

```
// *****

const Web3 = require('web3');

const Tx = require('ethereumjs-tx').Transaction;

var abijs = require('ethereumjs-abi')

var BN = require('bn.js')

// *****

const rpcURL = "wss://goerli.infura.io/ws/v3/6a04ad695d1b4522a654dd20f16bf9c9";

const web3 = new Web3(rpcURL);

const CONTRACT_ADDRESS = "0x19050e739cAdDffad706b8f3CfD4B94D1DdDa39b";

const CONTRACT_ABI = [
  {
    "inputs": [
      {
        "internalType": "uint256",
        "name": "voto",
        "type": "uint256"
      }
    ],
    "name": "firmaDelVoto",
    "outputs": [
      {
        "internalType": "bytes32",
        "name": "",
        "type": "bytes32"
      }
    ],
    "stateMutability": "view",
    "type": "function"
  },
  {

```

```

        "inputs": [
            {
                "internalType": "uint256",
                "name": "voto",
                "type": "uint256"
            },
            {
                "internalType": "bytes",
                "name": "_mensajeFirmado",
                "type": "bytes"
            }
        ],
        "name": "quienFirmo",
        "outputs": [
            {
                "internalType": "address",
                "name": "",
                "type": "address"
            }
        ],
        "stateMutability": "view",
        "type": "function"
    }
];

const contract = new web3.eth.Contract(CONTRACT_ABI, CONTRACT_ADDRESS);

// *****

const cuentaEmisora = '0xdD870fA1b7C4700F2BD7f44238821C26f7392148';

const clavePrivadaEmisora =
Buffer.from('71975fbf7fe448e004ac7ae54cad0a383c3906055a65468714156a07385e96ce',
'hex');

```

```

const clavePrivada =
"0x71975fbf7fe448e004ac7ae54cad0a383c3906055a65468714156a07385e96ce";

// *****

var hashVoto = "0x" + abijs.soliditySHA3(
    [ "address","uint", "address"],
    [ new BN("0x4B0897b0513fdC7C541B6d9D7E929C4e5364D2dB", 16), 1,
    CONTRACT_ADDRESS ]
).toString('hex');

console.log("--- Hash Voto ---");
console.log(hashVoto);
console.log("0xe87f128d97ffc1d10e1231a6ff0267d035a1f1520842e8be1430b1809cb19d50");

var firmaTransaccion = web3.eth.accounts.sign(hashVoto, clavePrivada);
console.log("--- Firma Transaccion ---");
console.log(firmaTransaccion.signature);

// *****

const firmaFuncion = 'firmaDelVoto(uint256)';
const parametro1 = 1;
const datosCodificados = web3.eth.abi.encodeFunctionSignature(firmaFuncion) +
    web3.eth.abi.encodeParameters(['uint256'], [parametro1]).slice(2);
console.log(datosCodificados);

// *****

const firmaVotoFunction = contract.methods.firmaDelVoto(1);
const data = firmaVotoFunction.encodeABI();
console.log(data);
console.log("--- Metodo FirmaVoto Encode ---");

```

```
const nonce = web3.utils.toHex(web3.eth.getTransactionCount(cuentaEmisora));
const gasPrice = web3.utils.toHex(web3.eth.gasPrice);
const gasLimit = web3.utils.toHex(21000);
const value = web3.utils.toHex(web3.utils.toWei('1', 'ether'));
const transaccion = new Tx(
  {
    nonce: nonce,
    gasPrice: gasPrice,
    gasLimit: gasLimit,
    to: CONTRACT_ADDRESS,
    value: value,
    data: data
  },
  { chain: 5 }
);
```

```
transaccion.sign(clavePrivadaEmisora);
```

```
const transaccionFirmada = '0x' + transaccion.serialize().toString('hex');
console.log("Transacción firmada:", transaccionFirmada);
```

```
const hashTransaccionFirmada = web3.utils.keccak256(transaccionFirmada);
console.log("Hash de la transacción firmada :", hashTransaccionFirmada);
```