

# Actividades desenchufadas

Al final de este módulo vamos a:

Ejercitar la resolución de problemas complejos a través del pensamiento computacional

Analizar las diferentes destrezas que componen el pensamiento computacional

Diseñar retos complejos para que nuestro alumnado active y desarrolle las destrezas necesarias

Conocer actividades de aula para trabajar con el alumnado la resolución de problemas mediante el pensamiento computacional

- Cómo hacerlo en el aula
- Ejemplos de actividades y propuestas para el aula
- Actividades de conceptos de computación
- Juegos de mesa

# Cómo hacerlo en el aula

## Introducción

El Pensamiento Computacional consiste en **ser capaz de pensar de forma lógica, algorítmica, recursiva y abstracta**.

La teoría está clara, pero, **¿cómo podemos llevarlo a la práctica real en el aula?**

Lo haremos mediante diversos **recursos**, algunos los podrás descargar directamente. Pero no sólo queremos darte peces, sino que aprendas a pescar, así que te enseñaremos **cómo generar tus propios recursos**.

Nos hemos basado en las tarjetas de la bianual "Desafío Bebras", adaptándolas y re-diseñándolas.

Los retos incluidos en las tarjetas están centrados en aplicar y desarrollar habilidades de pensamiento computacional para resolver problemas y responder preguntas. Están diseñadas para **ser utilizadas sin recursos adicionales** y pueden ser trabajadas por estudiantes **individuales** o en pequeños **grupos**.

Cada tarjeta de retos incluye una **sugerencia de respuesta al reto** y se indica qué **destreza** de pensamiento computacional es la que más se usa para resolver el problema. No obstante, se utilizan **múltiples habilidades** de pensamiento computacional en todas las preguntas. La habilidad **destacada** es solo una solución, utilizada para generar discusión sobre las habilidades de pensamiento computacional.

Es muy, muy importante que tú también estés entrenado/a en estas destrezas, con lo que recomendamos que sigas los mismos pasos que el alumnado: primero resolviendo retos en grupo y posteriormente de forma individual.

En la mayoría de los casos, **no hemos sido formados** en estas destrezas, destinadas a la resolución de retos complejos, con lo que es muy importante que nos formemos en las mismas, que nos entrenemos.

# Objetivo

El objetivo fundamental de estas actividades es que las alumnas y alumnos **“activen”** estas destrezas. Para ello es necesario que **practiquen**, se entrenen en el uso de las mismas para que cuando se enfrenten a problemas complejos, de la índole que sea, puedan resolverlos.

El desarrollo de estas destrezas **no es exclusivo** para la resolución de retos informáticos o de programación. Estas destrezas son necesarias para la resolución de problemas complejos en general.

# Guía de uso

Como ya hemos mencionado en la introducción, estas tarjetas de retos se pueden usar de manera **individual** o en pequeños **grupos**. No obstante, creemos que es **importante**, ya sea con las alumnas y alumnos más jóvenes, o sea en las primeras sesiones, empezar trabajando en grupo.

El **tamaño de los grupos** recomendado es de **dos** personas para infantil y primera etapa de Primaria, y de **tres** personas para más mayores.

## Forma de proceder recomendada:

- Entrega a cada grupo los **recursos** necesarios para poder trabajar en equipo: papeles, pizarras, bolis, lápices, pinturas...
- Explica a cada grupo **cómo deben trabajar** en grupo, cómo deben comunicarse mediante preguntas abiertas destinadas a entender lo que nuestra compañera o compañero nos quiere decir y no a cuestionarle, es decir, practicando la escucha activa.
- Entrega una **tarjeta** de retos a cada grupo o persona.
- Indica a todos los grupos que el **tiempo** para la resolución del reto es limitado.
- Enfatiza que **no es una carrera** por resolver el reto, que no importa qué grupo termina primero.
- Permite que cada grupo **analice** el reto y lo entienda.
- Pide al alumnado que apunten en un papel la solución a su reto y que **justifiquen** la misma. Indicando, si es posible, qué destreza o destrezas consideran que ha sido fundamental para la resolución del mismo.

