

Universidad Nacional de Quilmes

Elementos de Programación y Lógica

Solucionario del Cuadernillo del Estudiante

por Alan Rodas Bonjour

Ver. 0.6.0 (24/04/2022)

Copyright © 2019 Alan Rodas Bonjour

PUBLICADO ONLINE POR ALAN RODAS BONJOUR

HTTP://ELEMENTOSDEPROGRAMACIONYLOGICA.WEB.UNQ.EDU.AR



Licenciado bajo los términos de la siguiente licencia de contenidos: Licencia Creative Commons Atribución-NoComercial-CompartirIgual 4.0 Internacional (la “licencia”). Puede obtener una copia completa de ella en <https://creativecommons.org/licenses/by-nc-sa/4.0/deed.es>.

Usted tiene derecho a:

- **Compartir:** copiar y redistribuir el material en cualquier medio o formato
- **Adaptar:** remezclar, transformar y construir a partir del material

El licenciatario no puede revocar estas libertades en tanto usted siga los términos siguientes.

- **Atribución:** Usted debe dar crédito de manera adecuada, brindar un enlace a la licencia, e indicar si se han realizado cambios. Puede hacerlo en cualquier forma razonable, pero no de forma tal que sugiera que usted o su uso tienen el apoyo de la licenciatario.
- **NoComercial:** Usted no puede hacer uso del material con propósitos comerciales.
- **CompartirIgual:** Si remezcla, transforma o crea a partir del material, debe distribuir su contribución bajo la misma licencia del original.

No hay restricciones adicionales — No puede aplicar términos legales ni medidas tecnológicas que restrinjan legalmente a otras a hacer cualquier uso permitido por la licencia.

No tiene que cumplir con la licencia para elementos del material en el dominio público o cuando su uso esté permitido por una excepción o limitación aplicable.

No se dan garantías. La licencia podría no darle todos los permisos que necesita para el uso que tenga previsto. Por ejemplo, otros derechos como publicidad, privacidad, o derechos morales pueden limitar la forma en que utilice el material.

Primera edición, Febrero 2019



Índice general

I

Computadoras

1	Computadoras	7
1.3	Actividades	7
2	Historia de las computadoras	13
2.5	Actividades	13

II

Información

3	Bajo nivel	23
3.3	Actividades	23
4	Informática	27
4.5	Actividades	27

III

Lógica

5	Introducción a la lógica	41
5.1	Actividades	41



Computadoras

1	Computadoras	7
1.3	Actividades	
2	Historia de las computadoras	13
2.5	Actividades	

1. Computadoras

Placa base de una computadora con sus circuitos impresos.

Fotografía de [Blickpixel](#).

1.3 Actividades

Responda las siguientes preguntas. Puede investigar en Internet si lo cree conveniente.

Ejercicio 1.1 Mire el cartel a continuación



Determine.

¿Cuáles de los elementos mencionados corresponden a software?

¿Cuáles de los elementos mencionados corresponden a hardware? ■

En el cartel, los siguientes elementos aluden directamente al software:

- Filtros y colores en Photos
- FaceID para reconocimiento facial
- iOS 11 con Siri
- Llamadas de video con FaceTime

Mientras que los siguientes elementos aluden al hardware:

- Chip A11 Bionics de 64 bits
- Pantalla OLED multitouch de 5.8"
- Cámaras de 12MP con gran angular y teleobjetivo
- WiFi 802.11ac con MIMO
- Bluetooth 5.0
- NFC con modo lectura
- Bocina estéreo integrada y micrófono de alta calidad
- Hasta 21hs de batería en conversación

Cada uno de los elementos de hardware requiere de software para funcionar. Por lo tanto, incluso si refieren a hardware, siempre hay software involucrado para que el dispositivo funcione y se integre con el sistema operativo. El software no necesariamente es una aplicación que viene por separado, sino que son partes específicas del sistema operativo que dar soporte a ese hardware, lo cual se realiza en forma de firmware, drivers u otros.

Ejercicio 1.2 ¿Qué sistemas operativos conoce?

Discuta su respuesta con compañeros y el docente para descubrir nuevos sistemas. ■

Existen muchos sistemas operativos, algunos ya mencionados en el libro. Repasemos algunos de ellos, e indagemos sobre otros:

Microsoft Windows

El más conocido es tal vez **Microsoft Windows**, debido a que es el sistema que viene instalado por defecto en muchas computadoras. **Microsoft** hizo un acuerdo con la empresa **IBM**, en donde sus máquinas se vendían con Windows pre-instalado en sus equipos, que en los 80s y principios de los 90s eran el estándar de la industria. A medida que aparecían otros jugadores en la industria del hardware, Microsoft establecía contratos draconianos para que Windows fuera el sistema por defecto en sus equipos, a la vez que las empresas de hardware buscaban estos contratos en pos de captar usuarios ya fieles al sistema operativo de las ventanas.

Microsoft establecería una fuerte campaña de eliminación de la competencia y monopolio del mercado, al iniciar acuerdos con diversas dependencias del Estado en las que entregaban Windows de forma gratuita (solo inicialmente), por ejemplo en el área de educación.

Esto hizo que muchas personas solo hayan sido expuestas a este sistema operativo, y por tanto es el único que conocen. Peor aún, es el único que saben manejar, y por tanto, cuando necesitan una nueva computadora, esta debe venir con Windows (el usuario debe pagar un adicional) o debe instalarse luego (se debe comprar la licencia).

Las primeras versiones de Windows estaban basadas en **MS-DOS**, otro popular sistema de Microsoft de los 80s. Tuvo varias versiones a lo largo de su historia, como **Windows 1.0**, **Windows 2.0** y **Windows 3.1** (Las primeras versiones bastante primitivas, pero que sentaron las bases para las demás, y donde se desarrolla el concepto de ventana), **Windows 95** (El primero en tener el “botón



Bliss, la imagen de fondo por defecto de Windows XP se ha transformado en un ícono de ese sistema.
Imagen de Azuresh.

de inicio”), **Windows 98** (primero con soporte para acceso a Internet por defecto), **Windows NT** (para computadoras de uso empresarial), **Windows XP** (el primero en mezclar tecnologías de NT, con las de 98, que hasta el momento eran tecnologías separadas. Fue tal vez de las versiones más populares, saliendo en el año 2000), **Windows 8** (donde se introdujo la interfaz Metro) y **Windows 10** (la versión actual, que mezcla la interfaz Metro con una más tradicional).

MacOS

Históricamente **Apple** ha desarrollado sus propios sistemas operativos, para sus equipos (Ellos se definen como una empresa que vende computadoras, no sistemas). Las primeras computadoras Apple, como la **Apple II**, venían con **Apple DOS** y luego **Apple ProDOS**. Más tarde lanzaría lo que posteriormente se conocería como **Mac OS**, en ese momento denominado simplemente **System**, acompañando su nueva línea de computadoras Macintosh.

Llegando el 2000, su sistema empezaría a sentir el paso de los años, un nuevo sistema diseñado desde cero sería lanzado con toda su nueva línea de computadoras, **Mac OS X**, sistema que hoy a recibido un rebranding, para denominarse simplemente **macOS**, y que acompaña a todas las computadoras de la marca.

A diferencia de otros sistemas operativos, macOS es un sistema que viene instalado por defecto en los equipos de Apple, y que no se encuentra disponible para otros equipos, ni siquiera como compra por separado.

GNU/Linux

Cuándo inició el movimiento de **software libre** en 1984, **Richard Stallman**, el fundador del movimiento, se propuso realizar un sistema operativo completamente libre, compatible con **UNIX** (Un popular sistema de la época, que era propiedad de AT&T). Así, Stallman funda el **Proyecto GNU**, bajo cuya ala aparecen cientos de herramientas necesarias para el funcionamiento de un sistema operativo realmente utilizable, como compiladores, editores, terminales, etc. Sin embargo, el proyecto **Hurd**, un sub-proyecto de GNU que desarrollaría el **núcleo** del sistema operativo, seguía teniendo cientos de retrasos, idas y vueltas, sin nunca terminar con una versión utilizable. Ya en 1991, **Linus Torvalds**, un estudiante de la Universidad de Helsinki en Finlandia, quería utilizar en su computadora hogareña un sistema compatible con UNIX. Ante la ausencia de alternativas, se dispuso a programar su propio sistema, y tras anunciar públicamente en un foro sobre su desarrollo, Stallman se le acercó para convencerlo de que **Linux**, nombre que recibiría el sistema operativo, debía ser libre.

Linux evolucionó, y al ser libre, miles de personas pueden contribuir al proyecto. Su código es la base para múltiples sistemas operativos empaquetados para el usuario final, que utilizan el núcleo Linux, junto con las herramientas de GNU, y otros conjuntos de herramientas, imágenes, iconos, etc. (Lo que se conoce como una **distribución**). Hay miles de distribuciones, como **Ubuntu**, **Mint**, **Fedora**, **Debian**, **Arch**, **Zorin**, **Huayra**, entre otros tantos. Cualquiera con los conocimientos técnicos suficientes puede crearse su propia distribución. Lo más interesante es que existen distintas distribuciones, que se enfocan en distintas cosas. Así por ejemplo, hay distribuciones para computadoras de escritorios, otras para teléfonos celulares, otras para grandes servidores, para routers, para computadoras viejas, para consolas de videojuegos, etc.

