

```
/* INSTALAR PRIMERO nodejs
entrar en una consola
c:/ node -v
esto debe dar la version de nodeJs instalada.
Luego instalar las dos librerias
c:/ npm i web3
c:/ npm i exceljs
```

```
Crear un account en infura.io y crear un nuevo proyecto.
Usar la clave personal ejemplo
wss://mainnet.infura.io/ws/v3/6a04ad695d1b4522a654dd20f16bffff
buscar el contrato inteligente a probar
ejemplo 0xC02aaA39b223FE8D0A0e5C4F27eAD9083C756Cc2
lanzar el comando para grabar las transacciones y los eventos
c:/ node bot.js
*/
```

```
const Web3 = require('web3'); // Utilizar la libreria web3 para llamar
a contratos inteligentes
const rpcURL =
"wss://mainnet.infura.io/ws/v3/6a04ad695d1b4522a654dd20f16bf9c9"; // TU
DIRECCION CREADA EN INFURA
const web3 = new Web3(rpcURL); // crear el objeto de llamada
const CONTRACT_ADDRESS = "0xDaF88906aC1DE12bA2b1D2f7bFC94E9638Ac40c4"; //
EL CONTRATO QUE QUIERES LLAMAR
const CONTRACT_ABI =
[{"constant":true,"inputs":[],"name":"name","outputs":[{"name":"","type":
"string"}],"payable":false,"stateMutability":"view","type":"function"},{"
constant":false,"inputs":[{"name":"guy","type":"address"},{"name":"wad","
type":"uint256"}],"name":"approve","outputs":[{"name":"","type":"bool"}],
"payable":false,"stateMutability":"nonpayable","type":"function"},{"const
ant":true,"inputs":[],"name":"totalSupply","outputs":[{"name":"","type":"
uint256"}],"payable":false,"stateMutability":"view","type":"function"},{"
constant":false,"inputs":[{"name":"src","type":"address"},{"name":"dst","
type":"address"},{"name":"wad","type":"uint256"}],"name":"transferFrom","
outputs":[{"name":"","type":"bool"}],"payable":false,"stateMutability":"n
onpayable","type":"function"},{"constant":false,"inputs":[{"name":"wad","
type":"uint256"}],"name":"withdraw","outputs":[],"payable":false,"stateMu
tability":"nonpayable","type":"function"},{"constant":true,"inputs":[],"n
ame":"decimals","outputs":[{"name":"","type":"uint8"}],"payable":false,"s
tateMutability":"view","type":"function"},{"constant":true,"inputs":[{"na
me":"","type":"address"}],"name":"balanceOf","outputs":[{"name":"","type"
:"uint256"}],"payable":false,"stateMutability":"view","type":"function"},
{"constant":true,"inputs":[],"name":"symbol","outputs":[{"name":"","type"
:"string"}],"payable":false,"stateMutability":"view","type":"function"},{
"constant":false,"inputs":[{"name":"dst","type":"address"},{"name":"wad",
"type":"uint256"}],"name":"transfer","outputs":[{"name":"","type":"bool"}
],"payable":false,"stateMutability":"nonpayable","type":"function"},{"con
stant":false,"inputs":[],"name":"deposit","outputs":[],"payable":true,"st
```

```

    ateMutability":"payable","type":"function"}, {"constant":true,"inputs":[{"name":"","type":"address"}, {"name":"","type":"address"}], "name":"allowance", "outputs":[{"name":"","type":"uint256"}], "payable":false, "stateMutability":"view", "type":"function"}, {"payable":true, "stateMutability":"payable", "type":"fallback"}, {"anonymous":false, "inputs":[{"indexed":true, "name":"src", "type":"address"}, {"indexed":true, "name":"guy", "type":"address"}, {"indexed":false, "name":"wad", "type":"uint256"}], "name":"Approval", "type":"event"}, {"anonymous":false, "inputs":[{"indexed":true, "name":"src", "type":"address"}, {"indexed":true, "name":"dst", "type":"address"}, {"indexed":false, "name":"wad", "type":"uint256"}], "name":"Transfer", "type":"event"}, {"anonymous":false, "inputs":[{"indexed":true, "name":"dst", "type":"address"}, {"indexed":false, "name":"wad", "type":"uint256"}], "name":"Deposit", "type":"event"}, {"anonymous":false, "inputs":[{"indexed":true, "name":"src", "type":"address"}, {"indexed":false, "name":"wad", "type":"uint256"}], "name":"Withdrawal", "type":"event"}];

const contract = new web3.eth.Contract(CONTRACT_ABI,
CONTRACT_ADDRESS); // crear el contrato con direccion y abi

/* VALORES INICIALES DEL CONTRATO QUE PUEDES VER EN READ DEL CONTRATO */
console.log('**VALORES INICIALES DE LECTURA DEL CONTRATO**');
contract.methods.totalSupply().call((err, result) => { console.log("Total supply = " + result) });
contract.methods.name().call((err, result) => { console.log("Nombre del contrato = " + result) });
contract.methods.symbol().call((err, result) => { console.log("Simbolo del contrato = " + result) });
contract.methods.decimals().call((err, result) => { console.log("Decimales del contrato = " + result) });
contract.methods.balanceOf('0x7BB4057d35598B34f60f4668290F49cf450B8D01').call((err, result) => { console.log("Balance de la cuenta = " + result/1000000000000000000) });
/* -----
*/

/* CREACION DEL EXCEL */
const Excel = require('exceljs'); // Agregamos la libreria exceljs
const fileName = 'ballenas.xlsx'; // nombre del archivo excel que crearemos
const workbook = new Excel.Workbook(); // crea el objeto excel
/* -----
*/

/* PROPIEDADES DE LA HOJA Excel Transfer , HEADERS, FILTROS , COLORES , LETRAS */
const worksheetTransfer = workbook.addWorksheet('Transfer', {properties:{tabColor:{argb:'FFC000'}}});
worksheetTransfer.columns = [
    { header: 'Transfer', key: 'Transfer', width: 64 },
    { header: 'Destination', key: 'Destination', width: 64 },

