

CIENCIAS DE LA  
**COMPUTACIÓN**  
PARA EL AULA  
Anexo para docentes

**1º CICLO SECUNDARIA**

# ANEXO I

---

## INTERNET

### SECUENCIA DIDÁCTICA 1

INTERNET, COSA DE CABLES Y PAQUETES  
Un mundo interconectado... ¿cómo?  
¿Cómo viaja la información?

---

### SECUENCIA DIDÁCTICA 2

¡ME SIGUE UNA NUBE!  
¿Y la nube dónde está?  
La libreta de direcciones

---

### SECUENCIA DIDÁCTICA 3

CIUDADANÍA DIGITAL  
La llegada del hombre a la Luna  
Información ¿Personal?

---

### MODELO DE EVALUACIÓN

ARMAMOS UNA RED...PERO DE  
CONCEPTOS

A diario escuchamos hablar de Internet o de “la *nube*” como algo que existe, aunque no sabemos dónde se encuentra. Parece estar en todos lados y se nos hace cada vez más difícil imaginar la vida sin su ayuda. Nos permite solucionar muchas de nuestras tareas cotidianas: escuchar la última canción de nuestra cantante favorita, mirar un capítulo de la serie que nos atrapa, hacer un *backup* de todas nuestras fotos y videos o editar colaborativamente un documento. Los dispositivos móviles que usamos ya vienen conectados a Internet, los televisores “inteligentes” también y poco a poco también lo están haciendo los electrodomésticos más clásicos como las heladeras o los lavarropas.

A pesar de esto, rara vez nos hemos cuestionado sobre este fenómeno: ¿Dónde está físicamente la información que guardamos en la *nube*? ¿Tiene dueños Internet? ¿Puedo crear mi propia *nube*? ¿Qué costos tiene y quién los paga? ¿Alguien puede ver los sitios por los que navego en Internet? A lo largo de este capítulo intentaremos motivar esas preguntas, para comenzar a esbozar algunas respuestas.

## Secuencia didáctica 1

## Internet, cosa de cables y paquetes

Internet poco a poco va convirtiéndose en la principal fuente de comunicación y de acceso a la información, y por eso resulta más que necesario comprender algunos principios básicos de su funcionamiento.

Las computadoras interconectadas conforman redes y, cuando hablamos de **Internet**, nos referimos a un conjunto de esas redes, descentralizado y con alcance mundial. A medida que se vuelven predominantes las tecnologías inalámbricas, como el WiFi o las redes móviles, se pierde de vista el carácter concreto de Internet: el grueso de las comunicaciones se da mediante cables, que tienen dueños, ocupan lugar, atraviesan territorios, involucran grandes cantidades de tecnología y cuestan mucho dinero. La comunicación a través de estos cables no es tan directa como solemos pensar. Cuando enviamos un mensaje a través de Internet da la impresión de que la comunicación es instantánea. Sin embargo, ese mensaje se descompone en muchos fragmentos que viajan por muchas computadoras a gran velocidad, para luego recomponerse al llegar al destino. El grupo de reglas que regula las comunicaciones se denomina protocolo. El conjunto de protocolos TCP/IP es el utilizado para las comunicaciones de Internet.

Es importante entender cómo se transportan los mensajes por Internet para comprender cómo se llevan a cabo las comunicaciones y cuáles son sus potencialidades y vulnerabilidades.

### Objetivos

- Comprender las nociones básicas del funcionamiento de las redes de computadoras.
- Reconocer la forma en que computadoras de distintos países se conectan a través de Internet.
- Identificar las principales características de los protocolos de comunicación TCP e IP.

## Actividad 1

### Un mundo interconectado... ¿cómo?

#### Objetivo

- Reconocer a Internet como una red mundial de computadoras.  
Identificar la existencia de cables submarinos transoceánicos y su rol dentro de la infraestructura física para garantizar el funcionamiento de Internet.

#### Modalidad de trabajo

En grupos de 4 personas, y la clase completa para el intercambio final.

#### Materiales

- Pizarrón, proyector y parlantes.

### Desarrollo

En esta actividad buscaremos una reflexión grupal para construir el concepto de Internet y discutir sobre las formas en que viaja la información en la red. Para indagar sobre los conocimientos e ideas previas de los estudiantes y guiar los intercambios de la clase utilizaremos una serie de preguntas y un video disparador.

A lo largo de toda la actividad intentaremos reforzar la idea de Internet como una red de computadoras interconectadas, su alcance mundial y la variedad de los medios por los que viaja la información: las conexiones locales, a través del aire o cables (Wi-Fi, conexión telefónica o fibra óptica) y las conexiones entre países o regiones a través de grandes cables, en su mayoría submarinos, y en menor medida a través de satélites.

Para comenzar, vamos a repartir las fichas y pedirle a los estudiantes, organizados en grupos de aproximadamente 4 personas, que discutan la primera pregunta de la ficha. Recorriendo las discusiones buscaremos orientar las reflexiones grupales sobre el concepto de Internet. ¿Cómo usan la red? ¿Mandan mails, navegan y buscan información, utilizan redes sociales, bajan contenidos? Sin brindar una respuesta acabada, intentaremos que todos los grupos identifiquen que hay dos o más computadoras (entendiendo a los celulares u otros dispositivos como tales) involucradas y conectadas que realizan un intercambio de datos y distintos tipos de información. Luego, en la puesta en común de las respuestas escritas por cada grupo, enfatizaremos estos conceptos.

La segunda pregunta nos permite profundizar sobre los medios por los que se intercambia la información. Cuando los grupos expongan sus respuestas, al llegar a esta pregunta podemos escribir en el pizarrón los medios que mencionen: a través de cables, por el aire, etc.

Al terminar la lectura de las respuestas de los grupos vamos a proyectar un video, "El Mapa Digital - Un millón de kilómetros de cable para conectar al mundo", <https://www.youtube.com/watch?v=cQVuJte-1dQ>

Utilizando el video de referencia podemos concluir que no toda la información se transmite "por el aire", que la red de cables submarinos moviliza enormes volúmenes de información digital y es un componente clave para el funcionamiento de Internet. Para ilustrar el despliegue de la construcción física y técnica que posee Internet mostraremos un mapa con los cables submarinos que conectan las distintas regiones del mundo, similar al que se muestra en la Figura 1, y mencionaremos que las conexiones con cables abarcan aproximadamente un 90% de la información que se transmite por la red y solo un 10% utiliza satélites de soporte.

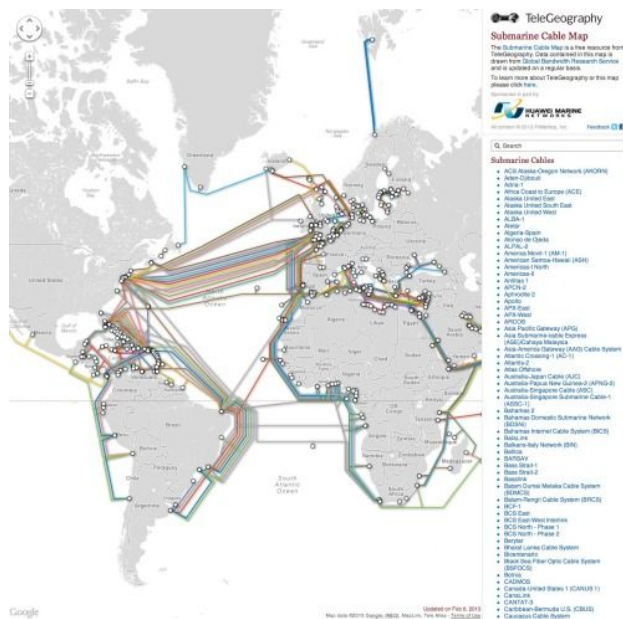


Figura 1. Mapa de la red de cables submarinos que conecta las diferentes regiones del mundo y da soporte físico a Internet.

Finalizada la proyección del video y los intercambios en torno a las preguntas de la ficha, pediremos a las estudiantes que revisen y amplíen, de manera individual, las respuestas que escribieron al comienzo de la actividad.

### Cierre

Para cerrar la actividad, haremos una última puesta en común para repasar los conceptos claves de la actividad y conversar sobre los cambios en las respuestas antes y después de ver el video y escuchar las reflexiones de los grupos. De esta manera se podrá poner en evidencia que pese a la familiaridad con el uso de Internet no se posee un conocimiento profundo respecto a su funcionamiento y la infraestructura física que lo permite.

## Ficha 1 para el alumno

### Un mundo interconectado... ¿cómo?

Seguro alguna vez escuchamos sobre Internet o la usamos para algo. Pero... ¿Qué pasa cuando mandamos información por la red? ¿Viaja directamente de un celular a otro y si los ponemos más cerca llega más rápido? ¿Mis fotos y videos dan la vuelta al mundo?

#### **Parte 1:**

Reunite en un grupo de 4 integrantes y entre todos piensen y escriban una posible respuesta a cada pregunta:

- ¿Qué es Internet? ¿Cómo la definirían con sus palabras?
- ¿Cómo viaja la información en Internet? Cuando se envía un mensaje desde una computadora a otra ¿qué medios se utilizan? Por ejemplo, cuando se escribía una carta el medio era el papel y podía enviarse de mano en mano, con un intermediario o por correo ¿Y en Internet?

Tómense un tiempo para pensar y discutir, y nombren una persona para que presente las ideas del grupo cuando su docente lo indique.

#### **Parte 2:**

Luego de ver el video, revisá las respuestas que dieron a las preguntas anteriores. ¿Creés que podrías completarlas o que deberías modificar algo? Anotá los cambios que le harías a las respuestas, y comparalas con las de tu grupo para escribir entre todos una respuesta más completa para entregarle a su docente.

## Actividad 2

### ¿Cómo viaja la información?

<p>Objetivo</p> <ul style="list-style-type: none"><li>• Analizar cómo se conforman las rutas que utilizan los datos para viajar por Internet. Comprender la importancia del uso de protocolos para organizar el funcionamiento de las comunicaciones y el caso particular de los protocolos TCP e IP para las que se realicen a través de redes de computadoras.</li><li>• Identificar conceptos propios de las comunicaciones a través de Internet.</li></ul>	<p>Modalidad de trabajo</p> <p>Grupal</p>
<p>Materiales</p> <ul style="list-style-type: none"><li>• Papel y lápiz; sobres y un taco de papeles.</li></ul>	

### Desarrollo

En esta actividad las y los estudiantes podrán recrear y analizar los mecanismos que utiliza Internet para el intercambio de información.

El comienzo de la actividad tiene como consigna trasladar un mensaje desde un **origen**, que puede ser el docente o algún estudiante, hasta un **destino** dentro del aula, que puede ser otro estudiante del curso ubicado lo más alejado posible de quien le envíe el mensaje. El emisor del mensaje va a escribir una palabra o frase en un papel que deberá llegar al destino establecido. El papel se pasará de mano en mano entre los estudiantes sin que ninguno se levante y para establecer el recorrido del papel cada estudiante debe elegir a quién dárselo de manera espontánea, sin decidirlo de antemano y sin indicaciones externas. Cuando el papel llegue a destino, el receptor deberá leer el mensaje, escribir una respuesta y volver a pasarlo de la misma forma (pero no necesariamente por el mismo camino) hasta que llegue nuevamente al origen.

Puede repetirse esta dinámica una vez más para que sea más evidente que el camino o *ruta* que recorre la información es afectado por las decisiones que toman los participantes del recorrido. Si la persona asignada al origen se cambia de lugar cuando está esperando el mensaje de vuelta será inevitable tener que cambiar el recorrido.

Luego de enviar los mensajes propiciaremos un breve intercambio destacando la existencia de un **origen** y un **destino** y cómo la ruta que toma el mensaje no fue la decisión de una persona sino una construcción a partir de decisiones individuales. Si antes de comenzar a pasar el mensaje se hubiese decidido qué ruta debía seguir el papel, organizar esta información hubiera implicado trabajo extra y se habría restado flexibilidad al recorrido.

El proceso a través del cual se decide la ruta por la que viaja la información se denomina **enrutamiento**. En particular, cuando la ruta no está definida de antemano y es elegida en el transcurso del envío, nos referimos a un **enrutamiento dinámico** que además es la manera que se utiliza en Internet para llevar mensajes de un lugar a otro a través de varias

computadoras. A modo de cierre se puede mencionar que la forma en que se realizó el envío de mensajes en el aula es una simplificación de la forma de transmitir mensajes que utiliza Internet, y utilizando de ejemplo la telefonía móvil: los celulares están constantemente en movimiento y sin embargo la información de Internet se puede recibir a pesar de cambiar de ubicación. Esto ocurre porque las rutas que siguen los paquetes correspondientes son calculadas a medida que se acercan hacia su destino.

El segundo ejercicio de la actividad nos permite analizar cómo se organiza la información en las comunicaciones por Internet. La transmisión de datos entre distintas computadoras utiliza cables que soportan una cantidad fija de información, por esta razón, los mensajes deben adaptarse a la dimensión del medio que los transmite. Cada mensaje se divide en fragmentos más pequeños de tamaño fijo denominados **paquetes**.

Le presentaremos al curso el siguiente problema: un ilustrador realiza dibujos en acuarela para ilustrar los libros de su saga favorita. La saga está organizada en muchos capítulos y tomos y los dibujos reflejan todo el desarrollo de sus historias. El ilustrador quiere mandar los originales en una carpeta en la que los tiene ordenados según el desarrollo de la saga así el autor y el editor, que se encuentran en otro país, pueden incluirlos en la próxima edición. Pero el papel acuarela no puede doblarse y el total de las ilustraciones ocupa mucho más lugar y supera el peso que tiene el sobre más grande que permite enviar el correo. Utilizando un taco de hojas de papel de oficina y sobres comunes donde no entren todas las hojas a la vez, vamos a pedirle a la clase que haga de cuenta que cada papel es una ilustración y que piensen cómo harían para solucionar el problema del ilustrador.

La solución que intentaremos que encuentre el grupo, contando con nuestra guía, es enviar las ilustraciones por separado en distintos sobres. Como los dibujos no van a enviarse juntos y el correo postal no siempre funciona como se espera, los dibujos deberán poder volver a ordenarse cuando lleguen a destino para recrear el desarrollo de la historia de los libros. Pero para poder reconstruir la carpeta, el editor y el autor deben esperar a que lleguen **todas** las ilustraciones. Esto es similar a lo que sucede cuando se envía mucha información por Internet (por ejemplo, un archivo con una imagen grande o una canción): la información se divide en paquetes numerados que son recibidos por el receptor, y que antes de poder utilizarlos debe volver a juntarlos para reconstruir el archivo original. Además, las limitaciones que planteaba el correo son equivalentes a las restricciones de tamaño dadas por los cables de comunicación en Internet.

En el tercer ejercicio, combinaremos los ejercicios anteriores: la carpeta que representa el taco de papeles tiene que ir desde un origen (nuevamente el docente, por ejemplo) a un destino (nuevamente un estudiante en el fondo del curso), pero cada persona solamente puede pasar una hoja por vez. Como las ilustraciones tienen que tener un orden coherente con el de la historia, cada papel va a tener un número que represente su ubicación en la carpeta y van a ser distribuidas en cualquier orden. Las personas que están en el medio no necesitan reconstruir la carpeta en cada paso, sino que pueden ir pasando las ilustraciones hasta el destino. Por su parte, el destinatario de la carpeta sí tendrá que reconstruirla completa, para lo cual necesitará esperar a que lleguen todos los dibujos, y utilizar los números que aparecen escritos para ponerlos en orden. Un hecho a destacar es que no hace falta que todos los papeles sigan el mismo camino para llegar a su destino; para ilustrar esto, daremos distintas hojas a distintas personas al iniciar la ruta.



Luego de presentar los tres ejercicios, explicaremos que las características trabajadas en ellos son los principios básicos según los cuales viaja la información en Internet: cada persona en los ejercicios representa a una computadora (ya sea personal o de otras funciones), y la carpeta representa a la información que debe viajar entre dos computadoras específicas.

La información digital está compuesta por números, y las tiras de números que representan la información digitalizada se dividen en partes y forman diferentes paquetes. Los paquetes se transmiten de máquina en máquina hasta llegar a un destino donde se ordenarán para reconstruir el mensaje original cuando se hayan recibido todos. Cada par de computadoras conectadas intercambia paquetes que contienen los distintos fragmentos del mensaje original que ha sido enviado.

Esta forma de establecer la transmisión de mensajes entre computadoras sigue una serie de reglas que reconocen todos los dispositivos que participan de la comunicación. Al *conjunto de reglas básicas que usan las computadoras para comunicarse* se las conoce con el nombre de **protocolo** de comunicación. En Internet, la comunicación se basa principalmente en dos protocolos, los protocolos TCP e IP, que se conocen en conjunto como TCP/IP.

El **Protocolo de Control de Transmisión** (TCP por sus siglas en inglés, *Transmission Control Protocol*, o de manera informal, “**protocolo TCP**”) establece cómo intercambiar mensajes entre una computadora de origen y una de destino, independientemente de cómo se realiza la conexión entre ambas. Este protocolo es el responsable, por ejemplo, de establecer las reglas que determinan cómo dividir mensajes grandes en paquetes para poder enviarlos de acuerdo a las características físicas de la red. Gracias a él, podemos enviar mensajes entre dos máquinas como si estuvieran directamente conectadas, incluso cuando estas máquinas estén en diferentes continentes y haya que pasar por muchísimas computadoras intermedias para llegar de una a otra.

Para que esta comunicación entre dos computadoras sea posible hace falta calcular de forma dinámica la ruta entre ellas. Las formas de calcular la ruta dinámicamente también responden a un protocolo que las regulan. El protocolo que permite hacerlo se denomina **Protocolo de Internet** (IP por sus siglas en inglés, *Internet Protocol*, o de manera informal, “**protocolo IP**”).

### Cierre

Para concluir con la actividad haremos una síntesis en el pizarrón de los conceptos puestos en juego en las dinámicas con los papeles y su relación con las comunicaciones en Internet.

En nuestro uso cotidiano de Internet parece que los mensajes salen de un **origen** y llegan a **destino** completos y de forma directa, pero en realidad los cables que componen las conexiones de Internet tienen limitaciones de tamaño, y por esta razón los mensajes se dividen en **paquetes**. La organización y reconstrucción de los mensajes requiere de un número asignado a cada fragmento y su funcionamiento depende del **protocolo TCP**. Durante las transmisiones de datos los paquetes pasan por muchas de computadoras intermedias ya que no es posible conectar de manera directa cada computadora con todas las demás. Cada una se conecta con un grupo pequeño de ellas y siguiendo las reglas del **protocolo IP** conforman una red donde todas pueden llegar a cualquier destino por algún

<Program.AR/>

camino de conexiones entre computadoras intermedias. Las **rutas** que siguen los paquetes no están prefijadas de antemano, sino que se van calculando de manera **dinámica** a medida que los mensajes avanzan, lo que permite, por ejemplo, que se pueda usar Internet en teléfonos móviles.

## Secuencia didáctica 2

## Me sigue una nube

Cuando hacemos referencia a la información que almacenamos o compartimos fuera de nuestros dispositivos solemos decir que están en **"La nube"**. La "nube" es en realidad un conjunto de servicios que pueden accederse en Internet y aunque su nombre se aleja de los aspectos concretos que la componen, en muchos casos su funcionamiento requiere una infraestructura costosa y de grandes dimensiones.

Las computadoras que proveen los servicios que requerimos, por ejemplo almacenar información, vídeos y fotografías, se denominan **servidores**. A lo largo de esta Secuencia didáctica replicaremos el funcionamiento de un servidor y profundizaremos en algunos aspectos relacionados a la forma en que las computadoras, sean servidores o no, se identifican en la red y cómo es el sistema para poder encontrarlas, reconocerlas y acceder a ellas.

### Objetivos

- Reconocer el concepto de **servidor** como una computadora que ofrece servicios a otros dispositivos conectados a la misma red.
- Reflexionar sobre las condiciones necesarias para poder instalar y mantener un servidor de manera profesional.
- Comprender los mecanismos que permiten identificar y acceder a un servidor en la red: la dirección IP y el sistema de nombres de dominio o DNS.

## Actividad 1

### ¿Y la nube dónde está?

<p>Objetivo</p> <ul style="list-style-type: none"><li>• Comprender que cualquier dispositivo electrónico con conexión a la red puede proveer servicios a otros dispositivos.</li><li>• Construir un modelo de servidor para compartir archivos e imágenes predeterminados.</li><li>• Reconocer la importancia y utilidad de utilizar direcciones IP y puertos.</li></ul>	<p>Modalidad de trabajo</p> <p>Grupal</p>
<p>Materiales</p> <ul style="list-style-type: none"><li>• Una computadora con Python 2.7 instalado y con conexión a una red (la de la escuela, o bien otra red wifi).</li><li>• Varias computadoras en la misma red que la computadora anterior.</li><li>• Opcionalmente, un proyector para mostrar cómo se instala el servidor, y cómo se realizan los cambios.</li></ul>	

- El programa que ofrece el servicio y sus archivos, que puede encontrarse en `iniciar-servidor.py` (<http://bit.ly/CCau1SAn1>).

## Desarrollo

En esta actividad vamos a armar nuestro propio servicio de Internet para compartir algunos archivos e imágenes predeterminados. Como primer paso, se puede empezar un breve intercambio para empezar a reconocer cómo es el funcionamiento de los servicios que utilizamos cotidianamente en Internet. Por ejemplo, preguntar a la clase qué creen que sucede cuando queremos leer un foro de preguntas, ver una foto en una red social o consultar un documento en Wikipedia; dónde creen que se encuentra esa información y si no es parte de sus respuestas, qué piensan que es la nube. Utilizando las respuestas de la clase como referencia, iremos construyendo definiciones más precisas y nuevas preguntas respecto a la ubicación y características físicas de la nube.

La nube es un conjunto de servicios relacionados con Internet, que generalmente incluyen el almacenamiento de archivos y datos que, contando con los permisos adecuados, pueden ser accedidos desde cualquier computadora. Los archivos y datos de la “nube” se encuentran físicamente en una computadora conectada a Internet a la que llamamos **servidor**. La computadora que cumple el rol de servidor ocupa espacio y consume electricidad y si se quiere garantizar que sus servicios estén disponibles tiene que estar prendida a toda hora. Eso requiere personas trabajando para cuidarla y mantenerla en funcionamiento. Si un servidor se rompe, especialistas técnicos deben colocar otro en su reemplazo mientras dure la reparación porque el servicio que brinda es más importante que el servidor en sí.

Cuando queremos utilizar alguno de los servicios que provee un servidor, como por ejemplo acceder a una foto de una red social, nuestra computadora envía un mensaje a otra computadora, que puede estar ubicada en cualquier lugar del mundo, y al recibirlo nos responderá con la información solicitada.

Le explicamos a la clase que la propuesta de esta actividad es configurar nuestro propio servidor en una computadora local para acceder durante la clase. La computadora elegida no estará prendida las 24 horas del día, pero sí nos permitirá poner ciertos archivos a disposición de otras computadoras para que sus usuarios puedan acceder a ellos. Es importante resaltar que cualquier computadora conectada a una red puede funcionar como servidor y que su funcionamiento va a recrear el de otros servidores de Internet.

La computadora que oficiará de servidor tiene que tener instalado Python en su versión 2.7. Python ya se encuentra instalado en las distribuciones más populares de Linux y no presenta una instalación compleja en Windows. Para iniciar el funcionamiento del servidor, tenemos que ejecutar el programa que se encuentra en `iniciar-servidor.py` y esperar a ver un mensaje similar al siguiente:

```
¡El servidor está en línea! :)
```

Para acceder a él, abrí un navegador y escribí `192.168.43.242:8001` en la barra de

direcciones.

Para apagarlo, presioná Ctrl + C o cerrá esta ventana.

El número que tenemos que escribir va a variar en cada red y depende de su configuración (su significado y características específicas se abordarán más adelante). Como indica el mensaje, podemos acceder al servidor para recorrer o “navegar” los contenidos que contiene utilizando el navegador de nuestros celulares o computadoras, pero para poder hacerlo los dispositivos deberán estar en la misma red que el servidor, ya sea conectados por un cable a la red escolar o en la misma red WiFi.

Indicamos a las estudiantes que para acceder al servidor sigan las instrucciones impresas en el mensaje que dio el servidor: escribir el número largo en la barra de direcciones de un navegador exactamente igual a como aparece en pantalla, sin omitir ni agregar ningún número ni símbolo. Luego de acceder pueden empezar a explorar la estructura de archivos que se encuentra disponible con el objetivo de que identifiquen que lo que ven es aquello que se encuentra en el servidor y luego vamos a trabajar sobre la ficha para que se refuerce la idea de que quien maneja el servidor decide cuáles archivos compartir y cuáles no.

