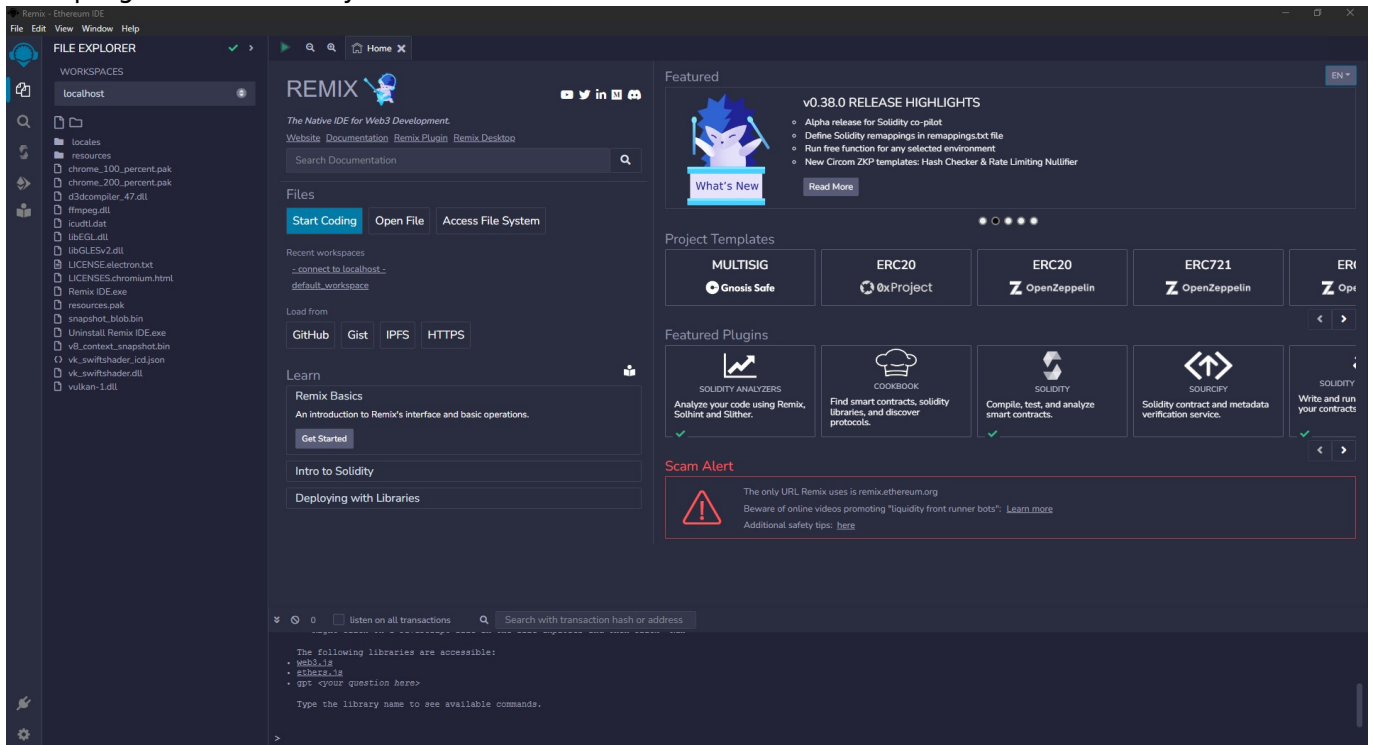


Introducción a la Programación en Solidity

Para programar en solidity se utiliza el IDE [Remix](#):



componentes de un Smart Contract

Pragma

Lo primero que se escribe en un contrato inteligente es la versión pragma, que indica la versión del compilador que debe usar el código. Generalmente se debe poner un rango de versiones que sean compatibles con el código, por ejemplo, si quisiéramos que el código fuera compilable por las versiones entre la 0.6.12 y la 0.9.0 escribiríamos:

