**¡Hola Querid@ Candidat@!**

A continuación, te presentamos el siguiente desafío para postular al puesto de desarrollador@ Software Support Engineer. 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*

➢ *Arnardold Gana la pelea y aun le queda 1 de energía*

**Otros ejemplos:**

Gana Tonyn

{"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.**

Si ya realicé el push y me di cuenta de que olvidé un archivo, primero agrego el archivo olvidado al área de preparación utilizando el comando **git add <nombre\_del\_archivo>**. Luego, creo un nuevo commit utilizando **git commit --amend -m "Mensaje de commit actualizado"**, lo que reemplazaría el commit anterior y agregaría el archivo olvidado. Por último, realizaría un push forzado utilizando **git push --force origin <nombre\_de\_la\_rama>** para sobrescribir el commit anterior en el repositorio remoto y mantener solo un commit con los cambios completos.

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

He trabajado principalmente con dos flujos de trabajo en control de versiones, que son basados en ramas.

Con el flujo de trabajo basado en ramas, desarrollo cada funcionalidad en una rama separada y luego subo el cambio y el líder de equipo se encarga de fusionar los cambios en la rama principal, para después realizar el despliegue correspondiente. Sin olvidar mencionar que después de subir los cambios se revisa el código para cualquier corrección de errores, gestionando el desarrollo de nuevas funcionalidades, preparar lanzamientos y solucionar problemas.

A modo personal, no tengo problemas en adaptarme a los flujos de trabajo específicos de su empresa, ya que considero importante seguir las prácticas establecidas por el equipo para una colaboración efectiva.

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

La situación más compleja que he tenido fue cuando entre a trabajar a Celeris (mi último empleador), entre y comenzó la tormenta perfecta, partamos que renuncio el compañero más antiguo, el que sabia todo el funcionamiento de los proyectos de todos los clientes, y justo él estaba con un cliente importante que era el banco estado, sumando que después de él, el colega más antiguo tenía 8 meses aproximadamente, la cosa que a la semana renuncia el gerente de desarrollo y con eso el equipo quedo a la deriva, y yo con dos semanas aproximadamente tome el proyecto del banco, sin capacitación, si saber que era, fueron 4 meses de aprendizaje full, pero logre sacar adelante el proyecto aunque no voy a negar que hubieron momentos en el que me amanecí trabajando, pero al final todo salió bien (ojo que resumí la situación).

**4. ¿Qué experiencia has tenido con los microservicios?**

Hasta el momento, mi experiencia se ha centrado principalmente en la exploración e investigación de microservicios lambda en AWS. Aunque no he tenido la oportunidad de trabajar en proyectos de producción con ellos, he adquirido un conocimiento sólido en este ámbito a través de mi enfoque investigativo.

He explorado cómo implementarlos, desplegarlos y escalarlos utilizando los servicios de AWS, como AWS Lambda, Amazon API Gateway.

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

AWS es mi servicio favorito por varias razones. En primer lugar, AWS ofrece una amplia gama de servicios en la nube que cubren casi todas las necesidades de desarrollo y despliegue de aplicaciones. Además, AWS tiene una gran escalabilidad y flexibilidad. Puedo escalar mis recursos de manera rápida y eficiente para satisfacer las demandas cambiantes de mis aplicaciones, sin preocuparme por la infraestructura. La seguridad es otra razón por la que prefiero AWS y Por último, pero no menos importante, AWS cuenta con una comunidad activa y un soporte técnico excepcional

**¡Te deseamos mucha suerte!**