1. Con el servidor en funcionamiento indicamos al curso que complete el árbol de directorios que está en la ficha. Esto va a permitir reconocer los archivos que brinda el servidor.
2. Vamos a borrar, renombrar y agregar archivos en el servidor sin indicar a la clase cuáles son. De esta manera solo podrán ver los cambios accediendo al servidor y recorriendo su estructura de archivos otra vez. Es importante realizar al menos una vez cada una de las operaciones. Por ejemplo, se podrían borrar los archivos `documento-secreto.txt` y `otras-cosas/hola.png`, renombrar el archivo `bebida.jpg` (de la carpeta `una-carpeta/con-otra-adentro/`) a `mate.jpg`, y agregar un archivo llamado `sencillo.txt` en el mismo directorio que `documento-secreto.txt` (por ejemplo, copiando `documento-secreto.txt` y renombrando la copia a `sencillo.txt`).
3. Pedimos a los estudiantes que cierren el navegador, lo vuelvan a abrir y accedan al servidor otra vez.
4. Utilizando de referencia el árbol que completaron en el punto 1, le indicamos a la clase que llene la segunda parte de la ficha registrando los archivos que fueron borrados, renombrados y agregados.
5. Apagamos el servidor, siguiendo las instrucciones que indica el programa.
6. Pedimos a los estudiantes que nuevamente cierren y abran el navegador e intenten acceder al servidor. Como el servidor no se encuentra activo ya no debería ser posible.

El primer registro del árbol de archivos que completen los estudiantes debería ser similar al siguiente:

```
|— documento-secreto.txt
|— iniciar-servidor.py
|— leer-hasta-el-final.txt
|— LEER.png
|— LEER.svg
```

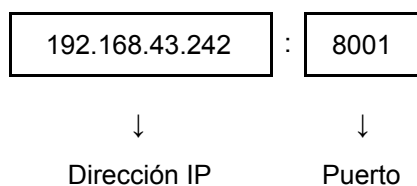
<Program.AR/>

```
|— otras-cosas
|   |— hola.png
|— una-carpeta
|   |— con-otra-adentro
|       |— bebida.jpg
```

Luego de las modificaciones, se espera que la tabla que completen sea como esta:

Archivo	Operación realizada (elegí la que corresponda)	Nuevo nombre (si corresponde)
documento-secreto.txt	<del>Renombrado</del> / Eliminado / <del>Agregado</del>	
otras-cosas/hola.png	<del>Renombrado</del> / Eliminado / <del>Agregado</del>	
una-carpeta/con-otra-adentro/bebida.jpg	Renombrado / <del>Eliminado</del> / <del>Agregado</del>	una-carpeta/con-otra-adentro/mate.jpg
sencillo.txt	<del>Renombrado</del> / <del>Eliminado</del> / Agregado	

Luego de que todos completen sus fichas, recordamos al curso el “número largo separado por puntos” que tuvieron que escribir para establecer la conexión con el servidor. Lo escribimos en el pizarrón diferenciando bien sus dos partes y los dos puntos que las separan.



Explicamos que ese número identifica al servidor y que está compuesto por:

- La **dirección IP**, compuesta por cuatro números entre 0 y 255 separados por puntos, que identifican a un dispositivo dentro de la red. Cada dispositivo conectado a una red tiene su propia dirección IP.
- El **puerto**, un número que identifica al servicio que queremos acceder dentro del dispositivo. Cada servicio tiene asociado un número de puerto específico para permitir que un mismo servidor brinde varios servicios. En las redes, cada uno de los diferentes servicios (envío de mails, recepción de mails, intercambio de archivos, sitios web, etc.) utilizan un puerto diferente de comunicación.

## Cierre

La primera parte de la actividad servirá para presentar el concepto de servidor, su relación con la nube, la necesidad de ubicarlo físicamente en una computadora que es administrada por una o varias personas que deciden qué información contiene y cuándo está disponible. La existencia de un número específico para identificar un dispositivo de la red nos invitará a reflexionar sobre la posibilidad de accederlo no sólo desde aquellas computadoras que integran la misma red sino a través de las redes que se conecten a ella utilizando Internet.

En un intercambio grupal, pediremos al curso que mencione empresas reconocidas que presten servicios en Internet. Lo ideal es que se mencionen algunas que permitan compartir archivos, como Google Drive o Dropbox y otras vinculadas a las redes sociales a las que podamos subir fotos y videos, como Facebook o Instagram. Iremos orientando las reflexiones colectivas con algunas preguntas: ¿Dónde se encuentra físicamente la información que compartimos? ¿Dónde están esas empresas y sus servidores? Esta actividad nos permite identificar la existencia de servidores destinados a almacenar y poner a disposición esa información. Además, cuando apagamos el servidor ya no pudimos acceder a sus archivos. Esto significa que quien sea responsable de los servidores debe garantizar que estén en funcionamiento las 24 horas para poder ser accedidos en todo momento y desde cualquier lugar del mundo. Mantener activa toda la infraestructura física que requieren estas grandes empresas implica que tengan conectividad, electricidad y resguardos frente a cualquier inconveniente. Si bien suelen ofrecerse estos servicios de manera gratuita, el mantenimiento de sus servidores parecería ser algo sumamente costoso. Por esta razón, los servidores son máquinas especialmente potentes que brindan otros servicios asociados y las empresas que los mantienen también son lucrativas gracias a la venta de publicidad entre otros mecanismos de uso de la información que reciben.

## Ficha 1 para el alumno

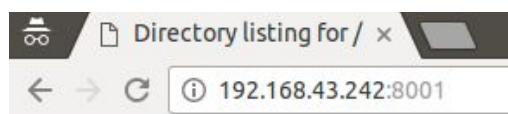
¿Y la nube dónde está?

### Bajada

¿Alguna vez te preguntaste dónde está la información que subís a Internet? ¿Una vez que la borrás de tu celular, la imagen desaparece? ¿Cómo funciona un servidor?

Tu profesor/a te va a indicar a qué red debés conectarte y te va a compartir una dirección numérica. Ese número representa la dirección del servidor con el que vamos a trabajar. Para accederlo, tenés que escribir la dirección en la barra de direcciones del navegador que estés utilizando (Google Chrome, Chromium, Firefox, Internet Explorer, etc.) exactamente como la vez, con todos los números y símbolos y sin agregar espacios ni caracteres.

Vas a ver una pantalla parecida a la de la imagen aunque es posible que el número en la barra de direcciones sea diferente (porque depende de la configuración de la red de la escuela).



### Directory listing for /

- [documento-secreto.txt](#)
- [iniciar-servidor.py](#)
- [leer-hasta-el-final.txt](#)
- [LEER.png](#)
- [LEER.svg](#)
- [otras-cosas/](#)
- [una-carpeta/](#)

Tu primera tarea será completar el árbol de directorios que te presentamos a continuación, para lo cual deberás navegar por los distintos archivos y carpetas. Para volver atrás podés utilizar el botón que provee tu navegador (cuyo ícono es una flecha, y suele estar a la izquierda de la barra de direcciones) o podés también presionar simultáneamente las teclas Alt y Flecha Izquierda.

Lo primero que vas a hacer es completar lo que ves en la lista de abajo, que es un listado de los archivos y directorios que se encuentran en el servidor al que accedimos.

Para volver atrás podés utilizar el botón de tu navegador (cuyo ícono es una flecha hacia la izquierda) o podés también presionar simultáneamente las teclas Alt y Flecha Izquierda.

```
.
|----- .txt
|----- .py
|----- .txt
|----- .png
|----- .svg
|-----
```





## Actividad 2

### La libreta de direcciones

<p>Objetivo</p> <ul style="list-style-type: none"><li>• Identificar las direcciones IP como la forma en que se denominan en Internet los servidores de un sitio y que pueden utilizarse para accederlos.</li><li>• Comprender los principios básicos de funcionamiento del sistema de nombres de dominio (DNS) y cómo este mecanismo facilita la navegación por Internet.</li></ul>	<p>Modalidad de trabajo</p> <p>Grupal</p>
<p>Materiales</p> <ul style="list-style-type: none"><li>• Dispositivos con acceso a Internet.</li><li>• El navegador Google Chrome. (Opcional)</li></ul>	

### Desarrollo

Todo dispositivo que se encuentra en Internet tiene una dirección que lo identifica y permite acceder al mismo, llamada **dirección IP**. Esta dirección consta de cuatro números entre 0 y 255 separados por puntos, como por ejemplo 186.33.232.19, e identifica a un dispositivo conectado a Internet dentro de la red. Si dicho dispositivo provee algún servicio como una página web, correo electrónico, o intercambio de archivos, hace falta conocer la dirección IP del mismo para acceder al servicio. Utilizando únicamente 4 bytes puede representarse cualquier dirección IP.

En la actividad anterior accedimos al servidor del aula utilizando su dirección IP, pero en general es dificultoso identificar distintas computadoras utilizando solo estos números. Para facilitar la navegación, recordar y distinguir mejor los sitios y dispositivos conectados a Internet utilizamos otra forma de distinguirlos: los **nombres de dominio**<sup>1</sup>, que pueden estar compuestos por palabras y caracteres en lugar de utilizar únicamente números. Los dominios ocupan como mínimo 8 bytes, es decir que pueden ocupar más del doble de espacio que una dirección IP. Su gran ventaja frente a ellas no tiene que ver con el almacenamiento u otro aspecto técnico sino con la facilidad para que puedan recordarlos las Personas.

Si las computadoras utilizan direcciones IP para identificarse en Internet y las personas utilizamos nombres de dominio es necesario algún mecanismo para que cuando ingresamos una URL en la barra de direcciones de un navegador la computadora pueda solicitar los servicios a la dirección IP correspondiente. Esta función de “traducción” es realizada por un sistema denominado **DNS**, por sus siglas en inglés, **Domain Name System** (Sistema de Nombres de Dominio).

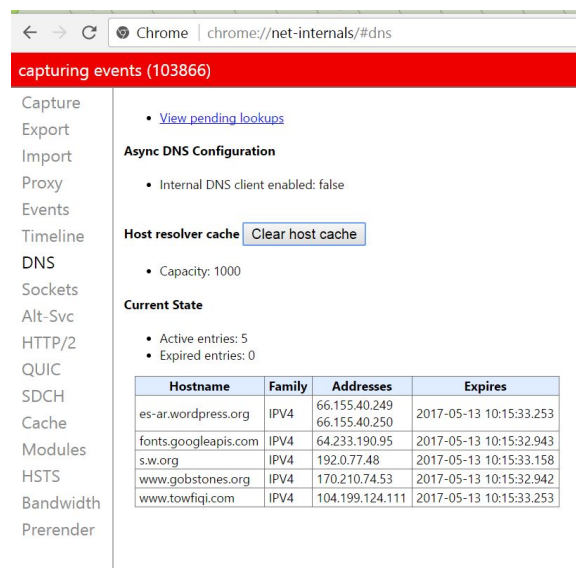
---

<sup>1</sup> A veces, para referirse a los nombres de dominio, se usan términos como “dirección web” o URL. Si bien existen ciertas diferencias técnicas entre estos términos, en el habla cotidiana es normal utilizarlos como sinónimos.

Como introducción a esta actividad indagaremos sobre la forma en que los estudiantes utilizan Internet, cómo acceden a un sitio web conocido, si utilizan Google y entran a sus resultados, si escriben directamente la dirección de la página que buscan o si al igual que cuando tuvieron que acceder al servidor, utilizan la dirección IP. El objetivo de este intercambio es reflexionar respecto a que si todos los servidores de los sitios web que visitamos tienen una dirección IP pero al momento de accederlos solemos hacerlo a través de su nombre de dominio, debe existir un mecanismo que se ocupe de asociar esos dominios a la direcciones IP correspondientes.

Invitamos a las y los estudiantes a indagar sobre el funcionamiento del sistema DNS, el encargado de traducir los dominios en direcciones IP. Utilizando los sitios que aparecen en la Ficha, deben completar la dirección IP que le corresponde al servidor que permite acceder a cada uno de ellos. Para ello, podemos presentar distintas herramientas

1. Desde el navegador Google Chrome, se puede acceder a la memoria de DNS escribiendo `chrome://net-internals/#dns` en la barra de navegación. En dicha pantalla se muestran todos los dominios accedidos recientemente junto a su correspondiente IP. Recomendamos borrar esta memoria antes de comenzar la actividad para que sea más fácil encontrar los sitios que interesan; esto se logra haciendo clic en el botón "Clear host cache". Ver la Figura 1.

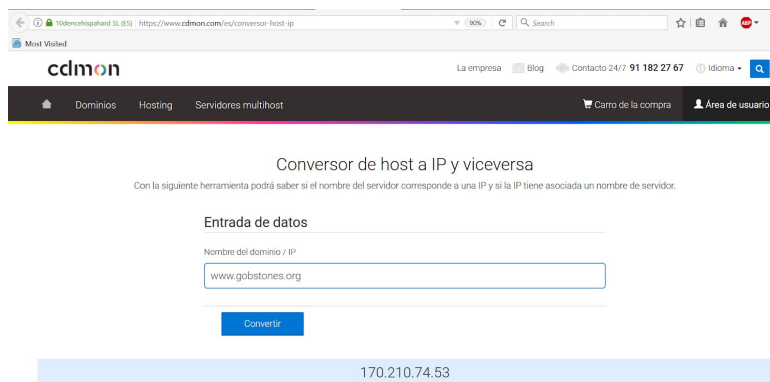


Navegador Google Chrome mostrando los DNS de los últimos sitios accedidos y el botón de "Clear host cache"

2. Si se dispone de equipos con GNU/Linux instalado, el comando `host` permite conocer la IP de un dominio. Basta con escribir ese comando en una terminal, por ejemplo:

```
host www.google.com.
```

3. Si ninguna de las opciones anteriores funciona, puede utilizarse un servicio online como <https://www.cdmon.com/es/conversor-host-ip> o cualquiera que surja de la búsqueda de "encontrar ip de host". Ver Figura 2.



Ejemplo de página de conversión de hosts a IP o viceversa.

Luego de que todos los estudiantes hayan podido utilizar al menos un mecanismo para obtener la dirección IP de los sitios, explicaremos cómo realizan las computadoras la traducción del dominio.

Cuando se ingresa una dirección de dominio, el navegador la busca en una tabla de direcciones para verificar si ya la tiene registrada, y sólo si no la reconoce realiza una consulta a otras máquinas. Por esta razón, si bien todos los mecanismos pueden ser utilizados en esta actividad, recomendamos que los estudiantes intenten analizar al menos una dirección de dominio con las herramientas del navegador que indicamos en el punto 1.

Pedimos a los estudiantes que ingresen las direcciones IP obtenidas en la barra de direcciones del navegador. De esta forma pueden comprobar que es posible acceder a cada uno de los sitios tal como podían hacerlo usando el nombre de dominio correspondiente.

Luego de acceder a los sitios con las direcciones IP que registraron, organizamos el curso en grupos para que discutan y respondan las preguntas de la última parte de la Ficha.

## Cierre

Para finalizar la actividad, haremos un intercambio reflexionando sobre la necesidad de tener nombres de dominio, direcciones IP y un mecanismo para poder asociarlos y profundizaremos sobre cómo funciona realmente el sistema DNS de nombres de dominios. Vamos a leer las preguntas una a una y guiar los intercambios para arribar a conclusiones similares a las siguientes:

La cantidad de nombres de dominio es tan grande que hace imposible almacenar toda esa información en una tabla en nuestra computadora. Para resolver este inconveniente existen servidores dedicados a eso. Cada vez que accedemos a un sitio nuevo, la computadora busca esa dirección IP consultando a un servicio que se encarga de ello, el cual, si no la tiene dentro de sus registros, consultará a su vez a otros servidores que repetirán este mecanismo hasta dar con el resultado, o concluir en que el dominio no existe.

El sistema de nombres de dominio es jerárquico, y cada nivel agrupa los servicios o sitios según algún criterio. Si leemos una dirección de derecha a izquierda, vamos a encontrar información desde lo más general a lo más particular. Por ejemplo, `canaima.softwarelibre.gob.ve` nos sirve para identificar los niveles dentro del

dominio. Si lo analizamos de derecha a izquierda, sabremos que se trata de un sitio de Venezuela (.ve), que pertenece a una entidad gubernamental (.gob), que está relacionado con el software libre (.softwarelibre) y, que su contenido está relacionado con el sistema operativo Canaimá (canaima).

Los países suelen tener su propio dominio: .ar es el de Argentina, .br el de Brasil. Otros dominios pueden estar asociados a tareas y no a países, como .com (de *comercial*) para empresas, .edu para instituciones educativas, .org para organizaciones diversas y .gob, como vimos en el ejemplo, para organismos gubernamentales

Para concluir, destacaremos que seguramente podemos recordar los nombres de dominios de los sitios que investigamos en la actividad, reconocer información a partir de sus nombres y repetirlos sin revisar las tablas, pero no ocurre lo mismo con las direcciones IP.

## Ficha 2 para el alumno

### La libreta de direcciones

#### Bajada

Dicen los libros de historia que nuestros abuelos recordaban muchos números de teléfono y los que no, los anotaban en una agenda. Con la ayuda de los celulares y la información digital, cada vez necesitamos menos recordar números largos.

En la actividad anterior accedimos al servidor usando un número largo separado por puntos... ¿Te lo acordás?

Ese número representa una dirección IP, que es la forma en que se identifican los servicios que utilizamos en Internet, pero que no solemos ver ¡y menos recordar!

Nuestro primer objetivo es rastrear la verdadera dirección IP de algunos sitios.

Para encontrar la dirección IP correspondiente a un nombre de dominio podemos utilizar el navegador Google Chrome, escribiendo `chrome://net-internals/#dns` o acceder a algún servicio online como <https://www.cdmon.com/es/conversor-host-ip>.

Una vez elegido el método para encontrar las direcciones IP, completá la siguiente tabla:

Nombre del dominio	Dirección IP
educ.ar	
www.gobstones.org	
facebook.com	
canaima.softwarelibre.gob.ve	
www.convosenlaweb.gob.ar	
gnu.org	

Luego, organizados en grupos, discutan y respondan las siguientes preguntas:

1. ¿Es lo mismo entrar a una página escribiendo el nombre de dominio que escribiendo su IP?
2. ¿Por qué creen que se inventaron los nombres de dominio?
3. ¿Cómo hará la computadora para descubrir qué IP le corresponde a un dominio?
4. ¿Es posible que una sola computadora tenga guardados todos los dominios que existen en el mundo?
5. ¿Por qué algunos dominios terminan en *.com*, otros en *.org* y otros en *.ar*?

#### El dato

Si las IPs son tan difíciles de recordar, ¿para qué existen? Los motivos son varios, pero hay uno que resulta particularmente interesante: el poco espacio que ocupan. Cualquier IP se puede representar con tan solo 4 bytes, mientras que para guardar texto se utiliza un byte por carácter; si quisiéramos representar algo tan corto como `gnu.org` necesitaríamos 7 bytes... ¡casi el doble!

<Program.AR/>

Por esto y por la facilidad que tiene la computadora para hacer cuentas con estos números es que se utilizan las IPs. Las personas recordamos las palabras con más facilidad ya que además permiten asociar el nombre de dominio con el contenido de la página, por esta razón existen ambas formas.

### Secuencia didáctica 3

## Ciudadanía digital

Cuando utilizamos tecnología informática, surgen no solo aspectos tecnológico sino otros humanos, culturales, políticos y sociales que también debemos tener en cuenta. La **ciudadanía digital** se relaciona con la comprensión de estos aspectos, y nos permite aplicar conductas seguras, responsables y orientadas por principios éticos y legales cuando utilizamos Internet, las redes sociales y otras tecnologías disponibles.

La presente secuencia pretende crear concientización acerca de los riesgos o problemas que podríamos enfrentar cuando utilizamos celulares, computadoras, o diferentes medios tecnológicos y compartimos dicha información a través de Internet. El primero de tales riesgos tiene que ver con los resultados que arrojan los motores de búsqueda. Un **motor de búsqueda**, o **buscador**, es una aplicación que permite buscar información en Internet. Los motores de búsqueda son la principal puerta de acceso a la información y por lo tanto se vuelve imprescindible comprender cómo funcionan, de qué dependen los resultados que arrojan y cómo nuestros hábitos modifican los resultados que vemos. De esa forma podremos tener más conciencia sobre la calidad de la información que encontramos en Internet y observarla con una mirada crítica.

La utilización responsable de Internet y la ciudadanía digital comprenden aspectos fundamentales para poder desenvolverse en la vida cotidiana y para ello requieren ser abordadas con información apropiada.

#### Objetivos

- Visibilizar los riesgos de compartir información sensible en Internet.
- Comprender que los resultados que arroja un motor de búsqueda no son neutrales, sino que dependen de múltiples factores. En particular, con información que se recolecta de los usuarios.

### Actividad 1

## La llegada del hombre a la Luna

#### Objetivo

- Comprender que la información que encontramos en Internet depende en gran medida de cómo se realiza la búsqueda.
- Reflexionar sobre los aspectos que condicionan los resultados que arroja una búsqueda en Internet: la visión hegemónica sobre el tema que se está investigando, los intereses del motor de búsqueda, la información sobre el usuario y sus preferencias, etc.

#### Modalidad de trabajo

Grupal



#### Materiales

- Dispositivos con acceso a Internet

## Desarrollo

Cuando realizamos búsquedas en Internet podemos pensar que siempre vamos a obtener los mismos resultados y en el mismo orden. Aunque no siempre resulte evidente, los algoritmos que utilizan los motores de búsqueda tienen la capacidad de personalizar los resultados, teniendo en cuenta factores que van desde el contexto en el cual buscamos (nuestra ubicación, qué dispositivo estamos usando, qué busca la gente que está en el mismo lugar que nosotros) hasta datos personales (edad y género, historial de búsquedas, preferencias).

En esta actividad vamos a poner de manifiesto estos mecanismos, a partir de una búsqueda sobre un hecho que aún hoy genera controversias: la llegada del hombre a la luna. Como las condiciones de búsqueda que mencionamos en el párrafo anterior son difíciles de recrear, intentaremos "forzar" esta personalización utilizando un buscador que permite simular que estamos buscando desde otro país.

Para comenzar la actividad realizamos un breve debate con la clase con preguntas similares a las siguientes:

- Si dos personas buscamos exactamente lo mismo en un motor de búsqueda, ¿veremos los mismos resultados?
- ¿Da lo mismo buscar en Google desde Argentina que, por ejemplo, desde Francia?
- ¿Pueden nuestras búsquedas pasadas afectar a las búsquedas nuevas que hacemos?

En general solemos creer que los buscadores actúan con más neutralidad de la que realmente lo hacen, si las respuestas de los estudiantes dan cuenta de esta creencia no haremos demasiadas aclaraciones al respecto para no condicionar el desarrollo de la actividad.

Dividimos la clase en una cantidad par de grupos y repartimos las fichas definiendo desde qué región buscará cada equipo: Rusia o Estados Unidos. En esta actividad utilizamos el buscador DuckDuckGo (<https://duckduckgo.com>), cuya principal característica es que no condiciona automáticamente las búsquedas a partir de los datos personales del usuario, como sí lo hacen los buscadores tradicionales como Google, Yahoo o Bing. Además, permite que habilitemos algunas características de personalización en forma voluntaria (localización).

Utilizando el buscador para simular que se encuentra en el país asignado, cada grupo va a hacer la búsqueda que se indica en la Ficha. Para cambiar de país deben realizar una búsqueda y después hacer click en la sección de arriba a la izquierda, en el lugar donde se observa el nombre de un país y un pequeño selector, y se desplegará un menú con la lista de países posibles. Allí pueden elegir el país correspondiente a su grupo.

Cada grupo va a completar la Ficha con el contenido de al menos tres recursos (artículos, videos, etc) que hayan surgido de la búsqueda que realizó y luego deberá compararlo con los resultados que obtuvieron en otro grupo al simular la búsqueda desde un país diferente. El resultado esperado es que los equipos que utilizan el buscador como si estuvieran en Estados Unidos encuentren la "historia oficial" de la llegada del Apolo 11 a la Luna en el año 1969, mientras que los que buscan como si se encontraran en Rusia tendrán más resultados sobre un supuesto montaje televisivo, detalles sobre la carrera espacial entre EEUU y Rusia, etc.

Al finalizar las comparaciones indicaremos al curso que realice una búsqueda con sus propios dispositivos, comparando los resultados y las publicidades que aparecen. Deberían observar que las publicidades no son las mismas para todos: cada persona recibe publicidades en base a las búsquedas que ha realizado antes y a otros datos disponibles en Internet.

Motivaremos una reflexión en torno al funcionamiento de los buscadores: ¿Es conveniente que las respuestas dependan de quién es el que busca? ¿Cómo saber si el buscador que estoy utilizando está excluyendo resultados para mi búsqueda porque considera que no serían relevantes para mí? ¿O cómo saber si no está preparando los resultados para que yo crea algo que puede no ser cierto? Intentaremos concluir que al buscar en Internet las respuestas dependen de muchos factores, incluyendo búsquedas anteriores, ubicación, y datos personales, lo que en muchos casos contribuye a acercarse a nuestros intereses pero también hace que nos encontremos frente a un recorte de la información que no siempre elegimos.