```

```

    { header: 'Tokens', key: 'Tokens', width: 32 },
  ];
  worksheetTransfer.autoFilter = {
    from: 'A1',
    to: 'C1',
  };
  /* -----
  */

  /* PROPIEDADES DE LA HOJA Excel Deposit , HEADERS, FILTROS , COLORES ,
  LETRAS */
  const worksheetDeposit = workbook.addWorksheet('Deposit',
  {properties:{tabColor:{argb:'FFC0000'}}});
  worksheetDeposit.columns = [
    { header: 'Wallet', key: 'Wallet', width: 64 },
    { header: 'Tokens', key: 'Tokens', width: 32 },
  ];
  worksheetDeposit.autoFilter = {
    from: 'A1',
    to: 'B1',
  };
  /* -----
  */

  /* PROPIEDADES DE LA HOJA Excel Approval , HEADERS, FILTROS , COLORES ,
  LETRAS */
  const worksheetApproval = workbook.addWorksheet('Approval',
  {properties:{tabColor:{argb:'FFC0000'}}});
  worksheetApproval.columns = [
    { header: 'Wallet', key: 'Wallet', width: 64 },
    { header: 'Contrato', key: 'Contrato', width: 64 },
    { header: 'Tokens', key: 'Tokens', width: 32 },
  ];
  worksheetApproval.autoFilter = {
    from: 'A1',
    to: 'C1',
  };
  /* -----
  */

  /* PROPIEDADES DE LA HOJA Excel Withdrawal , HEADERS, FILTROS , COLORES ,
  LETRAS */
  const worksheetWithdrawal = workbook.addWorksheet('Withdrawal',
  {properties:{tabColor:{argb:'FFC0000'}}});
  worksheetWithdrawal.columns = [
    { header: 'Wallet', key: 'Wallet', width: 64 },
    { header: 'Tokens', key: 'Tokens', width: 32 },
  ];
  worksheetWithdrawal.autoFilter = {

