

```
// SPDX-License-Identifier: GPL-3.0
```

```
pragma solidity ^0.8.0;
```

```
contract CodeCriptoHash {
```

```
    function firmaDelVoto(uint256 voto) public view returns (bytes32) {
```

```
        bytes32 hashFirmadoDelVoto =  
keccak256(abi.encodePacked(msg.sender,voto,address(this)));
```

```
        return hashFirmadoDelVoto;
```

```
    }
```

```
    function quienFirmo(uint256 voto,bytes memory _mensajeFirmado) public view returns  
(address) {
```

```
        bytes32 hash = keccak256(abi.encodePacked(msg.sender,voto,address(this)));
```

```
        hash = keccak256(abi.encodePacked("\x19Ethereum Signed Message:\n32", hash));
```

```
        return _recoverSigner(hash, _mensajeFirmado);
```

```
    }
```

```
    function _recoverSigner(bytes32 message, bytes memory sig) internal pure returns (address)  
{
```

```
        uint8 v;
```

```
        bytes32 r;
```

```
        bytes32 s;
```

```
        (v, r, s) = _splitSignature(sig);
```

```
        return ecrecover(message, v, r, s);
```

```
    }
```

```
    function _splitSignature(bytes memory sig) internal pure returns (uint8, bytes32, bytes32) {
```

```
        require(sig.length == 65);
```

```
        bytes32 r;
```

```
bytes32 s;  
uint8 v;  
  
assembly {  
    r := mload(add(sig, 32))  
    s := mload(add(sig, 64))  
    v := byte(0, mload(add(sig, 96)))  
}  
  
return (v, r, s);  
}  
}
```