## Puesta en común:



- Apunta en la **pizarra** la solución de los distintos grupos y la destreza que han considerado más relevante para la solución del reto propuesto
- Solicita a los distintos grupos que **expliquen el proceso** que han seguido para llegar a la solución del reto

Es importante tomar notas de cómo ha resultado el reto cada grupo y qué tipo de destrezas están menos desarrolladas. Estas destrezas menos desarrolladas deberemos trabajarlas más.

Se recomienda realizar este tipo de retos con cierta **frecuencia** para conseguir que el alumnado asimile esta forma de trabajar y de afrontar los retos complejos. Como mencionábamos al principio, **es importante el “entrenamiento”**.

## Destrezas, habilidades computacionales

En este apartado nos vamos a centrar en **6 destrezas principales**. Algunos autores consideran más destrezas y pasos a dar en la resolución de un reto complejo. O incluso algunas de las destrezas que estamos considerando, las dividen en otras más simples.

Las 6 destrezas que considera “Bebras” son:

# DESTREZAS

Extraído de <https://www.bebbras.uk>

## DESCOMPOSICIÓN

Descomponer los datos, el proceso o el programa en partes más pequeñas y manejables.



## RECONOCIMIENTO DE PATRONES

Observar patrones, tendencias y elementos regulares que den sentidos a los datos.



## ABSTRACCIÓN

El proceso de identificar y extraer la información relevante e ignorar o despreciar la información innecesaria.



## ALGORITMOS

Crear una serie ordenada de instrucciones para resolver problemas similares, o para hacer una tarea.



## MODELADO Y SIMULACIÓN

Desarrollo de un modelo para imitar procesos y problemas.



## EVALUACIÓN

Determinar la eficacia de una solución, generalizando y aplicando esa información a nuevos problemas.

# Ejemplos de actividades y propuestas para el aula

## Tarjetas Bebras

Hemos adaptado, traducido y re-diseñado una serie de fichas de Bebras que puedes **descargar**. En cada nivel encontrarás una ficha para cada destreza. Las **edades** asociadas a cada nivel son **orientativas**. Te recomendamos que comiences siempre con los retos correspondientes a un **nivel inferior** a la edad de tu alumnado. Es importante que adquieran confianza, que se sientan capaces de resolver los retos y que aprendan a fijarse en los detalles.

**Descargables** por niveles y edades (orientativo):

**Nivel 0: 3-4 años**

**Nivel 1: 5-6 años**

**Nivel 2: 7-8 años**

**Nivel 3: 9-10 años**

**Nivel 4: 11-12 años**

*Esta información está basada en Bebras Unplugged, de la “Australia’s National Science Agency”.*

## Diseña tu propia actividad

Como hemos visto, hay **muchas actividades** para trabajar las destrezas del pensamiento computacional. No obstante, siempre es interesante poder **crear** o **adaptar** alguna a lo que **necesita tu aula**.

Aquí puedes **descargarte** una actividad para practicar reconocimiento de patrones **contextualizada**:

### **Mi propia tarjeta (formato pdf)**

### **Mi propia tarjeta (plantilla de Canva)**

Hay multitud de **herramientas** que puedes utilizar para diseñar tus propias actividades. Nosotras hemos utilizado **CANVA**.

Con ella podrás diseñar **cualquier actividad** en un formato similar al de las tarjetas del punto anterior, o en cualquier otro formato.

Si no estás familiarizada/o con CANVA mira el siguiente video-tutorial donde encontrarás la información básica para trabajar con esta herramienta y un ejemplo de cómo crear tu propia actividad.

[https://www.youtube.com/embed/v6pl8Mz\\_Jo8](https://www.youtube.com/embed/v6pl8Mz_Jo8)

# Actividades de conceptos de computación

## Introducción

Esta información está basada en [csunplugged](#).

*“La Ciencia de la Computación no tiene que ver con las computadoras más que la Astronomía con los telescopios”. Edsger Dijkstra*

Tal y como hemos visto, el pensamiento computacional trata sobre **cómo poder enfrentarnos a un reto complejo**. Es una manera de gestionar la información y una serie de habilidades humanas lo que ponemos en marcha a la hora de hacer funcionar un ordenador. **No trata sobre cómo piensa un ordenador**. El ordenador no piensa.