BSD

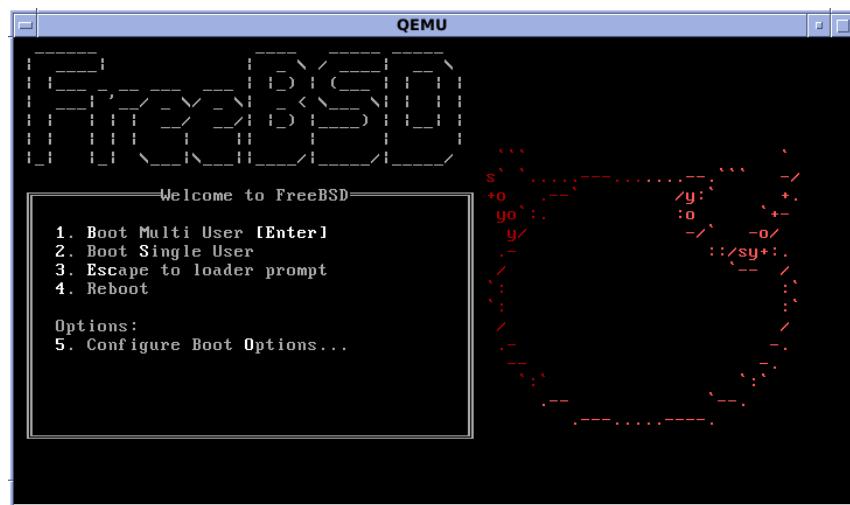


Imagen de inicio de FreeBSD, corriendo sobre QEMU.

Imagen de Huihermit.

Los sistemas **BSD (Berkeley Software Distribution)** comprenden, al igual que Linux, un núcleo, y varias distribuciones derivadas de dicho núcleo (Aunque incluso hay núcleos distintos e incompatibles entre si). El sistema surgió en la **Universidad de Berkley**, en donde **Bill Joy** compiló un conjunto de programas creados en la universidad que corrían sobre la versión 6 de UNIX. Otros universidades que utilizaban UNIX, podían entonces instalar dichos programas en sus

computadoras. Los programas comenzaron a crecer y a incluir cambios en el mismo sistema UNIX. Sin embargo, por ser solo modificaciones a partes del sistema, aunque Berkley distribuía todo el sistema UNIX modificado, AT&T seguía siendo el dueño del sistema, y por tanto, quien quisiera utilizarlo debía pagar a AT&T.

Tras una disputa legal que tomaría años resolver, Berkley dejaría de distribuir el software. Tras la resolución del conflicto en 1992, y tras quitar todo rastro de código del sistema UNIX original, se liberaría **BSD 4.2**. Berkley distribuye el sistema bajo una licencia llamada **Licencia BSD**, que permite que cualquiera pueda tomar el código de BSD y utilizarlo para lo que quiera, inclusive, en proyectos comerciales de código cerrado.

Hoy hay varias distribuciones BSD, como **FreeBSD**, **OpenBSD**, y **NetBSD**, pero hay muchas otras a su vez derivadas de estas tres. No solo eso sino que **parte del código de BSD se encuentra en Windows, y es la base para macOS de Apple**.

iOS

iOS es un sistema operativo creado por **Apple** para que corra de forma exclusiva en el **iPhone** y luego en **iPad**. El sistema utiliza una gran cantidad de código de BSD, y también de macOS.

Este sistema se crea pensando en la optimización de recursos y de batería, así como en una interfaz que pueda ser utilizada de forma sencilla en un dispositivo que no cuenta ni con teclado ni mouse.

Android

Android es un sistema creado por **Google**, basándose en el núcleo de **Linux** (Aunque el mismo está altamente modificado para funcionar de forma óptima en dispositivos móviles y tablets). El sistema, al estar basado en Linux es software libre, pero la mayoría de los fabricantes de teléfonos que venden dispositivos con Linux agregan gran cantidad de herramientas no libres sobre el sistema, comenzando por **Google**, que incluye no solo **Play Store**, sino también aplicaciones como **Gmail**, **Calendar**, etc.

Otros sistemas

Estos no son los únicos sistemas operativos. Existen cientos, y siempre aparecen nuevos (Aunque muchos son altamente experimentales, o son solo prototipos que fracasan). Ejemplos interesantes son **ReactOS** (Un sistema libre que intenta generar un sistema completamente compatible con Windows XP), **Haiku** (Un sistema que intenta revivir BeOS, y ser compatible con este), **MenuetOS** (Un sistema operativo que pesa solo 3.25MB) y **Plan9** (Un sistema operativo experimental de Bell Labs).

A lo largo de la historia han habido varios otros sistemas que han ganados varios adeptos, y cuyas ideas innovadoras en muchos casos fueron luego adoptadas por los sistemas comerciales más grandes, como **Amiga OS**, **BeOS**, **Symbian**, entre otros.

Finalmente podemos mencionar que existen sistemas operativos mucho más oscuros, experimentales y casi sin usuarios. Un caso muy interesante es el **TempleOS**, un sistema operativo con contenidos bíblicos, creado por Terry A. Davis, quien lo creó con la intención de transformar al sistema operativo en el tercer templo profetizado en la biblia. Davis comenzó a sufrir una serie de episodios esquizofrénicos y alusiones sobre invasiones extraterrestres y conspiraciones gubernamentales, motivo por el cual estuvo hospitalizado en reiteradas ocasiones. Tras una auto-

proclamada “revelación” hacia su persona por Diós, con quien él dice tener comunicación directa, comenzó el trabajo de desarrollar dicho sistema operativo, pues consideraba a otros sistemas como impuros y llenos de pecado. TempleOS ha sido criticado por expertos como una excelente pieza de ingeniería y un trabajo excelente para ser producto de una sola persona, aunque no sea un sistema operativo capaz de ser utilizado para el día a día.

Ejercicio 1.3 ¿Se le ocurren otros dispositivos de hardware que use de forma cotidiana?
¿Son periféricos de entrada, de salida, de entrada y salida, o de almacenamiento? ■

Existen cientos de dispositivos. Por ejemplo, como entrada es común encontrar emisores infrarrojos. Esto es fácil de encontrar en consolas de videojuegos (Zappers, Control de Wii, entre otros). También son comunes los sensores de presión (Botones táctiles, o plataformas sobre las cuales uno se puede parar).

En la parte de salida, existen diversas cosas, como rotores y motores que pueden generar movimiento, abriendo o cerrando tuberías u otros. También hay incluso proyectores holográficos.



2. Historia de las computadoras

Museo de Historia de las Computadoras en Mountain View, California, EE.UU..

Michael Kappel.

2.5 Actividades

Ejercicio 2.1 ¿Cuál fue la primer computadora en argentina? ¿Para que se usaba? ¿Quién la trajo al país?

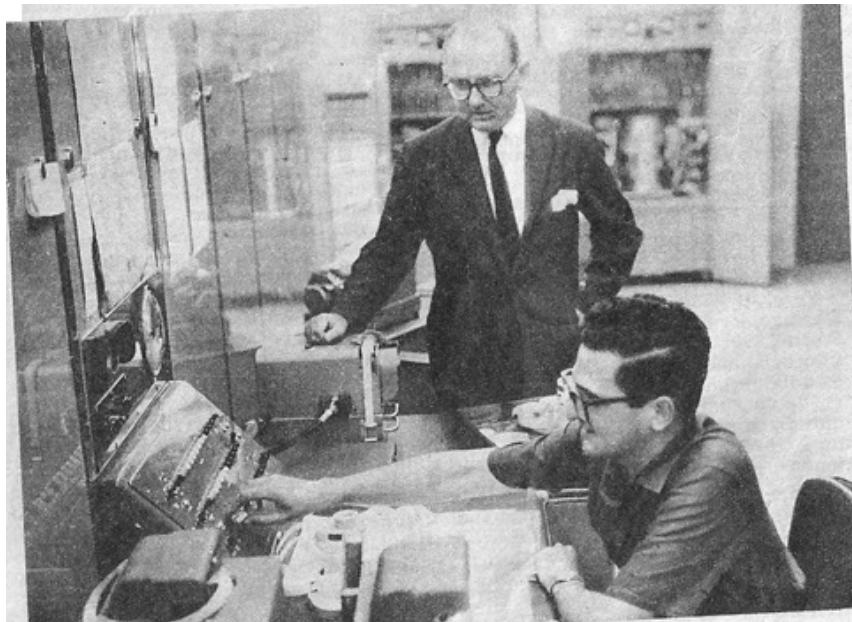
La primer **computadora para fines científicos** que funcionó en Argentina fue **Clementina**, una **Ferranti Mercury**, creada en 1957, de la cual se produjeron solo 19 unidades.

Manuel Sadosky fue quien lideró las gestiones para su adquisición en 1959. La computadora llegó el 24 de Noviembre de 1960, aunque recién comenzó a operar en Enero de 1961. Ferranti (una empresa inglesa) ganó la licitación por sobre firmas como IMB, Remington y Philco. La máquina costó 152.099 libras esterlinas (equivalentes a unos 4.500.000 dólares actuales, aproximadamente), lo cual representó el desembolso más grande en ciencia y tecnología en la historia hasta ese momento.

La computadora se ubicó en el **Instituto de Cálculo**, dependiente de la **Universidad de Buenos Aires**, en el **Pabellón I de Ciudad Universitaria, en Nuñez**.

El equipo tenía más de 4500 válvulas termiónicas (que debían ser reemplazadas en varias oportunidades durante el funcionamiento de la máquina), y memoria de núcleos magnéticos de 4 KWords (de 10 bits). Se constituía de 14 gabinetes de 60cm que tenían las funciones de procesador y memoria, 4 gabinetes con cilindros magnéticos (para un total de 64 KWords de 10 bits), que ocupaban toda una habitación. A esto había que sumarle otros 5 racks en otra habitación que contenían las fuentes de alimentación. Medía en total una 50.000 veces más que un gabinete de computadora moderna.

