

# El funcionamiento interno

Esta secuencia didáctica está compuesta por una única actividad, en la que mediante un juego de rol las/os estudiantes simularán la ejecución de bajo nivel de un programa sobre una computadora imaginaria. Esto permite comprender la función de cada componente, observar que las instrucciones que ejecutan las computadoras son sencillas, y reflexionar sobre lo que se puede hacer programando computadoras.

## Actividad 1. La máquina criptográfica

Presentación de la arquitectura de una computadora imaginaria y descripción de cada uno de sus componentes y sus respectivas funciones. Simulación de su funcionamiento interno con un juego de rol, en el que van a participar las y los estudiantes.

## Actividad 2. La memoria rápida y la memoria grande

Análisis de distintos tipos de memoria de acuerdo con si la información que contiene es o no volátil, el volumen de información que pueden almacenar y la velocidad a la que se puede acceder a su contenido.

## Datos curriculares

**Nivel:** Primario, segundo ciclo y Secundaria básica

**Área:** Infraestructura tecnológica

**Eje:** Organización y arquitectura de computadoras

### Contenidos

- La computadora como un sistema integrado por *hardware* y *software*.
- Modelo de máquina programable: arquitectura von Neumann.
- Componentes de *hardware*.
- Ejecución a bajo nivel: lenguaje de máquina, representación binaria e impulsos eléctricos.

## Objetivos de aprendizaje

- Exhibir el funcionamiento interno de una computadora.
- Mostrar que las computadoras solo ejecutan un conjunto de instrucciones muy simples.

## Materiales necesarios

- Tizas o marcadores de pizarra.
- Borrador.
- Cartulina.
- Fichas para estudiantes.

Todos los recursos necesarios para esta secuencia están disponibles en: <https://curriculum.program.ar/>  
Podés buscarlos por el título de la secuencia.

# Acerca de esta iniciativa

Desde el sitio [curriculum.program.ar](http://curriculum.program.ar) tenemos por objetivo acompañar a la comunidad docente de habla hispana en el desafío de llevar las Ciencias de la Computación al aula.

Para ello, construimos un repositorio que reúne diversos recursos para el aula que desde la Iniciativa Program.AR de la Fundación Sadosky impulsamos desde 2013.

Organizados a partir de los saberes a promover con nuestras y nuestros estudiantes y los conceptos de la disciplina presentados en la [Propuesta curricular para la inclusión de las Ciencias de la Computación \(CC\) en el aula](#), encontrarán en [curriculum.program.ar](http://curriculum.program.ar) proyectos, secuencias didácticas y actividades desarrollados por una diversidad de autores y docentes en conjunto con instituciones y universidades de América Latina.

Estos materiales, que han sido desarrollados para responder a necesidades de diferentes contextos y países y que son heterogéneos en su formato y extensión, comparten un mismo propósito: integrar las Ciencias de la Computación en la escolaridad obligatoria para promover en el conjunto de los y las estudiantes la construcción de saberes que les permitan comprender, apropiarse y transformar la tecnología digital y computacional y así participar de manera crítica del mundo contemporáneo.

## Cómo utilizar este recurso

Siguiendo la Propuesta curricular, es posible organizar una planificación escolar para el grado o el año a abordar y, a partir de ella, seleccionar del universo de recursos para el aula que ofrecemos los que sean adecuados al contexto y la realidad de cada grupo de estudiantes.

Al acceder a esta secuencia en el sitio [curriculum.program.ar](http://curriculum.program.ar), encontrará los enlaces para descargar los materiales anexos que fueren necesarios.

## Instituciones



Ministerio de Ciencia,  
Tecnología e Innovación  
Argentina



FAMAF  
Facultad de Matemática,  
Astronomía, Física y  
Computación



UNC  
Universidad  
Nacional  
de Córdoba

## Fuente

Areces, C., Benotti, L., Cortez, J. J., et.al (2018). *Ciencias de la computación para el aula: 2do. ciclo de primaria: libro para docentes*. Ciudad Autónoma de Buenos Aires: Fundación Sadosky.

<http://program.ar/manual-segundo-ciclo-primaria/>



Fundación  
SADOSKY

<Program.AR/>



## Secuencia Didáctica 2

# EL FUNCIONAMIENTO INTERNO

Esta secuencia didáctica está compuesta por una sola actividad llamada “La máquina criptográfica”. En ella se presenta la arquitectura de una computadora imaginaria y se describe cada uno de sus componentes y sus respectivas funciones. Los componentes del *hardware* son la memoria rápida, la memoria grande, la unidad central de procesamiento y el puntero de instrucción.

Luego se simula su funcionamiento interno con un juego de rol, en el que van a participar distintos estudiantes del curso. Debido a la extensión de la actividad, puede hacer falta más de una clase para completarla.

.....

### OBJETIVOS

- Exhibir el funcionamiento interno de una computadora.
- Mostrar que las computadoras solo ejecutan un conjunto de instrucciones muy simples.

.....



## Actividad 1

### La máquina criptográfica



GRUPAL (4)

#### OBJETIVOS

- Emular el funcionamiento de bajo nivel de una computadora.
- Presentar el funcionamiento de la unidad central de procesamiento.

#### MATERIALES



Tizas o marcador de pizarra



Borrador



Cartulina



Ficha para estudiantes

#### DESARROLLO

En esta actividad presentamos la máquina criptográfica, una computadora especialmente diseñada para poder cifrar mensajes. Mostramos cada uno de sus componentes y explicamos sus funciones. Además, observamos cómo se integran dentro de la arquitectura general de una computadora imaginaria.