Al acceder a Internet con un buscador moderno, la experiencia es muy subjetiva. Por ejemplo, como vimos, un estadounidense leerá sobre lo maravilloso de la llegada del hombre a la luna, mientras que un ruso leerá sobre la carrera espacial y sobre el posible fraude realizado por Estados Unidos. Dos personas se van a encontrar con información diferente, y seguramente van a llegar a conclusiones diferentes sobre el mismo hecho. Además, seguramente también van a ser diferentes las noticias sugeridas o las publicidades que vea cada uno, recortando sus perspectivas frente a la realidad e incitándolos a comprar distintos productos.

## Cierre

Para concluir la actividad, abordaremos nuevamente un aspecto trabajado en actividades anteriores: los costos que requiere montar un servidor. Reflexionaremos sobre la relación entre el costoso despliegue de infraestructura que requieren algunos servicios que para nosotros son (o parecen ser) gratuitos, entre ellos los buscadores. ¿Cómo se pagan esos costos? En la mayoría de los casos son nuestros datos personales lo que damos a cambio, muchas veces sin ser demasiado conscientes de ello. Diferentes empresas pagan para poder saber cómo hacer mejor publicidad sabiendo qué intereses tiene cada persona, de esta manera será mucho más fácil lograr venderle diversos productos o servicios. Nuestro paso por Internet deja huella y tiene consecuencias, y debemos tener presente si queremos o no dejar ese registro de nuestros datos a disposición de esas empresas.

## Ficha 1 para el alumno

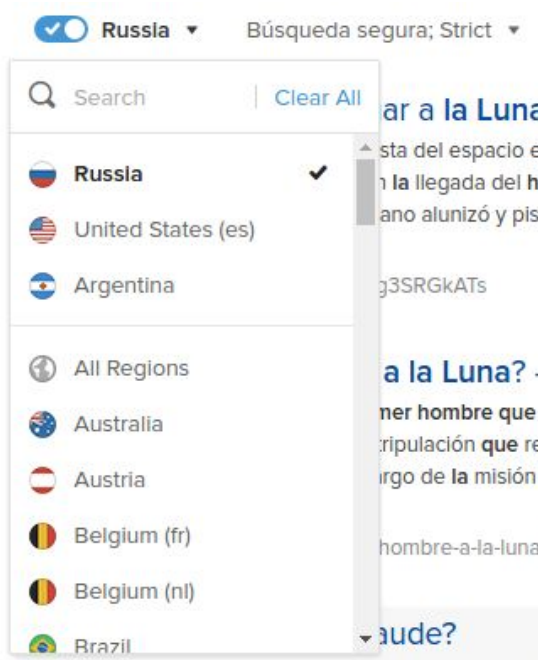
### La llegada del hombre a la Luna

¿Escuchaste hablar sobre la llegada del hombre a la Luna? ¿Y alguna versión que desmienta que realmente haya pasado? ¿Siempre es confiable lo que dice Internet? ¿Y si se contradice?

Para esta actividad vamos a usar un buscador que no es tan conocido como otros pero que nos permite hacer las búsquedas como si estuviéramos en otro país. Se llama DuckDuckGo y se puede acceder desde cualquier dispositivo entrando a <https://duckduckgo.com/>.

Según le toque a tu grupo, van a buscar simulando estar en Estados Unidos o en Rusia; es importante que no comparen los resultados con los grupos del otro país hasta que termine la actividad.

Para empezar tenés que hacer cualquier búsqueda y después vas a poder cambiar de país. En la imagen se ve cómo elegirlo: tenés que hacer click en la sección de arriba a la izquierda, y cuando se despliega el menú de países, elegir el que fue asignado a tu grupo.



Una vez seleccionado el país correspondiente, tienen que buscar la frase "quién fue el primer hombre que llegó a la Luna" y seleccionar tres recursos (pueden ser artículos, videos, etc.) que les llamen la atención y explicar brevemente qué cuentan.

Recurso 1	
Título	

Tipo de recurso	
Resumen del contenido	

<b>Recurso 2</b>	
Título	
Tipo de recurso	
Resumen del contenido	

<b>Recurso 3</b>	
Título	
Tipo de recurso	
Resumen del contenido	

Cuando el profesor/a te lo indique, acercate con tu grupo a otro al que le haya tocado configurar el buscador desde otro país y comparen los resultados. ¿Arribaron a historias muy diferentes? ¿Por qué creés que dos personas que buscan en Internet pueden llegar a resultados distintos? ¿Además del país, se les ocurren otros factores que condicionen los resultados de una búsqueda?

Discutan las preguntas para poder pensarlas entre todos.

## Actividad 2

### Información ¿Personal?

<p>Objetivo</p> <ul style="list-style-type: none"><li>• Tomar conciencia de la información que hay disponible en la red y de la posibilidad de ser accedida por cualquier persona.</li><li>• Comprender que todo lo que se sube a Internet puede ser usado con cualquier fin, y que es imposible controlarlo.</li></ul>	<p>Modalidad de trabajo</p> <p>Grupal e individual</p>
<p>Materiales</p> <ul style="list-style-type: none"><li>• Computadoras con acceso a Internet</li></ul>	

### Desarrollo

Vamos a desarrollar esta actividad en torno a los datos personales que se pueden acceder desde Internet y cómo ser más cuidadosos al momento de compartir información personal en las redes.

Planteamos unas preguntas disparadoras: ¿Qué son los datos personales? ¿Cuántos de esos datos se encuentran en Internet? ¿Yo elegí compartirlos? Esperamos que surjan algunos tipos de información como DNI, nombre, apellido, edad, dirección, entre otros y que muchas veces pueden accederse en Internet incluso en contra de la voluntad de la persona propietaria de esos datos.

En grupos de a dos, y utilizando celulares o computadoras y las herramientas de búsqueda que nos brinda Internet (incluidas las redes sociales, pero sin loguearse para usar la amistad de Facebook ni de ninguna otra red social en que los estudiantes estén vinculados entre sí) cada estudiante va a buscar información de su compañero y algún adulto de su familia, como puede ser el nombre, edad, domicilio, lugar de estudios, entre otros. Con esta información contestarán las preguntas planteadas: ¿Qué datos tiene la web sobre cada uno? ¿Cuáles fueron ingresados intencionalmente por la persona que fue investigada, y cuáles no?

Haremos una puesta en común para toda la clase donde conversaremos acerca de las conclusiones de los grupos y la necesidad de protección de los datos personales, concientizando sobre la información que subimos a Internet y el carácter “público” que adquiere. Nos centraremos en la responsabilidad que tenemos en la publicación tanto de datos propios, como aquella información que publicamos o comentamos de otras personas, incluso desconocidas, como videos, fotos, comentarios, memes, etc. También haremos foco en la voluntad de compartir esa información ¿Qué datos fueron ingresados intencionalmente? ¿Estaban destinados a todas las personas que pudieron acceder a ellos? Dentro de este intercambio vamos a establecer relaciones con algunos contenidos abordados en actividades anteriores. Al utilizar servicios en Internet, por ejemplo navegar por la web, uno brinda información de manera no consciente. Esto sucede porque la computadora o el celular envían constantemente datos a los servidores. Algunos ejemplos de esto son los siguientes:

- Las computadoras tienen un número IP que las identifica. Cuando visitamos un sitio web, esta "etiqueta" es registrada por el servidor que nos brinda el acceso al sitio. Esto permite determinar cuáles son los sitios que fueron visitados desde una misma computadora.
- Algunos sitios almacenan pequeños archivos en nuestra computadora, denominados "cookies", que registran información sobre las actividades que realizamos en ellos. De esta forma, los sitios web pueden detectar cómo navegamos y registrar nuestros perfiles de comportamiento.
- El GPS de los celulares permite que el dispositivo pueda informar los lugares por los que circulamos, información que también puede ser almacenada y hasta accedida por terceros. Por ejemplo, la aplicación Google Maps tiene una funcionalidad llamada "Tu cronología", que indica, si dispone de los datos, en qué lugares estuvo el usuario en un día determinado.

Luego de estos intercambios, proyectaremos un video que nos permitirá completar el tema entre todos, reflexionando sobre la manipulación que se puede realizar con los datos que se encuentran en las redes. El video muestra a una persona que simula ser un adivino que a partir de simples datos como el nombre de la persona entrevistada, acierta detalles de la vida de las personas a las que interroga. El impacto que genera el video es descubrir, al finalizar la entrevista con cada persona, que trabaja con un equipo de personas que realizan la búsqueda de información en Internet, brindando al supuesto adivino datos certeros. La idea es concientizar acerca del peligro de la información que subimos a la red y de su carácter público una vez que lo hicimos. El video, titulado "Adivino Dave - Privacidad Redes Sociales", puede encontrarse en el siguiente link:

<https://www.youtube.com/watch?v=qmnhclKgVg0>.

### Cierre:

Generalmente los alumnos no tienen información clara sobre los peligros y la necesidad de la protección de los datos personales que circulan en la web. Intentaremos que el curso tome conciencia de que el alcance de la información es muy difícil de saber, por esta razón hay que ser sumamente cuidadosos al momento de publicarla.

Para concluir esta actividad abriremos un debate acerca de lo que sintió cada estudiante durante los ejercicios, qué opinan sobre el derecho a la privacidad que tiene cada persona y cómo se ve afectado por lo visto anteriormente. ¿Cómo les parece que puede afectar que haya tanta información sobre una persona en Internet? ¿Podría alguien malintencionado utilizar esa información para su provecho? Si aparecen alumnos apáticos respecto a sus datos, concientizar acerca del etiquetamiento de otras personas o de brindar información sobre salidas o costumbres, por ejemplo. Se puede ejemplificar con una situación posible: si publicamos que salimos 15 días de vacaciones o que recibimos muchos regalos costosos, alguien puede usar esa información para decidir entrar en nuestra casa a robar. Otro aspecto importante es respetar la voluntad de las personas que eligen compartir información con nosotros. Que nos manden fotos, videos o lo suban a las redes que tenemos en común de manera privada debería ser motivo suficiente para no difundirlas. Organizaremos el debate a través de las preguntas buscando que se tome dimensión de la

<Program.AR/>

importancia del uso responsable de la información en Internet y de respetar la voluntad de otras personas de no compartir su información más allá del destinatario original.

## Ficha 2 para el alumno

### Información ¿Personal?

Muchas veces existe publicada en Internet mucha más información sobre nosotros que la que creemos. En esta actividad vamos a buscar en Internet la mayor cantidad de datos de un compañero que podamos encontrar, completando la ficha que se adjunta a continuación.

Nombre del compañero: \_\_\_\_\_

Datos que pueden encontrarse en la web	¿Fueron ingresados intencionalmente?

Para esta actividad podemos utilizar las redes sociales (Facebook, Instagram, Twitter, etc.), buscadores, o cualquier otra forma de encontrar información que esté disponible en Internet. Podés comenzar buscando el nombre de esa persona en Google, y ver si tiene perfil de Facebook, o si tiene cuentas en Google u otros correos, y también ver si tiene fotos públicas (el buscador de Google busca exclusivamente fotos en su pestaña "Imágenes"). Fijate si podés obtener el DNI, la dirección, el CUIL, lugares que visitó, quiénes son sus familiares o amigos, y si podés conseguir fotos de esa persona (sin usar tu cuenta de Facebook, Instagram o cualquier otra red social, porque la idea es buscarlo como si no lo conocieras).

En caso de que no encuentres nada, preguntale el nombre de algún adulto de su familia y fijate si existen datos comerciales de ellos. Muchas veces se filtra la información que tienen los bancos y otras entidades comerciales que utilizan los adultos.

Para terminar, si tenés cuenta en Google (lo cual es posible si tenés un teléfono con Android), entrá en tu cuenta y en la aplicación Google Maps buscá la opción "Mi cronología". Si la tenés activada vas a descubrir que Google sabe mucho más de lo que imaginabas sobre los lugares donde estuviste.

#### ¿Sabías que...?

Muchos sitios de Internet almacenan pequeños archivos llamados 'cookies' en nuestras computadoras, registrando información sobre las actividades que realizamos.

Podés editar la configuración de tu navegador de Internet para tener control sobre tu historial y editar las cookies que se almacenan en tu computadora, habilitando o deshabilitando las mismas. Al deshabilitar las cookies hacemos más difícil que los sitios recolecten información sobre nosotros, pero al mismo tiempo muchos de los sitios que visitamos, como Facebook, las precisan para recordar ciertas preferencias.

### Modelo de evaluación



## Armamos una red... pero de conceptos

<p>Objetivo</p> <ul style="list-style-type: none"><li>• Realizar una síntesis integradora de los conceptos del capítulo completando un mapa conceptual a modo de rompecabezas.</li></ul>	<p>Modalidad de trabajo</p> <p>Grupos de a 4</p>
--	--

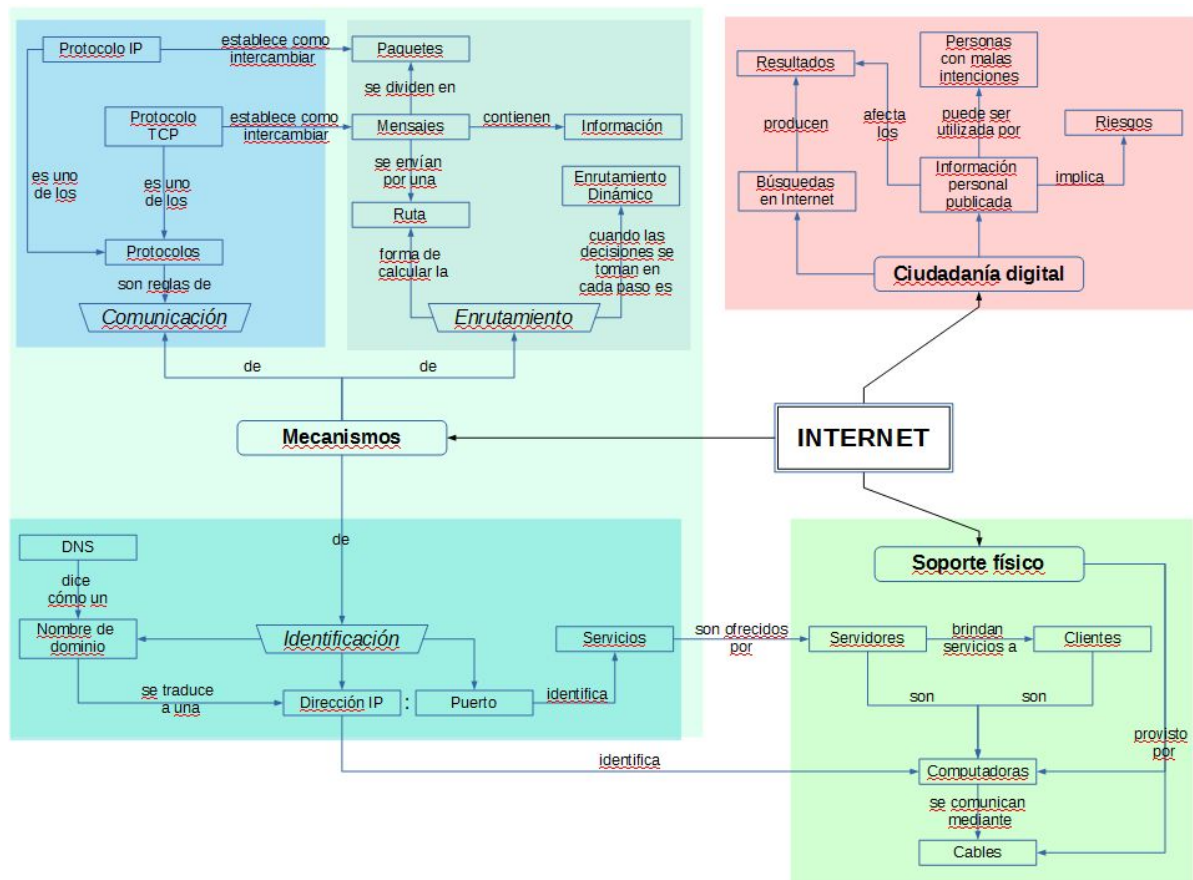
### Desarrollo

Para esta actividad de evaluación, vamos a formar grupos de 4 estudiantes como mínimo. Cada grupo debe completar un mapa conceptual según las instrucciones de la Ficha, conectando la palabra clave INTERNET con los diferentes conceptos abordados durante el capítulo: ¿Cómo viaja la información? ¿Cuáles son los elementos físicos que posibilitan el tránsito de la información en Internet? ¿Cómo influye la tecnología informática en los aspectos sociales y culturales?

Para presentar la actividad explicamos que un mapa conceptual es una representación gráfica de los conceptos de un tema que nos muestra la jerarquía de los mismos y las relaciones entre ellos. En la Ficha se encuentra un mapa conceptual donde los conceptos fueron separados como piezas de un rompecabezas y los estudiantes deben volver a poner en su lugar. Al hacerlo, deben tenerse en cuenta las jerarquías de conceptos y cómo cada concepto se relaciona con otros en diversas categorías.

Luego de unos minutos para que la clase comprenda y lea el mapa conceptual propuesto y sus elementos, es conveniente que recomendemos al curso que además de leer con atención los textos que conectan las flechas observen la forma de las piezas, porque es un detalle que también brinda pistas sobre la forma de vincular los conceptos. También les podemos sugerir que a medida que escriban en los espacios vacíos del mapa tachen las piezas que utilizan. Si pasado cierto tiempo, y a pesar de estas ayudas, resulta difícil completar el trabajo, daremos pistas adicionales, por ejemplo dónde colocar algunas de las piezas que pueden resultar más dificultosas (Por ejemplo "DNS", "Resultados", "Clientes" y "Riesgos").

El resultado esperado es el siguiente:



## Cierre

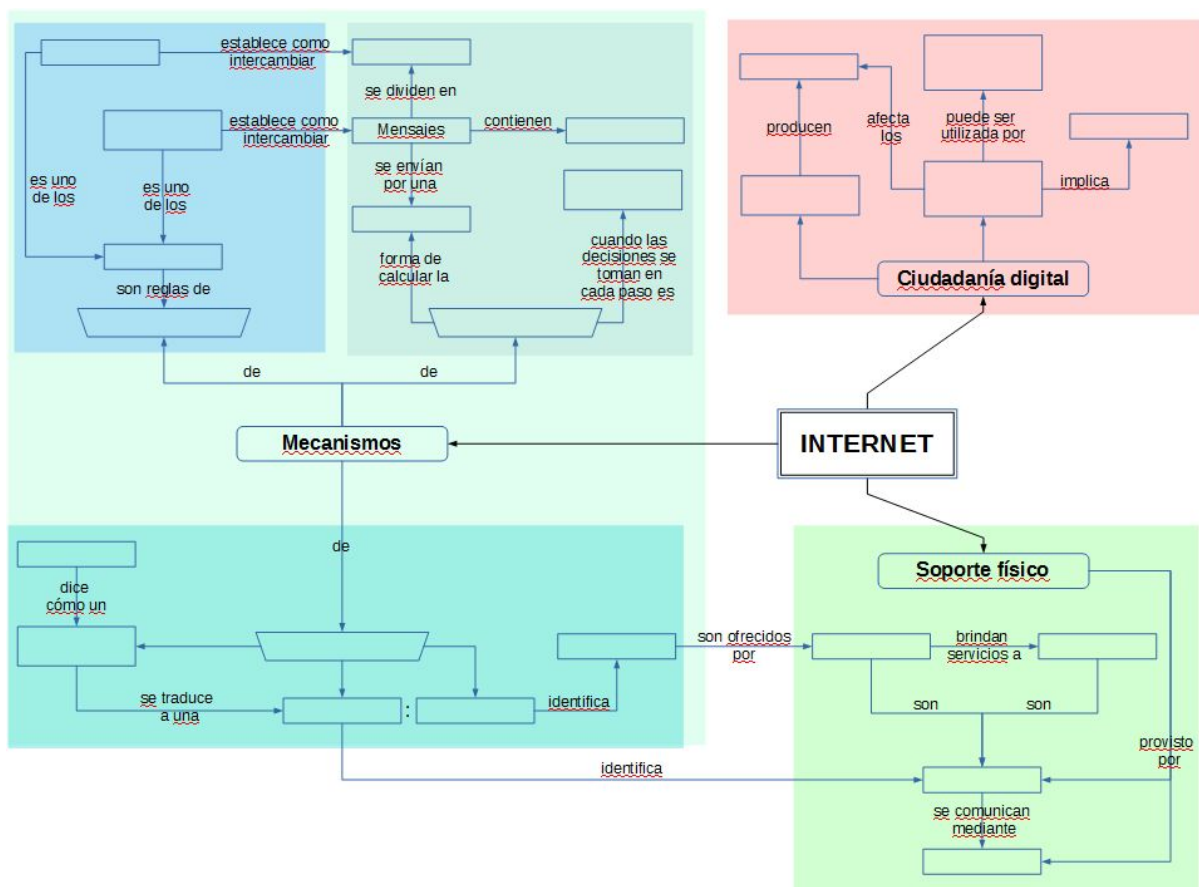
Para concluir, vamos a hacer una puesta en común compartiendo las construcciones realizadas, verificando si se armaron correctamente y reconociendo las diferencias que puedan aparecer para discutir entre todos el significado del mapa, fundamentar las decisiones que se tomaron y estableciendo relaciones entre cada concepto con las actividades correspondientes realizadas durante el capítulo.

## Ficha para el alumno

Armamos una red... pero de conceptos

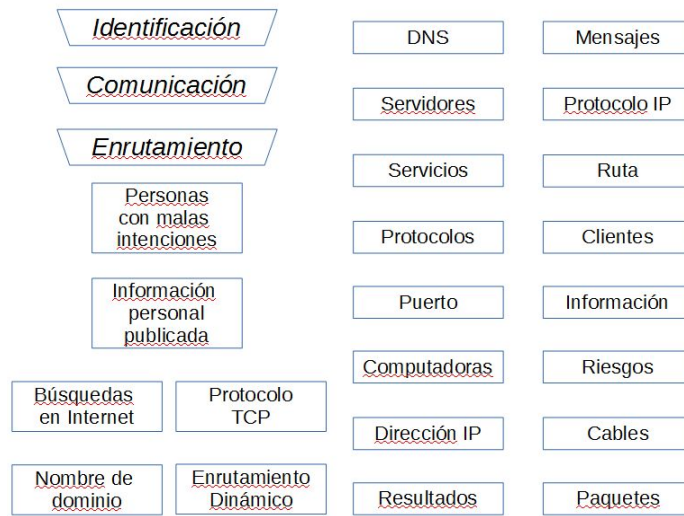
Para terminar el capítulo habíamos armado un mapa conceptual (o sea, una representación gráfica de ciertos conceptos y las relaciones entre ellos) con todas las ideas que trabajamos en él, pero tuvimos un accidente, ¡y todos los conceptos se volaron del mapa! Ayudanos a reconstruirlo, poniendo cada una de las piezas que representan conceptos de manera tal que cada idea quede asociada correctamente con las demás. Las piezas representan las ideas que trabajamos en las diferentes actividades del capítulo, todo lo que fuimos trabajando en clase va a servir de ayuda para completarlo.

Para hacer el trabajo vas a discutir en un grupo de al menos 4 integrantes dónde poner cada pieza en base a lo que significa y a cómo se vincula con las demás.



Las piezas para completar el mapa conceptual son las siguientes. Pueden recortarlas o escribirlas para reconstruir el mapa.

Al terminar conversen entre todos sobre el resultado final y sobre la relación de cada idea con lo trabajado en el capítulo para poder compartirlo con el resto de sus compañeros.



# ANEXO II

---

# SISTEMAS OPERATIVOS

## **SECUENCIA DIDÁCTICA 1:**

¿QUÉ ES UN SISTEMA OPERATIVO?

Imprimiendo una O  
Filósofos comensales

---

## **SECUENCIA DIDÁCTICA 2:**

INTERFACES DE USUARIO

Explorando archivos  
Explorando por línea de comandos

---

## **SECUENCIA DIDÁCTICA 3:**

SEGURIDAD

Usuarios, permisos y privilegios  
¿Qué es un virus?

---

## **SECUENCIA DIDÁCTICA 4:**

¿QUÉ SISTEMAS OPERATIVOS HAY Y EN QUÉ SE DIFERENCIAN?