Carecía de monitor y de teclado. La entrada de instrucciones se hacía con un lector fotoeléctrico de cinta de papel perforado, y los resultados se emitían por una perforadora de cinta a 30 caracteres por segundo, opcionalmente alimentando una teletipo a la velocidad standard de 7 caracteres por segundo. Más adelante se le pudo adaptar un lector de tarjetas perforadas de fabricación nacional, siendo este un método de ingreso de datos más práctico que el original.



Manuel Sadosky en el Instituto de Cálculo trabajando con la computadora Clementina junto a su colega Juan Carlos Angio.
Fotografía de Archivo. Diario Chaco

El lenguaje de programación que utilizaba era Mercury Autocode, especialmente desarrollado para este modelo. Sobre Clementina se creó el primer lenguaje de computación argentino, llamado **COMIC**. Fue creado por **Wilfred Duran** y estaba adaptado a problemas de simulación socio económicos.

La computadora prestó servicios para varias dependencias del Estado, trabajando en cálculos astronómicos (verificación de los cálculos manuales hechos por el astrónomo ítalo-argentino Francisco J. Bobone sobre el pasaje del cometa Halley en 1904), modelos matemáticos de cuencas fluviales y económéticos, desarrollo en computadora del método de camino crítico (CPM), estudios de mecánica del sólido, problemas lingüísticos y problemas estadísticos.

El nombre de Clementina surgió de una canción popular estadounidense **Oh My Darling, Clementine** que venía entre los programas de muestra provistos por Ferranti. La computadora tenía la posibilidad de accionar un parlante ubicado en la consola, lo que permitía generar tonos muy rudimentarios por software. Luego, utilizando dicho parlante se produjeron programas que tocaban tangos.

Clementina siguió funcionando hasta mediados del año 1971, cuando su mantenimiento por falta de piezas se hizo imposible. Sería desmantelada y los restos dispuestos para su eliminación como simples residuos. Tan sólo unos pocos módulos fueron rescatados por personal técnico de la facultad antes de que se los vendiera como chatarra, y aún los conservan como piezas de colección.

Ejercicio 2.2 ¿Qué es la ley de Moore? ¿Se cumple actualmente?

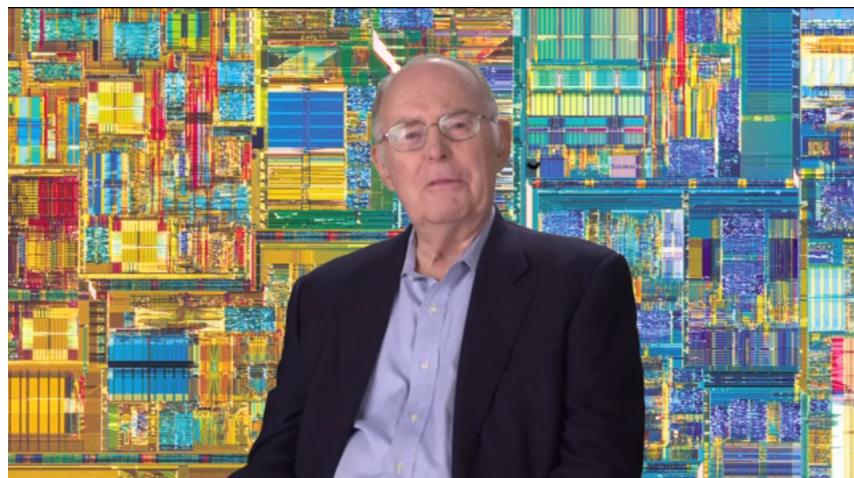
La **ley de Moore** es una ley empírica postulada con el cofundador de Intel **Gordon E. Moore** que expresa que **el número de transistores de un microprocesador se duplica aproximadamente cada dos años**.



Imagen de la computadora Clementina.
Fotografía de Archivo. Facultad de Ciencias Económicas, UBA

La ley fue postulada en 1965, por el joven ingeniero Gordon Moore era director de los laboratorios de **Fairchild Semiconductor**. Él observó en los primeros días de la microelectrónica, una tendencia que definía el mercado de los semiconductores. Su observación anticipaba que la cantidad de circuitos integrados se duplicaría cada año, con la reducción mensurable en costo. Poco después, en 1968, fundaría Intel junto con Robert Noyce.

En 1975, modificaría su propia ley al establecer que la duplicación no se realizaría cada año, sino cada dos años.



El científico y empresario Gordon E. Moore.
Captura del video “Científicos que debes conocer”, creado por la Chemical Heritage Foundation

La consecuencia directa de la ley de Moore es que los precios de los procesadores bajan al mismo tiempo que las prestaciones suben: la computadora que hoy vale 3000 dólares costará la mitad al año siguiente y estará “obsoleta” en dos años.

También funciona como guía para las empresas que producen semiconductores, por lo que es permite planificar a futuro que procesos deben realizar para que su tecnología se mantenga relevante con respecto a la competencia.

En Intel trabajamos duro para asegurarnos de que la ley de Moore continúe guiando a nuestra industria en el futuro. Ya hemos visualizado los próximos 10 a 15 años de adelantos en nuestros laboratorios de investigación.

Craig Barrett, CEO de Intel Corporation

La ley de Moore sigue vigente en la actualidad, a pesar de que hoy contamos con dispositivos que utilizan procesadores de la más diversa índole, como notebooks, tablets, teléfonos celulares, etc. Sin embargo, el mismo Moore ha dicho que su ley quedará eventualmente obsoleta (en 10 o 15 años) cuando una nueva tecnología reemplace a los semiconductores actuales.

Ejercicio 2.3 ¿De quién son los cables submarinos de Internet?

Hay miles de cables submarinos que son la columna vertebral de Internet. Estos cables, no son de acceso público y gratuito, sino se son privados, y una serie de empresas son dueños de los mismos.

Cuando uno contrata un servicio de Internet hogareño, como **Speedy**, **Fibertel**, **Telecentro**, **Claro**, etc. o incluso 4G, quien brinda el servicio (Conocido como **ISP**, por **Internet Service Provider**), debe poder comunicarse a cualquier lugar del mundo (para permitirte acceder a cualquier lado vía Internet). Para ello, debe utilizar los cables submarinos, y por tanto, debe pagarle a su dueño.

A su vez, a quien le paga nuestro ISP puede ser dueño de algunos cables (Por ejemplo, el que va de Argentina a Estados Unidos) pero no de otros (El que va de Estados Unidos a Europa), y por tanto, puede que este también deba pagar a un tercero para utilizar un cable.

De esta forma, se suele dividir a la columna vertebral de internet (llamada generalmente **Internet Backbone**), en capas (**tiers**). Así, nuestro ISP suele ser un considerado una empresa en el “tier 3”, que utiliza cables de una empresa en el “tier 2”, o de una del “tier 1”. Las del “tier 2” a su vez, utilizan los servicios de una empresa del “tier 1”. Una empresa se considera del “tier 1” cuando no debe pagar a nadie por el tráfico que transmite por los cables.

Así, hay una serie de empresas que son de cierta forma, dueñas finales de los cables que sostienen a internet. Algunas de ellas son **AT&T**, **Verizon**, **CenturyLink**, **Sprint**, **Deutsche Telekom AG**, **GTT Comunications**, **Orange**, **Telefónica**, **Tata Communications**, **Telia Carrier**, **Telecom Italia**, **NTT Communications**, **Liberty Global**, **KPN International**, **PCCW Global** y **Zayo Group**.

Ejercicio 2.4 Sabía que en Argentina se desarrollaron varias distribuciones de Linux. Averigüe el nombre y la historia de algunas de ellas.

Argentina fue lugar de nacimiento de la primera distribución en ser reconocida como totalmente libre por el Proyecto GNU, **Ututo Linux**, creada por **Diego Saravia de la Universidad Nacional**

de Salta en el año 2000. Su nombre remite a una especie de lagartija típica de la región noroeste. Lamentablemente, por falta de financiamiento y gente el proyecto dejó de actualizarse en el 2013. Hoy en día hay pequeños proyectos que tienen la idea de reflotar esta distribución.

Tuquito Linux, otra distribución argentina creada en Tucumán por **Ignacio Díaz, Chris Arenas, y Mauro Torres**. Tuquito remite al nombre con el que se conoce a un insecto de abdomen luminiscente en Tucumán, en general llamado luciérnaga. Tuquito intentó transformarse en una distribución de escritorio nacional, sin éxito. Tras encontrarse software malicioso en los servidores que alojaban a Tuquito, y la comunidad acusó a sus creadores de alojarlo allí de forma intensional. Tras el abandono de sus usuarios el proyecto murió en 2012.

Lihuen es una distribución Linux originalmente basada en GnuLinEx y luego en Debian, desarrollada por la **Facultad de Informática de la Universidad Nacional de La Plata**. El proyecto comenzó en 2004 con la intención de realizar una distribución de Linux especialmente diseñada para la educación en la universidad, incluyendo software pre-instalado y con la posibilidad de funcionar en software medianamente antiguo. El proyecto va hoy en día en su versión 6, y el LINTI, dependiente de la UNLP, es el encargado del mantenimiento y desarrollo del sistema.

Dragora es otro sistema 100% libre, recomendado por la FSF. El sistema fue creado desde cero, resultando en un sistema similar a Slackware, aunque incompatible en algunos aspectos con este.

Existen otros varios que han sido desarrollados por programadores argentinos en conjunto con programadores de otros lados del mundo, como **Musix, Wandoo, RXart, FriceOS o Urli**.



La vaca voladora, mascota de Huayra Linux.

Imagen de enREDando.

