

**¡Hola Querid@ Candidat@!**

A continuación, te presentamos el siguiente desafío para postular al puesto de desarrollador@ Software Support Developer. Te invitamos a hacer tu mejor esfuerzo, no importa si no terminas y solo lo resuelves parcialmente, desde ya te agradecemos por participar.

**Duración y en que consta la prueba:**

- 48 horas (Si terminas antes mejor, pero si necesita más tiempo, dale, nos enfocamos más en una prueba orientada a resultados a que solo medir el tiempo)
- Deberás desarrollar una solución que relate la pelea y diga el resultado final.
- Deberás responder unas preguntas para conocerte un poco más

**Forma de entrega:** El proyecto deberá estar en un repositorio compartido. Si lo entregas en docker mucho mejor.

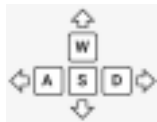
**Lenguajes, frameworks y herramientas:** Usa lenguaje que más te acomode (Si usas Python, mucho mejor, también nos gusta Go)

**Ejercicio**

**Talana Kombat JRPG**

Talana Kombat es un juego donde 2 personajes se enfrentan hasta la muerte. Cada personaje tiene 2 golpes especiales que se ejecutan con una combinación de movimientos + 1 botón de golpe.

**Los botones que se usan son**



(W)Arriba, (S)Abajo, (A)Izquierda, (D)Derecha,

(P)Puño, (K)Patada

**Golpes de nuestros personajes**

Tonyn Stallone		
Combinación	Energía que quita	Nombre del movimiento
DSD + P	3	Taladoken
SD + K	2	Remuyuken
P o K	1	Puño o Patada

Arnaldor Shuatseneguer		
Combinación	Energía que quita	Nombre del movimiento
SA + K	3	Remuyuken
ASA + P	2	Taladoken
P o K	1	Puño o Patada

**Información importante:**

Parte atacando el jugador que envió una combinación menor de botones (movimiento + golpes), en caso de empate, parte el con menos movimientos, si empatan de nuevo, inicia el con menos golpes, si hay empate de nuevo, inicia el player 1 (total el player 2 siempre es del hermano chico)

La secuencia completa del combate de cada jugador se entrega de una vez (consolidada en un json)

Cada personaje tiene 6 Puntos de energía

- Un personaje muere cuando su energía llega a 0 y de inmediato finaliza la pelea
- Tony es el player 1, siempre ataca hacia la derecha (y no cambia de lado)
- Arnaldor es el player 2, siempre ataca hacia la izquierda (y no cambia de lado)
- Los personajes se atacan uno a la vez estilo JRPG, por turnos hasta que uno es derrotado, los golpes no pueden ser bloqueados, se asume que siempre son efectivos.

Los datos llegan como un json con botones de movimiento y golpe que se correlacionan para cada jugada

Los movimientos pueden ser un string de largo máximo 5 (puede ser vacío)

Los golpes pueden ser un solo botón maximo (puede ser vacío)

Se asume que el botón de golpe es justo después de la secuencia de movimiento, es decir, AADSD + P es un Taladoken (antes se movió para atrás 2 veces); DSDAA + P son movimientos más un puño

**Para este desafío:** Desarrolla una solución que relata la pelea e informe el resultado final

Por ejemplo para el siguiente json de pelea:

```
{"player1":{"movimientos":["D","DSD","S","DSD","SD"],"golpes":["K","P","","K","P"]},"player2":{"movimientos":["SA","SA","SA","ASA","SA"],"golpes":["K","","K","P","P"]}}
```

- *Tonyn avanza y da una patada*
- *Arnaldor conecta un Remuyuken*
- *Tonyn usa un Taladoken*
- *Arnaldor se mueve*
- *Tonyn le da un puñetazo al pobre Arnaldor*
- *Arnaldor conecta un Remuyuken*
- *Arnaldor Gana la pelea y aun le queda 1 de energía*

**Otros ejemplos:**

Gana Tony

```
{"player1":{"movimientos":["SDD","DSD","SA","DSD"],"golpes":["K","P","K","P"]},"player2":{"movimientos":["DSD","WSAW","ASA","","ASA","SA"],"golpes":["P","K","K","K","P"]}}
```

"K"]}]}

Gana Arnaldor

```
{ "player1": { "movimientos": ["DSD", "S"], "golpes": ["P", ""],  
  "player2": { "movimientos": ["", "ASA", "DA", "AAA", "", "SA"], "golpes": ["P", "", "P", "K", "K", "K"] }}
```

## Preguntas generales

1. Supongamos que en un repositorio GIT hiciste un commit y olvidaste un archivo. Explica cómo se soluciona si hiciste push, y cómo si aún no hiciste.

De ser posible, que quede solo un commit con los cambios.

2. Si has trabajado con control de versiones ¿Cuáles han sido los flujos con los que has trabajado?

3. ¿Cuál ha sido la situación más compleja que has tenido con esto?

4. ¿Qué experiencia has tenido con los microservicios?

5. ¿Cuál es tu servicio favorito de GCP o AWS? ¿Por qué?

**¡Te deseamos mucha suerte!**