Open Cola  
Sabores de sistemas operativos

---

## **MODELO DE EVALUACIÓN**

Permisos en la escuela

---

## **GLOSARIO**

Una computadora de escritorio moderna tiene en general un procesador principal -o varios-, memoria RAM, discos rígidos, placa de red, monitores, teclado, mouse y otros dispositivos. Dependiendo del tipo de computadora y su uso, podemos encontrar distintas combinaciones; por ejemplo, esperamos ver un teclado y un mouse en una computadora de escritorio, mientras que nos conformamos con una pantalla táctil en una tablet. Llamamos **plataforma** al conjunto de componentes compatible con el propósito de un dispositivo dado.

Si al programar tuviéramos que tener en cuenta todas las combinaciones posibles de dispositivos y las particularidades de cada fabricante, nuestra tarea sería muy compleja. Para reducir esta complejidad las computadoras utilizan una pieza de software llamada el **sistema operativo**, que se encarga de proveer una interfaz homogénea a los distintos tipos de plataformas, es decir, que quien programa no tenga que conocer los detalles de cada dispositivo y fabricante, y administrar los recursos disponibles. El objetivo de este anexo es motivar y presentar la noción de sistema operativo.

## Secuencia didáctica 1

### ¿Para qué sirve un sistema operativo?

El **sistema operativo** es una pieza de software que administra los recursos de hardware de una computadora. A diferencia de los programas que venimos construyendo, el sistema operativo comienza a ejecutarse cuando prendemos la computadora y continúa en ejecución hasta que se apaga. Su primera función es la de detectar todos los dispositivos de hardware del equipo. Luego se dedica a cumplir otras dos funciones muy importantes: permitir a otros programas usar los dispositivos de forma simple, y administrar cuidadosamente los recursos disponibles para evitar conflictos entre varios programas que necesitan los mismos recursos.

El sistema operativo centraliza todo los accesos a los dispositivos de hardware. Por ejemplo, el **sistema de archivos** es un componente del sistema operativo que permite leer y escribir datos organizados en archivos. Un programa que desea leer un archivo no necesita saber si éste está almacenado en un disco rígido magnético, un DVD o una memoria flash; simplemente le indica al sistema operativo qué archivo desea leer y obtiene los datos. El sistema operativo posee distintos procedimientos para acceder a cada tipo de dispositivo, llamados **controladores** (o en inglés, *drivers*), y utiliza el que corresponde en cada caso.

Otras funciones de un sistema operativo tienen un impacto menos visible para un usuario. Por ejemplo, los programas pueden usar más memoria RAM de la que está disponible físicamente. Cuando la memoria real no es suficiente, el sistema operativo automáticamente escribe en disco los datos que no están actualmente en uso para liberar espacio, y cuando son requeridos los vuelve a leer del disco a la memoria. Esto se llama **memoria virtual**.

Además, la mayoría de los sistemas operativos permiten que ejecutemos múltiples programas simultáneamente. Cada programa en ejecución es llamado **proceso**. El **planificador** es el módulo del sistema operativo que se encarga de asignar tiempo de ejecución a los distintos procesos. Los procesos quedan pausados mientras no son elegidos, pero los cambios en la selección suceden tan rápidamente que no llegamos a notarlo. Para evitar problemas, el sistema operativo asigna a cada proceso un sector de memoria propio, evitando que puedan leer y escribir los datos de otro proceso; estos sectores se llama **páginas** y dan más seguridad y estabilidad al software. En resumen, el sistema operativo es el encargado de administrar los recursos de hardware en una computadora y proveer una abstracción adecuada para hacer uso de esos recursos.

#### Objetivos

- Presentar las principales funciones del sistema operativo.
- Comprender la necesidad de disponer de la capa de abstracción provista por los sistemas operativos y algunos de los problemas que pueden y no pueden resolver.

## Actividad 1

### Imprimiendo una O

<p>Objetivo</p> <ul style="list-style-type: none"><li>• Presentar al sistema operativo como un mediador para interactuar de una manera uniforme con distintos dispositivos.</li></ul>	<p>Modalidad de trabajo</p> <p>Individual</p>
<p>Materiales</p> <ul style="list-style-type: none"><li>• Computadoras con Gobstones</li><li>• Los proyectos "Imprimiendo una O con Julio Paca", "Imprimiendo una O con Brote-r" e "Imprimiendo una O con Ceros"</li></ul>	

### Desarrollo

En esta actividad se presentan tres modelos de impresoras cada una con un conjunto diferente de operaciones primitivas para imprimir, representadas cada una en un proyecto diferente. Debe señalarse que las tres impresoras tienen la misma capacidad para imprimir las mismas figuras, pero los procedimientos que ofrecen para hacerlo son distintos. En esta actividad el propósito es escribir tres programas distintos, uno para cada impresora, que permita imprimir una 'O'; posibles soluciones para cada una se presentan en la Figura 1.

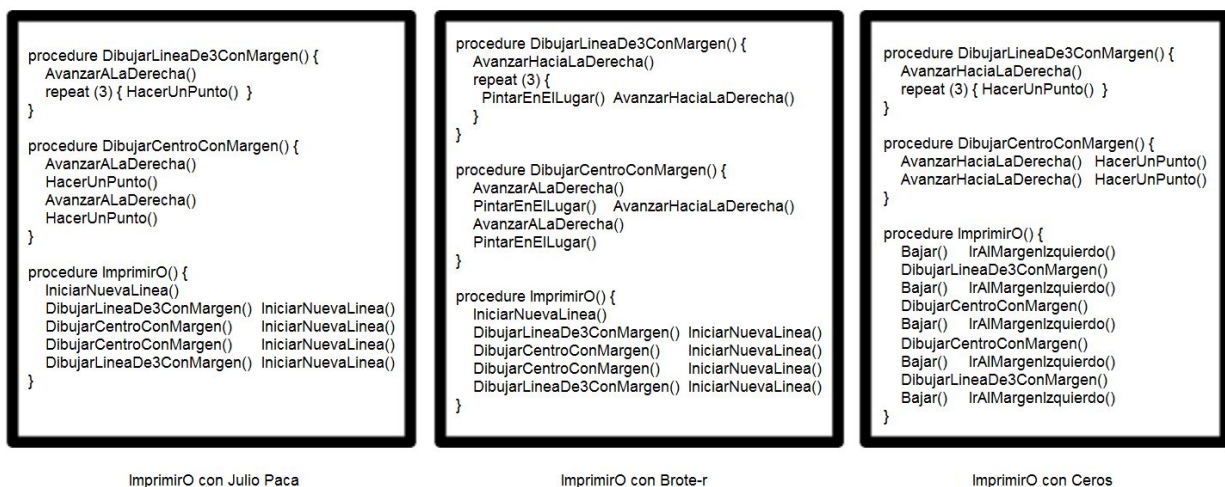


Figura 1. Soluciones para imprimir la O con cada una de las impresoras

Una vez terminados los 3 procedimientos para **ImprimirO** en cada uno de los proyectos, reflexionar sobre las diferencias y similitudes en los mismos y responder las siguientes preguntas:

- **¿Por qué cada tipo de impresora tiene diferentes comandos primitivos?**
  - Porque cada fabricante elige los mismos de manera independiente.
- **¿Cómo afecta esto la tarea de programar?**
  - Los programas pedidos son parecidos, pero cada uno es diferente.
- **¿Cómo podríamos lograr no ver las diferencias entre las impresoras?**
  - Si hubiese un único conjunto de comandos para dibujar, y una forma de traducir ese conjunto a los comandos primitivos de cada impresora.

El **sistema operativo** es un programa especial, encargado de tener un único conjunto de comandos para imprimir y realizar la traducción a cada una de las impresoras. Por ejemplo, el sistema operativo puede ofrecer como interfaz única:

- **HacerUnPunto**, que dibuja un punto y avanza hacia la derecha.
- **AvanzarALaDerecha**, que avanza hacia la derecha sin dibujar.
- **IniciarNuevaLinea**, que inicia una línea nueva, comenzando desde el margen izquierdo.

Al conectarse con cada impresora, detecta de qué tipo es, y traduce cada uno de estos procedimientos en los comandos primitivos propios de la impresora. De esta forma, solo es necesario realizar un único programa para imprimir, pero cada una de las impresoras recibe los comandos adecuados. Lo mismo sucede con otros tipos de dispositivos, como por ejemplo los de almacenamiento, discos magnéticos, de estado sólido y memorias SD, dispositivos de entrada, teclados, mouse, etc.

## Cierre

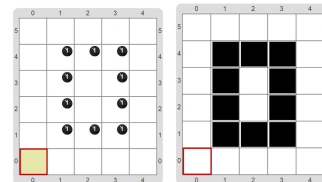
Como conclusión reflexionar sobre la facilidad que provee el sistema operativo al ofrecer una única interfaz para manejar cualquier impresora y lo mismo para otros dispositivos, y lo complejo que resultaría trabajar sin él sabiendo que hay miles de modelos y marcas diferentes de impresoras y de todos los dispositivos que usamos a diario.



## Ficha 1 para el alumno

### Imprimiendo una O

Abri de a uno los proyectos “*Imprimiendo una O con Julio Paca*”, “*Imprimiendo una O con Brote-r*” e “*Imprimiendo una O con Ceros*” y mirá qué procedimientos traen definidos. Cada uno de los procedimientos primitivos de cada proyecto representan los comandos primitivos de 3 impresoras de distintos modelos, a los que vamos a llamar Julio Paca, Brote-r y Ceros para poder distinguirlas. El propósito de la actividad es escribir el procedimiento **ImprimirO**, que “imprima”, o sea, dibuje con bolitas, una “O” como la que se muestra en la figura, en cada uno de los tres proyectos. Cada uno de estos procedimientos tiene que utilizar una de las “impresoras”, o sea, utilizar solamente los procedimientos que representan los comandos primitivos de una de las impresoras.



Los procedimientos de cada una de las impresoras son los siguientes:

Julio Paca:

- **HacerUnPunto**, que dibuja un punto y avanza hacia la derecha
- **AvanzarALaDerecha**, que avanza hacia la derecha sin dibujar
- **IniciarNuevaLinea**, que baja una línea, yendo al margen izquierdo

Brote-r:

- **PintarEnElLugar**, que pinta un punto y se mantiene en el mismo lugar
- **AvanzarHaciaLaDerecha**, que avanza hacia la derecha sin dibujar
- **IniciarNuevaLinea**, que baja una línea, yendo al margen izquierdo

Ceros:

- **HacerUnPunto**, que dibuja un punto y avanza hacia la derecha
- **AvanzarHaciaLaDerecha**, que avanza hacia la derecha sin dibujar
- **Bajar**, que baja a la siguiente línea manteniéndose en la misma columna
- **IrAlMargenIzquierdo**, que va al borde izquierdo manteniéndose en la misma fila.

Al terminar, el docente te va a guiar en una reflexión sobre la solución a la que llegaste.

## Actividad 2

### Filósofos comensales

<p>Objetivo</p> <ul style="list-style-type: none"><li>• Comprender alguno de los problemas que pueden surgir cuando múltiples procesos utilizan recursos compartidos.</li></ul>	<p>Modalidad de trabajo</p> <p>Grupos de 5 personas</p>
<p>Materiales</p> <ul style="list-style-type: none"><li>• Palillos</li></ul>	

### Desarrollo

Los sistemas operativos usualmente trabajan con muchos programas ejecutando al mismo tiempo. A cada uno de esos programas en ejecución lo llamamos un proceso. Cada proceso requiere recursos, y es el sistema operativo el encargado de proveer mecanismos para asignar los recursos compartidos.

Un problema clásico que surge al tener múltiples procesos ejecutando en un contexto con recursos compartidos es el de que dos o más procesos quedan detenidos esperando a otros, que a su vez están esperando al primero. Una forma clásica de representar este problema es simulando cada proceso mediante un filósofo que solamente sabe pensar y comer, y recreando el problema con este dominio.

En esta actividad, los estudiantes deberán actuar como dichos filósofos, que solamente piensan y comen. Para comer deben conseguir 2 palillos, que se encuentran uno a su izquierda y uno a su derecha, y de forma que cada palillo puede ser utilizado por dos filósofos; ver la Figura de la Ficha. Hay que guiar a los alumnos a descubrir que si todos los filósofos toman el palillo a su izquierda y luego intentan tomar el palillo a la derecha se puede generar una espera circular donde todos esperan por siempre a que otro termine de usar el palillo, pero nadie los puede utilizar porque no tiene dos. Para solucionar este problema basta con que uno de los filósofos tome los palillos en el orden inverso, es decir, primero el de la derecha y luego el de la izquierda.

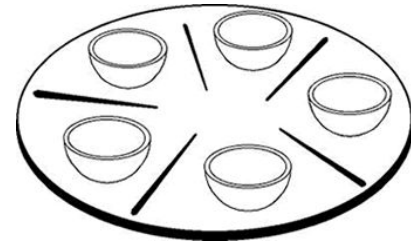
### Cierre

Para completar la actividad, explicar que el sistema operativo provee los mecanismos necesarios para la asignación de recursos compartidos (en el ejemplo los palillos), pero que a pesar de que la asignación de recursos se realiza correctamente, pueden existir problemas como el de la espera circular. Es tarea de los programadores detectar estos problemas en los programas y corregirlos.

## Ficha 2 para el alumno

### Filósofos comensales

En esta actividad vas a cumplir el papel de un filósofo que se reúne con otros a comer arroz y filosofar. Los filósofos alternan entre las actividades de filosofar y comer. Para comer necesitan dos palillos, el que tienen a su izquierda y a su derecha, pero con la particularidad de que cada palillo se comparte con el filósofo de al lado. Cuando un filósofo intenta tomar un palillo que está en uso, espera, sin hacer nada más, hasta que se libere. La imagen muestra la mesa donde los filósofos se acercan a comer, teniendo cada uno un único lugar asignado.



El siguiente código muestra cómo es la actividad del filósofo:

Filósofo:

Repetir por siempre:

Filosofar

Tomar el palillo de la izquierda

Tomar el palillo de la derecha

Comer

Dejar el palillo de la izquierda

Dejar el palillo de la derecha

Respondé las siguientes preguntas:

¿Es posible que se llegue a un punto en donde al menos un filósofo no pueda comer nunca?

---

¿Cómo puede evitarse este problema, modificando el comportamiento de al menos un filósofo?

---

**Pista:** ¿qué pasa si todos tratan de comer al mismo tiempo?

## Secuencia didáctica 2

## Interfaces de usuario

Los sistemas operativos proveen múltiples funcionalidades a sus usuarios, pero para poder acceder a ellas es necesaria una pieza de software llamada **interfaz** o cáscara (*shell*, en inglés). La interfaz es el componente que hace visible el sistema al usuario, y puede ser de dos tipos: gráfica o de línea de comandos. Las **interfaces gráficas** están diseñadas para ser fáciles de aprender y usar, mientras que las **interfaces de línea de comandos** están diseñadas para la automatización de tareas.

Las interfaces suelen estar diseñadas teniendo en cuenta el tipo de plataforma donde serán utilizadas. Así, la de una tablet o un celular está orientada a actuar a través de una pantalla táctil, mientras que la de una computadora de escritorio está mejor preparada para utilizar un *mouse*. Dependiendo del uso esperado, un sistema operativo podría no incluir una interfaz gráfica; por ejemplo los sistemas operativos para servidores sólo proveen una interfaz por línea de comando.

La mayoría de las interfaces gráficas están basadas en la metáfora de un escritorio virtual, donde el usuario puede ejecutar programas, guardar y abrir archivos. El escritorio sirve como punto de entrada al resto de las funciones del sistema como el sistema de archivos y la administración de procesos. Múltiples procesos pueden ser utilizados simultáneamente alternando entre ventanas.

### Objetivos

- Distinguir a la interfaz gráfica de un sistema operativo tan sólo como un módulo más.
- Introducir la noción de sistema de archivos jerárquico.

## Actividad 1

### Explorando archivos

<p>Objetivos</p> <ul style="list-style-type: none"><li>• Comprender y acceder a la estructura de carpetas en sus computadoras.</li><li>• Distinguir dispositivos y carpetas importantes.</li></ul>	<p>Modalidad de trabajo</p> <p>Individual</p>
<p>Materiales</p> <ul style="list-style-type: none"><li>• Un sistema operativo con interfaz gráfica (e.g. Windows, Linux)</li></ul>	

## Desarrollo

En esta actividad los alumnos deben usar la interfaz del sistema operativo que dispongan en sus computadoras para hacer una serie de tareas simples. La solución variará de computadora en computadora, especialmente si tienen distintos sistemas operativos instalados. Además, la experiencia en el manejo de la computadora puede impactar en lo rápido que resuelvan la actividad. Es necesario identificar a los alumnos con menos experiencia en el uso de la interfaz y guiarlos en la resolución del ejercicio. A continuación se presentan soluciones esquema para sistemas operativos comunes.

### En Windows:

Hay una raíz por cada disco rígido y partición. Normalmente como mínimo tiene una unidad llamada C que actúa como raíz del sistema de archivos donde está instalado el sistema operativo, como se puede observar en el lado izquierdo de la Figura 2.

El sistema de archivos, como se puede observar en el lado derecho de la Figura 2, tiene la siguiente estructura:

C: → *raíz del sistema de archivos*

| -C:\Archivos de programas

→ *los programas instalados por usuarios*

| -C:\Datos de programa

→ *datos de los programas instalados*

| -C:\Usuarios

→ *archivos de los usuarios del sistema*

| --C:\Usuarios\Alan1

→ *carpeta del usuario Alan1*

| --C:\Usuarios\Alan1\Descargas

→ *descargas de Alan1*

| --C:\Usuarios\Alan1\Escritorio

→ *archivos en el escritorio de Alan1*

| --C:\Usuarios\Alan1\Mis Documentos

→ *documentos de Alan1*

| --C:\Usuarios\Alan1\Público

→ *archivos públicos de alan1*

| -C:\Windows

→ *archivos del sistema operativo*

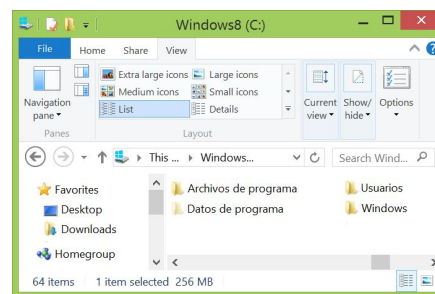
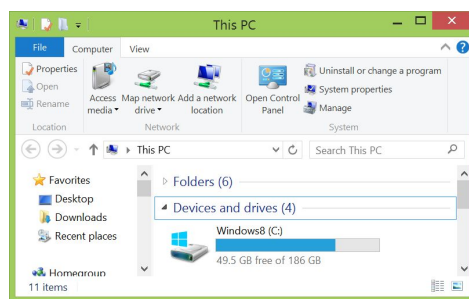


Figura 2. Ejemplos del sistema de archivos de Windows, en su interfaz gráfica.

### En Linux:

La raíz del sistema de archivos es el carpeta /.

El sistema, como se observar en la Figura 3, tiene la siguiente estructura:

/ → *raíz del sistema de archivos*

| -/bin → *contiene archivos ejecutables del sistema*

- |-/boot → *contiene la configuración de arranque del equipo*
- |-/cdrom → *aquí aparece el contenido del cdrom si se inserta uno*
- |-/dev → *contiene información sobre dispositivos*
- |-/etc → *contiene la configuración de distintos programas*
- |-/home → *contiene los archivos de cada usuario*
- |-/home/alan1 → *contiene los archivos del usuario alan1*
- |--/home/alan1/Desktop → *archivos en el escritorio de alan1*
- |--/home/alan1/Documents → *documentos de alan1*
- |--/home/alan1/Downloads → *descargas de alan1*
- |--/home/alan1/Public → *archivos públicos de alan1*
- |-/lib → *contiene funcionalidad compartida por muchos programas*
- |-/media → *contiene los dispositivos de almacenamiento detectados por el sistema*
- |-/mnt → *contiene los dispositivos de almacenamiento asignados por los usuarios*
- |-/proc → *contiene información sobre los procesos en ejecución*
- |-/root → *carpeta del usuario root*
- |-/sys → *contiene información de los dispositivos del sistema*
- |-/tmp → *contiene archivos temporales, se borran cada vez que se prende el equipo*
- |-/usr → *contiene archivos útiles a múltiples programas*
- |-/var → *contiene archivos de log del sistema*

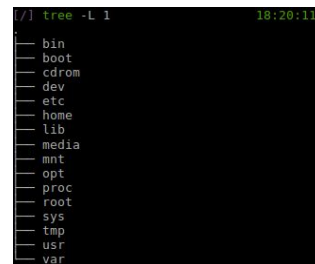
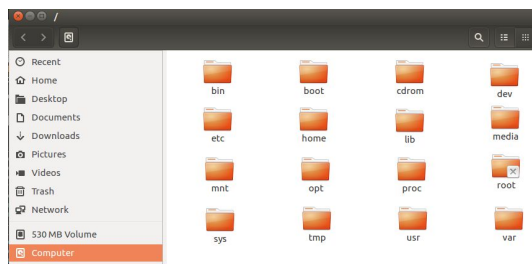


Figura 3. Ejemplos del sistema de archivos de Linux, en la interfaz gráfica y en la de línea de comandos.

En ambos casos ejecutar un programa es tan simple como hacer doble click sobre el archivo (aunque puede configurarse para hacerlo de otra forma). Al hacer doble click sobre un archivo no ejecutable el sistema operativo busca un programa que sepa abrir ese archivo y lo ejecuta, si no lo encuentra le pregunta al usuario con qué programa abrir el archivo.

## Cierre

El sistema operativo realiza múltiples funciones, pero principalmente se encarga de la administración de los recursos de una computadora. Los usuarios interactuamos con el sistema operativo a través de su interfaz, que es simplemente un programa diseñado para facilitarnos la tarea de utilizar la computadora. En particular, la mayoría de los sistemas operativos proveen una forma de manipular archivos en una estructura de carpetas jerárquica.

## Ficha 1 para el alumno

### Explorando archivos

En esta actividad tenés que explorar el contenido del sistema de archivos de tu computadora utilizando la interfaz. Buscá cómo hacer las siguientes cosas en orden:

1. Abrió un explorador de archivos y carpetas.
2. Identificá los elementos raíces del sistema de archivos, es decir los elementos más externos que contienen archivos y carpetas. Indicá de qué tipo de elemento se trata; por ejemplo: carpeta, disco rígido, DVD o CD-ROM, etc.
3. Dibujá el árbol de carpetas de tu computadora con un nivel de profundidad (es decir entrando una vez en cada carpeta, pero sin entrar en las subcarpetas) partiendo de los elementos raíces identificados anteriormente.
4. Identificá la carpeta que contiene los archivos correspondientes a tu usuario y dibujá también dos niveles de subcarpetas a partir de allí.
5. ¿Podés explicar cuál es el propósito de cada carpeta? Considerá las siguientes opciones:
  - a. Contiene los programas instalados por usuarios.
  - b. Contiene archivos propios de los programas.
  - c. Contiene archivos propios del usuario.
  - d. Contiene archivos descargados de internet.
  - e. Contiene los archivos que están en el escritorio del usuario.
  - f. Contiene archivos compartidos entre múltiples usuarios.
6. ¿Cómo podés ejecutar un programa a partir de su archivo? ¿Qué sucede si hacés lo mismo sobre un archivo que no es un programa?

## Actividad 2

### Explorando por línea de comandos

Objetivo	Modalidad de trabajo
<ul style="list-style-type: none"><li>Tener un contacto básico con la interfaz por línea de comandos.</li></ul>	Individual
Materiales	
<ul style="list-style-type: none"><li>Interfaz por línea de comandos</li></ul>	

### Desarrollo

Esta actividad es similar a la anterior, pero en lugar de usar una interfaz gráfica se debe usar una interfaz por línea de comandos. Es probable que los alumnos nunca hayan usando este tipo de interfaces por lo que se los debe orientar especialmente al principio de la actividad. A continuación se presenta la secuencia de comandos a ejecutar para resolver la actividad:

#### En Linux:

```
<CTRL>+<ALT>+<F1> → para pasar a modo consola
cd / → para posicionarse en la raíz
ls → para listar los archivos y carpetas
cd home → para acceder a la carpeta home
cd alan1 → para acceder a la carpeta del usuario
ls → para listar los archivos y carpetas
<CTRL>+<ALT>+<F6> → para volver al modo gráfico (puede ser F7 u otro)
```

#### En Windows:

```
Inicio -> Ejecutar -> cmd → para abrir una consola
C: → para posicionarse en la unidad principal (puede ser otra)
cd \ → para posicionarse en la raíz
dir → para listar los archivos y carpetas
cd Usuarios → para acceder a la carpeta de usuarios
cd Alan1 → para acceder a la carpeta del usuario
dir → para listar los archivos y carpetas
```

### Cierre

De la misma forma que podemos manipular el sistema de archivos a través de la interfaz gráfica, podemos operar usando la interfaz por línea de comandos. No obstante, mientras que al usar la interfaz gráfica las operaciones resultan intuitivas, para usar la línea de comandos es necesario conocer los comandos disponibles.



