Projecte

Creació d'un contracte intel·ligent de Crowdfunding o donacions usant BlockChain a MultiversX.

Les condicions del contracte serien les següents:

- Import mínim per donant (en 1 sola transacció): 0.1 $EGLD per transacció.
- Import màxim per donant (acumulat en totes les transaccions del mateix donant): 0.4 $EGLD en totes les transaccions del mateix donant.
- Import màxim per al projecte (sumant totes les transaccions de tots els donants):
  - Mínim de 50 $EGLD de finançament.
  - Màxim de 100 $EGLD.

Número de contracte

erd1qqqqqqqqqqqqqpgqacq4fpaah05j3ekpdst3e0xgdjpz4lxpw54sg424yy

Visualització a Devnet

https://devnet-explorer.multiversx.com/accounts/erd1qqqqqqqqqqqqqpgqacq4fpaah05j3ekpdst3e0xgdjpz4lxpw54sg424yy

Wallet

La wallet amb la què es realitza el contracte i les transaccions és la següent:
erd13p62plk4htgy3h0mqzhyzl2qp6sm0jdlpp9jzmw7f7g3vyqaw54sxcnz49

Codi del contracte (donacions.rs)

```rust
#![no_std]

use multiversx_sc::derive_imports::*;
#[allow(unused_imports)]
use multiversx_sc::imports::*;


#[type_abi]
#[derive(TopEncode, TopDecode, PartialEq, Clone, Copy)]
pub enum Status {
    FundingPeriod,
    Successful,
    Failed,
}


#[multiversx_sc::contract]
pub trait CrowdfundingSc {

    // --- MODIFICACIÓ 1: INIT ARA REP ELS NOUS PARÀMETRES ---
    #[init]
    fn init(
        &self,
```

```rust
        target: BigUint,
        deadline: u64,
        min_contribution: BigUint, // Mínim per transacció
        max_per_user: BigUint,     // Màxim acumulat per usuari
        max_cap: BigUint           // Màxim total del projecte (Hard
Cap)
    ) {
        require!(target > 0, "Target must be more than 0");
        require!(max_cap >= target, "Max cap must be >= Target");
        require!(max_per_user >= min_contribution, "Max per user must be
>= Min contribution");

        self.target().set(target);
        self.min_contribution().set(min_contribution);
        self.max_per_user().set(max_per_user);
        self.max_cap().set(max_cap);

        require!(
            deadline > self.get_current_time(),
            "Deadline can't be in the past"
        );
        self.deadline().set(deadline);
    }

    #[upgrade]
    fn upgrade(&self) {}

    #[endpoint]
    #[payable("EGLD")]
    fn fund(&self) {
        let payment = self.call_value().egld().clone_value();

        // --- CONDICIÓ 1: Import mínim per transacció ---
        require!(
            payment >= self.min_contribution().get(),
            "L'import es inferior al minim permès per transaccio"
        );

        let current_time = self.blockchain().get_block_timestamp();
        require!(
            current_time < self.deadline().get(),
            "cannot fund after deadline"
        );
```

```rust
        // --- CONDICIÓ 2: Màxim total del projecte (Hard Cap) ---
        // get_current_funds inclou el pagament actual perquè és un
mètode payable
        let current_funds = self.get_current_funds();
        require!(
            current_funds <= self.max_cap().get(),
            "S'ha superat el limit maxim de financament del projecte
(Hard Cap)"
        );

        let caller = self.blockchain().get_caller();
        let deposited_amount = self.deposit(&caller).get();
        let new_user_total = &deposited_amount + &payment;

        // --- CONDICIÓ 3: Màxim acumulat per usuari ---
        require!(
            new_user_total <= self.max_per_user().get(),
            "Superes el limit maxim d'aportacio per usuari"
        );

        // Si tot és correcte, actualitzem el dipòsit de l'usuari
        self.deposit(&caller).set(new_user_total);
    }

    #[endpoint]
    fn claim(&self) {
        match self.status() {
            Status::FundingPeriod => sc_panic!("cannot claim before
deadline"),
            Status::Successful => {
                let caller = self.blockchain().get_caller();
                require!(
                    caller == self.blockchain().get_owner_address(),
                    "only owner can claim successful funding"
                );

                let sc_balance = self.get_current_funds();
                self.send().direct_egld(&caller, &sc_balance);
            }
            Status::Failed => {
                let caller = self.blockchain().get_caller();
                let deposit = self.deposit(&caller).get();
```

```rust
            if deposit > 0u32 {
                self.deposit(&caller).clear();
                self.send().direct_egld(&caller, &deposit);
            }
        }
    }
}

#[view]
fn status(&self) -> Status {
    if self.get_current_time() <= self.deadline().get() {
        Status::FundingPeriod
    } else if self.get_current_funds() >= self.target().get() {
        Status::Successful
    } else {
        Status::Failed
    }
}

#[view(getCurrentFunds)]
fn get_current_funds(&self) -> BigUint {
    self.blockchain()
        .get_sc_balance(&EgldOrEsdtTokenIdentifier::egld(), 0)
}

// private

fn get_current_time(&self) -> u64 {
    self.blockchain().get_block_timestamp()
}

// storage

#[view(getTarget)]
#[storage_mapper("target")]
fn target(&self) -> SingleValueMapper<BigUint>;

#[view(getDeadline)]
#[storage_mapper("deadline")]
fn deadline(&self) -> SingleValueMapper<u64>;

#[view(getDeposit)]
```

```
    #[storage_mapper("deposit")]
    fn deposit(&self, donor: &ManagedAddress) ->
SingleValueMapper<BigUint>;


    // --- MODIFICACIÓ 2: NOUS STORAGE MAPPERS ---


    #[view(getMinContribution)]
    #[storage_mapper("minContribution")]
    fn min_contribution(&self) -> SingleValueMapper<BigUint>;


    #[view(getMaxPerUser)]
    #[storage_mapper("maxPerUser")]
    fn max_per_user(&self) -> SingleValueMapper<BigUint>;


    #[view(getMaxCap)]
    #[storage_mapper("maxCap")]
    fn max_cap(&self) -> SingleValueMapper<BigUint>;
}
```