Comenzamos la actividad repartiendo la ficha a los estudiantes. Allí se encuentra una descripción de la máquina criptográfica. Repasamos con los estudiantes todo lo referente a sus componentes y su funcionamiento. Además, les presentamos las reglas del juego de rol con el que simularán el comportamiento interno de esta particular computadora. Es importante que nos tomemos el tiempo necesario para que los estudiantes tengan un panorama claro sobre lo que harán. Sin un acuerdo en este sentido, difícilmente la actividad pueda llegar a buen puerto. A continuación, se ofrecen descripciones que pueden usarse como disparadores en el intercambio con los estudiantes.

#### EL HARDWARE DE LA MÁQUINA CRIPTOGRÁFICA

##### 1. Memoria rápida

La máquina criptográfica tiene dos memorias: una rápida y una grande. La memoria rápida consiste en un único casillero y la representamos como un cuadrado dibujado en el pizarrón. Allí se puede escribir de a un símbolo a la vez. El alfabeto disponible incluye las letras del abecedario, los dígitos del 0 al 9, el espacio, y los signos de exclamación e interrogación, tanto de apertura como de cierre.



Memoria rápida de un casillero

A, B, C, D, E, F, G, H, I, J, K, L, M, N, Ñ, O, P,  
Q, R, S, T, U, V, W, X, Y, Z, 0, 1, 2, 3, 4, 5, 6,  
7, 8, 9, , !, ., ?

Alfabeto de la máquina criptográfica

##### 2. Memoria grande

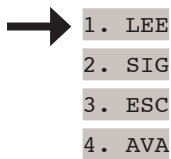
También podemos dibujarla en el pizarrón. Consiste en una tira de 10 celdas. En ellas se pueden guardar los mismos símbolos que en el casillero de la memoria rápida. Además, el componente tiene un cabezal que siempre se encuentra posicionado sobre una celda. Se puede leer el carácter almacenado en ella o escribir allí uno nuevo.



Memoria grande

### 3. Puntero de instrucción

En el pizarrón deberemos copiar el programa que ejecutará la máquina criptográfica. En esta máquina, un programa es una lista numerada de instrucciones (la descripción de las instrucciones se encuentra más adelante). Las escribiremos en el pizarrón, una debajo de la otra. La computadora cuenta con un puntero de instrucción, que representaremos con una flecha. Siempre apunta a una instrucción, que es la que la máquina tiene que ejecutar.



```
1. LEE
2. SIG
3. ESC
4. AVA
```

Puntero de instrucción

### 4. Unidad central de procesamiento

Por último, tenemos la unidad central de procesamiento (CPU), que dirige todo el funcionamiento. Es el componente que se encarga de ejecutar una a una las instrucciones de un programa. Para hacerlo, repite lo que se conoce como **ciclo de instrucción**: (i) lee la instrucción apuntada por el puntero de instrucción; (ii) la ejecuta; y (iii) mueve el puntero para que pase a apuntar a la próxima instrucción que el programa tiene que ejecutar. Estos ciclos se interrumpen solo cuando se ha completado una corrida del programa.

### Dinámica de funcionamiento del juego

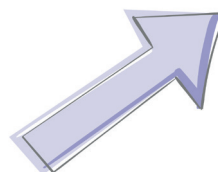
Para poner a funcionar la máquina criptográfica, hace falta que cuatro estudiantes pasen al frente. Cada uno de ellos interpretará un rol y recibirá uno o varios elementos específicos.

**OBJETOS NECESARIOS PARA EL JUEGO**

Recomendamos preparar previamente los siguientes elementos para que los estudiantes puedan interpretar cada uno de los roles:

**UNIDAD  
CENTRAL  
PROCESAMIENTO**

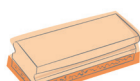

Rollo de cartulina a  
modo de megáfono

**PUNTERO DE  
INSTRUCCIÓN**


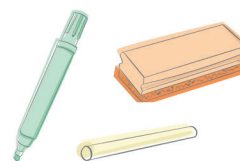
Flecha de cartulina  
para apuntar  
instrucción

**MEMORIA RÁPIDA**


Marcador o tiza  
(según tipo  
de pizarrón)



