

# Actividad refuerzo

## AAR2.17 Completa

**(C.ESP2 / CE2.2, CE2.3, CE2.4 / IC2-10p)**

### Pensamiento computacional

1. ● Completa las frases con las palabras que te ofrecemos:

ordenador    soluciones    creativas    computacional    problemas

- a) En ocasiones, trabajar como lo haría un [ ] nos puede ayudar a resolver problemas de la vida diaria y desarrollar [ ] sencillas y [ ] para problemas habituales.
- b) Esta forma de trabajo se denomina pensamiento [ ] y consiste en abordar los [ ] planteados como si fuéramos científicos informáticos.

2. ● Ordena del 1 al 5 los siguientes pasos, que describen el pensamiento computacional:

1.  
Desarrollamos un modelo que pueda ser una solución.
2.  
Identificamos el aspecto esencial de un problema (datos, condicionantes y restricciones).
3.  
La solución obtenida debe poder ser ejecutada por un sistema informático.
4.  
Hacemos deducciones, planteamos hipótesis, imaginamos situaciones... Utilizamos el pensamiento abstracto.
5.  
Simplificamos los elementos de un problema y lo dividimos en otros más sencillos.

**3.** ● Une cada palabra con su definición.

Patrón

Revisar los pasos y procedimientos que forman un algoritmo para que no haya errores, instrucciones poco precisas o innecesarias ni pasos duplicados.

Información

Conjunto de datos que nos aportan un significado de interés para un problema planteado.

Depurar

Conjunto de características que se repiten de forma determinada en los objetos; puede tratarse de repeticiones de colores, simetrías, cambios de forma o de posición, etc.

**4.** ● El método para solucionar problemas comprende cuatro fases. Une cada una de ellas con su explicación.

Comprende el problema

La solución al problema planteado tiene que ser completa, esto es, debe dar respuesta a todas las incógnitas. Por otra parte, no basta con llegar a una solución del problema, es necesario confirmar que la solución es la adecuada. Para confirmarlo, intenta llegar a ella por otro camino o estimar con antelación si es buena.

Establece un plan de acción

Lee el enunciado del problema varias veces hasta conocer la incógnita, los datos y los condicionantes que tienes que respetar.

Desarrolla el plan. Obtén una solución

Sigue detalladamente el plan que has planteado. Simplifica siempre que sea posible y haz dibujos o diagramas para presentar las soluciones.

Revisa la solución obtenida

Descarta los datos innecesarios o irrelevantes y decide qué ecuaciones vas a utilizar. Si el problema es complicado, divídelo en partes más pequeñas fácilmente resolubles.

5. ● A lo largo de la unidad hemos aprendido que los datos son fundamentales en el pensamiento computacional. Indica a qué tipo de datos se refiere cada una de estas dos definiciones:

Alfanuméricos

Numéricos

- a) Son representados mediante cifras. Pueden ser enteros (que permiten contar elementos) o decimales (que permiten expresar partes no completas de cantidades).

- b) Están formados por todo tipo de caracteres: cifras, letras, símbolos, etc. Ejemplos: las matrículas de los coches, las direcciones de correo electrónico y el número del NIF.

### Algoritmos

6. ● ● Une cada palabra con su definición.

Secuencia ordenada de pasos que resuelven un problema en un tiempo finito

Algoritmo

Serie o sucesión de cosas que guardan entre sí cierta relación

Selección

Acción de repetir un proceso para alcanzar un objetivo o resultado

Iteración

Acción de elegir una o más cosas entre varias

Terminal

Gráfico que muestra la secuencia de las acciones que se han de realizar

Proceso

Símbolo que, en un diagrama de flujo, representa el comienzo o el fin del desarrollo de un algoritmo

