

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.27;

import {IERC20} from "@openzeppelin/contracts/token/ERC20/IERC20.sol";

contract ICO {

    IERC20 immutable public token ;

    uint256 immutable public price;

    error TranferFail();

    event TokenBought(address indexed whois , uint256 quantity);
    event TokenSold(address indexed sell , uint256 indexed token, uint256 price , uint256 eth);

    constructor ( IERC20 _token ,uint256 _price){
        token = _token;
        price=_price;
    }

    function buy() external payable {
        ///@dev , aca hay que hacer una funcion que no pueda comprar mas de 100TK , si tiene , que no sume
        mas.
        uint256 amount = msg.value/price;

        token.transfer(msg.sender, amount);

        emit TokenBought(msg.sender, amount);
    }

    ///https://youtu.be/dVxu0alH2ys?list=PL_r-luCAIAeel21VCmwdOWBZhPPwKs2Qz

    function sell_FALLA_SEGU(uint256 _amount) external {
        ///@dev spread compra y venta

        /// aca nosotros se lo compramos le pagamos los eth. y despues nos da los tks.
        uint256 ehtAmount =( _amount * price *98)/100;
        ///@dev no oficial
        // payable(msg.sender).transfer(ehtAmount);
        ///@dev oficial
```

```
(bool succes, ) =payable(msg.sender).call{value: ehtAmount}("");
///@dev error personalizado
if(!succes) revert TranferFail();

/// en el paso anterior se sacaron los eth , ahora tiene quedar los token!!
/// aca es el msg.sender , es la persona que los esta vendiendo a los token
```

```

hacia To que somos nosotros.
    /// nostros es a nuestro contrato.
    token.transferFrom(msg.sender,address(this),_amount);
    /// la pregunta es el allowances tiene permiso para sacarme los token?
msg.sender yo y el contrato => address(this)
    ///lo pregunte y tiene cero , en el allowances , hay que quitarle o subirle el
allowances.
    /// el spender es el contrato !! y el sender yo!!(owner)

    ///@important aca hicieron un pregunta de seguridad , que primero deberiamos
sacarle los token , antes de comprarlos con ETH
    /// LO HIZO AL REVÉS APROPOSITO!!

    /// CHECK - efect -Interaction. osea primero chequear , luego efect cobrar y
despues enviar(token)

```

```

}

```

```

function sell_Security(uint256 _amount) external {
//Check
uint256 ehtAmount =( _amount * price *98)/100;

```

```

//interaccion (me pasas lo token)
token.transferFrom(msg.sender,address(this),_amount);
//efect. y aca cuando ya tengo los token te envio los eth
(bool succes, ) =payable(msg.sender).call{value: ehtAmount}("");
if(!succes) revert TranferFail();

emit TokenSold(msg.sender , _amount, _amount/ehtAmount , ehtAmount);

```

```

}

```

```

}

```