## Ficha 2 para el alumno

### Explorando por línea de comandos

En esta actividad tenés que explorar el contenido del sistema de archivos de tu computadora utilizando una interfaz por línea de comandos. Buscá cómo hacer las siguientes cosas en orden:

1. Abrí una interfaz de línea de comandos.
2. Identificá y posicionate en uno de los elementos raíces del sistema de archivos (es decir los elementos más externos que contienen archivos y carpetas).
3. Listá los archivos y carpetas en el carpeta raíz.
4. Ingresá a la carpeta correspondiente a su usuario y listá los archivos y carpetas que se encuentran allí.

**Ayuda:** Tené en cuenta que para operar con la línea de comandos se debe tipear un comando como los siguientes:

1. `cd carpeta` → se posiciona en la carpeta dado, tanto en Windows como en Linux
2. `C:` → se posiciona en el disco C: en Windows
3. `ls` → lista archivos y carpetas de la carpeta actual en Linux
4. `dir` → lista archivos y carpetas de la carpeta actual en Windows

## Secuencia didáctica 3

## Seguridad

Como ya vimos, el objetivo principal del sistema operativo es administrar los recursos de una computadora para que estos puedan ser usados fácilmente por los usuarios. No obstante, también es deseable que el sistema operativo pueda garantizar el uso seguro de los recursos; por ejemplo, que no puedan accidentalmente borrarse archivos de otros usuarios, o modificar datos que otro usuario tiene en la memoria de la computadora. Con este fin, la mayoría de los sistemas operativos modernos tienen un componente de gestión de usuarios con la capacidad de asignar distintos permisos y privilegios.

Los privilegios pueden asignarse sobre dispositivos: por ejemplo permiso de usar una impresora o incluso sobre el sistema de archivos. Los permisos sobre archivos más comunes son los de lectura, escritura (que incluye la creación y eliminación de archivos) y de ejecución. Esto permite que múltiples usuarios usen el mismo equipo sin el riesgo de modificar los archivos de otro. Además, el mismo usuario puede acceder al sistema con cuentas distintas dependiendo del rol que desea cumplir; normalmente quien administra el sistema se encarga de instalar programas, mientras que los usuarios comunes simplemente los usan.

Para agilizar la administración de los permisos, los sistemas operativos permiten definir grupos de usuarios con los mismos permisos. De esta forma, la asignación de permisos puede hacerse de forma individual (usuario por usuario) o por grupos con características comunes. Algunos sistemas operativos ofrecen niveles de seguridad adicionales que operan de forma transparente para el usuario. Por ejemplo, sistemas de archivos virtuales, que generan la ilusión de que se está alterando el sistema de archivos, pero en su lugar se almacenan los cambios en una carpeta alternativa.

### Objetivos

- Presentar los esquemas de permisos para un uso seguro del equipo, disminuyendo las chances de ejecutar programas dañinos como virus o alterar accidentalmente el estado del sistema.
- Concientizar sobre que los esquemas de permisos son efectivos solo en la medida que los usuarios los utilicen responsablemente.

## Actividad 1

### Usuarios, permisos y privilegios

<p>Objetivos</p> <ul style="list-style-type: none"><li>• Comprender cómo se aplican los esquemas de permisos en un sistema de archivos jerárquico.</li><li>• Concientizar sobre la utilidad de hacer un uso correcto de los esquemas de permisos.</li></ul>	<p>Modalidad de trabajo</p> <p>Individual</p>
<p>Materiales</p> <ul style="list-style-type: none"><li>• Lápiz y papel</li></ul>	

### Desarrollo

En esta actividad se presenta un sistema de archivos hipotético, con una cantidad de carpetas, archivos y usuarios muy reducida; los datos del mismo se brindan en la Ficha. La consigna requiere identificar todos los archivos y carpetas y comprender cómo se regula el acceso a ellos mediante la asignación de permisos. Los permisos tienen la particularidad de que se pueden dar individualmente a cada usuario o a grupos de usuarios, dándole a un usuario la suma de sus permisos individuales más los permisos de los grupos a los que pertenece. Este último punto debe ser aclarado en clase.

A continuación damos las respuestas a los distintos puntos:

1.

Archivo\Usuario	Alicia	Roberto	Carmen
notas.txt	dueño	-	-
jijiji.mp3	-	dueño	-
selfie.png	-	-	dueño
editor	lectura y ejecución	lectura y ejecución	lectura
reproductor	lectura y ejecución	lectura y ejecución	-
visualizador	lectura y ejecución	-	lectura y ejecución
materias.tx	-	lectura	dueño

t			
examen.txt	dueño	-	lectura
rogue	lectura, escritura y ejecución	-	lectura, escritura y ejecución
wump	lectura y ejecución	-	lectura y ejecución

2. Sólo el usuario admin puede instalar programas en la carpeta /programas. No necesita asignar ningún permiso adicional porque la carpeta ya tiene permiso de lectura y ejecución para todos los usuarios. Ningún usuario puede borrar el programa porque él no tienen permisos de escritura.
3. Además de Admin, sólo los usuarios del grupo 3 pueden instalar juegos (Roberto y Carmen). Hay que asignarle permisos de lectura y ejecución, pero no de escritura para el resto de los usuarios del grupo 3.
4. Alicia puede crear una carpeta /usuarios/alicia/compartido, con permisos de lectura y escritura para Roberto, y sin esos permisos para Carmen. Puede asegurarse que sólo ellos y el usuario Admin pueden acceder a esos archivos, pero no hay forma de negarle el acceso a Admin.

### Cierre

Los sistemas operativos modernos suelen manejar múltiples cuentas de usuario. Cada usuario es propietario de sus archivos y tiene la potestad de elegir qué otros usuarios pueden acceder a ellos. Es recomendable que sólo el administrador del equipo tenga la capacidad de instalar programas que puedan alterar permanentemente el estado del equipo (de esta forma se puede evitar el daño causado por algunos virus). Si bien con los permisos se puede restringir el acceso, esto no garantiza la privacidad de los datos. El administrador del sistema siempre puede acceder a todos los archivos y cualquier persona con acceso al disco rígido puede hacerse pasar por administrador. Para asegurar la privacidad de los datos es necesario tomar medidas adicionales como el cifrado criptográfico que imposibilita la comprensión de los datos a pesar de que puedan ser leídos.

## Ficha 1 para el alumno

### Usuarios, permisos y privilegios

Un sistema posee los siguientes usuarios:

- Alicia
- Roberto
- Carmen
- Admin

Admin es un usuario especial del sistema encargado del mantenimiento y como tal tiene permiso para leer y escribir cualquier dato del sistema de archivos. Los otros usuarios son usuarios comunes a los que se les ha agrupado en grupos de trabajo con los siguientes permisos específicos:

- Alicia → grupos: 1 y 2
- Roberto → grupos: 2 y 3
- Carmen → grupos: 1 y 3

El sistema de archivos utilizado es jerárquico, es decir, las carpetas están incluidos dentro de otros salvo la carpeta raíz (/) que es la base del sistema de archivos. Cada archivo y carpeta tiene asociado un conjunto de permisos que salvo se indique lo contrario se hereda de la carpeta superior a los inferiores. Es decir, si un usuario tiene acceso a una carpeta también tiene acceso a los archivos contenidos en él, salvo que se explicita lo contrario.

A continuación se indica la estructura de carpetas del sistema y los permisos sobre cada archivo/carpeta:

/	→	dueño admin, lectura grupos 1, 2 y 3.
/usuarios/	→	dueño admin, lectura: 1, 2 y 3, escritura: -, ejecución: -
/usuarios/alicia/	→	dueño alicia, lectura: -, escritura: -, ejecución: -
/usuarios/alicia/notas.txt		
/usuarios/roberto/	→	dueño roberto, lectura: -, escritura: -, ejecución: -
/usuarios/roberto/jijiji.mp3		
/usuarios/carmen/	→	dueño carmen, lectura: -, escritura: -, ejecución: -
/usuarios/carmen/selfie.png		
/usuarios/admin/	→	dueño admin, lectura: -, escritura: -, ejecución: -
/programas/	→	dueño admin, lectura: 1 y 2, escritura: -, ejecución: 2
/programas/editor		
/programas/reproductor	→	dueño: admin, lectura: 2, escritura: -, ejecución: 2
/programas/visualizador	→	dueño: admin, lectura: 1, escritura: -, ejecución: 1
/datos/	→	dueño admin, lectura: 1 y 3, escritura: 1, ejecución: -
/datos/materias.txt	→	dueño: Carmen, lectura: Roberto
/datos/examen.txt	→	dueño: Alicia, lectura: 1
/juegos/	→	dueño admin, lectura: 3, escritura: 3, ejecución: 3
/juegos/rogue		
/juegos/wump	→	dueño admin, lectura: 3, escritura: -, ejecución: 3

Resolvé los siguientes puntos:

1. Dibujá una tabla donde para cada archivo se indiquen los permisos que tiene cada usuario sobre ese archivo.
2. Se desea instalar el programa "Nimda" en la carpeta de programas ¿Qué usuarios pueden hacerlo? ¿Qué permisos hay que asignarle para que sólo el dueño pueda borrarlo?
3. Se desea instalar el juego "snake" en la carpeta de juegos ¿Qué usuarios pueden hacerlo? ¿Qué permisos hay que asignarle para sólo el dueño pueda borrarlo?
4. Alicia está mirando unas fotos en la computadora. Luego llega Carmen y se suma a verlas. En un momento a Carmen le gusta mucho una foto y le pide a Alicia que le guarde una copia en su carpeta personal. Cuando Alicia intenta hacerlo obtienen un mensaje que dice "acceso denegado". ¿Porqué les parece que sucede esto? ¿si incluso la misma Carmen está presente, porqué puede ser que en este momento no tengan acceso a su propia carpeta?

## Actividad 2

### ¿Qué es un virus?

<p>Objetivos</p> <ul style="list-style-type: none"><li>• Comprender que el software puede tener propósitos maliciosos.</li><li>• Evaluar las medidas necesarias para reconocer amenazas a la seguridad.</li></ul>	<p>Modalidad de trabajo</p> <p>Investigación</p>
<p>Materiales</p> <ul style="list-style-type: none"><li>• Internet</li></ul>	

### Desarrollo

En esta actividad se propone un trabajo de investigación sobre software malicioso. Las soluciones variarán de trabajo en trabajo, pero se espera que contengan lo siguiente:

1. Un programa malicioso, por ejemplo un virus, es un programa destinado a interrumpir la operatoria de una computadora, recolectar datos sensibles, obtener acceso a la computadora o simplemente mostrar publicidad no deseada.
2. Entre los programas maliciosos se encuentran los siguientes tipos:
  - a. **Virus**, programas que se replican a sí mismos e intentan dañar el equipo, por ejemplo borrando archivos; atentan contra la disponibilidad e integridad de los datos.
  - b. Caballo de troya o **troyano**, programas que permiten el acceso remoto al equipo; atentan principalmente contra la confidencialidad de los datos.
  - c. **Spyware**, programas que recopilan datos sobre el usuario, por ejemplo contraseñas; atentan principalmente contra la confidencialidad de los datos.
  - d. **Adware**, programas que muestran publicidad no deseada; podemos decir que atentan contra la disponibilidad del libre uso del equipo.
3. En la mayoría de los casos los programas maliciosos son accidentalmente instalados por el usuario; suelen venir ocultos en otros programas de origen dudoso. Muy rara vez un hueco de seguridad en el sistema operativo puede permitir su ejecución remota.
4. Para evitar instalar programas maliciosos se puede tomar varias medidas:
  - a. No instalar programas de origen dudoso.
  - b. Instalar un antivirus y ejecutar un análisis regularmente.
  - c. Ejecutar programas dudosos en un entorno seguro o *sandbox*.

<Program.AR/>

## Cierre

Los virus son programas que por lo general son instalados por negligencia del usuario. El objetivo de los virus es vulnerar la seguridad de un sistema con distintos fines maliciosos. Los sistemas operativos y otras aplicaciones permiten al usuario defenderse de la gran mayoría de los virus, pero requieren de un uso consciente y responsable.



## Ficha 2 para el alumno

### ¿Qué es un virus?

La seguridad que esperamos al usar una computadora tiene tres pilares fundamentales:

- **Confidencialidad:** que sólo nosotros podamos acceder a nuestros datos.
- **Integridad:** que nuestros datos no se dañen o se borren.
- **Disponibilidad:** que siempre que lo deseemos podamos acceder a nuestros datos.

No obstante programas maliciosos denominados comúnmente “virus” atentan contra estos aspectos. Investigá sobre este tipo de programas y respondé las siguientes preguntas:

1. ¿Qué es un programa malicioso?

---

---

---

2. Describí los siguientes tipos de programas maliciosos, identificando además cuál de los tres atributos de seguridad vulneran:

Nombre	¿Qué es?	¿Qué atributo/s vulnera?
Virus		
Troyano		
Spyware		
Adware		

3. ¿Cómo puede llegar un programa malicioso a nuestra computadora?

---

---

---

<Program.AR/>

4. ¿Qué podemos hacer para evitar instalarnos un programa malicioso?

---

---

---

---

---

## Secuencia didáctica 4

## Seguridad

Un sistema operativo es un programa y como tal es creado por programadores. Existen distintos sistemas operativos orientados a plataformas diversas: computadoras personales, servidores, celulares, consolas, etc. Como todo programa, un sistema operativo tiene un código fuente, que puede estar disponible a los usuarios o no. El movimiento **Open Source (código abierto)** propone que el código esté disponible para todo aquel que quiera estudiarlo, usarlo, modificarlo o extenderlo. Esta visión se opone al **Closed Source (código cerrado)** que es la forma en la que se distribuyen los programas comerciales más comunes.

La principal ventaja de los programas open source, también llamados de software libre, es que el código puede ser revisado por expertos y por la comunidad, y estos pueden proponer y realizar mejoras sin depender de la voluntad del dueño del software. El movimiento Open Source no obliga a que el software sea gratuito; de hecho el autor del software puede retener el derecho de propiedad intelectual. El autor puede elegir los mecanismos legales para proteger su creación eligiendo uno o más tipos de licencias. Las licencias indican con qué usos el software puede usarse gratuitamente (en general estudio, investigación y el desarrollo de otros productos gratuitos) y para qué usos el software debe pagarse (en general fines comerciales). Además de comercializar el software es posible ofrecer servicios relacionados, como el soporte y capacitación a clientes.

### Objetivos

- Destacar la disponibilidad de sistemas operativos Open Source.
- Difundir las variantes de sistemas operativos y resumir sus características principales.

## Actividad 1

### Open Cola

#### Objetivos

- Entender el concepto de código abierto en un contexto más tangible y no relacionado al software.
- Entender que Open Source no significa gratuito.

Modalidad de trabajo

Investigación

#### Materiales

- Internet

## Desarrollo

En esta actividad, nuevamente, se propone un trabajo de investigación. Las respuestas deberían ser similares a éstas:

- a) Open Source Cola, también llamada OpenCola, es una gaseosa cuya receta es abierta y compartida bajo una licencia de uso abierto.
- b) Se la puede encontrar en [https://web.archive.org/web/20010218075323/http://www.opencola.com/download/3\\_softdrink/formula.shtml](https://web.archive.org/web/20010218075323/http://www.opencola.com/download/3_softdrink/formula.shtml), (en inglés) o también en Wikipedia: <https://es.wikipedia.org/wiki/OpenCola>.
- c) No, las recetas y los procedimientos para elaborar una gaseosa son considerados secretos comerciales y no están disponibles.
- d) La empresa puede comercializar la gaseosa ya hecha o cobrar por su distribución.

Es importante lograr que los alumnos vean las diferencias entre una gaseosa cuya receta es desconocida y una que se puede preparar en casa aportando mejoras a la receta; y construir a partir de esta idea las ventajas que presenta el software de código abierto.

## Cierre

El software, al igual que una receta o un libro, es un producto intelectual que puede ser compartido de forma abierta o no. Hacerlo abierto no elimina la posibilidad de comercializarlo.

## Ficha 1 para el alumno

### Open Cola

En esta actividad te proponemos que investigues usando internet sobre un producto que quizás no sea tan conocido: la receta para una gaseosa, pero difundida en forma abierta. Para guiarte, te proponemos que contestes las siguientes preguntas:

- a) Investigar que es una Open Source Cola o OpenCola.
- b) ¿Cuál es la receta de OpenCola en este momento?
- c) ¿Es posible hallar en internet la receta de alguna de las gaseosas vendidas en los comercios?
- d) Si la receta está disponible para el público ¿Cómo puede ganar dinero una empresa vendiendo una cola open source?

## Actividad 2

### Sabores de sistemas operativos

<p>Objetivo</p> <ul style="list-style-type: none"><li>• Aprender que existen distintos tipos de sistemas operativos, con distintas disponibilidades (en cuanto a licencias) y que apuntan a distintas plataformas de hardware</li></ul>	<p>Modalidad de trabajo</p> <p>Individual</p>
<p>Materiales</p> <ul style="list-style-type: none"><li>• Internet</li></ul>	

### Desarrollo

En esta actividad, nuevamente, se propone un trabajo de investigación. Las respuestas deberían ser similares a éstas:

Nombre	Disponibilidad	Plataforma
<b>Android</b>	Open Source, Gratuito	móviles
<b>GNU/Linux</b>	Open Source, Gratuito	personales, servidores
<b>FreeBSD</b>	Open Source, Gratuito	personales, servidores
<b>iOS</b>	Closed Source, Comercial	móviles
<b>macOS</b>	Mezcla, Comercial	personales
<b>Solaris</b>	Open Source, Comercial	servidores
<b>Windows</b>	Closed Source, Comercial	personales, servidores

### Cierre

Los sistemas operativos no se limitan a Windows (generalmente el más conocido). Hay numerosos sistemas operativos, con distintas licencias, distintas prestaciones y que apuntan a diversas plataformas

## Ficha 2 para el alumno

### Sabores de sistemas operativos

Los sistemas operativos son programas, y como tales, hay una gran variedad. En esta actividad tenés que investigar en internet y completar la siguiente tabla, para tener idea de algunos de los sistemas operativos más conocidos, sus diferencias y sus formas de licenciamiento y comercialización:

Nombre	Disponibilidad	Plataforma
Android	Open Source, Gratuito	móviles

Para cada sistema operativo, en la columna de “Disponibilidad” tenés que indicar si es comercial o no y si de código abierto (open source) o código cerrado (closed source); y en la columna de “Plataforma” tenés que completar cuál es el o los tipos de dispositivos para los que ese sistema operativo está diseñado (por ejemplo, computadoras de escritorio, teléfonos celulares, etc.).

Los sistemas operativos a considerar son:

Android, GNU/Linux, FreeBSD, iOS, macOS, Solaris y Windows

## Modelo de evaluación

### Permisos en la escuela

<p>Objetivo</p> <ul style="list-style-type: none"><li>• Evaluar la comprensión respecto a la utilización del sistema de archivos y sus permisos.</li></ul>	<p>Modalidad de trabajo</p> <p>Individual</p>
<p>Materiales</p> <ul style="list-style-type: none"><li>• Ninguno</li></ul>	

### Desarrollo

En esta actividad se presenta un sistema utilizado en una escuela compartido entre varios usuarios pertenecientes a 3 grupos:

- Estudiantes
- Docentes
- Directivos

Los grupos tienen distintos permisos de acceso sobre el sistema de archivos según el rol que cumplen. La actividad plantea una serie de preguntas similares a las planteadas en la actividad 2.3.1. A continuación se muestran las respuestas esperadas:

1. Sólo los *Docentes* pueden instalar programas, pero un *Directivo* que pertenezca al grupo *Docentes* podría instalar programas.
2. No, sólo tiene permisos de ejecución.
3. Puede escribirlo en el carpeta `/publico`.
4. Puede copiarlo en su carpeta personal y asignarle permisos de lectura al otro usuario.
5. Pertenece a los grupos *Directivos* y *Docentes*.
6. Sí, las *Docentes* pueden instalar programas maliciosos.
7. Los *Estudiantes* no pueden instalar un programa malicioso, pero si ya está instalado podrían ejecutarlo.
8. Si el programa es de código abierto podría modificarse, pero sólo los *Docentes* podrían instalar el programa modificado.

### Cierre

Con los contenidos vistos en este capítulo esperamos generar una mayor conciencia en la utilización de las computadoras haciendo explícitos los mecanismos que a veces como usuarios pasamos por alto.



## Ficha 3 para el alumno

### Permisos en la escuela

Un sistema posee los siguientes grupos de usuarios:

- Estudiantes
- Docentes
- Directivos

Cada usuario es dueño de un carpeta personal que lleva su nombre, es decir, puede leer y escribir los archivos en esa carpeta; por ejemplo Alan es dueño de la carpeta `/usuarios/alan`. Los estudiantes tienen permiso de lectura y ejecución sobre la carpeta `/programas`, mientras que los *Docentes* tienen, además, permiso de escritura sobre esa carpeta. Existen dos carpetas compartidas, `/publico` que puede ser accedida por todos los usuarios y `/privado` que puede ser leída por los *Docentes* y escrita por los *Directivos*.

El sistema de archivos utilizado es jerárquico. Cada archivo y carpeta tiene asociado un conjunto de permisos que salvo se indique lo contrario se hereda de la carpeta superior a los inferiores.

Respondé las siguientes preguntas:

1. ¿Puede un *Directivo* instalar programas (en la carpeta `programas`)?
2. ¿Puede un *Estudiante* borrar un programa instalado?
3. ¿Cómo puede un *Estudiante* compartir un archivo con el resto de los usuarios?
4. ¿Cómo puede un *Docente* compartir un archivo con otro usuario de forma tal que el resto de los usuarios del sistema no puedan acceder al archivo?
5. ¿A qué grupos debe pertenecer el usuario de la directora del colegio si además da clases?
6. ¿Puede un *Docente* instalar un virus accidentalmente?
7. ¿Puede un *Estudiante* instalar un virus accidentalmente? ¿Puede ejecutarlo si ya está instalado?
8. ¿Puede un usuario modificar un programa instalado para adaptar su uso a las necesidades del curso?

## Glosario

**Código abierto (Open source)** - Esquema de distribución de software donde el código fuente está disponible para estudiarlo y modificarlo. El creador del software puede retener los derechos de propiedad intelectual y explotar el software comercialmente.

**Código cerrado (Closed source)** - Esquema de distribución de software donde sólo se distribuye el archivo ejecutable y no el código fuente. Normalmente se utilizan restricciones legales para evitar que el software pueda ser analizado.

**Interfaz (de un sistema operativo)** - Módulo del sistema operativo que le permite a un usuario acceder a la funcionalidad del sistema. Puede ser gráfica o de línea de comandos.

**Máquina virtual** - Programa encargado de abstraer las diferencias entre múltiples sistemas operativos, permitiendo escribir programas que se ejecuten en todos los sistemas soportados sin modificación.

**Memoria virtual** - Método por el cual el sistema operativo permite usar más memoria RAM que la disponible en hardware. Para ello hace una copia de la memoria RAM en disco, liberando la memoria, cuando los datos pasados a disco son requeridos se vuelven a copiar a la memoria.

**Módulo (de un sistema operativo)** - Pieza de software que forma parte de un sistema operativo y que tiene una función específica. Por ejemplo: el sistema de archivos, el planificador, la interfaz.

**Paginado de memoria** - Mecanismo por el cual el sistema operativo asigna porciones distintas de memoria a cada programa en ejecución garantizando que los datos de un programa no pueden ser leídos ni escritos por otro.

**Permisos y privilegios** - Esquema que permite regular el acceso a los distintos recursos administrados por un sistema operativo.

**Plataforma (de software/hardware)** - Conjunto de elementos (hardware y software) agrupados con fines comunes.

**Planificador** - Módulo del sistema operativo encargado de orquestar la ejecución concurrente de múltiples procesos.

**Proceso** - Programa en ejecución.

<Program.AR/>

**Programa malicioso** - Programa que busca comprometer la seguridad e integridad de un sistema.

**Servidor** - Computadora que corre programas que proveen servicios a otras computadoras; o sea, que brindan programas útiles para otras computadoras.

**Sistema de archivos** - Módulo del sistema operativo encargado de mantener datos en medios de almacenamiento. Provee una interfaz homogénea para leer y escribir a múltiples tipos de dispositivos de almacenamiento.

**Sistema operativo** - Programa encargado de la administración de los recursos disponibles en una computadora.

# ANEXO III

## ACTIVIDADES DE PROGRAMACIÓN

### REPRESENTACIÓN DE NÚMEROS

Números binarios con procedimientos

### PROCEDIMIENTOS CON PARÁMETROS

Dibujar figuras con subtareas  
Varias cosas de una sola vez

### REPETICIÓN CONDICIONAL Y RECORRIDOS

Hansel y Gretel van a la casa de la bruja  
Limpiemos imágenes

### CÁLCULO Y MEMORIZACIÓN DE DATOS

El radar

### PROGRAMAS INTERACTIVOS

Pidiendo empanadas  
Pacman

En este anexo se incluyen ocho desafíos de programación que complementan los del Manual para docentes - 1er ciclo secundaria. Con ellos, se podrá trabajar sobre representación de números, procedimientos con parámetros, repetición condicional y recorridos, cálculo y memorización de datos y programas interactivos.

