A person riding a horse

Description automatically generated with medium confidence**Punto 2 del Trabajo 2 (50%)**

**Solidity**

**Casa de Apuestas Hípica *Cowboy Dreams[[1]](#footnote-1)***

Cree un *smart contract* en Solidity con los siguientes requisitos para manejar una casa de apuestas sobre hípica.

Un usuario, denominado el anfitrión, instancia el contrato.

**Solamente el anfitrión puede:**

* Crear carreras de caballos. Cada carrera se identifica con un código y un nombre. Por ejemplo, la carrera 98 llamada “*Copa de Oro Chino*” y la carrera 75 “*Trofeo del Caballo Gordo*”. Cuando un anfitrión crea una carrera queda en estado “Creada”.
* Registrar caballos. Cada caballo se identifica con un código y un nombre. Por ejemplo, el caballo 764 llamado “Deacon Blue” y el caballo 888 llamado “Prefab Sprout”[[2]](#footnote-2).
* Registrar un caballo en una determinada carrera (que está en estado “Creada”). Por ejemplo, registrar el caballo 888 en la carrera 98. Nótese que un caballo puede participar en varias carreras y en una carrera pueden participar varios caballos, pero en una carrera pueden participar como máximo cinco caballos y deberán participar mínimo dos caballos (ver siguiente párrafo).
* Indicar que una carrera pasa al estado “Registrada”. Cuando una carrera se encuentra en este estado, ya no acepta que se registren más caballos en ella. Una carrera solamente puede pasar a este estado (“Registrada”) si tiene al menos dos caballos registrados.

Un usuario (exceptuando al anfitrión) puede hacer apuestas. Un usuario apuesta una cantidad de Ethers a un caballo dado en una carrera específica (la carrera debe estar en estado “Registrada” y por supuesto el caballo debe estar registrado en la carrera). Un usuario no puede apostar a más de un caballo en una misma carrera, pero sí puede apostar en diferentes carreras. Un usuario luego de haber apostado a un caballo en una carrera puede (si lo desea) incrementar (pero no disminuir) su valor apostado. Por ejemplo, el usuario con dirección 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4 puede haber apostado inicialmente 2 Ethers al caballo 888 (que está registrado en la carrera 98) y luego decide apostarle otros 3 Ethers a dicho caballo en dicha carrera para un total de 5 Ethers.

Cuando el anfitrión indica que una carrera termina, se genera un número aleatorio (Nroale) entre 1 y N, donde N es el número de caballos registrados en dicha carrera (1 < N < 6). Nroale indica cual fue el caballo ganador de la carrera. Ejemplo: supongamos que la carrera 98 tuvo en total tres caballos registrados: el 764 (primer caballo registrado en dicha carrera), el 888 (segundo caballo registrado en dicha carrera) y el 932 (tercer y último caballo registrado en dicha carrera). Acá, N = 3. Si por ejemplo Nroale = 2, entonces el caballo ganador de esa carrera fue el 888. Cuando se conoce el caballo ganador de una carrera se hace lo siguiente: se toma toda la cantidad de Ethers que se han apostado por parte de los usuarios en dicha carrera. La cuarta parte de dicha cantidad se transfiere a la cuenta del anfitrión y el resto se transfiere proporcionalmente entre los usuarios que le hayan apostado al caballo ganador (si nadie le apostó al caballo ganador, el anfitrión se queda con **todo** el dinero de la carrera). Ejemplo: supongamos que el total de dinero apostado en la carrera 98 fue 40 Ethers y que el ganador fue el caballo 888 al cual le apostaron en total 25 Ethers: 10 Ethers se los apostó el usuario 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4 y 15 Ethers se los apostó el usuario 0xAb8483F64d9C6d1EcF9b849Ae677dD3315835cb2. Entonces 10 Ethers se trasfieren a la cuenta del anfitrión y de los 30 Ethers restantes, 12 se trasfieren a la cuenta del usuario 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4 y 18 Ethers se trasfieren a la cuenta del usuario 0xAb8483F64d9C6d1EcF9b849Ae677dD3315835cb2. Cuando una carrera termina queda en estado “Terminada”.

Se deben además crear funciones o variables públicas que permitan a los usuarios (apostadores) saber cuántas carreras hay en total, cuáles caballos hay inscritos en cada carrera, cual es el estado de una carrera, y sobre una carrera que haya terminado: saber el caballo ganador, cuales apostadores ganaron y cuánto dinero se le trasfirió a cada uno.

**Notas adicionales:**

* Para enviar por email a *fjmoreno@unal.edu.co* y con copia a *jpenat@unal.edu.co* **el miércoles 26 de octubre hasta las 4 pm**. Solo se califican trabajos enviados a esos dos correos. No se reciben trabajos en hora posterior. No se reciben versiones “mejoradas”. No se califican trabajos enviados “por accidente” a otros correos.
* Grupos máximo de tres personas.
* Los trabajos deben ser independientes entre los grupos. Trabajos copiados así sea en una SOLA parte se califican con 0 (cero) en su totalidad para todos los integrantes.
* El monitor les puede ayudar con aspectos técnicos, pero su función no es hacerle el trabajo **ni está autorizado para cambiar las condiciones del trabajo**.
* Si trabaja con otro lenguaje, así su trabajo funcione y sea “espectacular”, el trabajo se califica con cero.
* Si encuentra errores en el enunciado por favor avisar lo más pronto posible para hacer las correcciones respectivas.
* Favor enviar en un solo mensaje todo el trabajo. Incluya en el mensaje los nombres de los integrantes. Nombre por favor el archivo con los apellidos de los integrantes, ejemplo: *Depp\_Electra\_Jolie.txt*.

Agradecimientos a ese género musical por darme la inspiración para crear este enunciado. ¡Qué privilegio haber crecido con él!

1. Esta nota no tiene que ver con el trabajo: ¡posiblemente fundada por Paddy McAloon de Prefab Sprout! [↑](#footnote-ref-1)
2. Esta nota no tiene que ver con el trabajo: los nombres de los caballos corresponden a grupos de *sophisti-pop*, véase [*https://en.wikipedia.org/wiki/List\_of\_sophisti-pop\_artists*](https://en.wikipedia.org/wiki/List_of_sophisti-pop_artists), “*Real Gone Kid*” de Deacon Blue y “*Cars and Girls*” de Prefab Sprout. [↑](#footnote-ref-2)