Finalmente, la distribución Linux argentina más ampliamente extendida en los últimos años ha sido **Huayra Linux**. Creada por **Educ.ar SE**, una empresa estatal que realizó el sistema en el marco del programa **Conectar Igualdad**, vieniendo el mismo integrado en las computadoras del programa. El sistema operativo tiene por mascota una vaca voladora, haciendo referencia a un chiste interno, en donde decían que las computadoras del programa contarían con un sistema operativo propio “el día que las vacas vuelen”. El sistema incluye herramientas de administración para docentes, permitiéndoles crear aulas virtuales, un reproductor de televisión digital terrestre (TDA), y herramientas para la enseñanza de programación, física, química, matemática, y otras áreas de enseñanza. Tras cancelar el plan conectar igualdad en 2016, el proyecto comenzó un proceso de desfinanciamiento por parte del Estado Nacional, quien era su principal promotor. Actualmente ya no se planifican nuevas versiones, y la situación del soporte a las versiones actuales varía entre escasa y nula.

Ejercicio 2.5 ¿Conoce alguna de las microcomputadoras emblemática? ¿Cuál? Si no conoce ninguna, investigue qué computadoras marcaron esa época. ■

Cuando iniciaron las microcomputadoras eran pocas las empresas que se dedicaban a fabricarlas. Hubieron ciertas marcas que vendieron un gran número de ejemplares y que marcaron a todas las computadoras por venir, además de forjar generaciones enteras de usuarios que aún hoy atesoran estas máquinas, las cuales suelen tener altos valores en el mercado de colección.

Apple desarrolló en 1977 el **Apple II**, su primer computadora fabricada en masa. Diseñada por **Steve Wozniak** el Apple II original marcaría una era de éxito para Apple, y la arquitectura de base de la máquina sería utilizada en diversas generaciones del equipo que se vendería hasta 1992 con modelos como el Apple II Plus, Apple IIe, Apple IIC y el Apple IIGS. También aparecerían cientos de clónicos de estos equipos. Apple descontinuaría los equipos tras reemplazarlos con sus **Macintosh**, que comenzaron a producirse en 1984, pero fueron un fracaso en ventas, logrando que la empresa despida a su fundador original, **Steve Jobs**. Macintosh también tendría varias ediciones, como Macintosh II, Macintosh Plus o Macintosh SE. Las ventas comenzarían a remontar recién en 1990, y en 1994 el sistema evolucionó cambiando completamente la arquitectura.



De izquierda a derecha, una TRS-80, una Commodore 64 y una Apple II.

Imagen del Museo de Historia de las Computadoras.

También en 1977, Commodore, otra gran empresa de microcomputadoras de la época lanzaría la **Commodore PET**, la primer computadora completamente equipada de la compañía, y que marcaría el diseño de toda la gama de computadoras de 8 bits de la empresa, incluyendo **Commodore VIC-20**, su sucesora, la **Commodore 64** y finalmente la **Commodore 128**. La Commodore 64 tenía esa denominación pues contaba solamente con 64KB de memoria RAM (hoy las máquinas tienen en general 128000 veces más memoria). Commodore 64 se transformó en un clásico, pasando a ser la principal competencia de la Apple II. Aunque se seguirían vendiendo hasta 1994, las ventas comenzarían a disminuir, por lo que en 1985 Commodore lanzaría al mercado la **Commodore Amiga**.

En 1977 también aparecía en los mercados la **TRS-80**, o Tandy Radio Shack 80. Tandy, empresa de tecnología dueña de una serie de tiendas de venta de tecnología al menudeo, Radio Shack, decidió diseñar y vender su propio equipo. El mismo se hizo sumamente popular por su escaso bajo costo. El equipo pasaría por diversos modelos y se vendería hasta mediados de los 80.

En Reino Unido salió en 1982 la **ZX Spectrum**, por parte de **Sinclair Research**. La computadora costaba solo 100 libras esterlinas (equivalente al costo de un celular hoy en día). Esto la transformó en la computadora más vendida del reino, dominando los mercados por los años venideros.

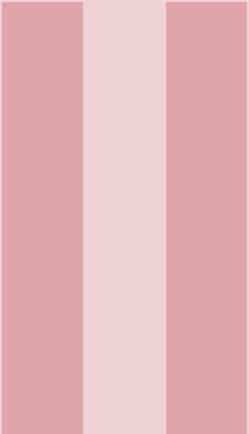
Si bien hay muchísimas otras máquinas iconicas, podemos terminar por mencionar la **IBM PC**, cuyo diseño de periféricos encastrables con ranuras estandarizadas transformó la venta de

hardware, abriendole la puerta a miles de productores más pequeños que podían dedicarse a fabricar componentes especializados en lugar de máquinas enteras.

Ejercicio 2.6 ¿Qué lenguajes de programación conoce? ¿Sabe para qué sirve y qué características tiene?

Hay cientos de lenguajes de programación, y constantemente aparecen nuevos lenguajes. Los lenguajes pueden agruparse en diversas categorías, según distintos criterios. Por ejemplo, podemos agruparlos en **lenguajes de propósito general** (Cuando permiten programar cualquier cosa) o **lenguajes de dominio específico** (Cuando sirven para programar algo en particular, por ejemplo, videojuegos). También se los puede agrupar según que tan bien ese lenguaje permite expresar las ideas del programador, y que tanto puedo olvidarme de como se compone internamente una computadora al usar ese lenguaje. De esta forma, tenemos **lenguajes de alto nivel** (Aquellos que permiten expresar muy bien las ideas, y me permiten olvidarme de la máquina) y de **bajo nivel** (Expresan mejor el funcionamiento interno de la máquina, pero se vuelve más difícil expresar las ideas del programador). Otra posible categorización es mediante paradigmas (Es decir, la forma general en la que se estructura un programa en dicho lenguaje), donde tenemos **lenguajes imperativos** (también llamados **estructurados**), **lenguajes orientados a objetos**, **lenguajes funcionales**, y **lenguajes lógicos**. Además hay lenguajes que entran en más de una de dichas categorías.

Entre los lenguajes más populares y conocidos se encuentran **C, C++, Java, C#, JavaScript, Python, Ruby, Kotlin, Swift, Visual Basic, PHP, Pascal, Pearl, SQL, Go, Smalltalk, Haskell, COBOL**. La agencia TIOBE, genera un listado con los lenguajes “más populares”.



Información

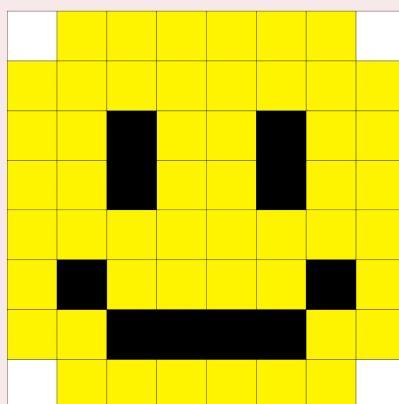
3	Bajo nivel	23
3.3	Actividades		
4	Informática	27
4.5	Actividades		

3. Bajo nivel

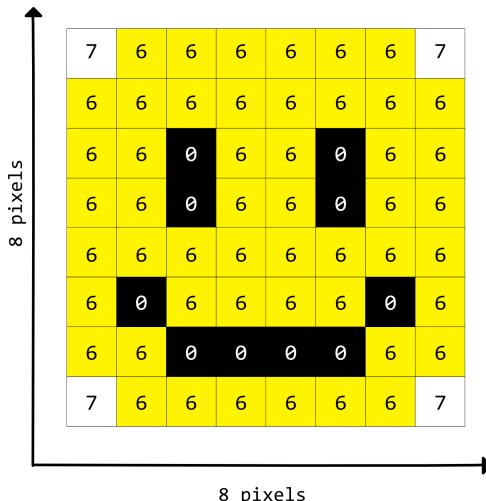
Fotografía artística que representa información.
Fotografía de Pixabay.

3.3 Actividades

Ejercicio 3.1 Dada la imagen a continuación, expresela como un único número utilizando la misma codificación que se aplicó en este capítulo.



En primer lugar, utilizamos la tabla de colores para marcar cada celda con el número correspondiente. También marcamos el ancho y el alto de la imagen.



Ahora podemos eliminar la imagen y quedarnos solo con los números.

```

8 8
7 6 6 6 6 6 6 7
6 6 6 6 6 6 6 6
6 6 0 6 6 0 6 6
6 6 0 6 6 0 6 6
6 6 6 6 6 6 6 6
6 0 6 6 6 0 6
6 6 0 0 0 0 6 6
7 6 6 6 6 6 6 7

```

Y ahora, podemos eliminar el formato para obtener el número que representa dicha imagen.

887666666766666666606606666066066666666606666066600006676666667

Ejercicio 3.2 La tabla mostrada en este capítulo para representar letras como números corresponde a la codificación ASCII. Utilizando esa tabla se pide que codifique la siguiente frase como una serie de números.

SOMOS LO QUE PROGRAMAMOS

El texto se corresponde con el siguiente código:

83 79 77 79 83 32 76 79 32 81 85 69 32 80 82 79 71 82 65 77 65 77 79 83

Ejercicio 3.3 Nuevamente, usando ASCII, se pide ahora que decodifique el siguiente mensaje expresado como una serie de números.

80 82 79 71 82 65 77 79 32 76 85 69 71 79 32 69 88 73 83 84 79

El código debería leer:

PROGRAMO LUEGO EXISTO

Ejercicio 3.4 Si contamos con una computadora con 16 cables para representar nuestros datos, ¿Qué cantidad de números distintos pueden representarse con ellos? ■

Si orestaron atención, cada cable nos permite 2 opciones, con electricidad, o sin ella. Si solo tengo un cable, tengo 2 números posibles. Si tengo 2 cables, puedo tener 4 números posibles representados de diferentes formas (ámbos cables sin electricidad, ámbos con electricidad, el primero con electricidad y el segundo sin, y finalmente el primero sin y el segundo con electricidad).