La **ciencia de la computación** es la ciencia del **procesamiento de la información**. El ordenador es una herramienta, como los telescopios lo son para la astronomía.

Ahora haremos una serie de **actividades para llevar al aula**, sin ordenador, para entender el procesamiento de la información y ver cómo funciona esta herramienta.

[https://www.youtube.com/embed/ZJ2R\\_IWFuDI](https://www.youtube.com/embed/ZJ2R_IWFuDI)

Al llevar a cabo las actividades en el aula es fundamental **dejar tiempo** para que el alumnado sea capaz de generar alguna respuesta. Recordemos que estamos en la fase de **instrucción**.

### Pautas recomendadas:

- No dar respuestas preparadas
- Jugar con preguntas
- Darles la oportunidad de encontrar soluciones por sí mismos/as



# Información adicional a las actividades

## Código binario

Imagina que quieres **comunicarte** con alguien, pero no puedes hablar ni escribir en letras y palabras como lo estamos haciendo ahora. En su lugar, puedes usar un lenguaje especial llamado **código binario**. Este lenguaje está compuesto únicamente por **dos números**: el 0 y el 1.

El código binario se basa en el **sistema de numeración posicional**, al igual que el sistema decimal que usamos normalmente, pero en lugar de tener 10 dígitos (0 al 9), el código binario solo tiene dos dígitos: 0 y 1. Estos dígitos se llaman "bits".

Cada número o carácter que queremos representar se traduce a una **secuencia de ceros y unos**. Por ejemplo, el número 5 en binario se representa como 101. Esto significa que tenemos un 1 en el lugar de las unidades, un 0 en el lugar de las decenas y un 1 en el lugar de las centenas.



## Código Binario

Un bit puede tener el valor

0

o

1

Según su posición cada bit vale el doble que el anterior

Valor

128

64

32

16

8

4

2

1

0

0

0

0

0

0

0

1

= 1

$$\begin{matrix} 128 & 64 & 32 & 16 & 8 & 4 & 2 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{matrix} = 0$$

$$\begin{matrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{matrix} = 3$$

$$\begin{matrix} 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{matrix} = 4$$

$$\begin{matrix} 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \end{matrix} = 25$$

$$\begin{matrix} 128 & 64 & 32 & 16 & 8 & 4 & 2 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \end{matrix} = 38$$

$$\begin{matrix} 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \end{matrix} = 73$$

$$\begin{matrix} 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \end{matrix} = 145$$

$$\begin{matrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{matrix} = 255$$

De esta manera, usando **combinaciones de ceros y unos**, podemos representar cualquier número, que queramos en código binario.

El código binario es **muy importante en informática** porque los ordenadores entienden y procesan la información en forma de ceros y unos, la corriente eléctrica pasa o no pasa. Todos los programas y datos en un ordenador se almacenan y procesan utilizando el código binario.

### Contenidos relacionados:

#### Matemáticas-números:

- Explorando números en otras bases
- Representando números en base dos

#### Matemáticas-álgebra:



- Siguiendo un patrón secuencial y describiendo una regla para ese patrón
- Patrones y relaciones en potencia de dos

**Destrezas:**

- Contar
- Comparar
- Secuenciar

**Edades:**

A partir de 7 años

**Materiales:**

Cartulinas, rotuladores

## Codificar texto

Hemos entendido cómo representar cualquier número, en esta ocasión aprenderemos **cómo se almacenan y transmiten los textos**.

Para esta tarea representaremos **cada letra con un código binario** y a partir de ahí podremos codificar un texto entero. Necesitamos un acuerdo previo, una **equivalencia común de número y letra**.

Como actividad para clase no usaremos la correspondencia estándar (ASCII), sino una simplificada.



A	0	Ñ	14
B	1	O	15
C	2	P	16
D	3	Q	17
E	4	R	18
F	5	S	19
G	6	T	20
H	7	U	21
I	8	V	22
J	9	W	23
K	10	X	24
L	11	Y	25
M	12	Z	26
N	13		

### Contenidos relacionados:

#### Matemáticas-números:

- Explorando números en otras bases
- Representando números en base dos

#### Matemáticas-álgebra:

- Siguiendo un patrón secuencial y describiendo una regla para ese patrón
- Patrones y relaciones en potencia de dos

### Destrezas:

- Contar
- Comparar
- Secuenciar

### Edades:

A partir de 7 años

### Materiales:

- Papel, cartulinas, rotuladores

## Codificar y comprimir imágenes

Además de números y letras, las **imágenes** son otra información importante que procesan los dispositivos tecnológicos. En esta actividad veremos cómo almacenar y transmitir imágenes de una manera eficiente.



