

## Parcial 1

La empresa de videojuegos universitarios Icesi Games tiene el proyecto de desarrollar un pequeño juego inspirado en el popular juego de STOP.

El juego a desarrollar empieza por la fase de conexión, donde el servidor espera a 2 jugadores. Una vez conectados, el servidor envía a los participantes una letra al azar, que es mostrada en la pantalla de cada jugador.

Luego de mostrar la letra, todos los jugadores empiezan a jugar, escribiendo palabras que comiencen por la letra recibida, en cuatro distintas categorías: Nombre o apellido, Animal, País o ciudad y Cosa.

La ronda finaliza cuando cualquiera de los jugadores pulse sobre el botón STOP. En ese momento, los jugadores ya no pueden escribir nada y reciben el puntaje según la regla tradicional: 50 puntos si la palabra está repetida, 100 puntos si la palabra no está repetida o 0 si no escribió palabra.

Es necesario que en la pantalla de cada jugador aparezcan las respuestas del usuario contendor cuando ya se haya anunciado el STOP.

Gráficamente se espera algo como la siguiente imagen.

Letra J		
Nombre o apellido	<input type="text"/>	
Animal	<input type="text"/>	
País o ciudad	<input type="text"/>	
Cosa	<input type="text"/>	
<input type="button" value="STOP"/>		

Resultados		
	TU	ADVERSARIO
Nombre o apellido	Juliana (100)	James (100)
Animal	Jabalí (50)	Jabalí (50)
País o ciudad	Jalisco (100)	Jordania (100)
Cosa	(0)	Jinete (100)
TOTAL	250	350

La empresa quiere limitar el juego para que cada ronda sólo tenga dos jugadores.

**Calificación**

**0.5 puntos** - El programa de los clientes tiene una interfaz gráfica simple con dos ventanas disponibles: ventana de juego y ventana de puntajes.

**0.5 puntos** - La conexión clientes-servidor funciona correctamente.

Por su parte el servidor NO tiene que tener interfaz gráfica, ni entrada por consola.

**0.5 puntos** - Todos los mensajes entre los clientes y el servidor se hacen usando JSON.

**0.5 puntos** - El programa está modelado correctamente. Sus clases del modelo

presentan variables suficientes y necesarias para el funcionamiento correcto del

juego.

**1 punto** - El servidor debe ser capaz de alojar múltiples partidas. Cada partida se compone de un par de jugadores. Por ejemplo si hay 5 usuarios que se conectan (cronológicamente Arnold, Bernardo, Carla, Daniela y Ernesto), las partidas se organizan en orden de llegada: Arnold juega con Bernardo, Carla con Daniela y Ernesto debe esperar a un sexto jugador para comenzar a jugar.

**1 punto** - El programa del cliente implementa el patrón Observer y el patrón Singleton. Reflexione sobre la utilidad de ambos patrones en la implementación del programa.

**1 punto** - El funcionamiento general del juego: reglas de puntuación, asignación aleatoria de letra al comienzo de la partida en donde ambos jugadores deben ver la misma letra, evitar que el jugador pulse STOP sin haber llenado todo y reglas de terminación de la partida.

**NOTA:** Para esta primera versión, el programa NO tiene que validar la existencia de una palabra escrita, ni tiene que validar que la palabra empiece por la letra requerida.

**Bonus**

**0.5 puntos** - Una vez terminada la partida se puede volver a jugar otra partida con el mismo contendor.

### **Presentación**

Presente el jueves la solución técnica.

El objetivo es explicar cómo planteó la solución, qué mensajes intercambia, cómo es el flujo de la aplicación, cuál es la estructura del programa, el modelo de datos o algún apartado del programa que le ayude a explicar a otros ingenieros su solución.