## Actividad

### Números binarios con procedimientos

<p>Objetivo</p> <ul style="list-style-type: none"><li>• Practicar la representación binaria de números mediante procedimientos en un programa.</li><li>• Vincular el trabajo concreto con las fichas de colores de las actividades previas con la idea en un lenguaje de programación, usando la ejecución como validación de los resultados.</li></ul>	<p>Modalidad de trabajo</p> <p>Individual</p>
<p>Materiales</p> <ul style="list-style-type: none"><li>• Computadoras con Gobstones</li><li>• El proyecto "Números binarios con procedimientos"</li></ul>	

## Desarrollo

Repartir las fichas con las consignas, e indicarles que abran el proyecto "Números binarios con procedimientos".

El propósito es construir procedimientos que pongan en una celda del tablero una cierta cantidad de bolitas rojas. Para eso, se brindan algunos procedimientos primitivos: los correspondientes a las cantidades de bolitas que son múltiplo de 2 (**Poner1BolitaRoja, Poner2BolitasRojas, Poner4BolitasRojas, Poner8BolitasRojas**).

El objetivo es que los estudiantes construyan un procedimiento para cada una de las siguientes cantidades de bolitas:

- 3 bolitas rojas
- 5 bolitas rojas
- 7 bolitas rojas
- 10 bolitas rojas
- 15 bolitas rojas

En cada uno de esos procedimientos, solo podrán utilizar los procedimientos primitivos provistos, pero no más de una vez cada uno; esta restricción es similar a la de la actividad anterior, donde la máquina no podía aceptar más de una única ficha de cada color. Tampoco podrán reutilizar otros procedimientos hechos por ellos (por ejemplo, al poner 5 bolitas, no deben reutilizar el procedimiento que pone 3 bolitas).

En caso de que lleguen a una solución que, o bien no define procedimientos, o bien utiliza nombres inapropiados para ellos, debe pedirse que se corrijan estos aspectos hasta llegar a

una solución aceptable. Se espera que los alumnos propongan las soluciones que se encuentran en la Figura 2.

<pre>program {   Poner3BolitasRojas() Mover(Este)   Poner5BolitasRojas() Mover(Este)   Poner7BolitasRojas() Mover(Este)   Poner10BolitasRojas() Mover(Este)   Poner15BolitasRojas() }  procedure Poner3BolitasRojas() {   Poner2BolitasRojas()   Poner1BolitaRoja() }</pre>	<pre>procedure Poner5BolitasRojas() {   Poner4BolitasRojas()   Poner1BolitaRoja() }  procedure Poner7BolitasRojas() {   Poner4BolitasRojas()   Poner2BolitasRojas()   Poner1BolitaRoja() }</pre>	<pre>procedure Poner10BolitasRojas() {   Poner8BolitasRojas()   Poner2BolitasRojas() }  procedure Poner15BolitasRojas() {   Poner8BolitasRojas()   Poner4BolitasRojas()   Poner2BolitasRojas()   Poner1BolitaRoja() }</pre>
---	--	---

Figura 2. Solución esperada para “Números binarios con procedimientos”

Es importante destacar la división en subtareas expresada por los procedimientos pedidos, ya que esto permite expresar la solución fácilmente. También es importante destacar la claridad de los nombres de cada procedimiento. Además se puede recapitular sobre el concepto de repetición simple ya que los procedimientos primitivos **Poner8BolitasRojas**, **Poner4BolitasRojas** y **Poner2BolitasRojas** se pueden expresar fácilmente utilizando este concepto.

Puede que surja como duda por qué no utilizar la repetición también para los procedimientos pedidos. En este caso debe explicarse que la idea es volver a realizar la actividad de números binarios, pero en lugar de usar fichas, usamos procedimientos primitivos. El procedimiento **Poner1BolitaRoja** es como usar una ficha de color rojo, el procedimiento **Poner2BolitasRojas** es como usar una ficha de color gris, etc.

### Cierre

Como conclusión, es importante destacar que esta actividad es equivalente a la anterior, cambiando las fichas de colores por los procedimientos. El mismo concepto es trasladable de una consigna a la siguiente, por lo que se espera que los alumnos puedan resolver este ejercicio sin dificultad siguiendo los mismos pasos usados en la solución del ejercicio anterior.

Recordar a los estudiantes cómo, en la actividad anterior, podía recuperarse la **representación binaria** de los números, escrita con ceros y unos, a partir de la presencia o ausencia de una ficha de cada color. Hacerles notar cómo, en este caso, se puede hacer algo similar con los procedimientos que programaron, teniendo en cuenta si utilizaron o no cada uno de los procedimientos primitivos provistos con el proyecto. Por ejemplo, para poner 10 bolitas, se usan los procedimientos **Poner8BolitasRojas** y **Poner2BolitasRojas**, pero no se usan **Poner4BolitasRojas** ni **Poner1BolitaRoja**; esto indica que la escritura binaria del número 10 es 1010, como ilustra la Figura 3.

<Program.AR/>

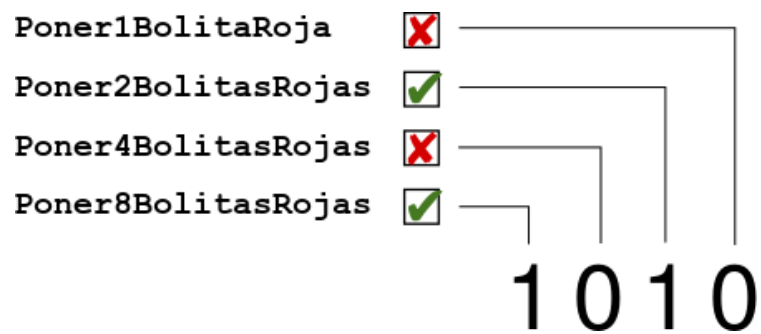


Figura 3. Representación binaria de un número obtenida a partir del procedimiento correspondiente.

## Ficha para el alumno

### Números binarios con procedimientos

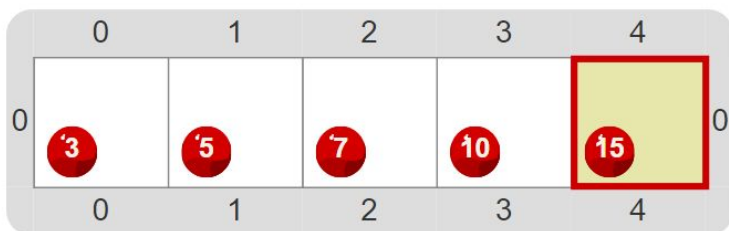
Abrí el proyecto "Números binarios con procedimientos" y fijate que vas a encontrar los siguientes procedimientos:

- **Poner1BolitaRoja**
- **Poner2BolitasRojas**
- **Poner4BolitasRojas**
- **Poner8BolitasRojas**

Tu tarea es construir un programa que ponga las siguientes cantidades de bolitas rojas en celdas consecutivas, como se vé en la imagen. Pero *para cada número*, solamente podés usar como *máximo una invocación* a cada uno de los procedimientos primitivos y nada más. Tenés que poner:

- 3 bolitas rojas
- 5 bolitas rojas
- 7 bolitas rojas
- 10 bolitas rojas
- 15 bolitas rojas

El tablero final debería quedarte como sigue:



¡Acordate de definir un procedimiento para cada subtarea!

¿Ves la similaridad de esta actividad con la anterior? ¿Podés encontrar qué ficha representa cada uno de los procedimientos que venían hechos?



**Actividad****Dibujar figuras con subtareas**

<p>Objetivo</p> <ul style="list-style-type: none"> <li>• Entender la descomposición dada de una tarea en subtareas y utilizar procedimientos con parámetros para definir cada una de ellas.</li> <li>• Practicar el uso de procedimientos parametrizados utilizando los argumentos correctos.</li> <li>• Descubrir que los parámetros de un procedimiento pueden utilizarse como argumentos de otros procedimientos.</li> </ul>	<p>Modalidad de trabajo</p> <p>De a 2</p>
<p>Materiales</p> <ul style="list-style-type: none"> <li>• Computadoras con Gobstones</li> <li>• Los proyectos "Dibujar figuras con subtareas" y "Dibujar figuras con subtareas, revisado"</li> </ul>	

**Desarrollo**

La tarea a solucionar en esta actividad es semejante a la de actividades anteriores: dibujar cuatro cuadrados de diferentes colores y tamaños. Sin embargo, esta vez para dibujar cada cuadrado se utilizarán procedimientos más sencillos para dibujar cada uno de sus lados. A su vez, estos podrán ser reutilizados luego para dibujar figuras distintas.

A) Como primer trabajo deben abrir el proyecto *"Dibujar figuras con subtareas"*. En el mismo ya se encuentra definido un procedimiento parametrizado para dibujar cuadrados dados su tamaño y su color. Los estudiantes deben completar los procedimientos auxiliares para dibujar líneas, y además descubrir la manera de utilizar el procedimiento primitivo **PosicionarseEnLaEsquinaParaUnCuadradoDeTamaño(dirección1,dirección2, tamaño)**, que sirve para ubicar el cabezal en el lugar adecuado para dibujar un cuadrado de cierto tamaño en la esquina indicada, pasándole adecuadamente los argumentos necesarios. Se espera que los estudiantes indaguen en el proyecto y descubran lo que deben hacer, por analogía con las soluciones anteriores. La solución se presenta en la Figura 7, con las partes a completar en color rojo.

Una cuestión sutil que debe tenerse en cuenta es que un parámetro solamente tiene sentido dentro del procedimiento que lo define. Esto es así porque el parámetro constituye un agujero *dentro* del procedimiento, y no tendría sentido que fuese utilizado en otro lugar. Es incorrecto, por ejemplo, tratar de utilizar **colorDelCuadrado** en el

procedimiento para dibujar líneas, ya que cada uno de éstos tiene su propio parámetro, **colorDeLaLineaAl...** Si algún estudiante intentase utilizar un parámetro fuera del procedimiento que lo define, es importante entender el problema para poder ayudarlo en la solución del mismo.

```
program {
  PosicionarseEnLaEsquina__ParaUnCuadradoDeTamaño_(Sur, Oeste, 4)
  DibujarCuadrado(Rojo, 4)

  PosicionarseEnLaEsquina__ParaUnCuadradoDeTamaño_(Sur, Este, 3)
  DibujarCuadrado(Negro, 3)

  PosicionarseEnLaEsquina__ParaUnCuadradoDeTamaño_(Norte, Este, 6)
  DibujarCuadrado(Azul, 6)

  PosicionarseEnLaEsquina__ParaUnCuadradoDeTamaño_(Norte, Oeste, 5)
  DibujarCuadrado(Verde, 5)
}

procedure DibujarCuadrado(colorDelCuadrado, longitudDelLado) {
  DibujarLineaAlNorte(colorDelCuadrado, longitudDelLado - 1)
  DibujarLineaAlEste(colorDelCuadrado, longitudDelLado - 1)
  DibujarLineaAlSur(colorDelCuadrado, longitudDelLado - 1)
  DibujarLineaAlOeste(colorDelCuadrado, longitudDelLado - 1)
}

procedure DibujarLineaAlNorte(colorDeLaLineaAlNorte, longitudDeLaLineaAlNorte) {
  repeat(longitudDeLaLineaAlNorte) { Poner(colorDeLaLineaAlNorte) Mover(Norte) }
}

procedure DibujarLineaAlEste(colorDeLaLineaAlEste, longitudDeLaLineaAlEste) {
  repeat(longitudDeLaLineaAlEste) { Poner(colorDeLaLineaAlEste) Mover(Este) }
}

procedure DibujarLineaAlSur(colorDeLaLineaAlSur, longitudDeLaLineaAlSur) {
  repeat(longitudDeLaLineaAlSur) { Poner(colorDeLaLineaAlSur) Mover(Sur) }
}

procedure DibujarLineaAlOeste(colorDeLaLineaAlOeste, longitudDeLaLineaAlOeste) {
  repeat(longitudDeLaLineaAlOeste) { Poner(colorDeLaLineaAlOeste) Mover(Oeste) }
}
```

Figura 7. Solución para “Dibujar figuras con subtareas” con las partes a completar en rojo.

B) En esta segunda parte, usando el proyecto “Dibujar figuras con subtareas, revisado”, el dibujo del cuadrado se sirve de un único procedimiento para dibujar líneas. La tarea de los estudiantes será definir este procedimiento. El mismo agrega un parámetro de dirección, para indicarle hacia dónde debe dibujar. Si se les complica descubrir cómo utilizar este parámetro, recordarles el trabajo de actividades anteriores, como “Tablero de Ludo” y “Ludo Loco”. La solución se muestra en la Figura 8.

```
procedure DibujarLinea(dirección, colorDeLaLinea, longitudDeLaLinea) {
  repeat (longitudDeLaLinea) {
    Poner(colorDeLaLinea)
    Mover(dirección)
  }
}
```

Figura 8. Solución para “Dibujar figuras con subtareas, revisado”, con un único procedimiento para dibujar líneas.

Para completar la actividad se les debe dar tiempo para que experimenten escribiendo procedimientos que permitan obtener otros dibujos, reutilizando los procedimientos para dibujar líneas. En la ficha se dan imágenes con algunas ideas. El código para completar dichas figuras se ofrece en la Figura 9.

```
program { DibujarRectángulo(Verde, 4, 8) }

program { DibujarRectángulo(Rojo, 8, 4) }

procedure DibujarRectángulo(colorDelRectángulo, baseDelRectángulo,
  alturaDelRectángulo) {
  DibujarLinea(Norte, colorDelRectángulo, alturaDelRectángulo - 1)
  DibujarLinea(Este, colorDelRectángulo, baseDelRectángulo - 1)
  DibujarLinea(Sur, colorDelRectángulo, alturaDelRectángulo - 1)
  DibujarLinea(Oeste, colorDelRectángulo, baseDelRectángulo - 1)
}
```

```
program { DibujarL(Azul) }

procedure DibujarL(colorDeLaL) {
  DibujarLinea(Norte, colorDeLaL, 7)
  DibujarLinea(Este, colorDeLaL, 2)
  DibujarLinea(Sur, colorDeLaL, 5)
  DibujarLinea(Oeste, colorDeLaL, 3)
  DibujarLinea(Sur, colorDeLaL, 2)
  DibujarLinea(Oeste, colorDeLaL, 5)
}
```

Figura 9. Programas para dibujar las 3 figuras pedidas al final de la Ficha.

## Cierre

Al finalizar, debe realizarse una reflexión global, analizando los programas, en especial el procedimiento para dibujar cuadrados, para entender cómo utilizar los parámetros de un procedimiento como argumentos de otros procedimientos parametrizados, y las ventajas de contar con procedimientos parametrizados para expresar subtareas. Utilizar los parámetros de un procedimiento como argumentos de otros es posible porque un parámetro representa un agujero cuyo contenido viene dado por el argumento correspondiente. Al usarlo como argumento, se utiliza el dato indicado en el argumento original.

El procedimiento para dibujar líneas sirve para dibujar un cuadrado, pero también para dibujar otras figuras, como rectángulos y otras. La flexibilidad conseguida en el programa de esta actividad es posible gracias a la identificación de subtareas generales y a la posibilidad de expresarlas utilizando procedimientos parametrizados.

## Ficha para el alumno

### Dibujar cuadrados con subtareas

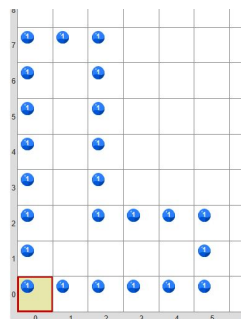
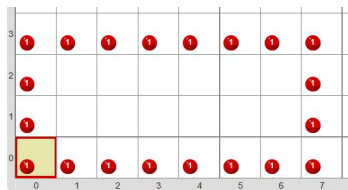
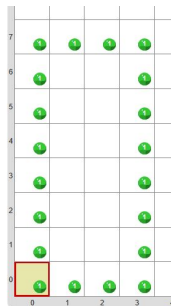
A) Abrí el proyecto “Dibujar cuadrados con subtareas” y completá el programa para lograr el mismo tablero que en la actividad anterior. La diferencia es que esta vez el procedimiento para dibujar cuadrados ya viene hecho, y vos tenés que hacer los procedimientos que definen las subtareas para ese dibujo. Además, fijate que para posicionar el cabezal antes de dibujar cada cuadrado ahora contás con un procedimiento primitivo parametrizado:

**PosicionarseEnLaEsquinaParaUnCuadradoDeTamaño(dirección1, dirección2, tamaño).**

Como antes, acordate que pasar argumentos es una forma de indicarle a un comando con qué dato queremos que trabaje, y tené en cuenta que la cantidad y orden de los argumentos debe coincidir con la cantidad y orden de los parámetros.

B) Para la segunda parte abrí “Dibujar cuadrados con subtareas, revisado”, que es casi igual, pero esta vez hay que definir un único procedimiento para dibujar líneas. Para hacerlo, tenés que descubrir cómo usar el nuevo parámetro de dirección.

Para completar la actividad probá hacer más dibujos utilizando el procedimiento para definir líneas, como los de la figura, u otros. ¿Ves qué útil que resulta tener un procedimiento con parámetros?



## Actividad

### Varias cosas de una sola vez

<p>Objetivo</p> <ul style="list-style-type: none"><li>• Practicar la definición y uso de procedimientos parametrizados.</li><li>• Ofrecer versiones más poderosas de los comandos primitivos.</li></ul>	<p>Modalidad de trabajo</p> <p>De a 2</p>
<p>Materiales</p> <ul style="list-style-type: none"><li>• Computadoras con Gobstones</li><li>• El proyecto “Varias cosas de una sola vez”</li></ul>	

## Desarrollo

En esta actividad, los estudiantes deben construir procedimientos que permitan, con una única invocación, usar un número determinado de veces los comandos primitivos de Gobstones; se podrían llamar **PonerMuchasBolitas**, **SacarMuchasBolitas** y **MoverMuchasCeldas**. Luego, deberán escribir un programa que transforme el tablero inicial provisto (de 7 celdas de ancho, 6 celdas de alto y con 17 bolitas negras en la celda de la esquina SurOeste) en el tablero final presentado en la Ficha, usando solamente llamadas a los procedimientos que se definieron y el procedimiento primitivo **IrALaEsquina**. Ver la Figura 10 para encontrar una posible solución al desafío.

```
program {  
  IrAlBorde(Sur)  
  IrAlBorde(Oeste)  
  SacarMuchasBolitas(17, Negro)  
  MoverMuchasCeldas(3, Este)  
  PonerMuchasBolitas(42, Rojo)  
}  
  
procedure PonerMuchasBolitas(cantidadDeBolitas, color)  
{ repeat(cantidadDeBolitas) { Poner(color) } }  
  
procedure SacarMuchasBolitas(cantidadDeBolitas, color)  
{ repeat(cantidadDeBolitas) { Sacar(color) } }  
  
procedure MoverMuchasCeldas(cantidadDeCeldas, direccion)  
{ repeat(cantidadDeCeldas) { Mover(direccion) } }
```

Figura 10. Solución para “Varias cosas de una sola vez”, con las partes a completar en color rojo.

La solución de los tres procedimientos requiere utilizar la repetición simple y el comando primitivo correspondiente, pero utilizando los parámetros dados como argumentos en la repetición y el comando. Puede suceder que algunos estudiantes aún no comprendan que pueden utilizar los parámetros como argumentos de otros comandos dentro del cuerpo de la definición. Las definiciones de los procedimientos ya provistas con el proyecto deberían servirles como pista para encontrar ese hecho, pero de no ser así, se les debe guiar con preguntas y sugerencias referenciándolos a las soluciones de actividades anteriores.

Para las preguntas finales de la Ficha, las respuestas son las siguientes.

La primera pregunta busca que los estudiantes escriban un procedimiento para sacar todas las bolitas de un color, usando **SacarMuchasBolitas**. Dicho procedimiento que puede

llamarse **SacarTodas**, deberá tomar un parámetro que indique el color a sacar, por lo cual un buen nombre para el mismo sería **colorASacar**. Para usar el procedimiento **SacarMuchasBolitas**, hace falta darle un número y un color. El color es el dado por el parámetro **colorASacar**, pero, ¿cómo saber el número? Dado que hay que sacar todas las bolitas, el número es la cantidad de bolitas que haya, lo cual puede obtenerse utilizando la expresión primitiva **nroBolitas**. El cuerpo del procedimiento quedará, entonces, **SacarMuchasBolitas(nroBolitas(colorASacar), colorASacar)**.

La segunda pregunta es sobre como sacar todas las bolitas de todos los colores; la respuesta es que hay que usar cuatro veces **SacarTodas**, una por color.

### Cierre

Para cerrar la actividad se puede reflexionar sobre la ventaja que tiene darle nombre a la tarea de poner muchas veces o mover muchas veces: a partir de ahora tenemos la posibilidad de definir un comando que haga esto, en lugar de tener que usar una repetición para hacerlo. Esto resulta mejor, porque es más claro al leer el programa.

## Ficha para el alumno

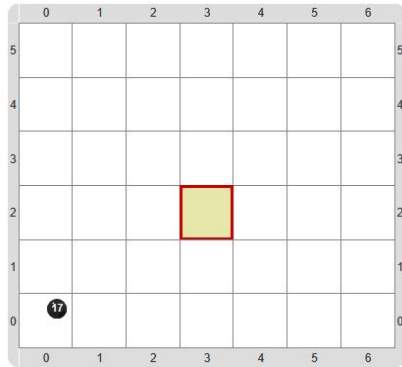
### Varias cosas de una sola vez

En esta actividad tenés que definir tres procedimientos parametrizados:

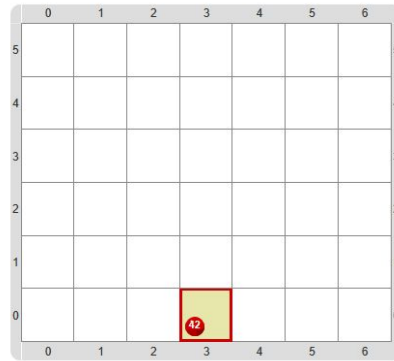
**PonerMuchasBolitas(cantidadAPoner,colorAPoner),**  
**SacarMuchasBolitas(cantidadASacar,colorASacar) y**  
**MoverMuchasCeldas(cantidadDeCeldas,dirección)**

El primero debe poner muchas bolitas, el segundo debe sacar muchas, y el último debe mover el cabezal muchas celdas. En los tres, la cantidad es la indicada en el parámetro correspondiente.

Después tenés que utilizarlos para transformar el tablero inicial dado en el tablero final que se ve en la figura; lo interesante es conseguirlo usando solamente los procedimientos que definiste y también el procedimiento primitivo **IrALaEsquina**.



Tablero inicial



Tablero final esperado

Después que completaste esta tarea, tenemos dos desafíos:

1. ¿Podés usar el procedimiento **SacarMuchasBolitas** para sacar todas las de un color determinado? ¿Cuántas tendrías que sacar? ¿Cómo llamarías a un procedimiento que haga esta tarea?

---

---

2. ¿Cómo harías para sacar todas las bolitas, de todos los colores, que haya en la celda actual?

---

---

---

**Actividad**

## Hansel y Gretel van a la casa de la Bruja

<p>Objetivo</p> <ul style="list-style-type: none"> <li>• Motivar la noción de recorrido a través de un problema que puede resultar muy complejo si se lo trabaja de otra manera.</li> <li>• Presentar la noción de recorrido, sus partes y la manera de utilizarla.</li> </ul>	<p>Modalidad de trabajo</p> <p>De a 2</p>
<p>Materiales</p> <ul style="list-style-type: none"> <li>• Computadoras con Gobstones</li> <li>• El proyecto “Hansel y Gretel van a la casa de la Bruja”</li> </ul>	

**Desarrollo**

Repartir las Fichas e indicarles que abran el proyecto correspondiente. El propósito del programa es recorrer el camino representado en el tablero, marcando las posiciones del mismo a medida que se recorre. Para que el problema resulte más familiar se utiliza como dominio el cuento clásico de Hansel y Gretel, donde las marcas en el camino son provistas por miguitas de pan. El proyecto incluye el procedimiento primitivo **AvanzarCarreta** para avanzar la carreta, y las funciones primitivas **llegamosALoDeLaBruja** que determina si aún falta recorrer camino o no, y **miguitas** que retorna el color que debe ponerse para poner miguitas de pan en el camino.

