

Contrato básico en Solidity para gestionar un mensaje

[crayon-6a9d4f7a2cf24933203069/]

The screenshot displays the Remix IDE interface, which is used for developing and testing smart contracts. The central pane shows the Solidity code for a contract named `HolaMundo`. The code includes a state variable `mensaje`, a constructor that initializes it with the string "Hola Mundo desde Solidity!", and two public functions: `obtenerMensaje()` to retrieve the message and `cambiarMensaje()` to update it.

```
1 // SPDX-License-Identifier: MIT
2 pragma solidity ^0.8.0;
3
4 contract HolaMundo {
5     // Variable de estado para almacenar el mensaje
6     string public mensaje;
7
8     // Constructor que inicializa el mensaje
9     constructor() {
10         mensaje = "Hola Mundo desde Solidity!";
11     }
12
13     // Función para obtener el mensaje
14     function obtenerMensaje() public view returns (string memory) {
15         return mensaje;
16     }
17
18     // Función para cambiar el mensaje
19     function cambiarMensaje(string memory _nuevoMensaje) public {
20         mensaje = _nuevoMensaje;
21     }
22 }
```

On the left, the "DEPLOY & RUN TRANSACTIONS" panel shows that the contract has been successfully deployed. It lists the deployed contract "HOLAMUNDO AT 0XD91..." and provides buttons to interact with it, such as "cambiarMen...", "mensaje", and "obtenerMen...". The "Low level interactions" section shows the raw transaction data.

The bottom right pane displays the execution results. It shows a successful transaction where the `cambiarMensaje` function was called, followed by a `CALL` to the `obtenerMensaje` function. The results include the transaction hash, gas used, and the return data.