Borrador

**MEMORIA GRANDE**


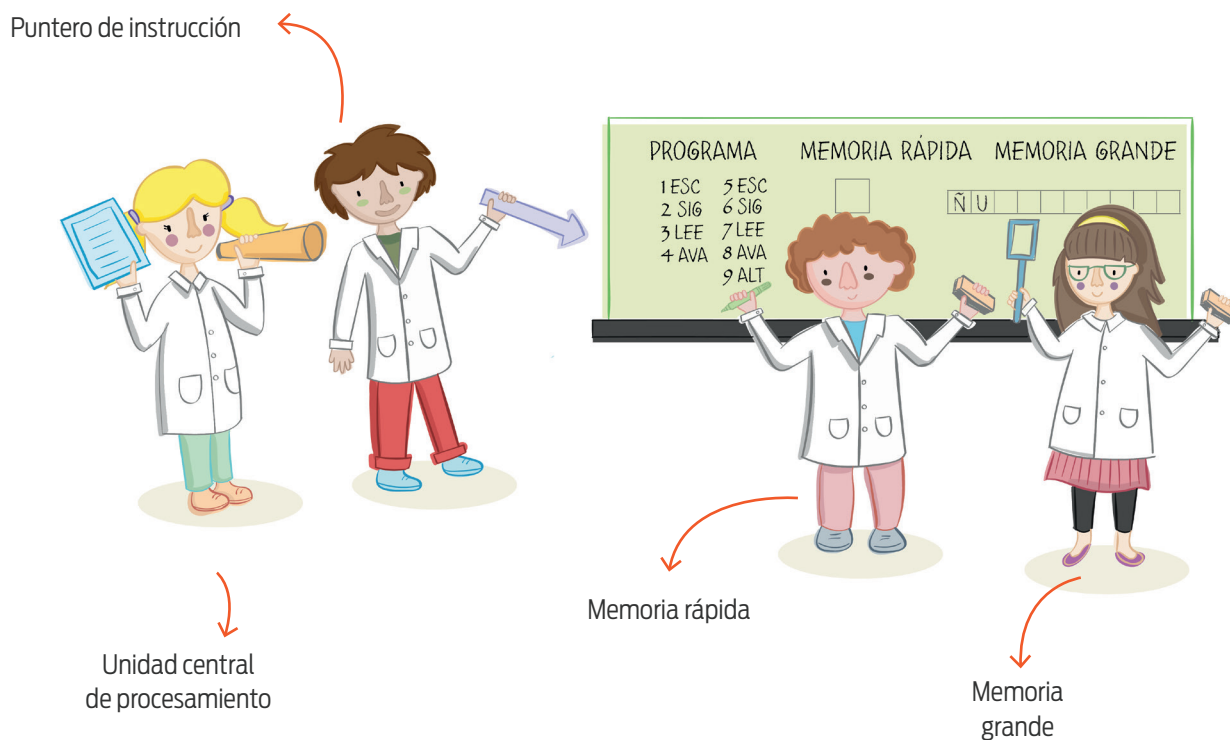
Marcador o tiza  
+ borrador



Cabezal para indicar  
celda (puede ser  
una vara, un rollo de  
cartulina, etc.)

Las tareas correspondientes al rol que desempeñará cada estudiante son las siguientes:

1. **Puntero de instrucción:** debe sostener la flecha que representa el puntero de instrucción. Al comenzar la ejecución de un programa, debe señalar la primera instrucción.
2. **Unidad central de procesamiento:** su función es ir haciendo avanzar sucesivas veces el ciclo de instrucción. Cada vez, debe (i) consultar al puntero de instrucción cuál es la instrucción del programa que tiene que ejecutarse; (ii) dar indicaciones al resto de los componentes para llevarla a cabo; y (iii) indicarle al puntero de instrucción a dónde apuntar, de forma tal de que quede señalando la próxima instrucción del programa que tiene que ejecutarse.
3. **Memoria rápida:** siguiendo las instrucciones que recibe de la unidad central de procesamiento, lee el contenido del casillero o escribe allí un nuevo símbolo.
4. **Memoria grande:** su tarea consiste en desplazar el cabezal y leer y escribir símbolos en las celdas. Siempre debe hacerlo siguiendo instrucciones impartidas por quien interprete el rol de la unidad central de procesamiento. Al comenzar una ejecución, el cabezal tiene que estar sobre la celda ubicada en el extremo izquierdo.



### El conjunto de instrucciones

Estas son las instrucciones que pueden formar parte de los programas de la máquina criptográfica.

#### AVA

Mueve el cabezal de la memoria grande una posición hacia la derecha. Si el cabezal se encuentra en la última celda, vuelve a posicionarse sobre la primera.

#### ESC

Escribe en la memoria rápida el contenido de la celda de la memoria grande apuntada por el cabezal.

#### LEE

Lee el contenido de la memoria rápida y lo escribe en la celda de la memoria grande apuntada por el cabezal.

#### SIG

Reemplaza el símbolo de la memoria rápida por el siguiente, de acuerdo con el orden del alfabeto que usa la máquina. Al aplicarlo sobre el símbolo ?, el contenido pasa a ser A.

#### ANT

Reemplaza el símbolo de la memoria rápida por el anterior, de acuerdo con el orden del alfabeto que usa la máquina. Al aplicarlo sobre el símbolo A, el contenido pasa a ser ?.

#### SAL [nro de línea]

Hace apuntar al puntero de instrucción hacia la línea **nro de línea**, que es un parámetro de la instrucción.

#### SI MR = [valor] SAL [nro de línea]

Si el contenido de la memoria rápida es **valor**, hace apuntar al puntero de instrucción a la línea **nro de línea**.

#### ALT

Termina la ejecución del programa.

### INSTRUCCIONES DE LOS LENGUAJES DE MÁQUINA

Los lenguajes de máquina provistos por computadoras reales están compuestos por unas pocas instrucciones, cada una de ellas de tres letras. Por tal motivo, en esta actividad usamos esa misma cantidad de letras para el nombre de cada instrucción.

Una vez que todo el grupo haya comprendido las características de la máquina criptográfica, pasamos a realizar las consignas de la ficha de la actividad. En ellas hay distintos programas que serán emulados por los grupos de estudiantes. Sugerimos que cada programa sea resuelto por un grupo distinto, de forma tal de que varios tengan la oportunidad tanto de interpretar a un componente de la máquina como de ser testigos oculares de la ejecución de un programa llevada a cabo por otros. Mirar la máquina desde afuera da una buena perspectiva sobre cómo funciona internamente una computadora.

### Primera ejecución

La primera consigna presenta el programa Encripto 2. Al ejecutarse, modifica el contenido de las dos primeras posiciones de la memoria grande. Reemplaza lo almacenado en cada celda por el símbolo siguiente de acuerdo con el orden del alfabeto usado en la actividad. Por lo tanto, hace que la máquina criptográfica ejecute el algoritmo César estudiado en el capítulo 1, esta vez sobre mensajes de dos símbolos.

