver (2015). "Enduring relations: Exploring a paradox of new materialism", Vol. 20(2) 127–148.

Harris, Oliver (2017). "Assemblages and Scale in Archaeology". En: Cambridge Archaeological Journal 27:1, 127–129, 2017.

Huhtamo Erkki y Parikka, Jussi (2011). *Media Archeology. Approaches, Applications and Implications*. California University Press.

Oeuvres de Descartes. Edition Ch. Adam y P. Tannery, Léopold Cerf, Paris 1897-1913. Nueva reimpression CNRS.

Parikka, Jussi (2015). *Una geología de los medios*. Maximiliano Gonnet (trad.), Caja Negra (ed), Buenos Aires (2021).

Robles, José y Benítez, Laura (1994). "La vía de las ideas". En *Enciclopedia Iberoamericana de Filosofía: del Renacimiento a la Ilustración I*, Ezequiel de Olaso (ed.), editorial Trotta.