Diagrama de flujo

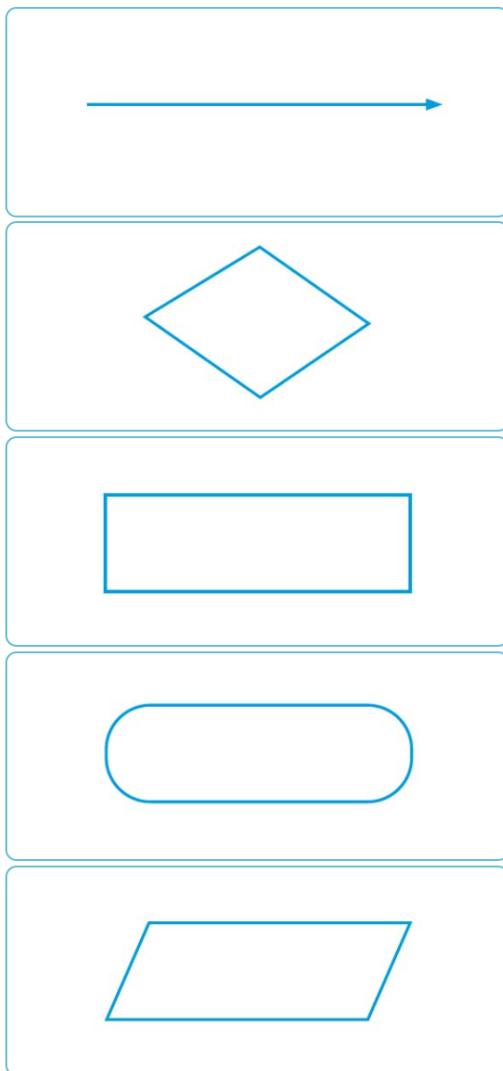
Símbolo que, en un diagrama de flujo, permite representar cada una de las acciones que hay que realizar para desarrollar un algoritmo

Secuencia

Símbolo que, en un diagrama de flujo, se utiliza cuando es necesario decidir entre dos o más opciones y señala el camino que habrá que seguir según cuál sea la opción elegida

Decisión

7. ● Une cada símbolo de los diagramas de flujo con la función que representa.



Decisión. Se utiliza cuando es necesario decidir entre dos o más opciones y señala el camino que habrá que seguir según cuál sea la opción elegida.

Proceso. Permite representar cada una de las acciones que hay que realizar para desarrollar el algoritmo.

Línea de flujo. Señala el orden en que se desarrollan las acciones en el algoritmo.

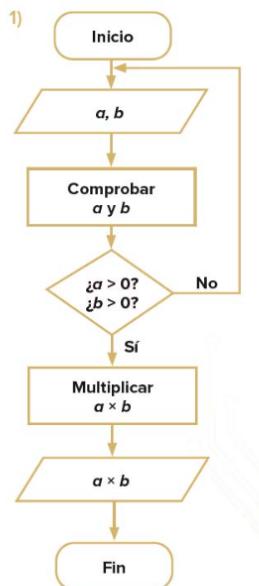
Terminal. Representa el comienzo o el fin del desarrollo de un algoritmo.

Entrada o salida de información. Se utiliza cuando es necesaria información (datos adicionales para desarrollar el algoritmo) o se presentan datos o resultados.

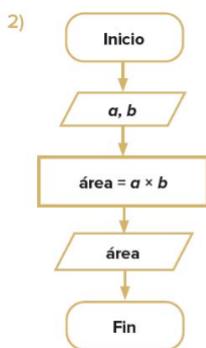
8. ● Di si las siguientes afirmaciones sobre tipos de algoritmos son verdaderas o falsas:

- a) En los algoritmos de estructura secuencial, las instrucciones se van cumpliendo en el orden en que aparecen.
- Verdadero       Falso
- b) En los algoritmos de estructura secuencial, las instrucciones se llevan a cabo sólo si se cumple una determinada condición.
- Verdadero       Falso
- c) Los algoritmos de estructura selectiva incluyen instrucciones que se repiten varias veces.
- Verdadero       Falso

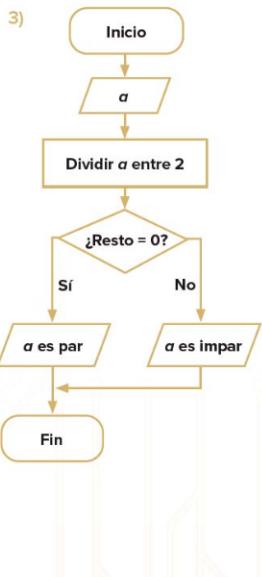
9. • • • ¿Para qué nos sirve cada uno de los siguientes diagramas de flujo?



