



## Ejercicio 01 -

David Olivares Muñoz - UDC  
31-10-2023

## Ejercicio 01 - FabricaContract.

```
pragma solidity ^0.8.0;

contract FabricaContract {
    uint idDigits = 16;

    struct Producto {
        string nombre;
        uint id;
    }

    Producto[] public productos;

    event NuevoProducto(uint ArrayProductoId, string nombre, uint id);

    mapping (uint => address) public productoAPropietario;
    mapping (address => uint) public propietarioProductos;

    function crearProducto(string memory _nombre, uint _id) private {
        productos.push(Producto(_nombre, _id));
        uint id = productos.length - 1;
        productoAPropietario[id] = msg.sender;
        propietarioProductos[msg.sender]++;
        emit NuevoProducto(id, _nombre, _id);
    }

    function _generarIdAleatorio(string memory _str) private view
returns (uint) {
        uint rand = uint(keccak256(abi.encodePacked(_str)));
        return rand % (10 ** idDigits);
    }
}
```

```
function crearProductoAleatorio(string memory _nombre) public {
    uint randId = _generarIdAleatorio(_nombre);
    crearProducto(_nombre, randId);
}

function getProductosPorPropietario(address _propietario) external
view returns (uint[] memory) {
    uint contador = 0;
    uint cantidadProductos = propietarioProductos[_propietario];
    uint[] memory resultado = new uint[](cantidadProductos);
    for (uint i = 0; i < productos.length; i++) {
        if (productoAPropietario[i] == _propietario) {
            resultado[contador] = i;
            contador++;
        }
    }
    return resultado;
}
```