Para el trabajo en el aula nos limitaremos a la codificación en **blanco y negro** ya que es fácilmente representable.

Las imágenes en blanco y negro están compuestas por una serie de puntos llamados **píxeles**. Cada píxel puede ser de dos colores: **blanco o negro**. La codificación en código binario se basa en representar cada píxel como un bit, que puede ser 0 o 1.

Imaginemos que tenemos una imagen en blanco y negro de **8 píxeles de ancho por 8 píxeles de alto**. Esto significa que hay un total de 64 píxeles en la imagen. Para codificarla, necesitaríamos 64 bits (unos y ceros).

Por ejemplo, si tenemos un píxel blanco, lo representamos con el número 0 en binario, y si tenemos un píxel negro, lo representamos con el número 1 en binario.

Así, si tuviéramos una imagen en blanco y negro completamente blanca, la codificación binaria sería una secuencia de 64 ceros (00000000...0000).

Si tuviéramos una imagen en blanco y negro completamente negra, la codificación binaria sería una secuencia de 64 unos (11111111...1111).

Solo hace falta **lápiz** y **papel** para codificar una imagen.

El código resultante puede ser muy largo y **repetitivo**. Gracias a ello tenemos la oportunidad de introducir la **compresión de imágenes**, y de entender que significa.

### Contenidos relacionados:

#### Matemáticas:

- Formas y ubicación

#### Tecnología:

- Usar números enteros para representar otros tipos de datos
- Reducir el espacio que se usa para datos repetitivos

### Destrezas:

- Contar
- Graficar

### Edades:

A partir de 7 años

### Materiales:

- Papel cuadriculado, rotuladores

# Control del error. Método de verificación

Cuando los datos se almacenan en un disco o se transmiten de un ordenador a otro, generalmente asumimos que los datos no cambian en el proceso. Pero, **a veces, las cosas salen mal** y los

datos se modifican accidentalmente.

Imagina que estás enviando un **mensaje** muy importante a una amiga, pero durante el proceso de envío, algunos bits cambian. Esto puede hacer que el **mensaje llegue incorrecto o incompleto**. Para evitar eso, se utiliza un método llamado **control de error o verificación**.

El control de error con código binario funciona **agregando** algunos bits extra al mensaje que se va a enviar. Estos bits extra se llaman **bits de paridad**. La paridad se refiere a si algo es par o impar.

En esta actividad te presentamos un ejemplo sencillo **para llevar al aula** de cómo funciona el control de error con código binario utilizando bits de paridad adecuada. Hay otros métodos más complejos y avanzados para detectar y corregir errores en la transmisión de datos, pero la idea principal es **utilizar bits extra** para verificar si los datos llegaron correctamente.

De esta manera, el control de error con código binario nos ayuda a asegurarnos de que los mensajes que enviamos lleguen sin errores y podamos **confiar** en la información recibida.

### **Contenidos relacionados:**

#### *Matemáticas-números:*

- Exploración de cómputo y estimación.

#### *Matemáticas-Algebra:*

- Exploración de patrones y relaciones, buscando un valor que falta

#### *Matemáticas:*

- Filas y columnas, coordenadas

#### *Tecnología:*

- Validar datos

### **Destrezas:**

- Contar
- Reconocer números pares e impares

### **Edades:**

- A partir de 7 años

**Materiales:**

- Un conjunto de 36 cartas con imanes y coloreadas de un solo lado
- Una pizarra metálica (un pizarrón blanco funciona muy bien) para la demostración

## Red de clasificación

A pesar de que los ordenadores son rápidos existe un **límite**, en cuanto a la rapidez, con la que pueden resolver problemas.

Una manera de **acelerar** el proceso consiste en utilizar diversos ordenadores para resolver diferentes **partes** de un problema.