```

```

    from: 'A1',
    to: 'B1',
  });
  /* -----
  */

let counterTransfer = 2;
let counterTransferEvents = 2;
let counterTransferBallenaEvents = 2;
let counterDepositEvents = 2;
let counterWithdrawalEvents = 2;
let counterDeposit = 2;
let counterApproval = 2;
let counterWithdrawal = 2;

const START_BLOCK = 13992430; // 13990310
const END_BLOCK = 4760000; // 7701000;

/* CREADOR DE FILE EXCEL PARTIENDO DE TODAS LAS TRANSACCIONES DESDE UN
BLOQUE INICIAL A UNO FINAL */
contract.getPastEvents("allEvents",{
  fromBlock: START_BLOCK, // Numero de bloque Inicial
  toBlock: 'latest' // Numero de bloque final o tambien
'latest' para ir al ultimo
}).then(events => {
// console.log(events); // VER TODAS LAS TRANSACCIONES COMPRENDIDAS EN
LOS BLOQUES EN PANTALLA

  for (const property in events) {
    console.log("INICIO " + events[property].event);
    if("Transfer" === events[property].event){
      console.log("Wallet de envio = " +
events[property].returnValues.src);
      console.log("Wallet de Destinacion = " +
events[property].returnValues.dst);
      console.log("Total de Tokens Transferidos = "
+ events[property].returnValues.wad / 1000000000000000000);
      worksheetTransfer.getRow(counterTransfer++).values =
[events[property].returnValues.src, events[property].returnValues.dst,
events[property].returnValues.wad / 1000000000000000000];
      workbook.xlsx.writeFile(fileName).then(() => {
        console.log('Agregado Record Transfer en file Excel');
      }).catch(err => {
        console.log(err.message);
      });
    } else if("Deposit" === events[property].event){

```

```

        console.log("Wallet de Deposito = " +
events[property].returnValues.dst);
        console.log("Total de Tokens Depositados = "
+ events[property].returnValues.wad / 1000000000000000000);
        worksheetDeposit.getRow(counterDeposit++).values =
[events[property].returnValues.dst, events[property].returnValues.wad /
1000000000000000000];
        workbook.xlsx.writeFile(fileName).then(() => {
        console.log('Agregado Record Deposit en file Excel');
    }).catch(err => {
        console.log(err.message);
    });
    } else if("Approval" === events[property].event){
        console.log("Wallet de Aprobacion = " +
events[property].returnValues.src);
        console.log("Contrato de Aprobacion = " +
events[property].returnValues.guy);
        console.log("Total de Tokens Aprobados = "
+ events[property].returnValues.wad / 1000000000000000000);
        worksheetApproval.getRow(counterApproval++).values =
[events[property].returnValues.src, events[property].returnValues.guy,
events[property].returnValues.wad / 1000000000000000000];
        workbook.xlsx.writeFile(fileName).then(() => {
        console.log('Agregado Record Approval en file Excel');
    }).catch(err => {
        console.log(err.message);
    });
    } else if("Withdrawal" === events[property].event){
        console.log("Wallet de Retiro = " +
events[property].returnValues.src);
        console.log("Total de Tokens Depositados = "
+ events[property].returnValues.wad / 1000000000000000000);
        worksheetWithdrawal.getRow(counterWithdrawal++).values =
[events[property].returnValues.src, events[property].returnValues.wad /
1000000000000000000];
        workbook.xlsx.writeFile(fileName).then(() => {
        console.log('Agregado Record Withdrawal en file Excel');
    }).catch(err => {
        console.log(err.message);
    });
    }
    console.log("FIN");
}
}).catch((err) => console.error(err));
/* -----
*/

/* PROPIEDADES DE LA HOJA Excel TransferEventos , HEADERS, FILTROS ,
COLORES , LETRAS */

```

```

const worksheetTransferEvents = workbook.addWorksheet('TransferEventos',
{properties:{tabColor:{argb:'FFC000'}}});
worksheetTransferEvents.columns = [
  { header: 'Transfer', key: 'Transfer', width: 64 },
  { header: 'Destination', key: 'Destination', width: 64 },
  { header: 'Tokens', key: 'Tokens', width: 32 },
  { header: 'Date', key: 'Date', width: 32 }
];
worksheetTransferEvents.autoFilter = {
  from: 'A1',
  to: 'D1',
};
/* -----
*/

/* ESCUCHADOR DEL EVENTO Transfer EN EL CONTRATO */
contract.events.Transfer().on('data', (event) => {
  //console.log(event.returnValues);
  worksheetTransferEvents.getRow(counterTransferEvents++).values = [event.returnValues.src, event.returnValues.dst,
event.returnValues.wad / 1000000000000000000,new
Date().toLocaleString()];
  workbook.xlsx.writeFile(fileName).then(() => {
    console.log('Agregado Record Evento Tranfer en file
Excel');
  }).catch(err => {
    console.log(err.message);
  });
}).on('error', console.error);
/* -----
*/

/* PROPIEDADES DE LA HOJA Excel DepositEventos , HEADERS, FILTROS ,
COLORES , LETRAS */
const worksheetDepositEvents = workbook.addWorksheet('DepositEventos',
{properties:{tabColor:{argb:'FFC000'}}});
worksheetDepositEvents.columns = [
  { header: 'Wallet', key: 'Wallet', width: 64 },
  { header: 'Tokens', key: 'Tokens', width: 32 },
  { header: 'Date', key: 'Date', width: 32 }
];
worksheetDepositEvents.autoFilter = {
  from: 'A1',
  to: 'C1',
};
/* -----
*/

/* ESCUCHADOR DEL EVENTO Deposit EN EL CONTRATO */