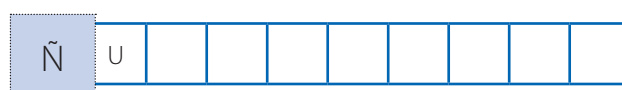
### PARA REFRESCAR EL MÉTODO DE CIFRADO CÉSAR

Como vimos en la secuencia didáctica 3 del capítulo 1, “Ciudadanía digital y seguridad”, existe una técnica para enviar mensajes secretos llamada **cifrado César** o **cifrado de desplazamiento**. En esta técnica se sustituye una letra por otra que se encuentra más adelante en el alfabeto, según el desplazamiento elegido. Por ejemplo, usando un desplazamiento de una sola posición, la A se reemplaza por la B, la B por la C y así sucesivamente.

Copiamos en el pizarrón el siguiente programa:

1 ESC	5 ESC
2 SIG	6 SIG
3 LEE	7 LEE
4 AVA	8 AVA
	9 ALT

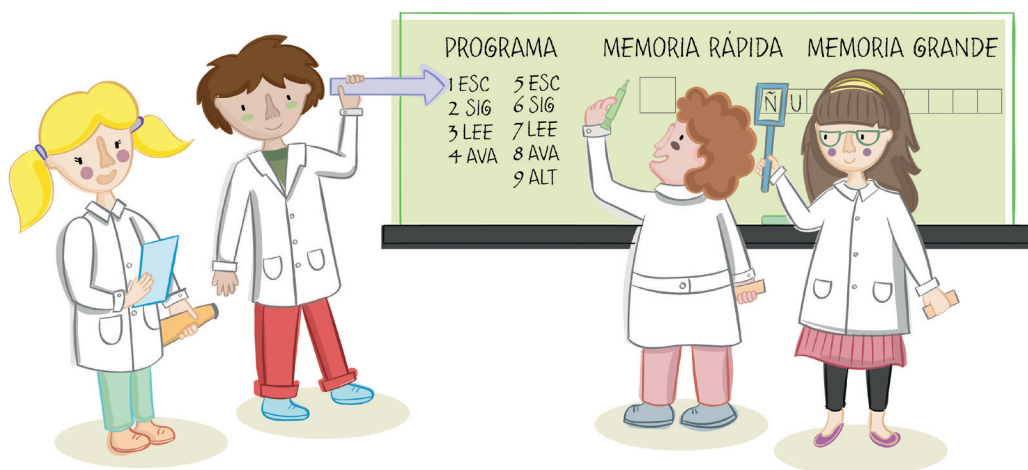
Inicialmente, en la memoria grande se encuentra escrita la palabra ÑU seguida de ocho espacios en blanco. Por su parte, el cabezal se encuentra sobre la celda que almacena la Ñ.



Al finalizar una ejecución, el contenido original de la primera celda habrá sido reemplazado por el símbolo O y el de la segunda, por V. El resto se mantendrá inalterado.



Hacemos pasar a cuatro estudiantes al frente y le asignamos un rol a cada uno. El primero debe sostener la flecha del **puntero de instrucción**, que comenzará apuntando a la primera instrucción. Otro tiene que leer y escribir la **memoria rápida**, que al empezar contiene un espacio en blanco. El tercero debe representar a la **memoria grande** manipulando el cabezal, que al iniciar un programa se encuentra posicionado sobre la celda del extremo izquierdo. Además, a medida que avance el juego, irá leyendo y escribiendo celdas siguiendo instrucciones impartidas por quien asuma el rol de la **unidad central de procesamiento**. Es conveniente que este último tenga a mano la ficha con la descripción del lenguaje de la máquina criptográfica para chequear rápidamente qué es lo que hace cada instrucción.



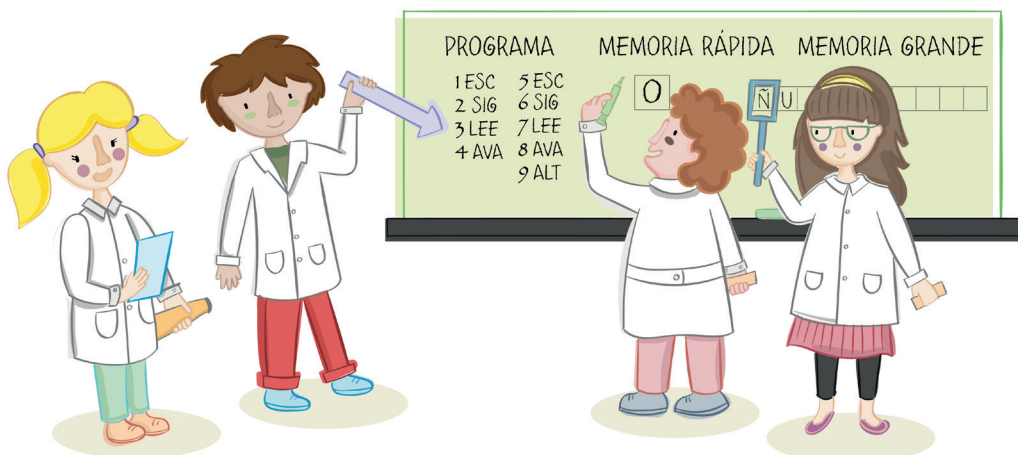
Estado inicial de la máquina criptográfica

Para dar comienzo a la ejecución del programa, el estudiante que interpreta a la unidad central de procesamiento le pregunta al puntero de instrucción qué instrucción del programa tiene que ejecutarse. En este caso, recibe como respuesta: “La instrucción actual es **ESC**”. A continuación, le pregunta a quien representa el rol de la memoria grande: “¿Qué hay en la celda bajo el cabezal?”. La memoria grande contesta: “La letra **Ñ**”. Luego, le dice a la memoria rápida: “Vos, escribí una **Ñ**.” El participante que actúa de memoria rápida escribe entonces el símbolo **Ñ** en la memoria monoceldaica. Finalmente, quien dirige la ejecución pide a quien interpreta al puntero de instrucción que señale la siguiente línea del programa. Con esto, completamos el primer ciclo de instrucción. O sea, llevamos a cabo la ejecución de la primera instrucción de Encripto 2.