En esta actividad vamos a utilizar una metodología llamada **Redes de Ordenamiento**, que realizan diversas **comparaciones** y **ordenamientos** al mismo tiempo con el objetivo de agilizar la tarea.

Imagina que tienes una caja llena de juguetes diferentes, y quieres ordenarlos por su tamaño. Una red de clasificación es como un **sistema** que te ayuda a hacerlo.

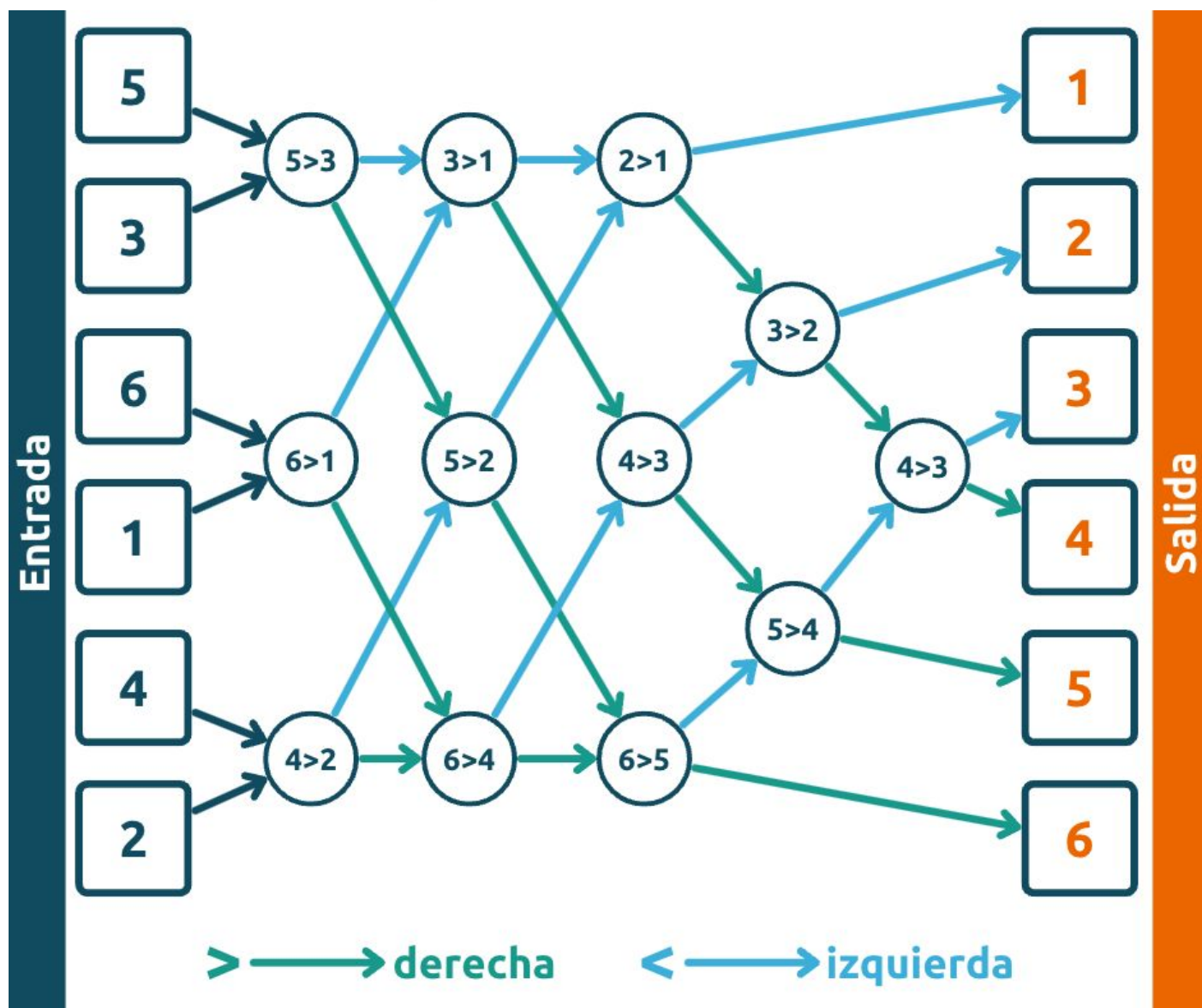
En una red de clasificación, tienes varios "nodos" o "cajas" que están conectados entre sí. Cada nodo realiza una tarea, por ejemplo, **comparar** el tamaño de dos juguetes y **decidir** cuál es más grande. Estos nodos trabajan juntos para ordenar los juguetes.

