

## Trabajo Practico 03: Contratos Inteligentes

### Objetivo:

Se requiere crear un contrato inteligente en solidity para que dos personas puedan jugar Piedra, Papel o Tijera con apuestas. Para ello se creará una moneda ficticia, y las apuestas se realizarán en dicha moneda. Esta moneda tendrá la particularidad de poder acumular saldo negativo. Se detallan a continuación los requerimientos de contrato inteligente:

**Fecha de entrega:** miércoles 27 de junio

### Detalle y requerimientos:

1. Creación de una moneda llamada fiubaCoin, que servirá para realizar apuestas. La moneda, deberá estar asociada, por supuesto, a direcciones validas de la blockChain.
2. El contrato deberá contar con dos objetos (*structs*) públicos de tipo “Persona”, siendo que una persona tiene los siguientes atributos:
  - CUIL: uint64
  - Nombre: String
  - Address (es decir una dirección valida en la BlockChain)
3. Deberá existir un par de métodos: “setApostador01” y “setApostador02” que permitan establecer estos dos objetos Persona respectivamente. Pero el único que podrá hacerlo es el dueño de la cuenta en la cual se creó el contrato. Si desde otra cuenta, quieren *setear* un apostador, el contrato no deberá hacer nada.
4. El contrato deberá guardar en un campo publico, la fecha de creación.
5. Deberá tener un método llamado “apostar”, dicho método recibirá dos parámetros: “objeto escogido” y dinero a apostar. Siendo “objeto escogido” uno de los siguientes: PIEDRA, PAPEL, TIJERA. Estos tres elementos deben ser parte de un tipo enumerado.
6. Al método “apostar” solo lo podrán invocar desde las cuentas de Apostador01 o Apostador02. Si se ejecuta desde otra cuenta el contrato no debe de hacer nada.
7. Cuando uno de los dos apostadores invoque al método “apostar”, se deberá *setear* un atributo privado del contrato de tipo Apuesta, el cual será un struct con los siguientes campos:
  - valor-escogido: (tipo enum Piedra, Papel Tijera)
  - dinero: (entero).Habrà dos atributos: apuesta01 y apuesta02. Cada uno con la apuesta de cada apostador respectivamente.
8. Una vez que se hayan establecido los dos objetos: apuesta01 y apuesta02 (es decir cuando el segundo apostador invoque al método apostar) se deberá resolver de inmediato la apuesta. Y según las reglas del piedra, papel o tijera determinar quien ganó y quien perdió. De ambas apuestas se tomará la menor, y ese monto se añadirá a la cuenta del apostador ganador y se restará de la cuenta del apostador perdedor. Recuerden que puede tener saldo negativo.

## 75.70 Sistemas de programación no convencional de robots

9. Se deberá de poder consultar el saldo de fiubaCoins de cualquier cuenta, usando la dirección de la blockCahin.

10. Al finalizar la transferencia de saldo, durante la apuesta se deberá emitir un evento, indicando de que cuenta a cual otra se transfiere y que cantidad. Así los usuarios podrán notificarse cuando este evento ocurra.

### **Entrega:**

Lo que se espera es que cada grupo entregue un solo archivo de extensión “.sol” escrito en solidity, el archivo debe contener toda la definición del contrato inteligente y debe de poder ser ejecutado en la interfaz web Remix: <http://remix.ethereum.org/>  
No hará falta desplegarlo en una blockChain real.

### **Material de ayuda:**

Para ayudarse en el desarrollo de este trabajo, deberán utilizar la IDE Remix, ya mencionada. Podrán encontrar un tutorial breve pero consiso de como utilizar dicha IDE en el siguiente vídeo de youtube: [https://www.youtube.com/watch?v=7I4E78\\_-B4M](https://www.youtube.com/watch?v=7I4E78_-B4M)  
Información para empezar con Solidty y crear una criptoMoneda podrán encontrarla acá: <https://solidity.readthedocs.io/en/v0.4.24/>