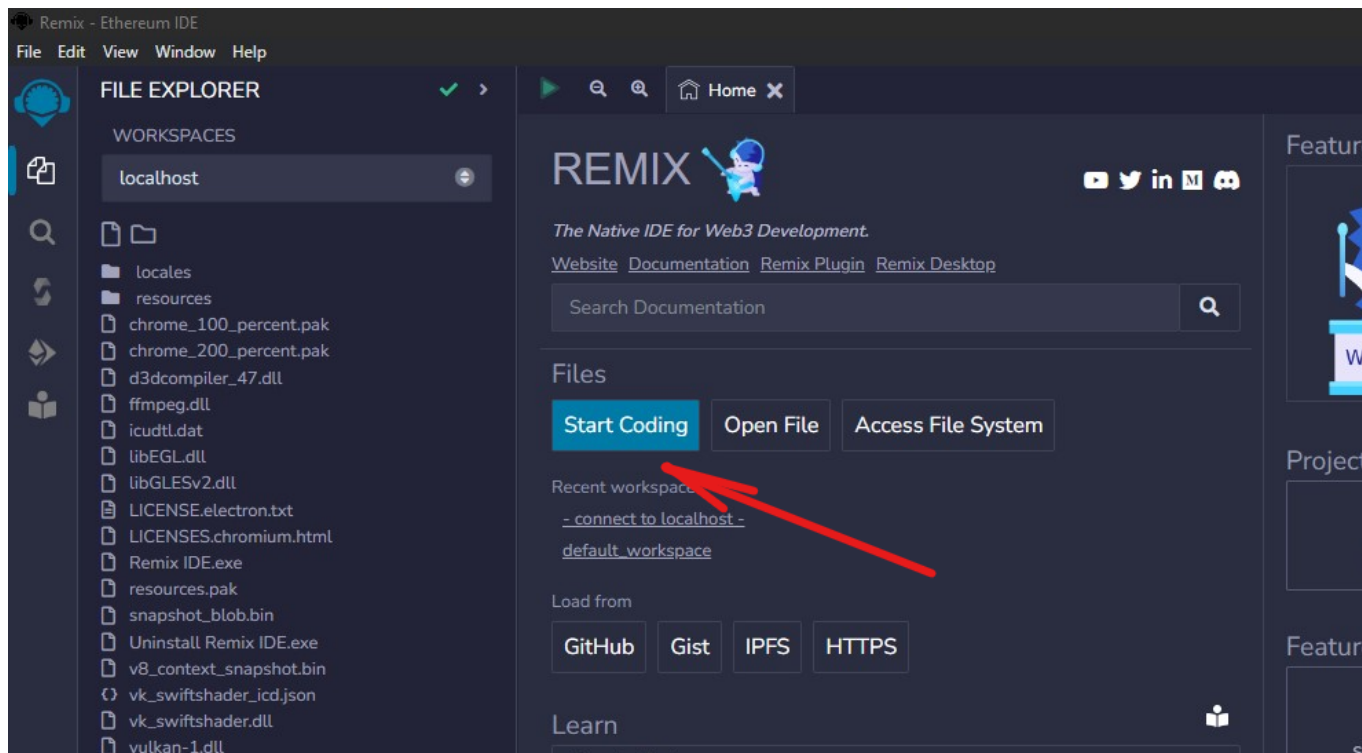
```
pragma solidity >=0.6.12 <0.9.0;
```

Para crear un contrato vacío, a continuación del pragma se puede introducir lo siguiente:

```
contract NombreDelContrato{  
  
}
```

Hola Mundo en un Smart Contract

Comenzaremos creando un Smart Contract de prueba con el típico “Hello World”, para ello pulsaremos en Start Coding:



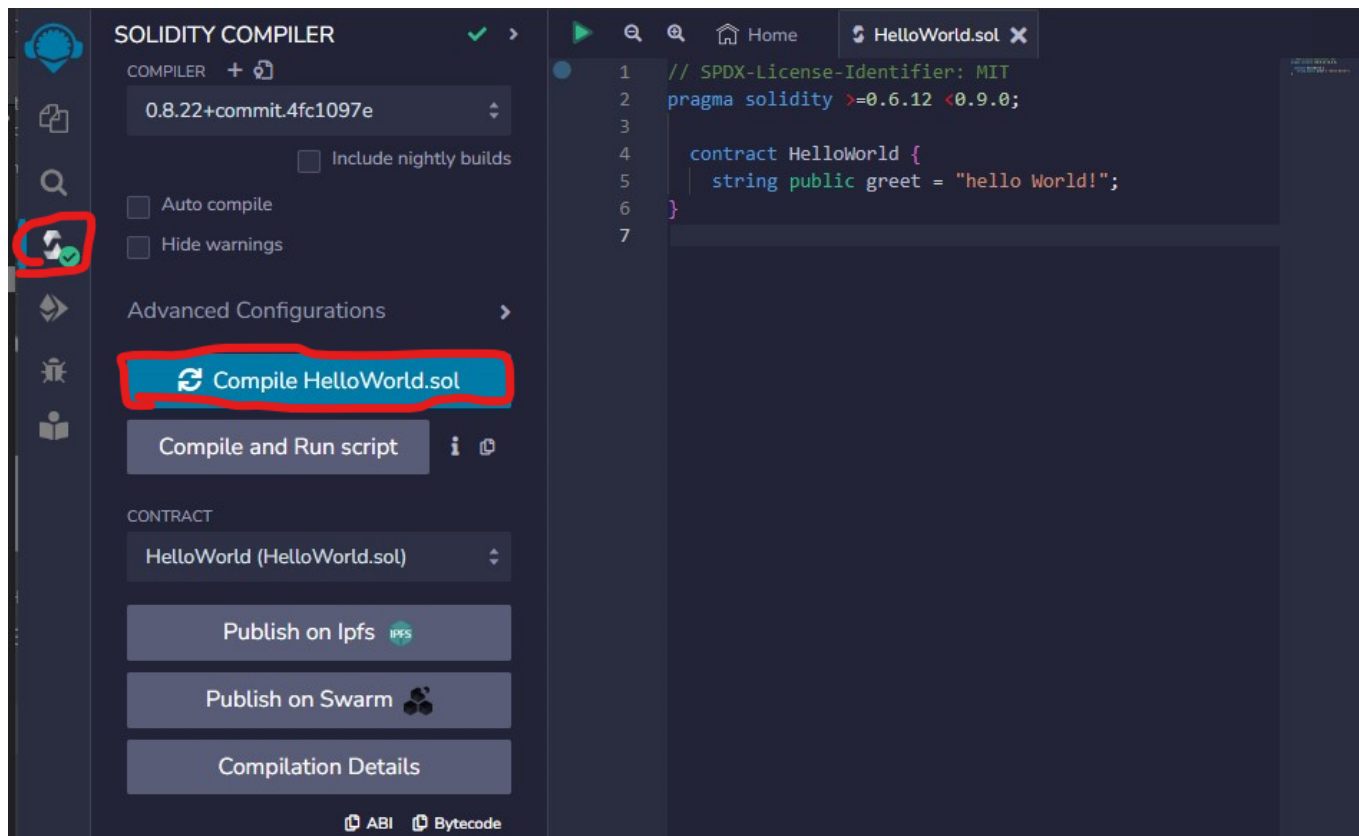
Para hacer un “Hola Mundo” escribimos el siguiente código:

[HolaMundo.sol](#)