1. **Realitzem la creació del nou contracte intel·ligent.**

2. Desplegament del contracte

```
mxpy contract deploy --bytecode ./output/donacions.wasm --recall-nonce
--pem=walletdiego.pem --gas-limit=60000000 --proxy=https://devnet-gateway.multiversx.com
--chain="D" --arguments 50000000000000000000 1767225600 100000000000000000
400000000000000000 100000000000000000000 --send
```

696e436f6e74726967275574696e746172276574465706f7369746d6178436170646561646c696e6e654c27696d706f727420657320696e66657269662206616
2616e73616363696f6f616c6e6e6f7420666f756e6420616674465720646561646c696e6e65532768612073570657261742720656c206c696d6974206d6178696d2064
6a656374465202848617264203631702953757065572657320206e206d6f6d656e6420642276746f617461706572206c206d6178696020
20304d617820636170206d6757374206265203e3d2054617276765744d617820706572207357365722206d7573374206265203e3d204d696e20636f6e7472696
20696e20746865205207371737463616e6f6f7420636c6169602726562665f726520646561646c696e65206566f6c79206f776e65722206361206e20636c61696207375
0418885B50B0b043Bffffff005000050002b5e3af16b18B00000696955b900001634578561dBa00000058d15e1762800000056bc75e2d63100000",
    "emittedTransactionHash": "d7bd6d7a391987214b3b82905dafe9cfcd62c93d6f3de1859dcc55297c24246b",
    "contractAddress": "erd1qqqqqqqqqqqqqpgqacq4fpaah05j3ekpdst3e0xgdjpz4lxpw54sg424yy"

pwd_is_123@lubuntu24-vbox:~/codigo/ejemplos/donacions$

Número de contracte:
erd1qqqqqqqqqqqqqpgqacq4fpaah05j3ekpdst3e0xgdjpz4lxpw54sg424yy
Contracte a Devnet:
https://devnet-explorer.multiversx.com/accounts/erd1qqqqqqqqqqqqqpgqacq4fpaah05j3ekpdst3e0xgdjpz4lxpw54sg424yy

3. Consulta del contracte
mxpy contract query
erd1qqqqqqqqqqqqqpgqacq4fpaah05j3ekpdst3e0xgdjpz4lxpw54sg424yy --proxy
https://devnet-gateway.multiversx.com --function getTarget

pwd_is_123@lubuntu24-vbox:~/codigo/ejemplos/donacions$ ^C
pwd_is_123@lubuntu24-vbox:~/codigo/ejemplos/donacions$ mxpy contract query erd1qqqqqqqqqqqqqpgqacq4fpaah05j3ekpdst3e0xgdjpz4lxpw54sg424yy --proxy https://devnet-gateway.multiversx.com --function getTarget
    "02b5e3af16b1880000"

pwd_is_123@lubuntu24-vbox:~/codigo/ejemplos/donacions$

4. Transaccions al contracte. (Donacions)
S'executen transaccions al contracte per tal de veure els límits posats.

Cas1: Transacció que supera el límit de la wallet amb la què es fa la donació.