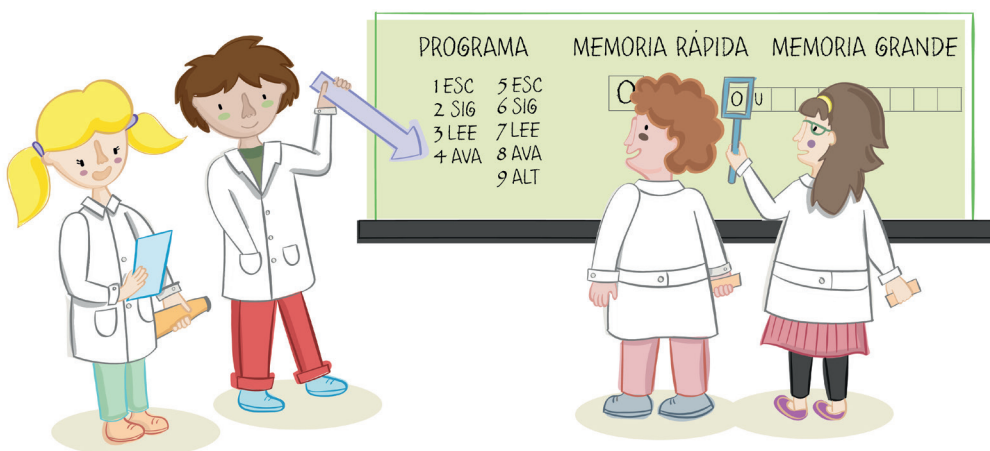
Estado de la máquina criptográfica luego de completar el primer ciclo de instrucción

El segundo ciclo de instrucción comienza con la unidad central de procesamiento preguntando a viva voz: “¿Cuál es la instrucción actual?”. El puntero de instrucción contesta: “La instrucción actual es **SIG**”. Entonces, le indica a la memoria rápida: “Pasá al siguiente símbolo”. Acto seguido, borra la Ñ y escribe una O. Entonces, el que interpreta a la CPU indica al puntero de instrucción que pase a apuntar a la siguiente instrucción, cosa que hace inmediatamente.



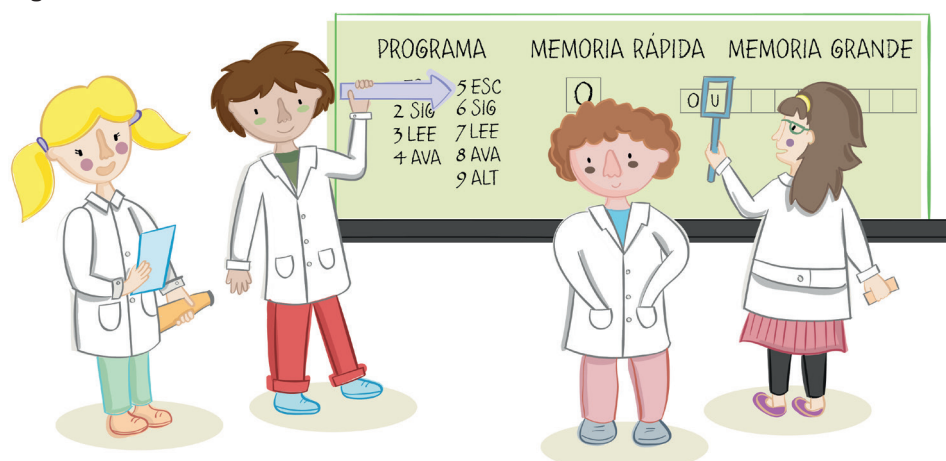
Estado de la máquina criptográfica luego de completar el segundo ciclo de instrucción

Al comenzar el tercer ciclo, la unidad central le consulta al puntero de instrucción: “¿Qué instrucción tenemos que ejecutar ahora?”. Este contesta: “La instrucción actual es **LEE**”. A continuación, le dice a la más pequeña de las memorias: “¿Qué hay allí?”, a lo que esta responde: “Una O”. Luego, le dice a la memoria grande: “Escribí una O”. La orden es obedecida inmediatamente. Finalmente, guía al puntero de instrucción para que avance un renglón.



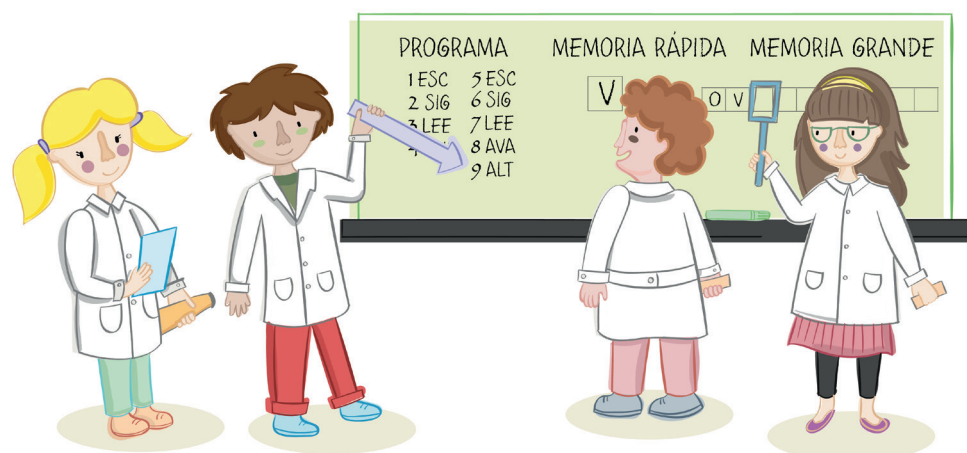
Estado de la máquina criptográfica luego de completar el tercer ciclo de instrucción

El cuarto ciclo de instrucción arranca con la CPU preguntando: “¿Cuál es la instrucción actual?”, a lo que el puntero de instrucción responde: “La instrucción actual es **AVA**”. Acto seguido, le dice a la memoria grande: “Avanzá el cabezal una posición”. El cabezal se mueve una celda hacia la derecha y queda apuntando a la que contiene la letra **U**. Como antes, esta vuelta finaliza indicando al puntero que señale la siguiente instrucción.



