

# Laboratorios

## Ejercicio 01

**Desarrollar un contrato en Solidity con las siguientes consideraciones:**

- Utilizará la versión 0.4.24 de Solidity
- Deberá asignar el owner en el constructor utilizando la variable global sender
- Deberá tener una lista de candidatos de tipo string como array dinámico
- En el constructor, deberá asignar tres opciones como candidatos
- Crear una función pública llamada totalVotesFor que devolverá el número de votos para el candidato que se recibirá por parámetro (de tipo string)
- La función totalVotesFor deberá efectuar un require con 0.05 ether para poder funcionar
- Generar la función pública voteForCandidate para registrar el voto en el candidato recibido por parámetro
- La función deberá requerir 0.15 ether para funcionar
- Liberar en remix y probar

## Ejercicio 02

**Desarrollar un contrato en Solidity con las siguientes consideraciones:**

- Utilizará la versión 0.4.24 de Solidity
- Deberá definir un evento llamado "NumberRequested" que recibirá como parámetro un uint256
- Deberá asignar el owner en el constructor utilizando la variable global sender
- Deberá tener una lista de candidatos de tipo string como array dinámico
- Definir un evento llamado ANewVoteForCandidate que reciba como parámetro el nombre del mismo
- En el constructor, deberá asignar tres opciones como candidatos
- Crear una función pública llamada totalVotesFor que devolverá el número de votos para el candidato que se recibirá por parámetro (de tipo string)
- Generar un modificador de función llamado requireEther en cuya definición se valide que el emisor del mensaje haya enviado la suma exacta de 0.1 ether
- Aplicar el modificador de función a las funciones de votación
- Invocar el evento ANewVoteForCandidate cada vez que haya un nuevo voto
- Exponer una función pública aplicándole un modificador de onlyOwner para efectuar el cambio de dueño del contrato
- Aplicarle a la función de cambio de owner el modificador de requireEther
- Liberar en remix y probar