Trabajo 2 – Bases de datos 2 (10%)

Solidity

Un organizador de rifas ha decidido modernizar su sistema, reemplazando las rifas tradicionales por rifas gestionadas mediante contratos inteligentes. Esta decisión surge tras escuchar de figuras públicas como Carmen Electra (inevitable) y Bolo Yeung (el Hulk chino) que esta tecnología ofrece una forma segura de gestionar las transacciones.

Inicialmente, el organizador instancia un contrato, convirtiéndose automáticamente en el **dueño** de este. El contrato queda en estado **Iniciado**. Posterior a la creación del contrato (y estando el contrato en estado **Iniciado**), el dueño debe hacer dos acciones fundamentales:

1. Nombrar a un **delegado** (un usuario de la red distinto al dueño).
2. Luego, debe definir la duración, en segundos, del período de inscripciones de usuarios participantes de la rifa.

Una vez nombrado el delegado y establecida la duración de las inscripciones, el contrato pasa al estado **Inscripciones**. En esta etapa, cualquier usuario —excepto el dueño y el delegado— se pueden inscribir pagando exactamente **1 Ether**. Al hacerlo, el sistema le asigna un **número secreto aleatorio entero** entre 1 y 10, visible únicamente por el delegado (por lo tanto, debe haber una función que permita que el delegado vea los números asignados a cada uno de los inscritos[1]). Después de que se haya cumplido el tiempo de inscripciones, ningún usuario se podrá inscribir.

Cuando finaliza el tiempo de inscripciones, la rifa pasa al estado **Apuestas** (activada por el delegado o por el dueño). Solo los usuarios inscritos pueden participar en esta etapa, y cada uno puede apostar una sola vez, eligiendo entre los siguientes montos (ver siguiente tabla) que le conceden un número de intentos para adivinar el número secreto asignado por el sistema al momento de su inscripción.

---

[1] **Nota**: En un sistema más realista dicha función no debería existir para evitar posibles filtraciones por parte del delegado a algunos usuarios…Pero acá se solicita hacerla para verificar cómo va el proceso. Lo mismo ocurre con otras funciones solicitadas en este trabajo.

Monto apuesta (Ethers)	Número de intentos que se conceden
5	1
10	2
15	3
20	4
25	5

Luego de que un usuario ha apostado, puede intentar adivinar su número secreto según el número de intentos que posee (ver tabla anterior).

Si el usuario acierta en alguno de sus intentos, se convierte en **candidato a ganador**.

En cualquier momento, el delegado puede cerrar la etapa de apuestas, lo que da paso a la fase de **Ganadores**. En esta nueva etapa:

- Cada candidato debe escoger un **número entero** entre 1 y el total de candidatos a ganadores (los candidatos podrán ver cuántos candidatos hay en total para así elegir dentro del rango permitido).
- No puede haber números repetidos entre los candidatos, es decir, si un usuario candidato escoge un número que ya fue elegido por otro, tendrá que escoger otro número.

Cuando todos[2] los candidatos hayan elegido su número, el delegado podrá activar la fase final: **Repartición** (de premios).

Nota: Debe haber una función que le permita solamente al delegado ver el número elegido por cada candidato.

En esta etapa, el sistema genera dos números enteros aleatorios distintos dentro del mismo rango (un número entre 1 y el total de candidatos a ganadores):

- El primer número determina al **ganador principal** (es decir, el candidato que posea el primer número que el sistema generó aleatoriamente).
- El segundo número determina al **segundo puesto** (es decir, el candidato que posea el segundo número que el sistema generó aleatoriamente).

---

[2] De nuevo, en un sistema más realista esta condición se podría flexibilizar o automatizar…

Nota: Debe haber una función que le permita solamente al delegado ver estos números.

Finalmente, se hace la distribución de premios de la siguiente manera:

- El **delegado** recibe la suma total de las tarifas de inscripción (1 Ether por cada usuario inscrito).
- El **dueño** recibe el **25% del total apostado** por todos los usuarios.
- El **ganador principal** recibe el **50% del total apostado** por todos los usuarios.
- El **segundo puesto** recibe el **25% del total apostado** por todos los usuarios.

Nota: Debe haber una función que permita observar los ganadores y cuánto se ganó cada uno. Esta función es pública (cualquier usuario puede invocarla y ver los resultados).

Nota: Usted es libre de decidir qué acciones tomar ante situaciones no descritas en el enunciado, por ejemplo, ¿qué pasa si solamente se inscribe un usuario y no acierta el número? ¿qué hacer si no se inscriben usuarios? Etc.

Notas adicionales:

• Puede usar todas las estructuras de datos de Solidity que desee.
• Para entregar por email a *fjmoreno@unal.edu.co* con copia al monitor **Alejandro Orozco Ochoa** <aorozcooc@unal.edu.co>, el viernes 13 de junio hasta las 5 pm.
• No se reciben trabajos en hora posterior. No se reciben versiones "mejoradas".
• El trabajo debe incluir un informe breve (máximo 4 páginas) en PDF donde se describa que funciones tiene el contrato, que parámetros tiene cada una y que retorna. Este informe hace parte de la calificación del trabajo.
• Grupos máximo de tres personas, mínimo de dos.
• Los trabajos deben ser independientes entre los grupos. Trabajos copiados así sea en un SOLO punto se califican con 0 (cero) en su totalidad para todos los integrantes. El trabajo debe ser desarrollado por los integrantes del grupo no por personas ajenas a él.
• Cualquier duda consultarla personalmente o por email con el profesor.
• El monitor les puede ayudar con aspectos técnicos pero su función no es hacerles la práctica **ni está autorizado para cambiar las condiciones del trabajo**.
• Si hay errores en el enunciado por favor informarme lo más pronto posible para corregirlos.

Atentamente,
Francisco Moreno, **Mayo 27, 2025**

Enunciado redactado por el monitor, revisado: Francisco Moreno.