El proceso funciona de la siguiente manera: colocas los juguetes en la red y comienzan a pasar de un nodo a otro. Cada nodo **compara** el tamaño de los juguetes que recibe y los **envía** al nodo adecuado en función de su tamaño. Por ejemplo, si tienes un nodo que compara juguetes y decide cuál es más grande, enviará el juguete más grande al nodo correspondiente para los juguetes grandes y el juguete más pequeño al nodo para los juguetes pequeños.

A medida que los juguetes pasan de nodo en nodo, se van **clasificando** y **ordenando** según su tamaño. Al final del proceso, los juguetes habrán sido colocados en diferentes "cajas" según su tamaño, **desde el más grande hasta el más pequeño**.

Una red de clasificación se puede utilizar para ordenar diferentes cosas, no solo juguetes. Por ejemplo, se puede usar para ordenar números de menor a mayor o para clasificar objetos en categorías diferentes, como clasificar frutas según su color.





Una red de clasificación puede ser mucho más **compleja** en la realidad, con muchos más nodos y funciones específicas. Pero la idea básica es que los nodos trabajan juntos para comparar y ordenar los elementos.

### Contenidos relacionados:

#### Matemáticas-números:

- Explorando números: Mayor que y menor que

#### Matemáticas-álgebra:

- Patrones y secuencias

**Destrezas:**

- Comparar números y trabajar con rangos
- Deducir
- Desarrollo de Algoritmos
- Cooperación en la solución de problemas

**Edades:**

A partir de 10 años

**Materiales:**

- Esta es una actividad grupal y se lleva a cabo al aire libre.
- Tiza
- Papel, rotuladores
- Reloj

**Para saber mas:** UC Computer Science Education:

<https://www.youtube.com/embed/VpDDPWVn5-Q>

# Juegos de mesa

Los juegos de mesa son siempre **un buen recurso** para practicar habilidades y destrezas como la **lógica**, la **estrategia** o la **resolución de problemas**. Muchas de estas destrezas están directamente **relacionadas** con el **pensamiento computacional**. Encontrarás en el mercado una **amplia oferta** de juegos que puedes llevar al aula.

En este apartado vamos a **destacar dos de ellos** creados para entender cómo funciona el pensamiento computacional sin necesidad de “estar conectado”.

Los dos han sido creados por el profesor en el **departamento de Telecomunicaciones de la Facultad de Ingeniería de la Universidad de Deusto** y miembro del **Deusto Learning Lab**, Pablo Garaizar.

## Moon



Fotos: Juego Moon. [Tibot](#)



+10 AÑOS



1-4



15-45 MIN

“MOON es un juego para 1 a 4 jugadores donde controlamos la computadora del **módulo lunar Eagle** durante el primer alunizaje. Solo 3 minutos antes de alunizar, **¡la computadora de la nave falló!** Sin embargo, un buen diseño desarrollado por los ingenieros de software del equipo liderado por **Margaret Hamilton** logró evitar un fallo fatal. En este juego deberemos **emular** esas inteligentes rutinas de software para ayudar a los astronautas a alunizar de forma segura.”

“Jugando a este juego aprenderás a **contar en binario**, realizar **operaciones lógicas** y **matemáticas** y **reparar** fallos técnicos que ocurrirán durante la misión.”

Este juego dispone de versión descargable **imprimible**, versión **online** y versión **app**.

Edad recomendada: a partir de 10 años

## Arqueras de Nand



Fotos: Juego Arqueras de Nand. Tibot

+10 AÑOS



1-2



20-30 MIN



“El Valle de Nand está siendo **invadido** por hordas salvajes de orcos. Mientras los guerreros locales pierden la batalla **cuerpo a cuerpo**, las **arqueras** de Nand aprovechan la altura de sus montañas para diezmar las tropas enemigas. En este juego, tendrás que decidir **cómo proteger el valle** gracias a la acción combinada de los guerreros y las arqueras de Nand antes de que los orcos lo destruyan todo.”