Así, se obtiene que la cantidad de números posibles depende de la cantidad de cables, siendo equivalente a:

$$2^{\text{cant. cables}} = \text{cant. números distintos} \quad (3.1)$$

En este caso, con 16 cables, tenemos:

$$2^{16} = 65536 \quad (3.2)$$

Es decir, **hay 65536 números distintos que pueden ser representados.**

Ejercicio 3.5 Un chiste de informáticos reza lo siguiente:

“Solo hay 10 tipos de personas en este mundo, los que entienden binario y los que no”

En qué radica la gracia del chiste. ■

Una códificación clásica para números utilizando binario (con dos bits) representa a los números de la siguiente forma:

Número	Binario
0	00
1	01
2	10
3	11

Así, 10 en binario, es 2. El chiste reza entonces que hay solo dos tipos de personas en el mundo, aquellos que entienden binario, y aquellos que no, pero emplea binario para decirlo, haciendo que solo uno de esos dos grupos pueda comprender el chiste.



4. Informática

Placa base de una computadora con sus circuitos impresos.

Fotografía de [Blickpixel](#).

4.5 Actividades

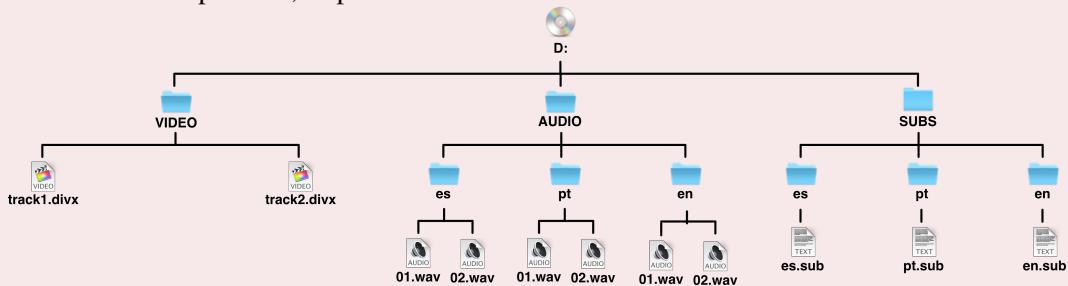
Ejercicio 4.1 Valiéndose de internet, determine para cada uno de los nombres completos de archivo a continuación, de qué formato de archivo se trata (imagen, audio, texto, programa ejecutable, etc.).

- | | |
|------------------------------|----------------------------|
| a) ACDC.mp3 | j) showcase.ppt |
| b) bumblebee.wav | k) cuadernillo.pdf |
| c) 2018_06_15_221653.jpg | l) day_of_the_tentacle.exe |
| d) logo.png | m) LEEME.txt |
| e) index.php | n) library.c |
| f) cv.docx | ñ) configuration.xml |
| g) materias.xls | o) run.py |
| h) subtitles_tbtt_s01e04.zip | p) LEEME.md |
| i) script.sh | q) casa.dwg |

- a) **ACDC.mp3** Un archivo de audio comprimido con codificación MP3.
- b) **LEEME.txt** Un archivo de texto plano
- c) **2018_06_15_221653.jpg** Una imagen fotográfica con compresión JPEG.
- d) **run.py** Un archivo de texto plano con código Python.
- e) **index.php** Un archivo de texto que contiene código en lenguaje de programación PHP.
- f) **cv.docx** Un documento de Microsoft Word en su versión 2007 en adelante.
- g) **materias.xls** Un documento de Microsoft Excel en su versión previo a 2007.
- h) **subtitles_tbtt_s01e04.zip** Un archivo comprimido en formato ZIP.
- i) **script.sh** Un archivo de texto con código de scripting en BASH.
- j) **showcase.odp** Una presentación de diapositivas de Libre Office Impress.
- k) **cuadernillo.pdf** Un documento PDF.
- l) **day_of_the_tentacle.exe** Un archivo ejecutable.
- m) **bumblebee.wav** Un archivo de audio sin comprimir.

- n) **library.c** Un archivo de texto plano contenido código en lenguaje C.
- ñ) **configuration.xml** Un archivo de texto plano con código del lenguaje de marcado XML.
- o) **logo.png** Una imagen o dibujo con compresión PNG.
- p) **LEEME.md** Un archivo de texto plano con código del lenguaje de marcado Markdown.
- q) **casa.dwg** Un archivo de Autodesk AutoCAD.

Ejercicio 4.2 Teniendo en cuenta el siguiente sistema de archivos de Windows que corresponde a un DVD de una película, se pide.



Se pide que escriba las siguientes rutas absolutas:

- Al archivo de subtítulos en inglés (en.sub)
- Al archivo de audio de la pista 02 en español (carpeta es, archivo 02.wav)
- Al archivo de video de la pista 01 (track1.divx)
- Al archivo de audio de la pista 01 en portugués.

- D:\SUBS\en\en.sub
- D:\AUDIO\es\02.wav
- D:\VIDEO\track1.divx
- D:\AUDIO\pt\01.wav

Ejercicio 4.3 Teniendo en cuenta el sistema de directorios del ejercicio anterior y sabiendo qué:

- Todo archivo de subtítulos pesa 500 KB.
- Los archivos de audio de la pista 01 pesan 5 MB.
- Los archivos de audio de la pista 02 pesan 25 MB, menos el de idioma portugués que pesa 27MB.
- El archivo de video de la pista 01 pesa 340 MB.
- El archivo de video de la pista 02 pesa 660 MB.

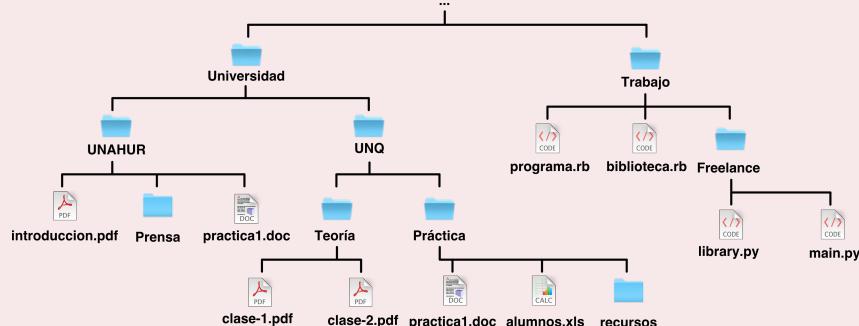
Se pide determine las siguientes:

- Cuanto pesa la carpeta VIDEO
- Cuanto pesa la carpeta SUBS
- Cuanto pesa en total el disco D:

Una carpeta pesa tanto como la suma de los elementos que contiene. El proceso es recursivo, es decir, si una carpeta contiene un archivo y una sub-carpeta, pesará tanto como lo que pesa el archivo y lo que pesan todos los archivos dentro de la sub-carpeta.

- a) $340 \text{ MB} + 660 \text{ MB} = 1000 \text{ MB} = \mathbf{1 \text{ GB}}$.
- b) Cada carpeta en SUBS pesa $500 \text{ KB} + 500 \text{ KB} = 1000 \text{ KB} = 1 \text{ MB}$. Las tres carpetas juntas pesan entonces **3 MB**.
- c) Son entonces 1000 MB de la carpeta VIDEO, sumado a los 3 MB de la carpeta de SUBS. A eso se le debe sumar la carpeta AUDIO, donde cada sub-carpeta pesa $25 \text{ MB} + 25 \text{ MB} = 50 \text{ MB}$, con la excepción de la carpeta pt que pesa $25 \text{ MB} + 27 \text{ MB} = 52 \text{ MB}$. El total de AUDIO es entonces $50 \text{ MB} + 50 \text{ MB} + 52 \text{ MB} = 152 \text{ MB}$. El total del disco es entonces: $1000 \text{ MB} + 3 \text{ MB} + 152 \text{ MB} = \mathbf{1155 \text{ MB}}$.

Ejercicio 4.4 Teniendo en cuenta la siguiente porción de un sistema de archivos de un sistema Linux:



Describa la ruta absoluta a las carpetas que se encuentren vacías, asumiendo que la carpeta superior es la raíz del sistema. ■

Solo se encuentran vacías las carpetas “recursos” y “prensa”. Las rutas asumiendo que el inicio es la raíz del sistema son:

/Universidad/UNAHUR/Prensa

/Universidad/UNQ/Práctica/recursos

Ejercicio 4.5 Utilizando el mismo sistema de archivos del ejercicio anterior, escriba las siguientes rutas relativas:

- a) Desde la carpeta más externa de la jerarquía hasta el archivo main.py
- b) Desde la carpeta más externa de la jerarquía hasta el archivo alumnos.xls
- c) Desde la carpeta UNQ hasta el archivo clase-2.pdf
- d) Desde la carpeta UNAHUR al archivo introducion.pdf
- e) Desde la carpeta UNQ al archivo introducion.pdf
- f) Desde la carpeta Teoría en UNQ al archivo programa.rb

- a) Trabajo/Freelance/main.py
- b) Universidad/UNQ/Práctica/alumnos.xls
- c) Teoría/clase-2.pdf
- d) introducion.pdf
- e) ../../UNAHUR/introducion.pdf
- f) ../../Trabajo/programa.rb

Ejercicio 4.6 Un programa “hola mundo” consiste en código que simplemente imprime en la pantalla las palabras “hola mundo”. Este tipo de programa se ha hecho popular como una forma de poder dar a los programadores una vista mínima y rápida de la sintaxis de un lenguaje, y en cursos que se enfocan en la enseñanza de lenguajes de programación en lugar de sus conceptos, es muy común encontrarlo como una de las primeras muestras de código o actividades a realizar.