mxpy contract call erd1qqqqqqqqqqqqqpgqacq4fpaah05j3ekpdst3e0xgdjpz4lxpw54sg424yy
--function fund --value 6000000000000000000 --recall-nonce --pem=walletdiego.pem
--gas-limit=10000000 --proxy="https://devnet-gateway.multiversx.com" --chain="D" --send

pwd_is_123@lubuntu24-vbox:~/codigo/ejemplos/donacions$ mxpy contract call erd1qqqqqqqqqqqqqpgqacq4fpaah05j3ekpdst3e0xgdjpz4lxpw54sg424yy --function fund --value 6000000000000000000 --recall-nonce --pem=wal
letdiego.pem --gas-limit=10000000 --proxy="https://devnet-gateway.multiversx.com" --chain="D" --send
WARNING cli: The --recall-nonce flag is DEPRECATED. The nonce is fetched from the network by default.   cli.py:70
INFO   utils: View this transaction in the MultiversX Devnet Explorer: https://devnet-explorer.multiversx.com/transactions/37278692b8780259b7eb4402a51f708fadd71d11e6ed420c13c7e9b56c59ff42   utils.py:161
{
    "emittedTransaction": {
        "nonce": 9,
        "value": "6000000000000000000",
        "receiver": "erd1qqqqqqqqqqqqqpgqacq4fpaah05j3ekpdst3e0xgdjpz4lxpw54sg424yy",
        "sender": "erd13p62plk4htgy3h0mqzhyzl2qp6sm0jdlpp9jzmw7f7g3vyqaw54sxcnz49",
        "senderUsername": "",
        "receiverUsername": "",
        "gasPrice": 1000000000,
        "gasLimit": 10000000,
        "data": "ZnVuZA==",
        "chainID": "D",
        "version": 2,
        "options": 0,
        "guardian": "",
        "signature": "22e5c1e3c059546a1e590db12f8331ac441759c549ac8a0a7abb78e659568f45bcd98d18e2b73359c1e05087dd77c86af0a4ae4c143e7cd9109dd5bb7c0a0109",
        "guardianSignature": "",
        "relayer": "",
        "relayerSignature": ""
    },
    "emittedTransactionData": "fund",
    "emittedTransactionHash": "37278692b8780259b7eb4402a51f708fadd71d11e6ed420c13c7e9b56c59ff42",
    "contractAddress": "erd1qqqqqqqqqqqqqpgqacq4fpaah05j3ekpdst3e0xgdjpz4lxpw54sg424yy"
}
pwd_is_123@lubuntu24-vbox:~/codigo/ejemplos/donacions$

Mirem les transaccions realitzades.

Missatge: *Insuficient funds:15540000000000*. (Superat el límit de la wallet).

Cas2: Transacció amb l'aportació del mínim de 0,1$EGLD per donant.

mxpy contract call erd1qqqqqqqqqqqqqpgqacq4fpaah05j3ekpdst3e0xgdjpz4lxpw54sg424yy
--function fund --value 100000000000000000 --recall-nonce --pem=walletdiego.pem
--gas-limit=10000000 --proxy="https://devnet-gateway.multiversx.com" --chain="D" --send

Mirem les transaccions realitzades.

Missatge: Transacció realitzada.

Cas3: Transacció que supera l'aportació màxima de 0,4 $EGLD per donant. (El donant ja havia donat 0,1 $EGLD)

Mirem les transaccions realitzades.

Missatge: *Superes el límit màxim d'aportació per usuari*.

Cas3: Transacció que supera l'aportació màxima de 0,4 $EGLD per donant. (El donant ja havia donat 0,1 $EGLD)
Transacció de 0,31 $EGLD. (Transacció màxima fins al límit del donant 0,3 $EGLD)

Mirem les transaccions realitzades.

Missatge: *Superes el límit màxim d'aportació per usuari*.

Cas4: Transacció amb l'aportació exacta restant de 0,3 $EGLD del donant. (El donant ja havia donat 0,1 $EGLD)

Mirem les transaccions realitzades.

Missatge: Transacció realitzada.

Cas5: Transacció (0,000000000000001$EGLD)que no arriba al límit de les transaccions permeses per donant. (El donant no arriba als 0,1 $EGLD)

Mirem les transaccions realitzades.

Missatge: *L'import és inferior al mínim permès per transacció*.

5. **Wallet amb la que s'han realitzat les donacions.**
   erd13p62plk4htgy3h0mqzhyzl2qp6sm0jdlpp9jzmw7f7g3vyqaw54sxcnz49

6. Transaccions realitzades per la Wallet amb la que s'han realitzat les donacions.