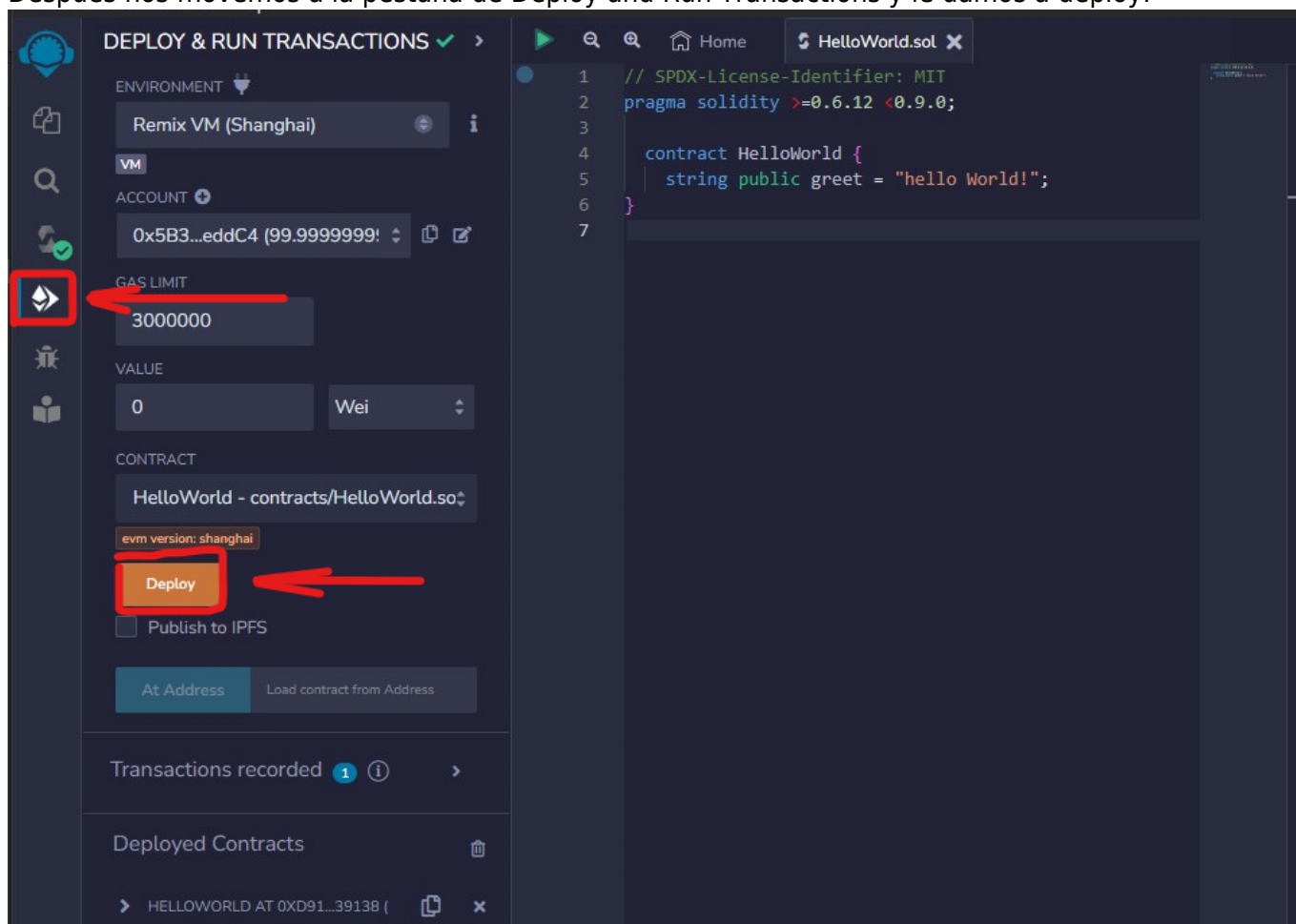
```
// SPDX-License-Identifier: MIT
pragma solidity >=0.6.12 <0.9.0;

contract HelloWorld {
    String public greet = "Hello World!";
}
```

Tras eso iremos a la pestaña de solidity compiler y le daremos a compile:



Después nos movemos a la pestaña de Deploy and Run Transactions y le damos a deploy:



Finalmente podemos ir a la pestaña de Deployed Contracts, seleccionar el contrato que acabamos de enviar y pulsar en el botón greet para ver el mensaje;

Transactions recorded 1 ⓘ >

Deployed Contracts ⓘ

▼ HELLOWORLD AT 0xD91...39138 | ⓘ x

Balance: 0. ETH

greet

0: string: hello World!

Low level interactions ⓘ

CALLDATA

Transact

0 ☐ listen on all transactions 🔍 Search with transaction hash or address

to: HelloWorld.greet() data: 0xcfa...e3217

call to HelloWorld.greet

[call]

CALL from: 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4 to: HelloWorld.greet() data: 0xcfa...e3217

Debug ▼

From:
<https://www.knoppia.net/> - Knoppia

Permanent link:
<https://www.knoppia.net/doku.php?id=bc:solidity&rev=1726668708>

Last update: 2024/09/18 14:11