Se pide entonces que busque en internet el código de un programa “hola mundo” para los siguientes lenguajes de programación. Puede encontrar una gran colección en el sitio <http://helloworldcollection.de>

- a) Python
- b) Lisp
- c) C
- d) Java
- e) Assembly ARM
- f) Assembly z80, Console

¿Qué reflexiones puede realizar después de ver los diferentes códigos? ■

a) Python:

```
1 # Hola mundo en Python 2
2 print "Hola mundo"
```

b) Lisp:

```
1 ;;=; Hola mundo en Common Lisp
2 (print "Hola mundo")
```

c) C:

```
1 /* Hola mundo en C */
2 #include <stdio.h>
3 #include <stdlib.h>
4
5 int main(void) {
6     // La función main es el punto
7     // de entrada del programa.
8     puts("Hola mundo");
9     return EXIT_SUCCESS;
10 }
```

d) Java:

```

1  /* Hola mundo en Java */
2  // Mandatorio, todo el código debe estar
3  // siempre en una clase
4  class HelloWorld {
5      // public static void main es el punto
6      // de entrada del programa.
7      public static void main( String args[] ) {
8          // System.out contiene el stream de salida
9          // al cual se le puede enviar el mensaje println
10         System.out.println( "Hola mundo" );
11     }
12 }
```

d) Assembly ARM

```

1  /*
2      Hola mundo en Assembler para procesadores
3      ARM (Dispositivos Android)
4  */
5 .data
6
7 msg:
8     .ascii      "Hola Mundo\n"
9 len = . - msg
10
11 .text
12
13 .globl _start
14 _start:
15     mov      %r0, $1
16     ldr      %r1, =msg
17     ldr      %r2, =len
18     mov      %r7, $4
19     swi      $0
20     mov      %r0, $0
21     mov      %r7, $1
22     swi      $0
```

e) Assembly z80, Consola

```

1 ; Este es un programa "Hola mundo" para procesadores Z80 y
2 ; TMS9918 / TMS9928 / TMS9929 / V9938 o V9958 VDP.
3 ; Eso significa que debería funcionar en SVI, MSX,
4 ; Colecovision, Memotech, y otras computadoras hogareñas
5 ; y consolas de videojuegos basadas en el procesador Z80.
6 ;
7 ; Como no sabemos en que sistema va a funcionar, no sabemos
8 ; donde está ubicada la memoria RAM, por lo que no podemos
9 ; utilizar stack en este programa.
```

```
10 ;  
11 ; Esta versión de "Hello World" fue escrita por  
12 ; Timo "NYYRIKKI" Soilamaa  
13 ; 17.10.2001  
14 ;  
15 ;-----  
16 ; Configure esta parte:  
17  
18 DATAP: EQU #98 ; VDP Data port #98 funciona en todos los  
19 ; modelos MSX (TMS9918/TMS9929/V9938 or V9958)  
20 ; #80 funciona en SVI  
21 ; (para otras plataformas debería ver el manual y cargar esto)  
22  
23 CMDP: EQU #99 ; VDP Command port #99 funciona en todos los  
24 ; modelos MSX (TMS9918/TMS9929/V9938 or V9958)  
25 ; #81 funciona en SVI  
26 ; (para otras plataformas debería ver el manual y cargar esto)  
27 ;-----  
28 ; El programa comienza acá:  
29  
30 ORG 0      ; El procesador Z80 comienza acá cuando se enciende  
31 DI          ; No sabemos como funcionan las interrupciones en este  
32           ; sistema, por lo que las deshabilitamos.  
33  
34 ; Configuremos la dirección de escritura de VDP a #0000  
35 XOR A  
36 OUT (CMDP),A  
37 LD A,#40  
38 OUT (CMDP),A  
39  
40 ; Ahora limpiemos los primeros 16Kb de memoria VDP  
41 LD B,0  
42 LD HL,#3FFF  
43 LD C,DATAP  
44 CLEAR:  
45 OUT (C),B  
46 DEC HL  
47 LD A,H  
48 OR L  
49 NOP      ; Esperemos ocho ciclos de reloj, solo en caso de  
50           ; que el VDP no sea lo suficientemente rápido  
51 NOP  
52 JR NZ,CLEAR  
53  
54 ; Ahora es momento de configurar los registros del VDP:  
55 ;-----  
56 ; Registro 0 a #0  
57 ;  
58 ; Poner el modo de selección del bit M3 (quizás también  
59 ; M4 y M5) en cero y deshabilitar el video externo e  
60 ; interrupciones horizontales  
61 LD C,CMDP  
62 LD E,#80  
63  
64 OUT (C),A  
65 OUT (C),E
```

```
66 ; -----
67 ; Registro 1 a #50
68 ;
69 ; Selecciona el modo de 40 columnas,
70 ; habilita la pantalla y deshabilita las
71 ; interrupciones verticales
72
73 LD A ,#50
74 INC E
75 OUT (C),A
76 OUT (C),E
77 ; -----
78 ; Registro 2 a #0
79 ;
80 ; Pone la tabla de patrones en #0000
81
82 XOR A
83 INC E
84 OUT (C),A
85 OUT (C),E
86 ; -----
87 ; Registro 3 es ignorado en modo 40 columnas
88 ; ya que no se requiere una tabla de colores
89 ;
90 INC E
91 ; -----
92 ; Registro 4 a #1
93 ; Poner el patrón de generación de tablas en #800
94
95 INC A
96 INC E
97
98 OUT (C),A
99 OUT (C),E
100 ; -----
101 ; Registros 5 (Atributos de sprites) y 6 (Patrones de
102 ; sprites) son ignorados en modo de 40 columnas
103 ; ya que este modo no tiene sprites
104
105 INC E
106 INC E
107 ; -----
108 ; Registro 7 a #F0
109 ; Pone los colores en texto blanco sobre fondo negro
110
111 LD A ,#F0
112 INC E
113 OUT (C),A
114 OUT (C),E
115 ; -----
116
117 ; Ponemos la dirección de escritura del VDP en #808
118 ; de forma de poder escribir los caracteres en memoria
119 ; (No hace falta escribir ESPACIO, ya es un carácter blanco)
120 LD A,8
121 OUT (C),A
```

```
122 LD A ,#48
123 OUT (C),A
124
125 ; Copiemos el juego de caracteres
126 LD HL,CHARS
127 LD B, CHARS-END-CHARS
128 COPYCHARS:
129 LD A,(HL)
130 OUT (DATAP),A
131 INC HL
132 NOP      ; Esperemos ocho ciclos de reloj, solo en caso de
133           ; que el VDP no sea lo suficientemente rápido
134 NOP
135 DJNZ COPYCHARS
136
137 ; Ponemos la dirección de escritura al comienzo de la tabla
138 XOR A
139 OUT (C),A
140 LD A,#40
141 OUT (C),A
142
143 ; Ponemos los caracteres en pantalla
144 LD HL,ORDER
145 LD B,ORDER-END-ORDER
146 COPYORDER:
147 LD A,(HL)
148 OUT (DATAP),A
149 INC HL
150
151 JR OVERNMI
152 NOP
153 NOP
154
155 ; Aquí está la dirección #66, es la entrada para NMI
156 RETN      ; Volver de NMI
157
158 OVERNMI:
159 DJNZ COPYORDER
160
161 ; Fin
162 HALT
163
164 ; Juego de caracteres:
165 ; -----
166 ORDER:
167 DEFB 1,2,3,3,4,0,5,4,6,3,7
168 ORDER-END:
169
170 CHARS:
171
172 ; H
173 DEFB %10001000
174 DEFB %10001000
175 DEFB %10001000
176 DEFB %11111000
177 DEFB %10001000
```

```
178 DEFB %10001000
179 DEFB %10001000
180 DEFB %00000000
181 ; e
182 DEFB %00000000
183 DEFB %00000000
184 DEFB %01110000
185 DEFB %10001000
186 DEFB %11111000
187 DEFB %10000000
188 DEFB %01110000
189 DEFB %00000000
190 ; l
191 DEFB %01100000
192 DEFB %00100000
193 DEFB %00100000
194 DEFB %00100000
195 DEFB %00100000
196 DEFB %00100000
197 DEFB %01110000
198 DEFB %00000000
199 ; o
200 DEFB %00000000
201 DEFB %00000000
202 DEFB %01110000
203 DEFB %10001000
204 DEFB %10001000
205 DEFB %10001000
206 DEFB %01110000
207 DEFB %00000000
208 ; W
209 DEFB %10001000
210 DEFB %10001000
211 DEFB %10001000
212 DEFB %10101000
213 DEFB %10101000
214 DEFB %11011000
215 DEFB %10001000
216 DEFB %00000000
217
218 ; r
219 DEFB %00000000
220 DEFB %00000000
221 DEFB %10110000
222 DEFB %11001000
223 DEFB %10000000
224 DEFB %10000000
225 DEFB %10000000
226 DEFB %00000000
227 ; d
228 DEFB %00001000
229 DEFB %00001000
230 DEFB %01101000
231 DEFB %10011000
232 DEFB %10001000
233 DEFB %10011000
```

```

234 DEFB %01101000
235 DEFB %00000000
236 chars_end:

```

Podemos ver que cada código es distinto, eso queda claro desde el primer momento. Sin embargo, también podemos ver aquí que hay lenguajes que son de alto nivel, y de bajo nivel. En los lenguajes de alto nivel, como Python o Lisp, el código refleja más claramente la intención del programador. En los lenguajes intermedios, como C, el código sigue reflejando de alguna forma la intención, pero empiezan a aparecer elementos de sintaxis que se vuelven obligatorios, y que no aportan a lo que uno quiere lograr. Java, por ser un lenguaje orientado a objeto, y ser fuertemente tipado, solicita un montón de pasos protocolares simplemente para poder escribir algo en pantalla (claramente no es un lenguaje pensado para tal fin). Los Assembly ya nos resultan esotéricos, y uno debe conocer muy bien el procesador, los chips de los equipos, y entender como se carga el programa en la memoria. En ARM, si bien extraño, uno puede ver un “Hola mundo” escrito en algún lado, en el de z80, solo aparecen ceros y unos en una tabla que refleja los pixeles de la pantalla (cuales estarán prendidos y cuales apagados), indicando el texto “hello world”.

