



¡Ya llegaste al tercer día!

¡Sigue así!

Por cierto, ¿qué te pareció el desafío de ayer? Al final de este correo te dejo mi idea de solución y me encantaría saber la diferencia con la tuya.

¿Alguna vez has jugado un **juego que te dé más de una opción** y, dependiendo de lo que elijas, el destino del personaje sea completamente diferente?

Hoy vas a desarrollar un ejemplo así con Javascript.

Quiero que trabajes con estructuras de control de flujo. Esta forma complicada de decirlo significa que, al igual que en los juegos, **la historia que desarrolles debe adaptarse a las respuestas dadas por quien está jugando.**

Para ello, necesitarás algunas estructuras capaces de alterar el flujo de la aplicación, como `for`, `while`, `if` y `else`. Todas estas pueden cumplir con este objetivo, dada una cierta condición.

El `if` y el `else`, que ya te he mostrado en los últimos días, son capaces de crear ramificaciones dentro de la aplicación para que se tome una u otra acción, dependiendo de la condición proporcionada.

Los **bucles** (como `for` y `while`) son capaces de transformar una tarea repetitiva en pocas líneas de código, sin importar cuántas veces necesites repetir esa tarea.

Tu desafío de hoy es crear los destinos posibles de un juego, en el que el usuario pueda elegir:

1. Si quiere **seguir hacia el área de Front-End** o **seguir hacia el área de Back-End**.
2. Si está en el área de **Front-End**, si quiere **aprender React** o **aprender Vue**. Si está en el área de **Back-End**, podrá **aprender C#** o **aprender Java**.
3. Después, independientemente de las elecciones anteriores, el usuario podrá elegir entre **seguir especializándose en el área elegida** o **desarrollarse para convertirse en Fullstack**. Debes mostrar en pantalla un mensaje específico para cada elección.
4. Finalmente, **pregunta en qué tecnologías le gustaría a la persona especializarse** o conocer. Aquí, la persona puede responder N tecnologías, una a la vez. Entonces, mientras continúe respondiendo ****ok**** a la pregunta: *"¿Hay alguna otra tecnología que te gustaría aprender?"*, sigue presentando el Prompt, para que complete el nombre de la tecnología en cuestión. Y, justo después, presenta un mensaje comentando algo sobre la tecnología ingresada.

Lo importante es que la persona que esté jugando siempre pueda **elegir qué decisión tomar para aprender** y desarrollarse en el área de programación.

Además, también es esencial que, al final del juego, pueda **ingresar tantas tecnologías como desee en la lista de aprendizaje**.

CONSEJO

Ya tienes una idea de cómo hacer que toda esta historia suceda, ¿verdad? ¡Principalmente **recordando cómo usar estructuras condicionales y bucles en Javascript!**

Si aún no sabes cómo imprimir y recibir valores en páginas web con HTML y CSS, puedes usar `console.log`, `prompt` y `alert` para desarrollar tu juego, como ya has visto en los últimos días.

Recuerda que siempre puedes **personalizar el juego de la forma que desees**.

EXTRA

Ya has visto estructuras condicionales en Javascript anteriormente, pero vamos a recapitular. El `if` se usa para verificar si una determinada condición es verdadera.

Ejemplo:

```
if (ciudad === "Roma") {  
    // muestra la foto del "Coliseo"  
}
```

Además, también puede usarse con uno o varios `else if`, que verificará si la condición anterior era falsa y comprobará si la actual es verdadera.

Finalmente, existe el `else` solo, sin ninguna condición, y el código dentro de él se ejecutará siempre que todas las condiciones anteriores encadenadas sean falsas.

```
if (ciudad === "Roma") {  
    // muestra la foto del "Coliseo"  
} else if (ciudad === "París") {  
    // muestra la foto de la "Torre Eiffel"  
} else {  
    // da la respuesta "No escribiste una ciudad válida"  
}
```

Además, **para la parte 4, necesitarás una estructura de repetición (bucle)** como el `while`. Para usarlo, es bastante fácil:

```
let edad = 0;  
  
while (edad < 8) {  
    // algún comando para imprimir la edad  
    edad = edad + 1;  
}
```

Este código comenzará con la edad en cero y, al entrar en el `while`, ese valor se imprimirá y, justo después, se incrementará en 1.

Es decir, después de la primera vez que pase, el valor de la edad será igual a 1, que es menor que 8, y por eso, **la condición del while tendrá un resultado verdadero** y continuará ejecutándose.

Solo se detendrá cuando el valor de la variable edad llegue a 8, que **no es menor que 8, y por eso la condición del while tendrá un resultado falso**.

También puedes aprender más sobre `'while'` en [este sitio](#).

RESPUESTA DEL DESAFÍO DEL DÍA 2

- [Solución del Día 2](#)
- [Solución del Día 2 \(con desafío Opcional\)](#)

¡Ya estamos llegando a la mitad de esta jornada, eh!

Te veo mañana, en el cuarto día del desafío, y claro, no olvides compartir tus códigos en tu GitHub y en tus redes sociales con el hashtag #7DaysOfCode.

¡Hasta luego!

Jose Gonzalez

Instructor Front-End en Alura Latam