Para guiar a los estudiantes en la solución de este problema, lo primero que se debe recomendar es que establezcan la estrategia de solución; una forma adecuada de expresarla sería “avanzar la carreta dejando miguitas hasta llegar a lo de la bruja”. Al enunciarlo de esta forma, queda claro que deberán utilizar una repetición condicional, y de ahí a determinar cuáles son las partes que deben utilizar hay poca distancia. Como condición para finalizar la repetición, se espera que empleen la función **llegamosALoDeLaBruja**, que ya viene provista. Dentro de la repetición deben realizar dos subtareas: para **AvanzarCarreta** los estudiantes también cuentan con un procedimiento ya provisto, pero deben definir su propio procedimiento para **DejarMiguitas** (lo cual sería más adecuado que simplemente usar “**Poner(miguitas())**” directamente, porque el procedimiento representa a la subtarea correspondiente). Si los estudiantes no definen este procedimiento, recordarles que siempre es mejor representar explícitamente las subtareas en forma de procedimiento.



Otra complejidad que puede aparecer es la de utilizar la función **miguita** en lugar de un color explícito. No sería correcto que escriban “**Poner(Rojo)**” para poner la miguita, pues esta última opción no resulta tan expresiva como la primera. Algunos estudiantes pueden encontrar extraño que la función represente a un color; si ese es el caso, recordarles las actividades del capítulo 3 donde se trabajó la noción de función como representación de información.

Una posible solución se encuentra en la Figura 15.

```
procedure HanselYGretelVanALoDeLaBruja() {
  while (not llegamosALoDeLaBruja()) {
    // "repetirHastaQue llegamosALoDeLaBruja"
    // hasta que no terminó el recorrido
    DejarMiguitas() // procesar elemento
    AvanzarCarreta() // pasar al siguiente
  }
}

procedure DejarMiguitas()
{ Poner(miguitas()) }
```

Figura 15. Posible solución para “Hansel y Gretel van a la casa de la Bruja”

Lo interesante de la actividad es realizar, una vez completado el programa, un análisis de cómo se utilizó la repetición condicional, y qué subtareas se utilizaron. Si la solución es correcta, las subtareas utilizadas son tres:

1. una función para determinar si se completó el recorrido del camino (**llegamosALoDeLaBruja**),
2. un procedimiento para realizar el procesamiento en el sector actual del camino (**DejarMiguitas** o similar), y
3. un procedimiento para avanzar al siguiente sector del camino (**AvanzarCarreta**).

A esta forma de utilizar la repetición condicional, repitiendo procesar y avanzar hasta que se cumpla cierta condición, la denominamos **recorrido**, pues se trata de *recorrer* una secuencia de elementos, en este caso, sectores del camino, para procesar cada uno de ellos de alguna forma, en este caso, dejar miguitas de pan. Entonces, denominamos recorrido a una forma de dividir un problema de procesamiento de elementos en tres subtareas: una función que determina si finalizó el recorrido, un procedimiento de procesamiento de elementos y un procedimiento para pasar al siguiente elemento.

### Cierre

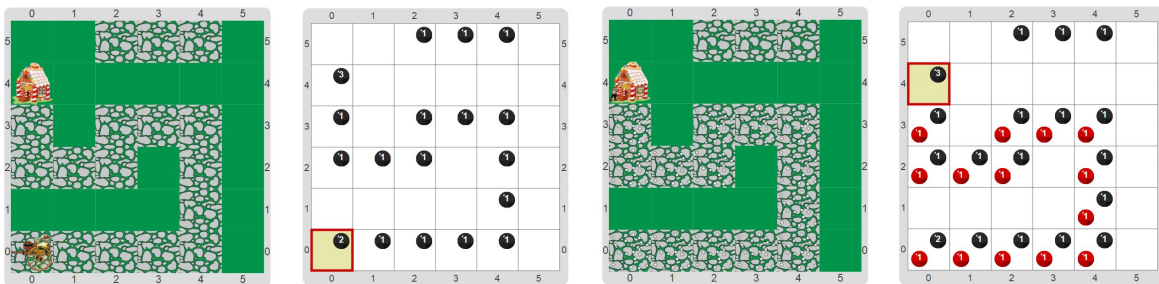
Para completar la actividad, presentar la idea de **recorrido**. Los recorridos son una forma algorítmica básica que se utiliza como guía para la realización de programas que involucran procesar los elementos de una secuencia, o sea una forma de guiar la división en subtareas en el caso de problemas que involucran secuencias. Las secuencias pueden variar ampliamente de naturaleza, y la complejidad de cada una de las partes también, pero en este tipo de problemas siempre podemos identificar una secuencia de elementos que deben ser procesados. El trabajo para resolver el problema consiste entonces en identificar cuáles son los elementos, cómo se pasa de uno al siguiente, cómo se procesa cada uno, y cuál es la condición de finalización. En las próximas actividades veremos más ejemplos de recorridos.

## Ficha para el alumno

### Hansel y Gretel van a la casa de la Bruja

Hansel y Gretel van en la carreta mágica hacia la casa de la bruja y deben dejar miguitas de pan en el camino para poder volver, porque no saben qué forma ni qué longitud tiene el camino. Cada grupo de miguitas se representa con una bolita de color "miguitas()", una función que devuelve el color correspondiente. El lugar donde Hansel y Gretel se suben a la carreta se representa con 2 bolitas negras, cada sección del camino se representa con 1 bolita negra y la casa de la bruja se representa con 3 bolitas negras. En el tablero puede haber también bolitas negras que no pertenecen al camino, y que no serán tenidas en cuenta en la solución.

Abri el proyecto, donde se provee también un procedimiento primitivo **AvanzarCarreta** que se encarga de avanzar la carreta y completá el programa para que avance la carreta hasta la casa de la bruja sin olvidar ir dejando miguitas en el camino. Un posible tablero inicial y su correspondiente tablero final, con y sin vestimenta, se observan en las imágenes:



Tablero inicial

Tablero final

¡Acordate de empezar pensando una estrategia de solución, y viendo qué subtarear son necesarias para expresarla! Al finalizar, el docente va a proponer un análisis de las soluciones propuestas, así que pensá bien las partes de tu programa, y cómo se arman para proveer la solución final.

## Actividad

### Limpiemos imágenes

Objetivo	Modalidad de trabajo
<ul style="list-style-type: none"><li>Realizar un recorrido por celdas.</li></ul>	Individual
Materiales	
<ul style="list-style-type: none"><li>Computadoras con Gobstones</li><li>El proyecto "Limpiemos imágenes"</li></ul>	

## Desarrollo

Repartir las fichas, indicarles que abran el proyecto "Limpiemos imágenes" y completen el programa. El tablero inicial del proyecto contiene una imagen en la codificación RGB vista en el capítulo 5, es decir, con los colores de sus píxeles representados mediante bolitas rojas, verdes y azules, pero algunas de las celdas tienen también bolitas negras que no deberían estar ahí. La imagen puede ser cualquiera, y las bolitas negras pueden estar en cualquier posición. El propósito del programa es limpiar la imagen, o sea, quitar impurezas de la codificación de la misma; para eso deben realizar un recorrido por todos sus píxeles, representados por las celdas del tablero, y en cada uno restaurar una codificación RGB válida.

La idea de recorrido, presentada en la actividad anterior, se puede aplicar a diferentes secuencias; en este caso, se toman todas las celdas del tablero como una secuencia. Parte del código ya viene comenzada, para guiarlos en la solución. La idea es que los estudiantes completen los procedimientos y funciones encargados de resolver cada una de las distintas subtarefas del problema, y puedan combinarlas para construir un recorrido. Las partes que deben completarse se muestran en la Figura 16.

```
program { LimpiarLaImagen() }
procedure LimpiarLaImagen() {
  ComenzarElRecorrido() \ \ ir al primer elemento
  while (not esLaÚltimaCelda()) {
    \ \ En bloques "repetirHastaQue esLaÚltimaCelda
    \ \ hasta que no terminó el recorrido
    EliminarCenizaDeLaCeldaActual() \ \ procesar elemento
    PasarALaSiguienteCelda() \ \ pasar al siguiente elemento
  }
  EliminarCenizaDeLaCeldaActual() \ \ caso de borde
}

procedure ComenzarElRecorrido()
{ IrAlBorde(Sur); IrAlBorde(Oeste) }

procedure EliminarCenizaDeLaCeldaActual()
{ SacarTodas(Negro) }

function esLaÚltimaCelda()
{ return (puedeMover(Este) || puedeMover(Norte) ) }
```

Figura 16. Solución para "Limpiemos imágenes", con las partes que deben completar en color rojo

## Cierre

Para concluir la actividad, repasar la idea de recorrido, y observar que es extremadamente flexible. Es decir, con la misma estructura básica pueden resolverse problemas que parecen muy distintos, lo cual queda claro al comparar esta actividad con la anterior. Para lograrlo es necesario definir con cuidado cuál es el siguiente elemento. En este caso, las celdas se recorren por filas (de Oeste a Este), y a su vez, las filas se recorren de Sur a Norte.

## Ficha para el alumno

### Limpiemos imágenes

¿Te acordás la codificación de imágenes RGB que aprendimos en el capítulo 5? Usábamos bolitas rojas, azules y verdes para codificar una imagen. Resulta que una de estas imágenes “se manchó con ceniza”, lo cual quiere decir que algunas celdas tienen bolitas negras que hacen que la imagen no se vea bien. Abrí el proyecto “Limpiemos imágenes” y completé el programa para que cumpla el propósito de eliminar las bolitas negras de todas las celdas, para que la imagen vuelva a ser una imagen RGB correctamente codificada.

La estrategia de solución que ya está empezada en el procedimiento **LimpiarLaImagen** es la de hacer un *recorrido* sobre las celdas del tablero. Tu trabajo consiste en completar ese recorrido, y también completar los procedimientos y funciones que representan las subtareas para las partes del mismo:

- el procedimiento **ComenzarElRecorrido** para posicionarse al principio de la imagen y no olvidarte ninguna celda,
- la función **esLaÚltimaCelda** para indicar el fin del recorrido,
- el procedimiento **EliminarCenizaDeLaCeldaActual** para procesar cada celda, y
- el procedimiento **PasarALaSiguienteCelda** que ya te lo damos hecho.

Te mostramos un tablero inicial posible, y su correspondiente tablero final.

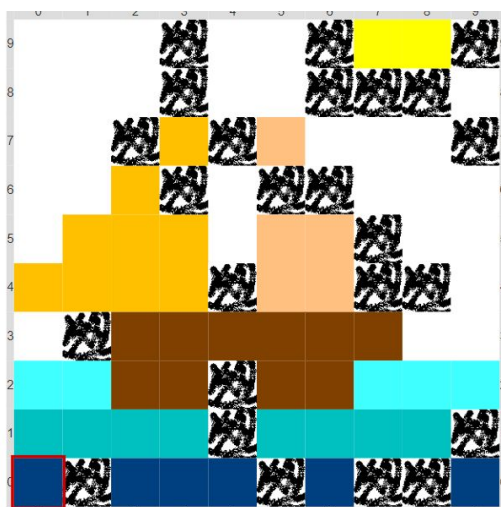


Imagen manchada

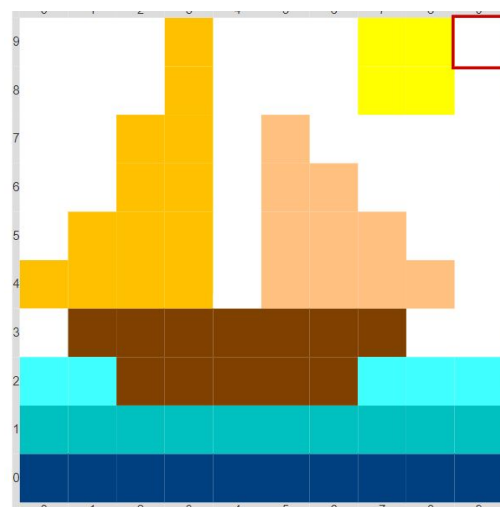


Imagen limpia

## Actividad

### El radar

<p>Objetivo</p> <ul style="list-style-type: none"><li>• Buscar en un cuadrante, para preparar el juego del Pacman</li></ul>	<p>Modalidad de trabajo</p> <p>De a 2</p>
<p>Materiales</p> <ul style="list-style-type: none"><li>• Computadoras con Gobstones</li><li>• El proyecto "El radar"</li></ul>	

## Desarrollo

El propósito en esta actividad es realizar un recorrido de búsqueda, pero con la particularidad de que no es en todo el tablero, sino en un cuadrante<sup>1</sup>. Deben construir un procedimiento **BuscarEnCuadrante(colorABuscar,direcciónHorizontal,direcciónVertical)** y una función **hayEnCuadrante(color, dirección1, dirección2)**. El procedimiento busca una celda con bolitas del **colorABuscar**, en el cuadrante determinado por las direcciones **direcciónHorizontal** y **direcciónVertical**, por ejemplo Norte y Este, y deja el cabezal sobre dicha celda si existe o sobre la celda final, si no. La función debe devolver verdadero si en el cuadrante indicado hay una celda con bolitas del color indicado, para lo cual debe utilizar el procedimiento. Básicamente, el procedimiento provee el procesamiento necesario para que la función con procesamiento pueda realizar su trabajo; esto es una aplicación de la división en subtareas: el procedimiento expresa la subtarea que tiene que hacer la función antes de retornar el resultado. Finalmente, el programa principal debe utilizar la función para indicar en qué cuadrantes hay bolitas rojas.

La solución para el procedimiento es un recorrido de búsqueda, pero con algunas particularidades: es necesario utilizar parámetros para indicar cuál es el cuadrante en el que se quiere buscar, y también recordar el ancho del cuadrante para que, al pasar a la siguiente celda, se pueda volver al borde del cuadrante; para calcular el ancho del cuadrante se puede utilizar la función **distanciaAlBorde**, definida en la actividad 8.3.3, "¿A qué distancia está el borde?". El código esperado se presenta en la Figura 22.

---

<sup>1</sup>Entendemos cuadrante como cada una de las cuatro porciones rectangulares que quedan determinadas por la posición del cabezal y cada uno de las esquinas del tablero. Así, podemos hablar del cuadrante NorEste (el que se encuentra entre la posición del cabezal y la esquina NorEste), del cuadrante SurEste, del cuadrante NorOeste y del cuadrante SurOeste. Ver las imágenes de la Ficha, que ilustran esta explicación.

```
program { SimularUnRadar() }

procedure SimularUnRadar() {
  if (hayEnCuadrante(Rojo, Norte, Oeste)) { Poner(Azul) }
  if (hayEnCuadrante(Rojo, Norte, Este)) { Poner(Negro) }
  if (hayEnCuadrante(Rojo, Sur, Este)) { Poner(Verde) }
  if (hayEnCuadrante(Rojo, Sur, Oeste)) { Poner(Rojo) }
}

function hayEnCuadrante(color, dirección1, dirección2) {
  BuscarEnCuadrante(color, dirección1, dirección2)
  return (hayBolitas(color))
}

procedure BuscarEnCuadrante(colorABuscar, direcciónPrincipal, direcciónSecundaria) {
  anchoDelCuadrante := distanciaAlBorde(direcciónPrincipal)
  // En bloques: "repetirHastaQue hayBolitas(colorABuscar)
  // oBien esLaÚltimaCelda(direcciónPrincipal, direcciónSecundaria)"
  while (not (hayBolitas(colorABuscar))
    && not (esLaÚltimaCelda(direcciónPrincipal, direcciónSecundaria)))
  { IrALaSiguienteCelda(anchoDelCuadrante, direcciónPrincipal, direcciónSecundaria) }
}

procedure IrALaSiguienteCelda(ancho, direcciónPrincipal, direcciónSecundaria) {
  if (puedeMover(direcciónPrincipal))
  { Mover(direcciónPrincipal) }
  else {
    Mover_Veces(opuesto(direcciónPrincipal), ancho)
    Mover(direcciónSecundaria)
  }
}

function esLaÚltimaCelda(dirección1, dirección2) {
  return (not (puedeMover(dirección1)) && not (puedeMover(dirección2)))
}

function distanciaAlBorde(direcciónAMedir) {
  cantidadDePasosAvanzados := 0
  // En bloques: "repetirHastaQue no puedeMover(direcciónAMedir)"
  while (puedeMover(direcciónAMedir)) {
    Mover(direcciónAMedir)
    cantidadDePasosAvanzados := cantidadDePasosAvanzados + 1
  }
  return (cantidadDePasosAvanzados)
}
```

Figura 22. Solución para “El radar”.

La solución tiene varios puntos que pueden resultar dificultosos, y en todo momento hay que ayudarlos con sugerencias y guías que los lleven a la solución adecuada. El consejo más útil es que realicen una división en subtareas, y luego se focalicen en cada subtaska de forma independiente, decidiendo cuáles herramientas de las aprendidas deben utilizar en cada caso. Los otros consejos se deberán regular según la respuesta de cada estudiante. Recordar también que si algún estudiante sigue un enfoque diferente para la solución, la idea es evaluarlo en función de si soluciona el problema y se puede entender leyendo el programa.

El primero de los puntos dificultosos es el uso de los parámetros **dir1** y **dir2** para determinar cuál es el cuadrante a buscar; todos los recorridos hechos hasta el momento usaron las direcciones Este y Norte, pero en este caso se hace necesario generalizarlas para decidir el cuadrante. Esto sigue el principio de procedimientos parametrizados utilizado en las actividades de la secuencia “Procedimientos con parámetros”, pero al tratarse de direcciones, puede resultar más complejo.

El segundo tiene que ver con el tema de buscar en una parte del tablero: en el procedimiento **IrALaCeldaSiguiente** no se puede usar **IrAlBorde** como antes, sino que es necesario saber el ancho del cuadrante, para volver al borde del cuadrante y no del tablero. Para esto, lo conveniente es que reciba el ancho; partes del código ya vienen comenzadas en el proyecto para guiarlos en este sentido.

El tercer punto conflictivo es el uso del procedimiento **BuscarEnCuadrante** como procesamiento para la función **hayEnCuadrante**, y la determinación de la condición a devolver. Dado que **BuscarEnCuadrante** mueve el cabezal hasta la celda que contiene bolitas del color buscado o hasta la última si no existe, la pregunta que debe hacer la función es si en la celda donde quedó el cabezal hay bolitas del color buscado. Por otra parte, y dado que es una función con procesamiento, la búsqueda en el cuadrante se hace solo de manera simulada, exactamente como si se tratase de un radar.

Finalmente, se les debe recordar en todo momento que tengan en mente los procedimientos y funciones hechos con anterioridad, para reutilizar los que hagan falta. En particular, en este programa se reutilizan la función **esLaÚltimaCelda** de los recorridos por celdas y la función **distanciaAlBorde** de la actividad “¿A qué distancia está el borde?”. Para reutilizar, pueden recordar la forma en que lo hicieron y volverlos a rearmar, o pueden utilizar una de

<Program.AR/>

las opciones del entorno para cargar código hecho con anterioridad; sería adecuado incentivarlos para que indaguen esta segunda opción.

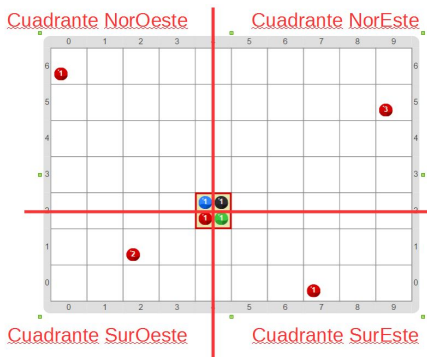
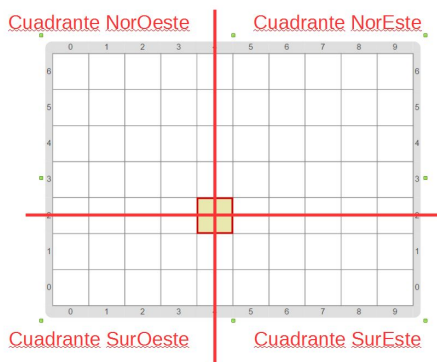
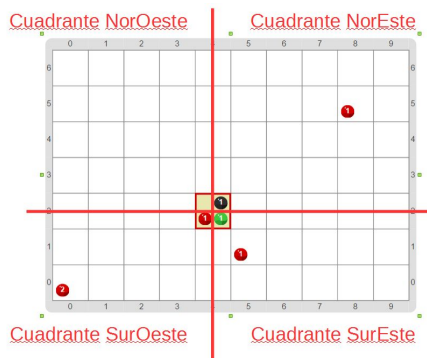
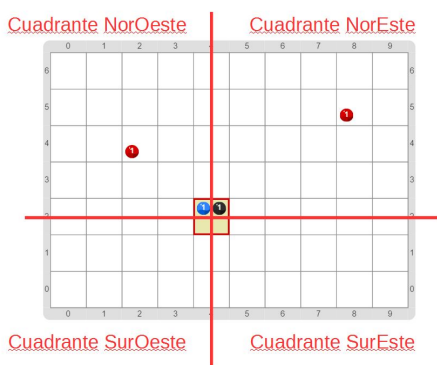
### Cierre

Para completar la actividad, se puede revisar la solución y explicitar cómo fueron utilizadas las diferentes herramientas aprendidas. También se puede marcar cómo el uso de cada herramienta es similar a lo realizado, siendo la complejidad el resultado de utilizarlas todas en combinación.

## Ficha para el alumno

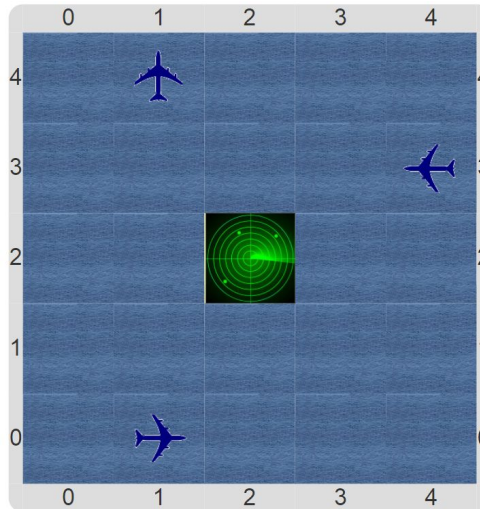
### El radar

¿Alguna vez te preguntaste cómo funciona un radar, como los que usan en las torres de los aeropuertos? Lo que hace es enviar una onda en una dirección, y si la misma vuelve rebotada es porque hay algo en ese lugar, con lo cual lo muestra como un punto en la pantalla; y repite esto en cada una de las direcciones. La idea en esta actividad es construir un programa que simule ser un radar, buscando en cada uno de los cuadrantes del tablero para ver si hay bolitas de un color dado. Si encuentra bolitas de ese color en el cuadrante NorEste, agrega una bolita negra; si las encuentra en el cuadrante NorOeste, agrega una azul; si las encuentra en el cuadrante SurEste, agrega una verde; y si las encuentra en el cuadrante SurOeste, agrega una roja; o sea, puede agregar entre cero y cuatro bolitas, según en qué cuadrantes encuentre lo que busca. Te mostramos algunos tableros finales; en el primero, se encontraron bolitas rojas en los cuadrantes noroeste y noreste y por eso hay una azul y una negra en la celda actual; en el segundo hay bolitas rojas en todos los cuadrantes menos en el noroeste; en el tercero no hay en ningún cuadrante; y en el último hay en todos.





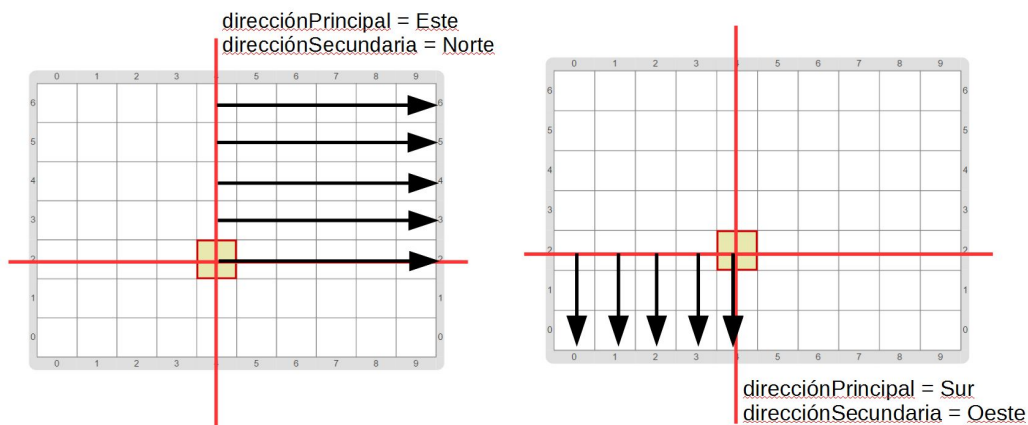
También te mostramos un tablero final con vestimenta:



Para llevar adelante la tarea, te proponemos que construyas:

- Un procedimiento **BuscarEnCuadrante**, que tome un color **colorABuscar** y dos direcciones, **direcciónPrincipal** y **direcciónSecundaria**, y busque una celda conteniendo bolitas del color dado en el cuadrante indicado por las direcciones. Si existe una celda como la buscada, el procedimiento deja el cabezal sobre ella; si no existe, lo deja en la última celda del cuadrante.
- Una función **hayEnCuadrante**, que tome un color y dos direcciones, y utilice **BuscarEnCuadrante** para determinar si hay o no bolitas del color indicado en ese cuadrante. Fijate que esta va a ser una función con procesamiento, por lo que la búsqueda es imaginaria, tal cual lo que hace el radar. Después de usar el procedimiento, ¿qué condición te va a decir si encontró lo que buscaba o no? Acordate que si hay una celda como la buscada, el procedimiento te deja sobre ella...