“ARQUERAS DE NAND es un juego con el que aprenderás los fundamentos básicos de las **bases de datos** (el Valle de Nand es una pequeña tabla de datos donde seleccionaremos parcialmente sus regiones para atacar a los orcos).”

Arqueras de Nand está diseñado para aprender **fundamentos básicos de bases de datos**.

Este juego dispone de versión descargable **imprimible**.

## Robot Turtles



Foto: Robot Turtles de Aprendiendomatematicas

**+ 4 AÑOS**



**2-5**



**5-10 MIN/**



Es un juego diseñado para introducir a niños y niñas en los **conceptos básicos de la programación**. Tienes que utilizar **tarjetas de movimiento** para guiar a las tortugas robóticas hacia los objetivos en el tablero. A medida que avanzas en los niveles, se agregan más reglas y desafíos, fomentando el **pensamiento lógico y algorítmico**.

## Bits & Bytes



Foto: Bits & Bytes de [Artijoc](#)

**+ 4 AÑOS**



**2-4**



**VARIABLE**

Este juego fomenta el **pensamiento estratégico, el razonamiento lógico y la resolución de problemas**, al tiempo que promueve el **trabajo en equipo** y la **colaboración**. Al jugar debes tomar **decisiones** inteligentes y **planificar** cuidadosamente tus movimientos para lograr el objetivo común.

El juego consiste en **construir circuitos electrónicos** utilizando tarjetas de componentes como **puertas lógicas, cables y microchips**. Quien juega se convierte en un/a ingeniero/a de software en busca de la combinación perfecta para completar los desafíos propuestos. Los desafíos van desde la creación de circuitos simples hasta los más complejos, lo que permite avanzar **gradualmente** en el juego y poner a prueba las habilidades de **resolución de problemas**. Además, el juego incluye tarjetas de desafío adicionales que permiten la creación de circuitos personalizados, agregando un elemento de creatividad y experimentación.



# Rush Hour



Foto: Rush Hour de [Aprendiendomatematicas](#)

**+6 AÑOS**



**1**



**VARIABLE**

En este juego, debes mover los coches en el tablero para **despejar** el camino y sacar el coche rojo del atasco. Te ayuda a desarrollar habilidades de **planificación, resolución de problemas y pensamiento estratégico**.

# Set

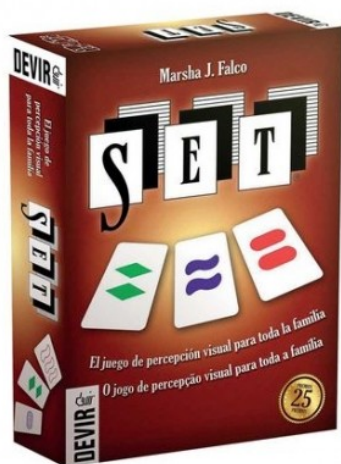


Foto: Set de [Aprendiendomatematicas](#)

**+6 AÑOS**



**1-10**



**30 MIN**

Como ya te comentamos en el módulo anterior debes observar **patrones** y **relaciones** entre las cartas para identificar los conjuntos. Te ayuda a desarrollar habilidades de **reconocimiento de patrones y razonamiento lógico**.

## Blokus





Foto: Blokus de [Zacatrus](#)

+6 AÑOS



2-4



30 MIN

En este juego, cada persona tiene un **conjunto de piezas** de diferentes formas y colores. El **objetivo** es colocar tus piezas **estratégicamente** en el tablero, bloqueando a tus oponentes mientras intentas ocupar la mayor cantidad de espacio posible. Te ayuda a desarrollar habilidades de **planificación, pensamiento estratégico y resolución de problemas**.

## Qwirkle





Foto: Qwirkle de [Zacatrus](#)

**+6 AÑOS**



**2-4**



**45 MIN**

Es un juego en el que se deben **crear líneas de fichas** que comparten el mismo color o la misma forma. Debes encontrar combinaciones lógicas y tácticas para maximizar tus puntos. Te ayuda a desarrollar habilidades de **reconocimiento de patrones, pensamiento abstracto y toma de decisiones**.

