// SPDX-License-Identifier: MIT
pragma solidity > 0.8.0;

interface IERC20 {

```
/// @dev no puede ser publicas las variables en Interfaces ya que marca como se
ven las cosas desde afuera
/// @dev  hay tener en cuenta que public es external + internal.
error InvalidadAllowances();
error ZeroValue();



event Transfer(address indexed _from, address indexed _to, uint256 _value);
/// @dev aca tiene 3 ya que esta el _owner que seria el sender.msn  , lo que
esta indexado sirve para buscar en los logs.
event Approval(address indexed _owner, address indexed _spender, uint256
_value);

function name() external  view returns (string memory);
/// @dev en ether son 18 , los usdc son 6
function symbol() external view returns (string memory);
/// @dev en ether son 18 , los usdc son 6
function decimals() external view returns (uint8);
/// @dev  cuanta cantidad de token hay en total.
function totalSupply() external view returns (uint256);


/// @dev  devuelve el balance de un usuario.
function balanceOf(address _owner) external view returns (uint256);


 /// @dev operacion de escritura  --con transferencia ==> evento de
transferencia
function transfer(address _to, uint256 _value) external returns (bool success);


/// @dev transferFrom seria como un debito automático. de from , aquien se
sacamos y a quien se lo enviemos _to
/// @dev  _value es la cantidad de token que queremos transferir.

/// @dev aca solo puedo transferir a una billetara que me aprueba ese retiro.

 /// @dev operacion de escritura  --con transferencia  ==> evento de
transferencia
function transferFrom(address _from, address _to, uint256 _value) external
returns (bool success);
```

```
    /// @dev y para retirar ese dinero tenemos una cantidad que solo podemos
    rehabilitar a traves del aprove()
    ///@dev     Allows _spender to withdraw from your account multiple times, up to
    the _value amount. If this function is called again it overwrites the current
    ///@dev  el approve solo la puede habilitar  a la persona que se le esta
    debitando el dinero , el _spender seria netflix osea quien nos saca , un monto
    ya aprobado.
    /// @dev operacion de escritura evento de consulta  --evento de approve
    function approve(address _spender, uint256 _value) external returns (bool
    success);



    ///@dev es la cantidad de plata de que tiene en la approve . ejemplo cuanta
    plata tengo aprobada a netflix ,
    /// lo saca netflix y despues tiene cero cuando se lo llama dice lo que se le o
    el remain.
    function allowance(address _owner, address _spender) external view returns
    (uint256 remain);
```

}