Estado de la máquina criptográfica luego de completar el cuarto ciclo de instrucción

Si observamos el programa con atención, podemos notar que las instrucciones entre las líneas 1 y 4 son las mismas que las que están entre las líneas 5 y 8. La máquina, entonces, reproducirá a continuación los mismos pasos ya hechos. Al finalizarlos, la segunda celda de la memoria grande contendrá el símbolo **V** y el cabezal estará sobre la celda de la tercera posición. Además, el puntero de instrucción señalará la instrucción **ALT**. Por lo tanto, la máquina se detendrá y habrá finalizado la ejecución del programa.



Estado de la máquina criptográfica al finalizar la ejecución del programa Encrypto 2

## Reflexión intermedia

Una vez concluida la ejecución de Encripto 2, es el momento apropiado para reflexionar sobre lo ocurrido.

En primer lugar, podemos hacer notar a los estudiantes que la unidad central de procesamiento lo único que hace es repetir mecánicamente una serie de pasos hasta que la ejecución de un programa finaliza. A su manera, este componente también ejecuta un programa que viene directamente incorporado en él. Es decir, viene programado de fábrica para repetir una y otra vez un ciclo de instrucción.

A continuación, subrayamos que cada uno de los pasos que realiza la CPU es extremadamente simple. Aun así, pudimos ver que un programa escrito en lenguaje de máquina puede llevar a cabo una tarea que tiene un interés práctico: encriptar mensajes. En general, las computadoras proveen un lenguaje con el que se puede programar su comportamiento. Suele tratarse de lenguajes con instrucciones muy rudimentarias, que permiten establecer cómo se procesan los datos mientras viajan por los componentes de la máquina. Vulgarmente, se los denomina *lenguajes de bajo nivel*.

Resulta oportuno aprovechar la ocasión para tender un puente entre lo hecho durante el desarrollo de la actividad y lo ejercitado en capítulos anteriores. Hasta aquí, cada vez que construimos programas para ejecutar en la computadora, hemos usado Scratch. Considerando que las computadoras solo cuentan con un conjunto de instrucciones muy básicas, ¿cómo hacen realmente para ejecutar nuestras producciones?

Scratch, como casi todos los lenguajes de programación de uso corriente, pertenece al grupo de los llamados *lenguajes de alto nivel*. Son lenguajes considerablemente más expresivos que los que las computadoras traen de fábrica. Casi todos los lenguajes de alto nivel omiten por completo cómo se procesa y circula la información entre los componentes físicos de la computadora. Al escribir un programa usando cualquiera de ellos, se produce un proceso de traducción. Este proceso se llama *compilación* y lo lleva adelante un programa llamado *compilador*. Toma como entrada un programa escrito en un lenguaje de alto nivel y produce como salida un programa en lenguaje de máquina, cuyo comportamiento es idéntico al del programa recibido como entrada.

Existe un mecanismo alternativo al de la compilación. En este caso, contamos con un programa llamado *intérprete*, generalmente programado en el lenguaje de la máquina. Al igual que el compilador, recibe un programa como entrada. Sin embargo, en lugar de generar una salida, emula paso a paso la ejecución del programa recibido.

## Segunda ejecución

La segunda consigna plantea hacer una ejecución de Encripto 4. Si bien es parecido a Encripto 2, no es idéntico. La secuencia de cuatro instrucciones que observamos que se repetían, en este programa aparece 4 veces.

1 ESC	5 ESC	9 ESC	13 ESC
2 SIG	6 SIG	10 SIG	14 SIG
3 LEE	7 LEE	11 LEE	15 LEE
4 AVA	8 AVA	12 AVA	16 AVA
			17 ALT

En este caso, una ejecución del programa aplicará el algoritmo César sobre cuatro caracteres. Al momento de arrancar, en la memoria grande se encuentra almacenada la palabra *PELO*. Al finalizar, *QFMP*.

## Tercera ejecución

La novedad de la tercera consigna es que el programa propuesto incorpora instrucciones del lenguaje que inspeccionan condiciones y permiten provocar saltos. Al igual que al programar en Scratch, un programa de la máquina criptográfica puede evaluar una condición y, de acuerdo a si es cierta o no, hacer una u otra cosa. En esta máquina, lo que puede evaluarse es si el contenido de la memoria rápida es igual a un determinado símbolo, y en caso de que así sea, hacer apuntar al puntero de instrucción a una instrucción arbitraria; caso contrario, el puntero avanza a la siguiente instrucción del programa.

```

1 ESC
2 SI MR = ! SAL 7
3 SIG
4 LEE
5 AVA
6 SAL 1
7 ALT

```

El programa propuesto se llama Corriendo hasta la exclamación, e inicialmente, el contenido de la memoria grande es "¡TERMINAR!". Al correr, el programa aplicará el algoritmo César sobre todos los símbolos que aparezcan antes del símbolo *!*. Por lo tanto, al finalizar la corrida la memoria grande contendrá el texto "¡UFSNÑBS!".

