

G01: Miguel Amato y Jorge Ortega

Practica 2

Ejercicio 1:

Después de llamar a la función HelloWorld tenemos los siguientes resultados:

✓ [vm] from: 0x583...eddC4 to: hello.helloWorld() 0xd91...39138 value: 0 wei data: 0xc60...5f76c logs: 1 hash: 0xfec...67901 Debug ^

```
transaction cost      22812 gas ⓘ
execution cost        1748 gas ⓘ
input                 0xc60...5f76c ⓘ
decoded input         {} ⓘ
decoded output         {} ⓘ
logs                  [
  {
    "from": "0xd9145CCE52D386f254917e481eB44e9943F39138",
    "topic": "0x241ba3bafc919fb4308284ce03a8f4867a8ec2f0401445d3cf41a468e7db4ae0",
    "event": "Print",
    "args": {
      "0": "Hello, World!",
      "message": "Hello, World!"
    }
  }
] ⓘ ⓘ
```

Ejercicio 2:

Para n = 50 la llamada a la funcion factorial es:

```
[vm] from: 0x5B3...eddC4 to: hello.(constructor) value: 0 wei data: 0x608...60033 logs: 0 hash: 0xefc...7ce63
call to hello.factorial

CALL [call] from: 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4 to: hello.factorial(uint256) data: 0x837...00032

from
    0x5B38Da6a701c568545dCfcB03FcB875f56beddC4

to
    hello.factorial(uint256) 0xf8e81D47203A594245E36C48e151709F0C19fBe8

execution cost
    4431 gas (Cost only applies when called by a contract)

input
    0x837...00032

decoded input
    {
        "uint256 n": "50"
    }

decoded output
    {
        "0": "uint256: 30414093201713378043612608166064768844377641568960512000000000000"
    }

logs
    []
```

Factorial de 50 =

30414093201713378043612608166064768844377641568960512000000000000

Ejercicio 3:

Después de hacer el primer deposito de 3 ETH la transacción es:

```
[vm] from: 0x5B3...eddC4 to: piggyBank0.deposit() 0xD7A...F771B value: 3000000000000000 wei data: 0xd0e...30db0 logs: 0
hash: 0xd19...09977

status
    0x1 Transaction mined and execution succeed

transaction hash
    0xd1919ad3ca8a8c20f3eeaacd1c5a60af7a8b90ca52f9d9755f865ed05da09977

block hash
    0xeda96755ccd8b0d39cbb9d64b53af69caf66886f82049eae068e283af7694a6

block number
    5

from
    0x5B38Da6a701c568545dCfcB03FcB875f56beddC4

to
    piggyBank0.deposit() 0xD7ACd2a9FD159E698b102A1ca21C9a3e3A5F771B

gas
    24387 gas

transaction cost
    21206 gas

execution cost
    142 gas

input
    0xd0e...30db0

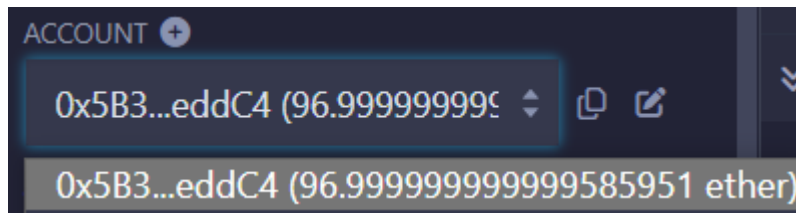
decoded input
    {}

decoded output
    {}

logs
    []

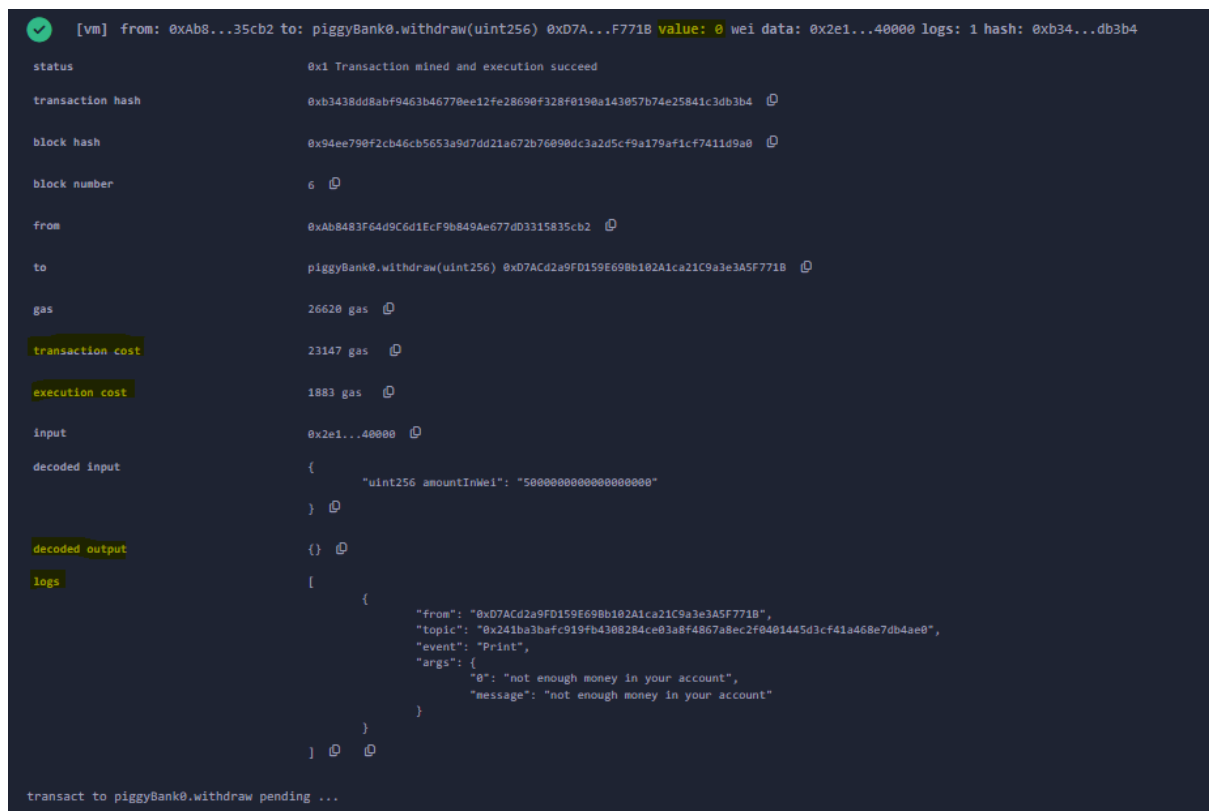
value
    3000000000000000000 wei
```