Una cosa interesante en el caso de Java, pero mucho más patente en el caso del z80, es que los programadores dejan mensajes para otros seres humanos, y que no son parte del programa mismo. Estos mensajes se llaman “comentarios”. El código del z80 es tan extraño a lo que uno quiere hacer realmente, que requiere una gran cantidad de comentarios para reflejar la idea que el programador tenía en la cabeza. Si quitamos los comentarios de ese programa, sería muy difícil (casi imposible) comprender qué ese programa es un “hola mundo”. Es importante que el código transmita ideas, y no solo que funcione, si el código funciona pero no se entiende, no podrá ser compartido. Peor aún, no podrá ser mantenido en el tiempo, ni siquiera por el mismo programador, pues este olvidará qué es lo que escribió con el tiempo.

Ejercicio 4.7 El sitio web <https://alternativeto.net> permite a sus usuarios buscar un programa por nombre, y brinda alternativas a ese programa con funcionalidades equivalentes o similares. Dentro de los resultados, se puede filtrar por sistema operativo y por el tipo de licencia (comercial, gratuita o libre).

Se pide que busque al menos una alternativa que sea software libre y/o open source para cada uno de los siguientes programas:

- a) Windows 10
- b) Microsoft Office Suite
- c) Adobe Acrobat Reader
- d) Internet Explorer
- e) Adobe Photoshop
- f) Autodesk AutoCAD
- g) Age of Empires II

a)

En general cualquier distribución de Linux o de BSD es una alternativa. Algunas están más asentadas y por tanto, son de mejor calidad que otras, pero pueden encontrarse alternativas, no solo a Windows 10, sino a cualquier otra versión de Windows, con distribuciones pensadas para correr en máquinas antiguas o de baja potencia, en tablets, etc.

Algunas de las alternativas son:

1. **Ubuntu**
2. **Linux Mint**
3. **Debian**
4. **Fedora**
5. **ZorinOS**

b)

1. **LibreOffice**

LibreOffice surge como un derivado de OpenOffice. A diferencia de este último utiliza una licencia GPL, por lo que se código no puede ser objeto de productos comerciales. Por eso, se ha transformado en la suite por defecto de la mayoría de las distribuciones Linux.

2. **Apache OpenOffice**

OpenOffice nació como una suite ofimática libre de la mano de la empresa Sun Microsystems. Cuando Sun fue adquirida por Oracle Corporation, los usuarios temieron que Oracle cerrara la rama de desarrollo de OpenOffice, algo que efectivamente sucedió. La fundación Apache se hizo cargo de OpenOffice, pero en el caótico proceso nació también LibreOffice.

3. **Calligra**

Calligra es una suite compuesta por aplicaciones tanto de ofimática, como de diseño gráfico. Es un conjunto de aplicaciones de KDE, por lo que viene instalado en muchas distribuciones Linux que utilizan ese entorno.

c)

En general cada distribución de Linux incluye uno. Hoy en día también Google Chrome y Firefox incluyen lector de PDF integrado, y su código es libre.

1. **Sumatra PDF**

Diseñado para Windows como una alternativa libre.

2. **Evince**

Lector por defecto en los sistemas Linux que utilizan el escritorio GNOME.

3. **Okular**

Lector por defecto en los sistemas Linux que utilizan el escritorio GNOME.

d)

1. **Mozilla Firefox**

Firefox existe desde hace ya muchos años como competencia de IE. Nació como una continuación del proyecto comercial Netscape, el cuál abrió su código tras dar quiebra la empresa. Hoy en día es una de las alternativas más elegidas entre los usuarios de software libre.

2. **Google Chromium**

Chromium es el proyecto libre de donde se produce Google Chrome (que agrega unas pocas características no libres al mismo). Surgió en Google con la intención de fomentar el uso de la web moderna y plantar una seria competencia a IE

También existen muchas otras alternativas más pequeñas y menos populares.

e)

1. **GIMP**

Es “él” editor libre. Posee gran cantidad de características avanzadas y una gran base de usuarios, tutoriales, foros, etc.

2. **Krita**

Krita es el editor por defecto en sistemas de escritorio KDE.

3. **Pinta**

Una alternativa libre en Windows, diseñada a partir del código de Paint.NET, posee características avanzadas, y una buena base de usuarios.

f)

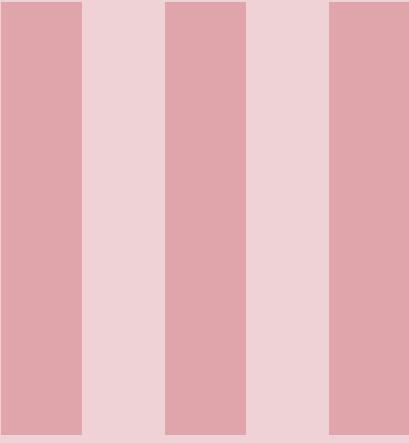
En general, si bien hay alternativas libres, están muy atrás con respecto a la alternativa privativa en este área. Algunas opciones son:

1. **FreeCAD**
2. **LibreCAD**
3. **OpenSCAD**

g)

Existen diferentes alternativas de juegos similares a AoE II, muchas con baja calidad. Cabe mencionar el proyecto, aún en desarrollo pero con una excelente calidad y potencial, 0 A.D.

1. **0 A.D.**
2. **MegaGlest**
3. **Widelands**



Lógica



5. Introducción a la lógica

Red Lógica de un Enrutador con LEDs.

Fotografía de tschoenemeyer.

5.1 Actividades

Ejercicio 5.1 Dados los siguientes razonamientos, indique si se tratan de razonamientos deductivos o inductivos.

- Dado que las versiones más nuevas de los navegadores soportan HTML5, y dado que el Microsoft Edge es la última versión del navegador de dicha empresa, se sigue que el Microsoft Edge soporta HTML5.
- Mi primo compró una computadora aquí y tenía instalado Windows 10. Mi hermano compró una computadora aquí y tenía instalado Windows 10. Mi tío compró una computadora aquí y tenía instalado Windows 10. Si compro una computadora aquí tendrá instalado Windows 10.
- Maria se inscribió el jueves y solo tenía habilitadas las materias del CI. Pablo se inscribió el jueves y solo tenía habilitadas las materias del CI. Juan se inscribió el jueves y solo tenía habilitadas las materias del CI. Por lo tanto, todo alumno que se inscriba el jueves tiene solo habilitadas las materias del CI.

■

Si bien basta con decir si es deductivo o inductivo, podemos además agregar qué tipo de razonamiento inductivo es, ya que podemos encontrar de dos tipos. En un razonamiento inductivo la conclusión puede ser un nuevo caso particular, que lo hace inductivo por analogía, o una generalización (es decir, afirmar que todos los casos futuros cumplirán una característica), lo cual lo hace inductivo por enumeración.

De esta forma, las respuestas son:

- Razonamiento deductivo.
- Razonamiento inductivo (por analogía).
- Razonamiento inductivo (por enumeración).

Ejercicio 5.2 Considerando las siguientes preguntas como básicas:

- ¿hay harina?
- ¿hay manteca?
- ¿hay aceite?
- ¿hay agua?
- ¿hay huevos?
- ¿hay yerba?
- ¿hay chocolate?
- ¿hay azúcar?

Se le pide que, utilizando las mencionadas como preguntas básicas^a y las conectivas lógicas vistas, exprese las preguntas generales a continuación^b:

- a) **¿hay para hacer una torta?** (Una torta requiere harina, huevos y manteca)
- b) **¿hay para hacer huevos fritos?** (Requiere huevos y aceite)
- c) **¿hay para hacer huevos duros?** (Requiere huevos y agua)
- d) **¿Puedo almorzar huevos?** (Ya sean duros o fritos)
- e) **¿hay para hacer una torta de chocolate?** (Idéntico a una torta, más chocolate)
- f) **¿Solo se puede tomar mate amargo?** (Cuando se puede tomar mate, es decir, hay agua y yerba, pero no hay azúcar)
- g) **¿No hay nada para el mate?** (Cuando se puede tomar mate, pero no hay torta de ningún tipo)

^aRecuerde que puede elaborar preguntas auxiliares que le ayuden a solucionar las pedidas

^bSe espera escriba algo de la forma “*¿hay manzanas? \wedge ¿hay bananas?*”

Las respuestas son:

- a) ***¿Hay harina? y ¿Hay huevos? y ¿Hay manteca?***
- b) ***¿Hay huevos? y ¿Hay aceite?***
- c) ***¿Hay huevos? y ¿Hay agua?***
- d) Acá comienza a volverse más divertido. El truco consiste en reutilizar preguntas para las cuales ya dimos una equivalencia en términos de las base, como *¿Hay para hacer huevos fritos?*. De esta forma, la mejor respuesta sería:

¿Hay para hacer huevos fritos? o ¿Hay para hacer huevos duros?