## Mastermind

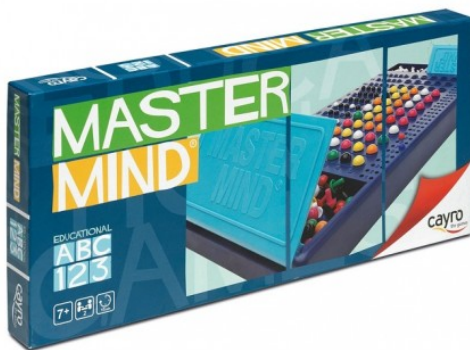


Foto: Mastermind de [Aprendiendomatematicas](#)

**+7 AÑOS**



**2**



**10 MIN**

Es un juego de **deducción** en el que uno de los/las jugadores/as crea un **código de colores oculto** y otra persona debe adivinarlo en base a **pistas limitadas**. Te ayuda a desarrollar habilidades de **razonamiento lógico, análisis de información y pensamiento analítico**.

## La Resistencia

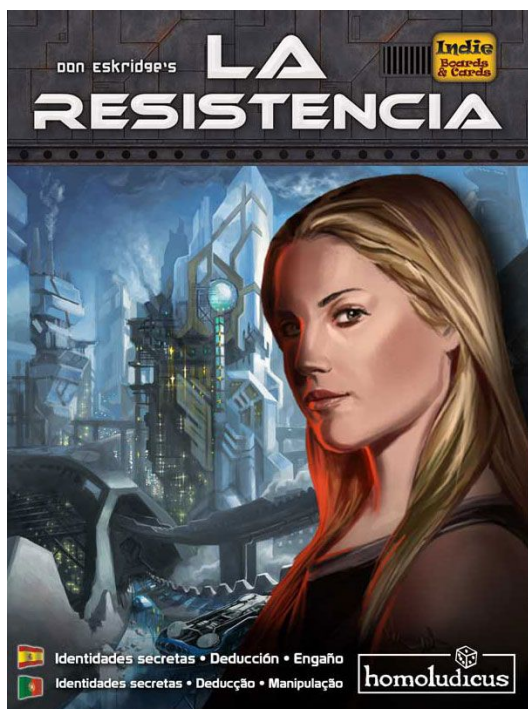


Foto: La Resistencia de Zacatrus

**+8 AÑOS**



**5-10**



**30 MIN**

Es un juego de **roles ocultos** en el que las personas jugadoras deben descubrir quiénes son los **espías** dentro del grupo de resistencia. Se requiere una **comunicación** cuidadosa y un **análisis** de comportamientos para determinar quién está diciendo la verdad y quién está engañando. Te ayuda a desarrollar habilidades de **razonamiento lógico, deducción y toma de decisiones** basadas en la información disponible.

## Code Master

Foto: Code Master de [Zacatrus](#)

+ 8 AÑOS



1



VARIABLE

En este juego, debes **programar** al personaje principal para que supere diferentes **desafíos**. Debes usar **cartas de programación** para planificar el camino correcto y superar obstáculos. Te ayuda a desarrollar habilidades de **planificación**, **secuenciación** y **resolución de problemas**.

## Ingenious

Foto: Ingenious de [masqueoca](#)



+10 AÑOS



1-4



30 MIN

En este juego, debes colocar **fichas de diferentes formas y colores** en un tablero **hexagonal**. El objetivo es acumular la mayor cantidad de puntos posibles al completar líneas y grupos de fichas del mismo color. Te ayuda a desarrollar habilidades de **planificación, reconocimiento de patrones y pensamiento estratégico**.

## Código Secreto



Foto: Código secreto de [Zacatrus](#)

+12 AÑOS



2-8



15 MIN

Es un juego de **palabras** en el que los jugadores deben comunicarse de forma estratégica para descubrir las palabras clave de su equipo. Una persona dará **pistas** relacionadas con varias palabras y las demás deberán **adivinar** cuáles son las palabras correctas. Te ayuda a desarrollar habilidades de **pensamiento lógico, comunicación y asociación de ideas**.