- a) Calcular el área de un rectángulo
- b) Calcular el perímetro de un rectángulo
- c) Multiplicar dos números
- d) Multiplicar dos números, siempre que ambos sean mayores que 0
- e) Saber si un número es par o impar
- f) Saber si un número es divisible entre dos o entre tres

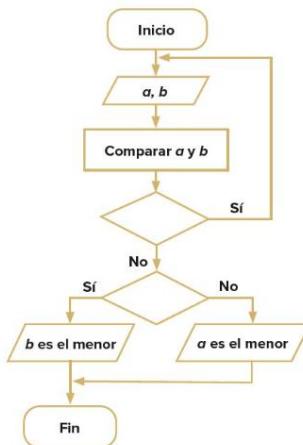


- 
- a) Calcular el área de un rectángulo
  - b) Calcular el perímetro de un rectángulo
  - c) Multiplicar dos números
  - d) Multiplicar dos números, siempre que ambos sean mayores que 0
  - e) Saber si un número es par o impar
  - f) Saber si un número es divisible entre dos o entre tres



- a) Calcular el área de un rectángulo
- b) Calcular el perímetro de un rectángulo
- c) Multiplicar dos números
- d) Multiplicar dos números, siempre que ambos sean mayores que 0
- e) Saber si un número es par o impar
- f) Saber si un número es divisible entre dos o entre tres

10. ● ● El siguiente diagrama de flujo incompleto sirve para comparar dos valores distintos y saber cuál es el menor de ellos. ¿Cuál es la pregunta en cada una de las dos decisiones que faltan?



a) La primera es  $a > b$ ? y la segunda es  $a < b$ ?

b) La primera es  $a = b$ ? y la segunda es  $a > b$ ?

c) Las dos preguntas son  $a > b$ ?

d) La primera es  $a > b$ ? y la segunda es  $a = b$ ?

11. • • Teniendo en cuenta las normas sobre cómo hacer diagramas de flujo, indica para cada imagen si es correcta o incorrecta.

a)



Correcta

Incorrecta

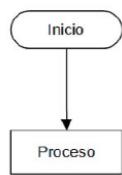
b)



Correcta

Incorrecta

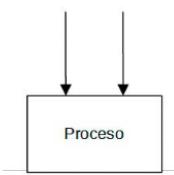
c)



Correcta

Incorrecta

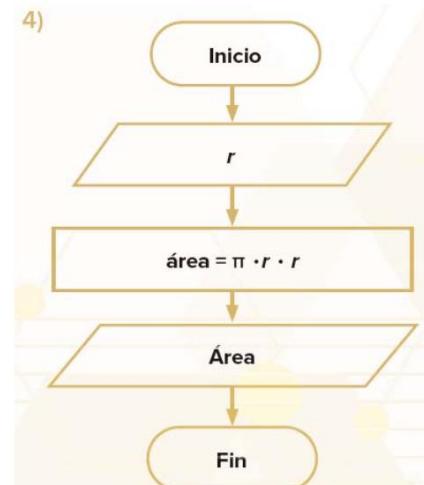
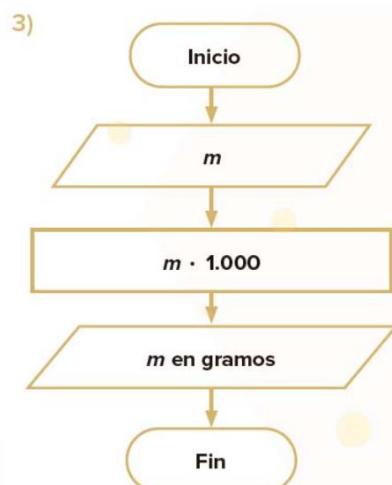
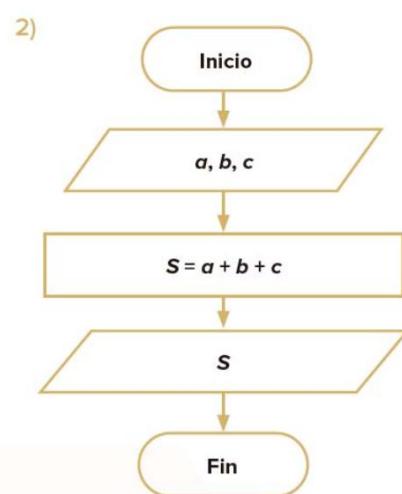
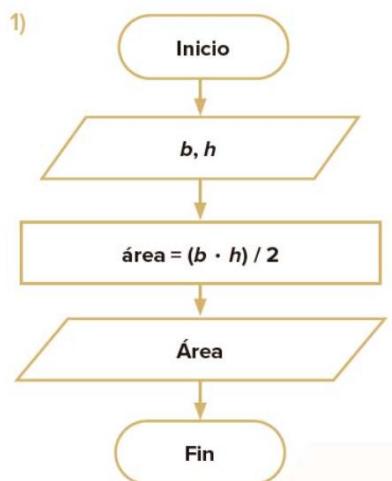
d)