Sin embargo es probable que no sea la respuesta de todos en una primer instancia. La primer aproximación tiende a ser emplear únicamente preguntas base, y por tanto se obtienen respuestas como:

(¿Hay huevos? y ¿Hay aceite?) o (¿Hay huevos? y ¿Hay agua?)

O también, quienes se percatan de que siempre deben haber huevos, que realizan algo como:

¿Hay huevos? y (¿Hay aceite? o ¿Hay agua?)

Luego de explicar que la mejor opción es reutilizar las preguntas ya analizadas, se espera respondan las siguientes siempre reutilizando.

- e) Este punto tiene la particularidad de que combinamos una pregunta ya realizada en otro ejercicio con una pregunta base. Es perfectamente posible hacer esto ya que todas son preguntas que portan un valor de verdad.

¿Hay para hacer una torta? y ¿Hay chocolate?

- f) En este caso es conveniente primero definir una nueva pregunta, ya que la idea de que se pueda tomar mate, es algo independiente de si se puede tomarlo amargo o no. Conviene entonces desglosar la pregunta, que de otra forma resulta compleja y con muchas partes, en preguntas más pequeñas. El resultado sería:

¿Hay mate? = ¿Hay yerba? y ¿Hay agua?

Y la pregunta original que queríamos definir sería entonces:

¿Hay mate? y no ¿Hay azúcar?

- g) Acá surge la ventaja de haber hecho *¿Hay mate?*, ya que podemos volver a usarla en este punto, resultando en:

¿Hay mate? y (no ¿Hay para hacer una torta de chocolate? y no ¿Hay para hacer una torta?)

Aunque, como si no hay para una torta común, tampoco hay para una de chocolate, varios podrían percatarse de que basta con:

¿Hay mate? y no ¿Hay para hacer una torta?**Ejercicio 5.3** Considerando las siguientes preguntas básicas y sus respuestas asociadas:

- ¿Plutón es un planeta? **FALSO**
- ¿Marte es un planeta? **VERDADERO**
- ¿Marte es un satélite? **FALSO**
- ¿Deimos es un satélite? **VERDADERO**
- ¿Ganímedes es un satélite? **VERDADERO**
- ¿Eros es un planeta? **FALSO**
- ¿Eros es un satélite? **FALSO**

Se le pide que, utilizando las mencionadas como preguntas básicas y las conectivas lógicas vistas, exprese las preguntas generales a continuación y determine el valor de verdad de cada una.

- a) **¿Son Marte y Plutón planetas?**
- b) **¿Es Marte un planeta o es Plutón un planeta?**
- c) **¿Es Marte un planeta o un satélite?**
- d) **¿Es cierto que Marte es un planeta y Plutón no lo es?**
- e) **¿Es cierto que Ganímedes y Eros son satélites?**
- f) **¿Es cierto que Eros no es un satélite, pero Deimos si lo es?**
- g) **¿Es Deimos un satélito o es cierto que Marte es un planeta?**
- h) **¿Es Eros un satélite o es cierto que es un planeta?**

- a) **¿Son Marte y Plutón planetas? FALSO**

Para determinarlo que es falso, la forma de solución es identificar que la pregunta dada es equivalente a:

¿Marte es un planeta? \wedge ¿Plutón es un planeta?

Y por tanto, considerando las respuestas a las preguntas base que tenemos, es equivalente a:

VERDADERO \wedge FALSO

De esta forma, y por la semántica de la conjunción, sabemos que si uno de los elementos es falso, la conjunción completa es falsa. Por lo que se obtiene que el resultado es:

FALSO

En los siguientes items presentaremos el resultado y el análisis línea a línea, sin explicación detallada de los pasos, pero todos siguen el mismo proceso.

- b) **¿Es Marte un planeta o es Plutón un planeta?**
- c) **¿Es Marte un planeta o un satélite?**
- d) **¿Es cierto que Marte es un planeta y Plutón no lo es?**
- e) **¿Es cierto que Ganímedes y Eros son satélites?**
- f) **¿Es cierto que Eros no es un satélite, pero Deimos si lo es?**
- g) **¿Es Deimos un satélito o es cierto que Marte es un planeta?**
- h) **¿Es Eros un satélite o es cierto que es un planeta?**

Ejercicio 5.4 Considerando las siguientes preguntas como básicas:

- | | |
|---|---|
| <ul style="list-style-type: none"> ■ ¿hay ejercito enemigo al Norte? ■ ¿hay ejercito enemigo al Este? ■ ¿hay ejercito enemigo al Sur? ■ ¿hay ejercito enemigo al Oeste? | <ul style="list-style-type: none"> ■ ¿hay ejercito aliado al Norte? ■ ¿hay ejercito aliado al Este? ■ ¿hay ejercito aliado al Sur? ■ ¿hay ejercito aliado al Oeste? |
|---|---|

Se le pide que, utilizando las mencionadas como preguntas básicas^a y las conectivas lógicas vistas, exprese las preguntas generales a continuación^b:

- a) **¿Se está amenazado?** (Cuando hay ejercito enemigo en alguna dirección)
- b) **¿Se está libre de peligro?** (Cuando no hay ejercitos enemigos en ninguna dirección)
- c) **¿Se tiene apoyo?** (Cuando hay un ejercito aliado en alguna dirección)
- d) **¿Se está hasta las manos?** (Cuando no hay apoyo y se está amenazado)
- e) **¿Se puede neutralizar alguna amenaza?** (Cuando hay un ejercito enemigo en alguna dirección, pero también hay un ejercito aliado allí)
- f) **¿Se puede neutralizar todas las amenazas?** (Cuando se puede neutralizar en todas las direcciones)

^aRecuerde que puede elaborar preguntas auxiliares que le ayuden a solucionar las pedidas

^bSe espera escriba algo de la forma “¿hay manzanas? \wedge ¿hay bananas?”

Ejercicio 5.5 Considerando las siguientes preguntas simples:

- ¿Se cumple P?
- ¿Se cumple Q?
- ¿Se cumple R?

Se le pide que analice que valuaciones darán verdadero y cuales falso para las siguientes preguntas compuestas^a. Además se le pide determine cuales de ellas se tratan de tautologías, cuales de contradicciones y cuáles de contingencias.

- a) **¿Se cumple P? \wedge \neg ¿Se cumple P?**

- b) $\text{¿Se cumple P?} \vee \neg \text{¿Se cumple P?}$
- c) $\text{¿Se cumple P?} \wedge \text{¿Se cumple Q?}$
- d) $\text{¿Se cumple P?} \wedge (\text{¿Se cumple Q?} \vee \text{¿Se cumple R?})$
- e) $\text{¿Se cumple P?} \vee \text{¿Se cumple Q?}$
- f) $\neg \text{¿Se cumple P?} \vee (\text{¿Se cumple Q?} \wedge \text{¿Se cumple R?})$
- g) $\neg \text{¿Se cumple P?} \vee \neg (\text{¿Se cumple Q?} \vee \text{¿Se cumple R?})$

“Si pensar en preguntas como “¿Se cumple P?” le resulta confuso, piense en reemplazarlas por preguntas que le sean más familiares, como “¿Se cumple que el lobo vive en el bosque?”. Lo interesante del ejercicio es que la pregunta en cuestión no es realmente relevante, sino la forma de la pregunta compuesta.”

Ejercicio 5.6 Sabiendo que las siguientes preguntas compuestas evalúan todas a **VERDADERO**

- ¿La palmera está creciendo torcida? y ¿Es cierto que el árbol no dio paltas este año?
- ¿Es cierto que no hay flores en el cantero? o ¿El árbol dio paltas este año?

Se pide que responda las siguientes preguntas simples

- a) ¿El árbol dio paltas este año?
- b) ¿Hay flores en el cantero?
- c) ¿La palmera está creciendo torcida?

Ejercicio 5.7 Dadas las siguientes preguntas base y sus respuestas:

- ¿Es Microsoft el creador de Windows? **VERDADERO**
- ¿Es Microsoft el creador de GNU? **FALSO**
- ¿Es Richard Stallman el creador de Windows? **FALSO**
- ¿Es Richard Stallman el creador de GNU? **VERDADERO**

Analice las preguntas a continuación, detecte las conectivas involucradas y reemplantee las preguntas para que estén elaboradas en términos de las preguntas dadas. Luego, analice el valor de verdad de las mismas.

- a) ¿Es cierto que no es cierto que Microsoft es el creador de GNU?
- b) ¿Es cierto que Richard Stallman es el creador de Windows o de GNU?
- c) ¿Es cierto que Microsoft no es el creador de GNU pero que Richard Stallman si lo es?
- d) ¿Es cierto que GNU no es del mismo creador que Windows?

Ejercicio 5.8 Se ha encontrado vida en otro planeta, y se ha decidido nombrar a los animales encontrados como “Woofle”, “Brlfks” y “Morlock”.

Cada animal tiene sus características distintivas (pueden ser grandes o chicos, con o sin pelo, acuáticos o terrestres).

Si las respuestas a todas las preguntas siguientes son **VERDADERO**, enumere qué características tiene cada animal. Para ello, realice la tabla de verdad de cada pregunta y analice las respuestas a las preguntas base en las valuaciones verdaderas.

- ¿El Woofle tiene pelo y no es acuático?
- ¿El Morlock es terrestre y es grande, y el Brlfks es acuático?

- ¿El Brlfks es pequeño o el Morlock es pequeño?
- ¿El Woofle no es grande o el Brlfks es grande? ¿Y es cierto que el Morlock no es pequeño?
- ¿El Woofle tiene pelo y el Brlfks no? ¿O es cierto que el Brlfks no tiene pelo y el Morlock tiene pelo?

■