El dinero de la cuenta con el que se ha hecho la transacción después del depósito es:



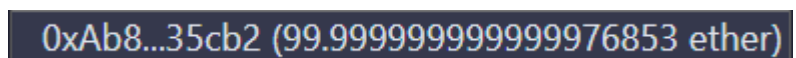
Podemos ver que hay 3 ETH menos y también hay menos por los costes de las transacciones.

Después de intentar retirar 5 ETH con otra cuenta y que no haya suficiente dinero, la transacción es:



Al no tener suficiente dinero, se emite un evento con un print en el que se le informa a la cuenta que no tiene suficiente dinero.

Este es el saldo de la cuenta tras no conseguir retirar el dinero porque no tenía suficiente:



Como se puede ver, solo se ha gastado el dinero de ejecutar la transacción.

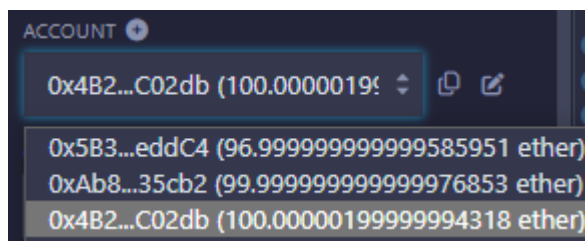
Ahora, con otra cuenta, retiramos 1 Szabo, la transacción es:

```
✓ [vm] from: 0x4B2...C02db to: piggyBank0.withdraw(uint256) 0xD7A...F771B value: 0 wei data: 0x2e1...51000 logs: 0 hash: 0x168...b32b1

status          0x1 Transaction mined and execution succeed
transaction hash 0x168fecb59607c2d019e6ac2a72dc7fdca6b7b6803cb346199903708f2bbb32b1
block hash      0x5a4c5fb132c2fda84b2e9cd53dace4371c1a8167c94140e4bfa1fe383b217109
block number    7
from            0x4020993Bc481177ec7E8f571ceCaE8A9e22C02db
to              piggyBank0.withdraw(uint256) 0xD7ACd2a9FD159E698b102A1ca21C9a3e3A5F771B
gas             32672 gas
transaction cost 28410 gas
execution cost   7170 gas
input           0x2e1...51000
decoded input    {
  "uint256 amountInWei": "1000000000000"
}
decoded output   {}
logs            []

transact to piggyBank0.withdraw pending ...
```

Exitosamente se retira el Szabo y como resultado la cuenta queda con un balance de:



Como se puede ver, ha aumentado la cantidad de dinero 10^{12} wei menos el coste de las transacciones.

También podemos ver el balance del banco con la siguiente transacción:

```
✓ [vm] from: 0x482...C02db to: piggyBank0.withdraw(uint256) 0xD7A...F771B value: 0 wei data: 0x2e1...51000 logs: 0 hash: 0xdd2...00647
call to piggyBank0.getBalance

CALL [call] from: 0x4820993Bc481177ec7E8f571ceCaE8A9e22C02db to: piggyBank0.getBalance() data: 0x120...65fe0

from      0x4820993Bc481177ec7E8f571ceCaE8A9e22C02db ⓘ
to        piggyBank0.getBalance() 0xD7ACd2a9FD159E69Bb102A1ca21C9a3e3A5F771B ⓘ
execution cost 193 gas (Cost only applies when called by a contract) ⓘ
input       0x120...65fe0 ⓘ
decoded input {} ⓘ
decoded output {
  "0": "uint256: 2999999800000000000"
} ⓘ
logs       [] ⓘ ⓘ
```

Se puede ver en el decoded output la cantidad de dinero que hay.

Al finalizar todas las cuentas quedan con el saldo:

```
0x5B3...eddc4 (96.999999999999585951 ether)
0xA8b...35cb2 (99.99999999999976853 ether)
0x4B2...C02db (100.00000199999994318 ether)
```

Siendo la primera la que depositó los 3 ETH, la segunda la que ha intentado retirar 5 ETH y la tercera la que ha logrado retirar 1 Szabo y ha comprobado el balance de la cuenta, todas ellas tienen menos wei por los costes de las transacciones.