Correcta

Incorrecta

12. • • ¿Para qué nos sirve cada uno de los siguientes diagramas de flujo?



## Programación

13. ● ● Selecciona la respuesta correcta en cada caso:

1. Son instrucciones de la categoría **Control** en Scratch:

- a) detener todos y operaciones
- b) repetir y apariencia
- c) por siempre y movimiento
- d) por siempre y esperar hasta que

2. Son instrucciones de la categoría **Movimiento** en Scratch:

- a) girar 15 grados
- b) mover 10 pasos
- c) apuntar en dirección 90
- d) Todas las respuestas anteriores son correctas.

3. En la categoría **Apariencia** encontramos bloques que nos permiten...

- a) Cambiar de un disfraz a otro.
- b) Apuntar en dirección 90.
- c) Crear variables.
- d) Ninguna de las respuestas anteriores es correcta.

4. En la categoría **Operadores** encontramos bloques que nos permiten...

- a) Realizar operaciones aritméticas.
- b) Unir textos.
- c) Realizar operaciones lógicas.
- d) Todas las respuestas anteriores son correctas.

14. De los siguientes programas, ¿cuál simulará mejor que un personaje ande?

a)



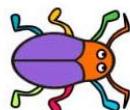
b)



c)



15. ● ● Queremos que el escarabajo se mueva describiendo un triángulo. ¿Cuál es el bloque de instrucciones correcto?



a)

```
al hacer clic en  
ir a x: 0 y: 0
deslizar en 1 segs a x: 100 y: 0
deslizar en 1 segs a x: 100 y: 100
deslizar en 1 segs a x: 0 y: 100
deslizar en 1 segs a x: 0 y: 0
```

b)

```
al hacer clic en  
ir a x: 0 y: 0
deslizar en 1 segs a x: 100 y: 0
deslizar en 1 segs a x: 50 y: 100
deslizar en 1 segs a x: 0 y: 0
```

c)

```
al hacer clic en  
mover 100 pasos
girar  120 grados
deslizar en 1 segs a posición aleatoria ▾
mover 100 pasos
girar  120 grados
```

16. ● ● ● Para el siguiente diagrama de bloques, señala si las afirmaciones son verdaderas o falsas.



a) Hay una variable creada, llamada Número final, que va almacenando el valor de la suma.

Verdadero  Falso

b) El programa suma cuatro números que va escribiendo el jugador.

Verdadero  Falso

c) El programa finaliza mostrando en pantalla la suma de los números durante 2 segundos.

Verdadero  Falso

17. ● ● Elige una palabra o grupo de palabras, entre las propuestas, de modo que sea correcta cada una de las afirmaciones siguientes:



- a) El bloque \_\_\_\_\_ es una estructura iterativa, que permite repetir un bloque de instrucciones hasta que finalice el programa. Si queremos que lo haga un número finito de veces, podemos utilizar el bloque \_\_\_\_\_ o bien el de repetir hasta que ...
- b) El bloque \_\_\_\_\_ es una estructura selectiva, que permite llevar a cabo una serie de instrucciones sólo si pasa algo.
- c) Hasta que no crees la variable, ésta no te \_\_\_\_\_ en los bloques de instrucciones correspondientes.
- d) Es importante \_\_\_\_\_ las variables cuando comienza un programa, que significa darle el valor 0. Si no, se quedará el valor que tenía la variable cuando jugaste la última vez.
- e) Para hacer aclaraciones sobre el funcionamiento de un programa puedes incluir \_\_\_\_\_ en él.
- f) Las \_\_\_\_\_ de Scratch son una serie de funcionalidades que te permiten programar tanto hardware como software.