El programa comienza copiando el contenido de la memoria grande bajo el cabezal en el casillero de la memoria rápida. La siguiente es una instrucción condicional que pregunta si el contenido de la memoria rápida es el signo de exclamación de cierre. En caso afirmativo, salta hasta la última instrucción del programa, que detiene la máquina. En caso contrario, pasa a la siguiente instrucción. Las instrucciones entre las líneas 3 y 5 se ejecutan una tras otra y no presentan innovación alguna en relación con las de



ambas versiones de Encrypto: reemplazan el símbolo de la celda bajo el cabezal de la memoria grande por el siguiente, de acuerdo con el orden del alfabeto de la máquina.

La instrucción de la línea 6, una vez alcanzada, provoca que el puntero de instrucción pase a señalar la primera instrucción, de modo que hace comenzar todo otra vez. Este proceso se repetirá hasta que el cabezal quede posicionado sobre una celda que contenga el cierre de una exclamación. Además, este es el único modo de alcanzar la última instrucción del programa. Si tal símbolo no se encontrase en la memoria al comenzar, el programa no se detendría, se ejecutaría hasta el fin de los tiempos.

### Reflexión de cierre

Combinando un salto condicional con uno incondicional conseguimos el mismo efecto que se produce con las instrucciones de un ciclo de cualquier lenguaje de alto nivel, como por ejemplo, Scratch. Al ejecutarse, el programa repite un bloque de instrucciones hasta que se alcanza una cierta condición.

Resulta interesante reflexionar sobre qué ocurriría si al comenzar la ejecución ninguna celda de la memoria grande tuviese impreso el símbolo !. Como vimos, en ese caso el programa no se detendría nunca. De acuerdo con la terminología de la teoría de la computabilidad, Corriendo hasta la exclamación calcula una función parcial. Es decir, una función que no está definida frente a ciertas entradas. En el plano de esta máquina, esto quiere decir que, al correr, si recibe ciertas entradas, se conseguirá encriptar un mensaje; y si recibe otras, la maquinaria no se detendrá nunca.

Desde la perspectiva del programa Corriendo hasta la exclamación, la entrada es el contenido de la memoria grande al comenzar una ejecución y la salida (en caso de existir) es lo que en esa memoria queda escrito al finalizar.

Desde la óptica de la máquina criptográfica, además del mensaje escrito en la memoria grande, el programa que tiene que emular también forma parte de la entrada. La salida, que existirá solo en el caso de que una corrida concluya, será el contenido de la memoria grande en el momento en que la máquina se detenga. Esta computadora puede, potencialmente, transformar cualquier mensaje de diez símbolos en cualquier otro de igual longitud. Es solo cuestión de construir el programa adecuado.

---

# LA MÁQUINA CRIPTOGRÁFICA

¿Querés saber cómo funciona por dentro una computadora? ¡Lo vas a ver con tus propios ojos! Para arrancar, andá mirando el manual de la máquina criptográfica.

## HARDWARE DE LA MÁQUINA CRIPTOGRÁFICA

La máquina criptográfica tiene dos memorias: una rápida y una grande. La **memoria rápida** consiste en un único casillero. Allí se puede escribir de a un símbolo por vez. El alfabeto disponible incluye las letras del abecedario, los dígitos del 0 al 9, el espacio y los signos de exclamación e interrogación, tanto de apertura como de cierre.

C

A, B, C, D, E, F, G, H, I, J, K, L, M, N, Ñ, O, P,  
Q, R, S, T, U, V, W, X, Y, Z, 0, 1, 2, 3, 4, 5, 6,  
7, 8, 9, , !, ., ¿, ?

Memoria rápida Alfabeto

La **memoria grande** consiste en una tira de 10 celdas. En cada una se pueden guardar los mismos símbolos que en el casillero de la memoria rápida. Además, el componente tiene un cabezal que siempre se encuentra posicionado sobre una celda. Se puede leer el carácter allí almacenado o escribir uno nuevo.

¿	C	H	<b>O</b>	R	I	Z	O	S	?
---	---	---	----------	---	---	---	---	---	---

Memoria grande

En esta máquina, un programa es una lista numerada de instrucciones que se escriben una debajo de la otra (las instrucciones de la máquina criptográfica se encuentran más adelante). La computadora cuenta con un **puntero de instrucción**. Se trata de una flecha que siempre apunta a la instrucción que la máquina tiene que ejecutar.

→

1. LEE
2. SIG
3. ESC
4. AVA

Puntero de instrucción

Por último, tenemos la **unidad central de procesamiento** (CPU), que dirige todo el funcionamiento. Es el componente que se encarga de ejecutar una a una las instrucciones de un programa. Para hacerlo, repite lo que se conoce como **ciclo de instrucción**: (i) consulta al puntero de instrucción cuál es la instrucción del programa que tiene que ejecutarse; (ii) da indicaciones al resto de los componentes para llevarla a cabo; y (iii) le indica al puntero de instrucción a dónde debe apuntar, de forma tal de que quede señalando la próxima instrucción del programa que tiene que ejecutarse. Estos ciclos se interrumpen solo cuando se ha completado una corrida del programa.





## EL CONJUNTO DE INSTRUCCIONES

Las instrucciones que pueden formar parte de los programas de la máquina criptográfica se describen a continuación.

### AVA

Mueve el cabezal de la memoria grande una posición hacia la derecha. Si el cabezal se encuentra en la última celda, vuelve a posicionarse sobre la primera.

### ESC

Escribe en la memoria rápida el contenido de la celda de la memoria grande apuntada por el cabezal.