Para construir el procedimiento vas a tener que usar un recorrido de búsqueda. Tenés que tener cuidado, porque la condición de finalización tiene que tener *dos* partes: una para ver si encontraste lo buscado, usando **hayBolitas(color)**, y otra para ver si todavía hay celdas para seguir buscando, usando una función como **esLaÚltimaCelda**, pero con parámetros. Los parámetros de dirección indican en qué dirección se debe buscar; por ejemplo, si los parámetros fueran Este y Norte, sería una búsqueda como las que ya hiciste; si en cambio fueran Sur y Oeste, la búsqueda sería en otras direcciones; mirá las imágenes para guiarte.



Un último detalle es que para el recorrido tenés que hacer un procedimiento como **IrACeldaSiguiente**. En los recorridos por celdas de todo el tablero, ese procedimiento usa **IrAlBorde** para volver al principio de la siguiente fila a buscar. Pero como un cuadrante no ocupa todo el tablero, vas a tener que usar **Mover\_Veces** en lugar de **IrAlBorde**, y para eso tenés que recordar cuál es el ancho del tablero. Te damos algunas partes ya empezadas para que te guíes. ¿Te acordás que ya habíamos hecho una función para medir la distancia al borde? ¡Reutilizala!

Este ejemplo puede parecer complicado, pero eso es porque utiliza todas las herramientas que vimos hasta ahora. Sin embargo, si prestás atención, cada una de las herramientas se usa para lo mismo que habíamos hecho en otras actividades. ¡Y estas ideas te van a servir para construir un juego de Pacman en el próximo capítulo!

## Actividad

### Pidiendo empanadas

<p>Objetivo</p> <ul style="list-style-type: none"><li>• Ejercitar división en subtareas en el contexto de programas interactivos.</li></ul>	<p>Modalidad de trabajo</p> <p>De a 2</p>
<p>Materiales</p> <ul style="list-style-type: none"><li>• Computadoras con Gobstones</li><li>• El proyecto "Pidiendo empanadas"</li></ul>	

## Desarrollo

Pedir a los alumnos que abran el proyecto "Pidiendo empanadas", El mismo contiene solamente un tablero inicial y la configuración necesaria; los estudiantes deben construir un programa para que cumpla el propósito esperado: permitir la confección de un pedido de empanadas. Para ello pueden asociar las teclas de flechas derecha (→) e izquierda (←) a subtareas que permitan mover la empanada seleccionada a derecha e izquierda, y las teclas de retorno (ENTER) y suprimir (DEL) a subtareas para incrementar y decrementar la cantidad a pedir en la selección actual. Cabe destacar la diferencia con la actividad anterior, donde a cada tecla se le asociaba un único comando (**Poner**, **Mover** o **Sacar**); en este caso, en cambio, cada tecla será vinculada a una subtask más compleja, es decir, a un procedimiento.

Se espera que los alumnos logren llegar a una buena descomposición en subtareas; si no lo consiguen luego de un tiempo, pueden sugerírseles los siguientes procedimientos:

- **Seleccionar(...)**, que (de ser posible) se desplaza una fila hacia la derecha o izquierda (no se puede desplazar hacia arriba o abajo), según un parámetro dado, actualizando la selección de la fila usando bolitas rojas.
- **Incrementar**, que incrementa la cantidad asociada a la fila seleccionada.
- **Decrementar**, que decrementa la cantidad asociada a la fila seleccionada.

No se debería poder incrementar más allá de 48 ni decrementar por debajo de 0.

Estos procedimientos deben trabajar según la codificación propuesta en la Ficha: bolitas azules para el tipo de empanadas, bolitas negras para las cantidades y una bolita roja para indicar la fila seleccionada.

Una sugerencia que puede simplificar el trabajo consiste en mantener la siguiente propiedad: al finalizar el trabajo asociado a cada tecla, el cabezal debe estar ubicado sobre

la primera fila, en la columna seleccionada. Esto permite que al comenzar el trabajo de cada nueva tecla, no haya que reposicionar el cabezal desde un lugar desconocido y facilita la programación de cada uno de los procedimientos a definir. Por ejemplo el procedimiento **Seleccionar** puede temporalmente desplazar el cabezal a la segunda fila para actualizar la selección, volviendo a posarse sobre la primera fila antes de terminar. Una posible solución, en la Figura 2.

<pre>interactive program {   K_ARROW_RIGHT -&gt; { Seleccionar(Este) }   K_ARROW_LEFT  -&gt; { Seleccionar(Oeste) }   K_ENTER       -&gt; { Incrementar() }   K_DELETE      -&gt; { Decrementar() } }</pre>	<pre>procedure Incrementar() {   if (nroBolitas(Negro) &lt; 48)   then { Poner(Negro) } } procedure Decrementar() {   if (nroBolitas(Negro) &gt; 0)   then { Sacar(Negro) } }</pre>	<pre>procedure Seleccionar(dir) {   if (puedeMover(dir)) {   then { Mover(Norte)  Sacar(Rojo)         Mover(dir)         Poner(Rojo)    Mover(Sur)         }   } }</pre>
---	---	--

Figura 2. Programa interactivo para pedir empanadas.

### Cierre

Para completar la actividad, reflexionar sobre la similitud de esta aplicación y una planilla de cálculo. Muchos programas, como por ejemplo las planillas de cálculo, organizan información en tablas. Estos programas pueden consultar o incluso procesar la información haciendo uso de las mismas herramientas conceptuales vistas durante el curso: procedimientos, funciones, tratamiento de secuencias, variables, interactividad, etc.

## Ficha para el alumno

### Pidiendo empanadas

Viernes a la noche, la familia decide pedir empanadas. Cada integrante de la familia quiere una determinada cantidad de empanadas de una variedad particular. Quien se encarga de hacer el pedido no puede recordar las preferencias de cada persona, por lo que decidimos hacer un programa en donde cada cual sume al pedido cuántas empanadas quiere de cada variedad. De esta forma facilitamos la organización y evitamos comer una empanada que no nos gusta.

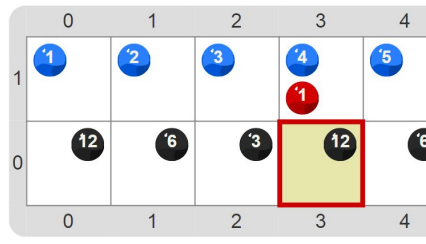
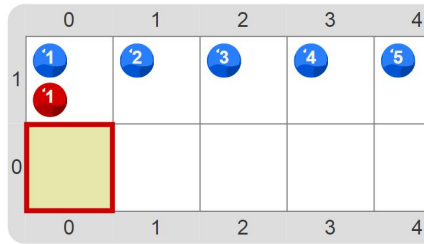
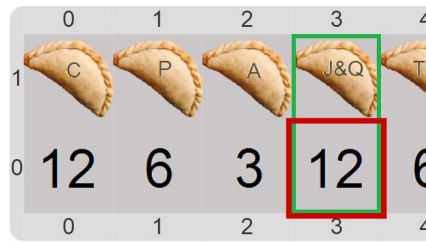
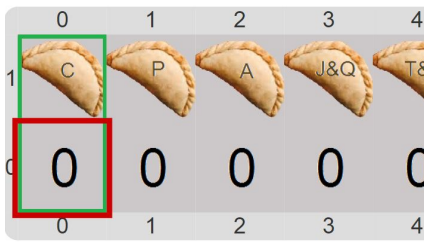
El programa trabaja sobre una "planilla" con 10 columnas, una por cada variedad de empanadas, y dos filas. La segunda fila indica la variedad de empanada y la primera, la cantidad a pedir. El cabezal siempre está ubicado sobre la primera fila, en la parte de la cantidad de una de las variedades de empanadas, y debe poder desplazarse a través de la lista hacia la derecha y la izquierda con las flechas de dirección. En la variedad seleccionada se puede indicar la cantidad de empanadas a pedir, incrementándola usando la tecla de retorno 'ENTER' o decrementándola usando la tecla de suprimir 'DEL'. No se debe permitir pedir más de 48 empanadas de una misma variedad ni menos de 0; esto se puede conseguir verificando la condición antes de realizar la operación de incremento o decremento. Observá que esta verificación se hace sobre el tablero actual, que debe ser modificado según la tecla que fue presionada; o sea, la tecla indica que se desea incrementar o decrementar *si se puede*, y la verificación establece si eso es posible. En caso de que no se pueda, no debe haber modificaciones.

#### Usamos la siguiente representación:

- Las variedades de empanadas se representan con entre 1 y 10 bolitas azules, ya presentes en el tablero inicial.
- La cantidad a pedir se representa con entre 0 y 4899 bolitas negras; inicialmente están todas en cero.
- El tipo de empanada actualmente seleccionado, sobre el cual se harán las operaciones de incremento y decremento, se representa con una bolita roja.

Te mostramos una parte del tablero inicial, y otra en medio de la ejecución (con y sin vestimenta). En el segundo tablero se ve que hay pedidas 12 empanadas del tipo 1, y 6 empanadas del tipo 2, 3 empanadas del tipo 3, 12 del tipo 4 y 6 del tipo 5; además la columna seleccionada es la cuarta.

<Program.AR/>



## Actividad

### Pacman

<p>Objetivo</p> <ul style="list-style-type: none"><li>• Ejercitar con un juego donde el oponente toma decisiones automáticamente.</li></ul>	<p>Modalidad de trabajo</p> <p>De a 2</p>
<p>Materiales</p> <ul style="list-style-type: none"><li>• Computadoras con Gobstones</li><li>• El proyecto "Pacman"</li></ul>	

## Desarrollo

Indicarles que deben abrir el proyecto "Pacman", y completarlo para que cumpla el propósito: permitir jugar una versión simplificada del juego Pacman, tal cual se describe en la Ficha. Como en toda actividad de programación avanzada es importante una correcta división en subtareas. Esta actividad además está dividida en dos partes bien diferenciadas: la primera concierne exclusivamente al movimiento del Pacman, mientras que la segunda, concierne al movimiento del Fantasma. Es importante guiar a los alumnos hacia la resolución ordenada de la actividad, ya que esto representa un desafío más gradual, primero resolviendo el juego sin el fantasma y recién después agregando la funcionalidad adicional.

En la Figura 7 se encuentra una posible solución a la primera parte, según la guía provista en la Ficha. En la solución propuesta la división en subtareas utiliza principalmente los procedimientos:

- **HacerJugadaDelPacman(direcciónDelPacman),**
- **MoverPersonaje(personajeAMover, direcciónAMover),**
- **PonerPersonaje(personajeAPoner, direcciónAMirar) y**
- **SacarPersonaje(personajeASacar),**

pero no es necesariamente la única forma de dividir el problema.

En la Figura 8 se encuentran los procedimientos necesarios para completar la segunda parte, también según la guía provista, agregando principalmente los procedimientos:

- **HacerJugadaDelFantasma y**
- **MoverFantasma(direcciónPrincipal, direcciónSecundaria).**

Los procedimientos reutilizados de ejercicios anteriores no se incluyen en la imagen.

Si algún estudiante ofrece una solución diferente a la propuesta, debe verificarse que el código sea legible y entendible además de que cumpla con el propósito. Si los estudiantes no llegan a una división en subtareas apropiada, guiarlos para que utilicen la división propuesta. La segunda parte de la actividad representa un desafío mayor, por lo que debe prestarse más atención a las dificultades que encuentren los estudiantes. Sin embargo,

debe darse un tiempo razonable para que puedan realizar su propia indagación, y las ayudas deben ser dadas en forma de sugerencia que los oriente y no que les resuelva el problema.

Una vez que completaron el código del programa, darles un tiempo razonable para que puedan jugar con él y ver su funcionamiento. Puede ser que durante este tiempo descubran pequeños errores en el código, que deberán ir buscando cómo solucionarlos. Este proceso de probar un programa se realiza también con programas profesionales; quizás los chicos sepan y mencionen a los que realizan tal tarea, conocidos como "testers", a partir de su experiencia con algunos juegos.

<pre>interactive program {   K_ARROW_UP   -&gt; { HacerJugadaDelPacman(Norte) }   K_ARROW_RIGHT -&gt; { HacerJugadaDelPacman(Este) }   K_ARROW_DOWN -&gt; { HacerJugadaDelPacman(Sur) }   K_ARROW_LEFT  -&gt; { HacerJugadaDelPacman(Oeste) }   TIMEOUT(1000) -&gt; { HacerJugadaDelFantasma() } }  procedure HacerJugadaDelFantasma() {   // Para la primera parte, el fantasma no se mueve }</pre>	<pre>function pacman() { return(Negro) } function fantasma() { return(Rojo) } function pastilla() { return(Verde) } function pared() { return(Azul) }  procedure HacerJugadaDelPacman(direcciónDelPacman) {   MoverPersonaje(pacman(), direcciónDelPacman)   if (hayPersonaje(fantasma()))     { BOOM("Game over: te comió el fantasma") }   if (hayPersonaje(pastilla()))     { Sacar(pastilla()) }   HacerJugadaDelFantasma() }</pre>	<pre>procedure MoverPersonaje(personajeAMover, direcciónAMover) {   SacarPersonaje(personajeAMover)   MoverSiPuede_(direcciónAMover)   if (hayPersonaje(pared())) { Mover(opuesto_(direcciónAMover)) }   PonerPersonaje(personajeAMover, direcciónAMover) }  procedure SacarPersonaje(personajeASacar) { SacarTodas_(personajeASacar) }  procedure PonerPersonaje(personajeAPoner, direcciónAMirar) {   if (direcciónAMirar == Norte) { Poner_Veces(personajeAPoner, 1) }   if (direcciónAMirar == Este) { Poner_Veces(personajeAPoner, 2) }   if (direcciónAMirar == Sur) { Poner_Veces(personajeAPoner, 3) }   if (direcciónAMirar == Oeste) { Poner_Veces(personajeAPoner, 4) } }</pre>
--	---	--

Figura 7. Juego de Pacman, sin movimientos del Fantasma.

<pre>procedure HacerJugadaDelFantasma() {   BuscarPersonaje_(fantasma())   if (hayPersonajeEnCuadrante(pacman(), Este, Sur))     { MoverFantasma(Este, Sur) }   if (hayPersonajeEnCuadrante(pacman(), Oeste, Sur))     { MoverFantasma(Oeste, Sur) }   if (hayPersonajeEnCuadrante(pacman(), Oeste, Norte))     { MoverFantasma(Oeste, Norte) }   if (hayPersonajeEnCuadrante(pacman(), Este, Norte))     { MoverFantasma(Este, Norte) }   BuscarPersonaje_(pacman()) }</pre>	<pre>procedure MoverFantasma(direcciónPrincipal, direcciónSecundaria) {   if (puedeMover(direcciónPrincipal))     { Mover(direcciónPrincipal)       if (hayPersonaje(pared())) { direcciónDefinitiva := direcciónSecundaria }       else { direcciónDefinitiva := direcciónPrincipal }       Mover(opuesto(direcciónPrincipal))     }   else { direcciónDefinitiva := direcciónSecundaria }    MoverPersonaje(fantasma(), direcciónDefinitiva)   if (hayPersonaje(pacman())) { BOOM("Game Over: te comió el fantasma") } }</pre>
---	--

Figura 8. Movimientos del Fantasma en el juego del Pacman.

En la Ficha se propone un desafío final, que es realizar un procedimiento para establecer si el Pacman ganó el juego, comiéndose todas las pastillas. Para este procedimiento deben hacer un recorrido sobre el tablero contando las pastillas, como se hizo antes en el Cap.8. Para utilizarlo, deben hacerlo después de que comieron una pastilla al mover el Pacman. Este desafío final puede no incluirse, sin afectar el propósito de la actividad. Una posible solución a este desafío se ofrece en la Figura 9 (incluyendo la modificación a **HacerJugadaDelPacman**).

<pre>procedure HacerJugadaDelPacman(direcciónDelPacman) {   MoverPersonaje(pacman(), direcciónDelPacman)   if (hayPersonaje(fantasma()))     { BOOM("Game over: te comió el fantasma") }   if (hayPersonaje(pastilla())) {     Sacar(pastilla())     VerificarSiGanóElJuego()   }   HacerJugadaDelFantasma() }  procedure VerificarSiGanóElJuego() { if (not hayMásPastillas()) { BOOM("¡Ganaste!") } }</pre>	<pre>function hayMásPastillas() {   totalDePastillas := 0   IrAlBorde(Sur) IrAlBorde(Oeste)   ancho := distanciaAlBorde_(Este)   while (hayCeldaSiguiente__(Este, Norte)) {     if (hayPersonaje(pastilla()))       { totalDePastillas := totalDePastillas + 1 }     IrACeldaSiguiente__(Este, Norte, ancho)   }   if (hayPersonaje(pastilla()))     { totalDePastillas := totalDePastillas + 1 }   return (totalDePastillas &gt; 0) }</pre>
---	--



<Program.AR/>

Figura 9. En rojo, código adicional para controlar si se ganó en el juego del Pacman.

## Cierre

Para cerrar se puede realizar una reflexión entre todos acerca de cómo las herramientas aprendidas durante el curso permiten construir una versión simplificada de un programa complejo y conocido. Además de lograr que se cumplan las reglas del juego podemos programar un juego con un jugador automático según la estrategia que nosotros programamos. Este comportamiento automático puede ser tan complejo como queramos. En este caso la estrategia del fantasma depende de un procedimiento de búsqueda sobre el tablero que nos permita detectar hacia qué lado debe moverse el fantasma para acercarse al Pacman.

## Ficha para el alumno

### Pacman

En esta última actividad del manual, vas a programar una versión simplificada del juego del Pacman. En este juego el personaje Pacman debe navegar por un laberinto hasta comer todas las pastillas que se encuentran desperdigadas. Los jugadores pueden controlar los movimientos del Pacman usando las flechas direccionales del teclado, pero como el Pacman no puede atravesar las paredes del laberinto, si el jugador hace un movimiento inválido, el Pacman debe quedarse en el lugar. Además de comer todas las pastillas en el laberinto, el Pacman debe evadir a un Fantasma que lo persigue. Por cada movimiento del Pacman el Fantasma también debe hacer un movimiento que lo acerque al Pacman, ya sea horizontal o verticalmente. Al igual que el Pacman, el Fantasma no puede atravesar las paredes, así que si intenta hacer un movimiento inválido debe quedarse en el lugar, aunque si tiene al menos una opción válida que lo acerque debe tomarla. Si el fantasma entra en el mismo casillero que el Pacman, el jugador pierde la partida y el programa debe hacer BOOM.

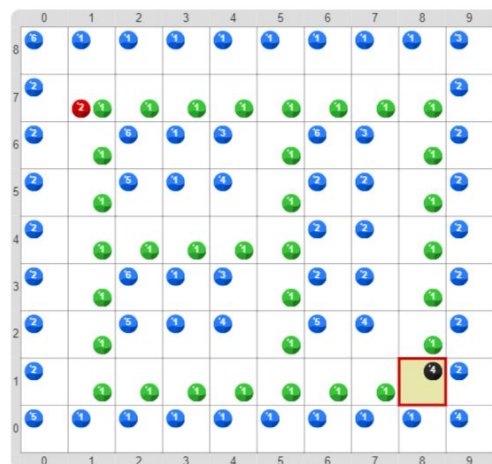
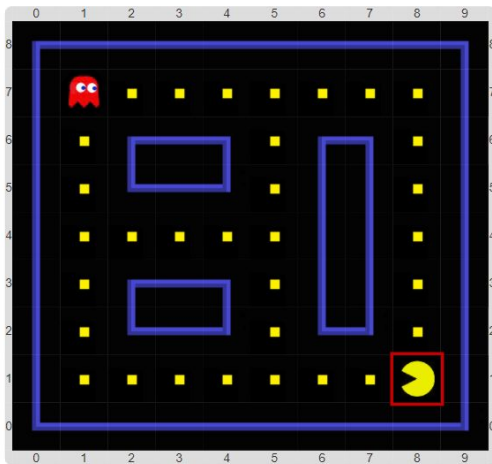
#### El dato

Pacman es un juego de consola desarrollado en 1980, pionero de los juegos de persecución en laberintos. Ese juego y sus derivados se convirtieron en uno de los juegos clásicos más populares en existencia.

A) En primer lugar te recomendamos resolver solamente los movimientos del Pacman, y dejar el tema del Fantasma para una segunda parte. Para eso, primero tenés que entender cómo representar cada uno de los elementos, y después hacer los procedimientos que mueven al Pacman.

La representación es la siguiente:

- Las paredes se representan con bolitas azules, entre 1 y 6 dependiendo de si es una pared vertical, horizontal o una de las cuatro esquinas.
- El Pacman se representa con bolitas negras, entre 1 y 4 dependiendo de si mira hacia el Norte, Este, Sur u Oeste, respectivamente.
- Las pastillas se representan con una bolita verde.
- El Fantasma se representa con bolitas rojas, también entre 1 y 4, con la misma codificación de direcciones que el Pacman.



Podrías comenzar haciendo funciones para decir **pacman** en lugar de Negro, **fantasma** en lugar de Rojo, etc. y también procedimientos para poner y sacar a los personajes, como ser **SacarPersonaje(personajeASacar)** y **PonerPersonaje(personajeAPoner, direcciónAMirar)**. También podés hacer un procedimiento para mover a un personaje; recordá que mover a un personaje es sacarlo de una celda y ponerlo en la celda siguiente, pero solamente si en la celda destino no hay una pared. Para ver si hay un personaje en una celda, ya te damos construida una función **hayPersonaje(personaje)**.

Luego tendrías que construir un procedimiento para hacer la jugada del Pacman; recordá que el Pacman después de moverse debe ver si perdió; si perdió, termina el programa (con un BOOM, por ejemplo), pero si no perdió, debe comer la pastilla si la encontró (el control para ver si ganó, dejalo para después). Para terminar esta parte, tendrías que asociar las teclas correspondientes con las jugadas del Pacman.

### Ayuda

Recordá que es fundamental una buena división en subtareas a la hora de programar. Y que podés reutilizar varios de los procedimientos y funciones hechos con anterioridad: **Poner\_Veces**, **Mover\_Veces**, **Sacar\_Veces**, **SacarTodas**, **MoverSiPuede**, etc. ¡Para que no tengas que hacerlos de nuevo, te los damos como procedimientos primitivos!

- B) En segundo lugar tenés que hacer un procedimiento para hacer la jugada del Fantasma. Para esto te recomendamos reutilizar el procedimiento **BuscarEnCuadrante(color, direcciónHorizontal, direcciónVertical)** y la función **hayEnCuadrante(color, dirección1, dirección2)** de la actividad "El radar" del Cap.8 (ya te las damos hechas, con los nombres un poquito más adecuados a esta actividad). También te recomendamos construir un procedimiento para decidir cómo mover al fantasma. Entonces, para hacer la jugada del Fantasma, primero llevamos el cabezal a dónde está el Fantasma, luego buscamos al Pacman en cada uno de los cuatro cuadrantes, y hacemos una movida del Fantasma hacia el cuadrante donde se encuentra el Pacman, y terminamos devolviendo el cabezal a la ubicación del Pacman (para que el cabezal siempre esté sobre un lugar fijo, en este caso, el Pacman).
- Para moverse hacia un cuadrante hay dos posibilidades, ya que hay dos direcciones: una principal y una secundaria (todo cuadrante queda dado por dos direcciones). Siempre que pueda moverse, debe hacerlo, por lo que hay que controlar si se puede mover en la

dirección principal, y si puede, moverse en esa dirección, pero si no puede, intentarlo en la dirección secundaria. No te olvides de controlar si el Fantasma se encontró con el Pacman, ya que en ese caso, el juego terminó...

- C) Un desafío final, pero que no es necesario para poder jugar, es hacer una función que cuente el número de pastillas que quedan, y controlar así, después de comer cada pastilla, si el Pacman ganó, cuando no quedan más pastillas. La cuenta de pastillas se haría de forma muy parecida a como hiciste en la segunda parte de la actividad *"Contar las baldosas"* del Cap.8 con una función con procesamiento usando un acumulador.