```

```

contract.events.Deposit().on('data', (event) => {
    //console.log(event.returnValues);
    worksheetDepositEvents.getRow(counterDepositEvents++).values
= [event.returnValues.dst, event.returnValues.wad /
1000000000000000000,new Date().toLocaleString()];
    workbook.xlsx.writeFile(fileName).then(() => {
        console.log('Agregado Record Evento Deposito en file
Excel');
    }).catch(err => {
        console.log(err.message);
    });
}).on('error', console.error);
/* -----
*/

/* PROPIEDADES DE LA HOJA Excel WithdrawalEventos , HEADERS, FILTROS ,
COLORES , LETRAS */
const worksheetWithdrawalEvents =
workbook.addWorksheet('WithdrawalEventos',
{properties:{tabColor:{argb:'FFC000'}}});
worksheetWithdrawalEvents.columns = [
    { header: 'Wallet', key: 'Wallet', width: 64 },
    { header: 'Tokens', key: 'Tokens', width: 32 },
    { header: 'Date', key: 'Date', width: 32 }
];
worksheetWithdrawalEvents.autoFilter = {
    from: 'A1',
    to: 'C1',
};
/* -----
*/

/* ESCUCHADOR DEL EVENTO Withdrawal EN EL CONTRATO */
contract.events.Withdrawal().on('data', (event) => {
    //console.log(event.returnValues);
    worksheetWithdrawalEvents.getRow(counterWithdrawalEvents++).v
alues = [event.returnValues.src, event.returnValues.wad /
1000000000000000000,new Date().toLocaleString()];
    workbook.xlsx.writeFile(fileName).then(() => {
        console.log('Agregado Record Evento Retiro en file
Excel');
    }).catch(err => {
        console.log(err.message);
    });
}).on('error', console.error);
/* -----
*/

```

```

/* PROPIEDADES DE LA HOJA Excel BallenasTransferEventos , HEADERS,
FILTROS , COLORES , LETRAS */
const worksheetTransferBallenaSeguimientoEvents =
workbook.addWorksheet('BallenasTransferEventos',
{properties:{tabColor:{argb:'FFC000'}}});
worksheetTransferBallenaSeguimientoEvents.columns = [
  { header: 'Transfer', key: 'Transfer', width: 64 },
  { header: 'Destination', key: 'Destination', width: 64 },
  { header: 'Tokens', key: 'Tokens', width: 32 },
  { header: 'Date', key: 'Date', width: 32 }
];
worksheetTransferBallenaSeguimientoEvents.autoFilter = {
  from: 'A1',
  to: 'D1',
};
/* -----
*/
/* ESCUCHADOR DEL EVENTO Transfer SEGUIMIENTO A UNA BALLENA EN EL
CONTRATO */
contract.events.Transfer().on('data', (event) => {
  //console.log(event.returnValues);
  if(event.returnValues.src ==
"0x68b3465833fb72A70ecDF485E0e4C7bD8665Fc45" ){
    worksheetTransferBallenaSeguimientoEvents.getRow(counterTrans
ferBallenaEvents++).values = [event.returnValues.src,
event.returnValues.dst, event.returnValues.wad / 1000000000000000000,new
Date().toLocaleString()];
    workbook.xlsx.writeFile(fileName).then(() => {
      console.log('Agregado Record Evento Seguimiento Ballena
Tranfer en file Excel');
    }).catch(err => {
      console.log(err.message);
    });
  }
}).on('error', console.error);
/* -----
*/

/*
contract.events.allEvents()
.on('data', (event) => {
  console.log(event);
}).on('error', console.error);
*/

```