### LEE

Lee el contenido de la memoria rápida y lo escribe en la celda de la memoria grande apuntada por el cabezal.

### SIG

Reemplaza el símbolo de la memoria rápida por el siguiente, de acuerdo con el orden del alfabeto que usa la máquina. Al aplicarlo sobre el símbolo ?, el contenido pasa a ser A.

### ANT

Reemplaza el símbolo de la memoria rápida por el anterior, de acuerdo con el orden del alfabeto que usa la máquina. Al aplicarlo sobre el símbolo A, el contenido pasa a ser ?.

### SAL [nro de línea]

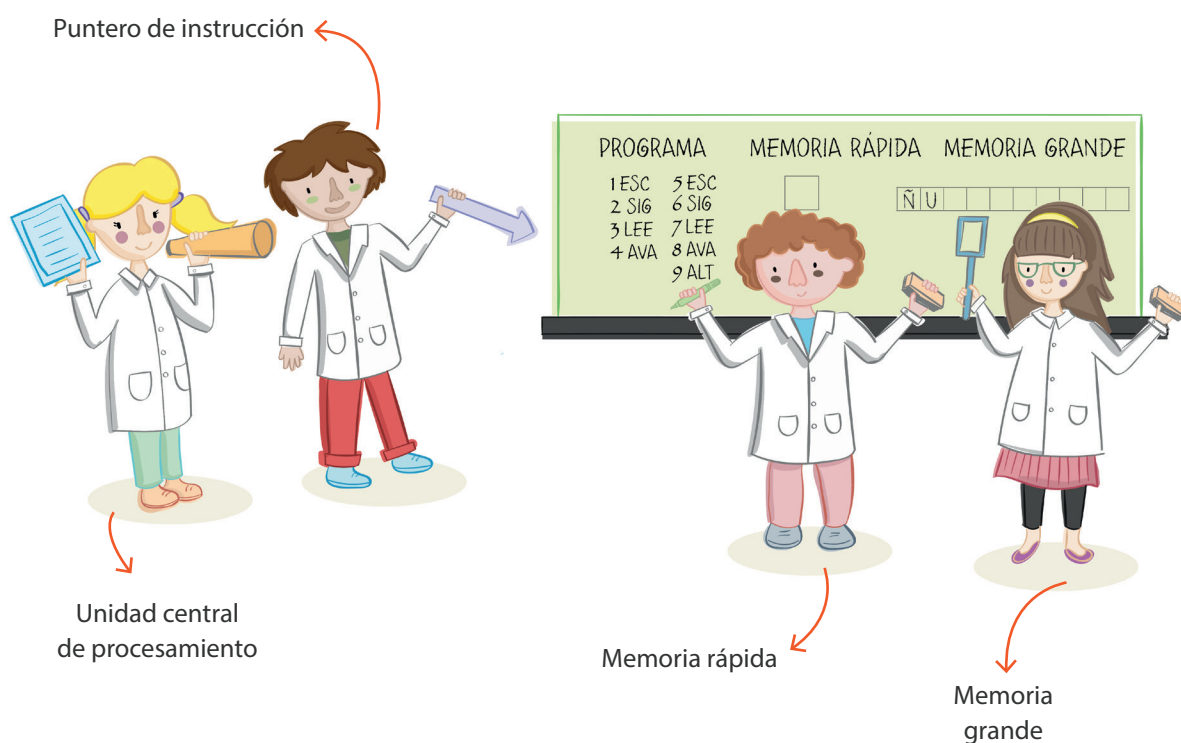
Hace apuntar al puntero de instrucción hacia la línea **nro de línea**, que es un parámetro de la instrucción.

### SI MR = [valor] SAL [nro de línea]

Si el contenido de la memoria rápida es **valor**, hace apuntar al puntero de instrucción a la línea **nro de línea**.

### ALT

Termina la ejecución del programa.



NOMBRE Y APELLIDO:

CURSO:

FECHA:

Cuatro compañeros van a simular la ejecución de distintos programas sobre esta máquina. ¡Asegurate de ser alguno de ellos en alguna oportunidad!

1. Te presentamos el programa Encripto 2.

1 ESC	5 ESC
2 SIG	6 SIG
3 LEE	7 LEE
4 AVA	8 AVA
	9 ALT

Ñ	U								
---	---	--	--	--	--	--	--	--	--

¿Qué ocurre cuando corren el programa Encripto 2 si al empezar la memoria grande contiene el mensaje *ÑU*?

2. Mirá el programa Encripto 4 y observá qué sucede si al comenzar la ejecución la memoria grande tiene impresa la palabra *PELO*.

1 ESC	5 ESC	9 ESC	13 ESC
2 SIG	6 SIG	10 SIG	14 SIG
3 LEE	7 LEE	11 LEE	15 LEE
4 AVA	8 AVA	12 AVA	16 AVA
			17 ALT

P	E	L	O						
---	---	---	---	--	--	--	--	--	--

¿Qué quedó?

3. Observá atentamente el programa Corriendo hasta la exclamación y fijate qué sucede si al comenzar a ejecutarlo en la memoria grande dice *¡TERMINAR!*

1 ESC	3 SIG	5 AVA
2 SI MR = ! SAL 7	4 LEE	6 SAL 1
		7 ALT

¡	T	E	R	M	I	N	A	R	!
---	---	---	---	---	---	---	---	---	---

¿Qué texto quedó impreso? ¿Qué pasa si ejecutamos Corriendo hasta la exclamación sobre un mensaje que no contenga el símbolo *!*?

#### CONDICIONALES Y CICLOS

Si prestás atención, vas a ver que las instrucciones que provocan saltos permiten incorporar condicionales y ciclos a los programas de la máquina criptográfica.