



Lectura

5.º de Primaria

Cómo se hace un videojuego

Quién lo escribe, quién lo dibuja, quién lo programa y quién lo rompe a propósito.



Nombre y apellidos: _____ Fecha: _____

Lee el texto

Un videojuego no lo hace una persona encerrada en su cuarto. En los juegos grandes trabajan **cientos de personas** durante **años**, y casi ninguna hace lo mismo que las demás. **Primero, la idea escrita.** Alguien decide de qué va: quién eres, qué quieres conseguir y qué te lo impide. Eso se escribe en un documento llamado **diseño**, que explica las reglas antes de que exista nada. Sin reglas claras, lo demás no se puede construir. **Después, el dibujo.** Los artistas diseñan los personajes, los escenarios y los objetos. Primero a lápiz o en la tableta; luego se convierten en figuras de tres dimensiones que se pueden mirar por todos los lados. **La programación.** Los programadores escriben las **instrucciones** que hacen que al pulsar un botón el personaje salte, que si cae de muy alto pierda vida y que el enemigo te persiga. El ordenador no adivina nada: hay que decírselo todo, paso a paso. **El sonido.** Se graban pisadas, puertas, lluvia y golpes. La música cambia según lo que pasa: si aparece un enemigo, suena distinto. Mucho de lo que sientes jugando viene del sonido, aunque no te des cuenta. **Y alguien que lo rompe a propósito.** Es el trabajo de **prueba**. Su tarea es hacer lo que no se debe: andar contra una pared mil veces, saltar donde no toca, guardar la partida en el peor momento. Cuando encuentra un fallo —un **bug**— lo apunta con todo detalle para que se pueda arreglar. **Y vuelta a empezar.** Un juego se prueba, se corrige y se vuelve a probar muchas veces. Cuando por fin sale, los jugadores encuentran fallos que nadie vio, y entonces llegan las **actualizaciones**. Curiosidad: la palabra bug significa «bicho» en inglés, y viene de los tiempos en que los ordenadores eran enormes y un insecto podía meterse dentro y estropear el funcionamiento.

Nivel 1. Los datos

1. ¿Cómo se llama el documento donde se escriben las reglas? _____
2. ¿Qué hacen los programadores?

3. ¿En qué consiste el trabajo de prueba?

4. ¿Qué significa la palabra bug en inglés? _____
5. ¿Qué son las actualizaciones y por qué aparecen?

6. Escribe cuatro oficios distintos que aparezcan en el texto.
_____, _____, _____ y _____
7. ¿Por qué se escriben las reglas antes de dibujar nada?

Nivel 2. Hay que deducir



8. El texto dice que «el ordenador no adivina nada». ¿Qué quiere decir?

9. ¿Por qué hace falta alguien que intente romper el juego, si el objetivo es que funcione?

Nivel 3. Interpretar

10. Une cada oficio con lo que hace.

diseño · · · graba pisadas, lluvia y música

arte · · · escribe las instrucciones

programación · · · decide las reglas del juego

sonido · · · crea personajes y escenarios

pruebas · · · busca fallos a propósito

11. El texto dice que mucho de lo que sientes jugando viene del sonido. Pon un ejemplo tuyo.

12. Verdadero o falso.

Un juego grande lo hace una sola persona. Verdadero [] Falso []

Los fallos se buscan antes de que salga el juego. Verdadero [] Falso []

Después de salir ya no se corrige nada. Verdadero [] Falso []

El trabajo de prueba consiste en jugar bien. Verdadero [] Falso []

13. De todos los oficios del texto, ¿cuál harías tú? Explica por qué y qué se te daría bien.

El reto

1. **Escribe las reglas.** Inventa un juego muy sencillo y escribe su diseño en cuatro líneas. Eres:

_____ Quieres: _____ Te lo impide: _____ Ganas

cuando: _____

2. **Piensa como un programador.** El ordenador no adivina. Escribe las instrucciones, en orden, para que un personaje **recoja una llave y abra una puerta.**

1. _____ 2. _____

3. _____ 4. _____

3. **Busca el fallo.** Lee estas instrucciones y di qué va mal.

1. **Abrir la puerta.** 2. **Recoger la llave.** 3. **Entrar.**

Qué falla: _____

Cómo lo arreglarías: _____



Solucionario

Ejercicio 1: el **diseño**. Ejercicio 2: escriben las **instrucciones** que dicen al ordenador qué tiene que pasar en cada caso. Ejercicio 3: en **buscar fallos a propósito**, haciendo lo que no se debe, y apuntarlos con detalle. Ejercicio 4: **bicho**. Ejercicio 5: son **correcciones que llegan después** de que el juego salga, porque los jugadores encuentran fallos que nadie había visto. Ejercicio 6: cuatro de estos: diseño, arte, programación, sonido, pruebas. Ejercicio 7: porque **las reglas deciden todo lo demás**. Si no se sabe qué se puede hacer en el juego, no se puede dibujar ni programar: habría que rehacerlo. Ejercicio 8: que hay que **decírselo todo paso a paso**; el ordenador no completa lo que falta ni entiende lo que quisiste decir, solo hace exactamente lo que está escrito. Ejercicio 9: porque **los jugadores harán cosas raras** de todos modos. Si no se buscan los fallos antes, los encuentran ellos, y entonces es peor. Ejercicio 10: diseño → reglas · arte → personajes y escenarios · programación → instrucciones · sonido → pisadas, lluvia y música · pruebas → fallos a propósito. Ejercicio 11: respuesta abierta. Valen la música que avisa del peligro, el silencio antes de un susto o el sonido de los pasos que dice si alguien se acerca. Ejercicio 12: **falso** · **verdadero** · **falso**, llegan las actualizaciones · **falso**, consiste en jugar mal a propósito. Ejercicio 13: respuesta abierta. Se valora que relacione el oficio con **algo que sabe hacer**, no solo con lo que le gusta. El reto: el diseño y las instrucciones son abiertos; lo que se corrige es que estén **en orden y completos**, por ejemplo: acercarse a la llave, recogerla, ir a la puerta, usar la llave. En el ejercicio 3 el fallo es el **orden**: no se puede abrir la puerta antes de tener la llave. Se arregla recogiendo la llave primero. Es el mismo error que cometen los programas cuando se les da un paso antes de tiempo.

Hoja de registro

Esta hoja es para el docente. Marca C en lo conseguido, EP en lo que va con ayuda y NC en lo que aún no aparece.

1. Localiza los oficios y sus tareas. C [] EP [] NC []
2. Explica para qué sirve el documento de diseño. C [] EP [] NC []
3. Entiende que el ordenador ejecuta, no interpreta. C [] EP [] NC []
4. Justifica el trabajo de buscar fallos. C [] EP [] NC []
5. Redacta unas reglas de juego claras. C [] EP [] NC []
6. Escribe instrucciones ordenadas paso a paso. C [] EP [] NC []
7. Detecta un error de secuencia y lo corrige. C [] EP [] NC []

Cómo leer el conjunto

El 8 y el reto 3 son la misma idea vista dos veces: **el orden importa y nada se da por supuesto**. Es pensamiento computacional puro, y aquí entra sin que el alumno note que está haciendo informática.

El 9 desmonta una idea cómoda: el que busca fallos no es un aguafiestas, es quien evita que el fallo lo encuentre el usuario. Sirve igual para un juego que para un trabajo de clase.

El 13 conecta la lectura con la orientación: detrás de una afición hay **oficios muy distintos**, y no todos consisten en jugar.