Date: May 2022
Students: Sebastian Pabon, Raphael Rotondari

Course: CP108 Plutus/Haskell II project

Problem statement:

Most African governments are trying different measures to eliminate corruption, bad governance, mismanagement and lack of accountability in their countries but these efforts are mostly unsuccessful. As a tool for change, blockchain can help in solving some of these governance issues plaguing Africa.

Task:

With this in mind, think of a way you can implement a smart contract that can help eliminate any of these ills, then implement it using Plutus.

Language:

Haskell and Plutus

Problem:

Child Labour

Problem description:

Child labour is a common practice in developing countries, a prominent symptom of communities with high poverty rates and low levels of development.

It is well known how the business of injustice works throughout the supply chain: situations occurring at one end of the chain (price-based competitive strategies, market pricing dynamics whose volatility tends to drive products and services into difficult price erosion cycles, and so on) tend to generate serious consequences throughout each link of the chain. The further the link is from the end, the darker the visibility and therefore the more permissible the means by which the goals are pursued: poor working conditions, serious environmental damage, exploitation of labor.

The visible end of the chain is not interested in what the opaque end of the chain has to do to fulfill its part, the important thing is that the supply chain does not stop moving. Time is money and money does not sleep.

It is this kind of environment that encourages the worst kinds of abuse against children in places around the world that have made child labour the engine of their economy. Complicity between governments and private companies has cultivated the atrocity we know as the traditional supply chain.

However, today's world is different. Or, in the words of the memorable baseball player and coach Yogi Berra, "The future ain't what it used to be". Today's consumer is starting to take a closer look at these things. He or she is beginning to pick from the shelves the coffee with the right mix of acidity and ethics; the clothes with the right texture, color and principles; the gadget with the right style, portability and level of justice. The greedy and shameless end of the supply chain is beginning to feel it: with increasing force and frequency, the consumer is discarding from his shopping list the products that are a leech on the artery of human decency and keeping in his mind and heart those products that give him the opportunity to contribute with his purchase to the development of a better humanity. This individual act, apparently isolated, is becoming part of a global sample of individual acts considerable enough to disturb the numbers on the balance sheets. The ship, little by little, is turning to avoid the obstacle in front of it.

The market's perception of a company's level of compliance with Environmental, Social, and Governance (ESG) criteria is playing a key role in the strategic decisions that must be made to avoid jeopardizing share prices and quarterly revenue numbers: a shareholder board will be ready to eviscerate management that threatens the company's market capitalization.

Solution description:

Solution title:

Anti Child Labour pledge campaign certified on Blockchain.

General description:

A social problem of the magnitude of child labour cannot be eliminated at a single stroke. It is fought by gradually weakening it through a process of disincentivization that progressively translates into awareness. The good news: the end result will be a dramatic drop in cases of child labour outside the legal limits of age, labor conditions and type of work. The bad news: we will still see cases of child labour because that's just the way human beings are. However, they will become less and less permissible.

“El Salvador is an example of a country which has made significant progress in removing hazardous child labour from the sugar cane industry. Harvesting cane is dangerous: workers use sharp machetes to cut the cane, fires are set to clear the fields, and workers have to drag heavy loads for long hours in the hot sun, breathing air that is thick with smoke. In 2002, the Sugarcane Producers Association signed a memorandum of understanding with the Government of El Salvador to put an end to child labour in the sugar cane industry. Its strategy included raising awareness, improving education and training, and organizing community-based monitoring schemes. As a result, it was able to reduce the number of child labourers in sugar cane production from 12,380 children in 2004 to 1,559 children in 2009.” Paper: Eliminating and Preventing Child Labour. International Labour Organization (ILO) 2016.

General solution approach:

- Evidence of systematic cases of child labour in social networks.
- Connecting evidence with the company contracting the suppliers involved in such evidence.
- Inform the public of the situation through social media.
- Pressure on the company involved to take measures to counteract the impact of bad publicity.
- With the support of local NGOs, company under the public eye formulates and initiates child labour disincentive campaign with its suppliers, demanding their commitment and compliance with the campaign.
- Under the general guidelines stipulated directly by the company under investigation, suppliers will engage internally in child labour disincentive campaign, fulfilling goals mainly related to the realization of talks and awareness-raising meetings around the problem: they will be characterized by their conciliatory and non-accusatory tone. Aditionally, initiatives that promote the importance of education and better working conditions (schools, protection equipment, and so on) for young people in working age and the working population in general will increase the rating received by the third party auditors hired to monitor monthly the activities developed by the suppliers within the framework of the Anti-Child Labour campaign.
- On a monthly basis, the third party auditors will inform the company under investigation of the score achieved by each supplier.
- In case of achieving the previously established minimum score, the company will proceed to mining a compliance NFT on behalf of the supplier.
- Once the NFT is in its wallet, the supplier will be able to release its financial incentive by waiting for it in a vesting contract.
- The company under the market's eye will widely publicize the results of its Anti Child Labor campaign, especially on its product packaging and marketing material: in a very non-discreet way the company will mention its social enterprise, will enable a QR code or a web address to which the consumer can easily enter, once on the website the consumer will be able to observe the documentary and audiovisual material reflecting its work and most of all, will publish a link that will allow the consumer to verify on the blockchain the transactions corresponding to the NFTs being mined and the rewards paid to the suppliers for distinguished services in the area of child labor prevention.

Sources of funding for the solution:

- Corporate grants to eliminate child labor in the international supply chain (e.g., see Child Labour Fund of the Netherlands Enterprise Agency).
- Reinvestment of part of the proceeds from increased sales due to Blockchain-certified compliance with Environmental, Social, and Governance (ESG) criteria on Child Labor.

Smart contract solution to be implemented. Specifically, smart contracts will be created for two features of the solution:

- Mining of compliance NFTs once the third party auditor has issued favorable monthly qualification. This contract is written in the AntiChildLaborToken.hs file (see smart contracts code section below).

- Vesting contract to allow the supplier to claim his reward once he has in his possession the token certifying the fulfillment of the month's objectives. This contract is written in the AntiChildLaborReward.hs file (see below smart contracts code section).

Secondary effects of issuing compliance NFTs:

- Potentially, these NFTs could be exchanged for some type of value between the supplier and the company, or between the supplier and international organizations, or between the supplier and other suppliers. A market of some kind or new business relationships could be forged through the issuance and existence of these tokens.

- Potentially, social campaign platforms such as AVAAZ (avaaz.org) could create their own DEX (decentralized exchange) where such NFTs or Native Tokens of a similar nature could be traded, creating a marketplace. Companies and individuals could assign value to compliance with ESG criteria through such transactions.

Smart contract scripts:

On a monthly basis, once the third party auditor has issued a favorable rating regarding the supplier's compliance with the monthly AntiChildLabour campaign objectives, the Lead Company under Child Labour suspicion proceeds to execute two (2) contracts, in order:

1. the AntiChildLaborToken.hs contract.

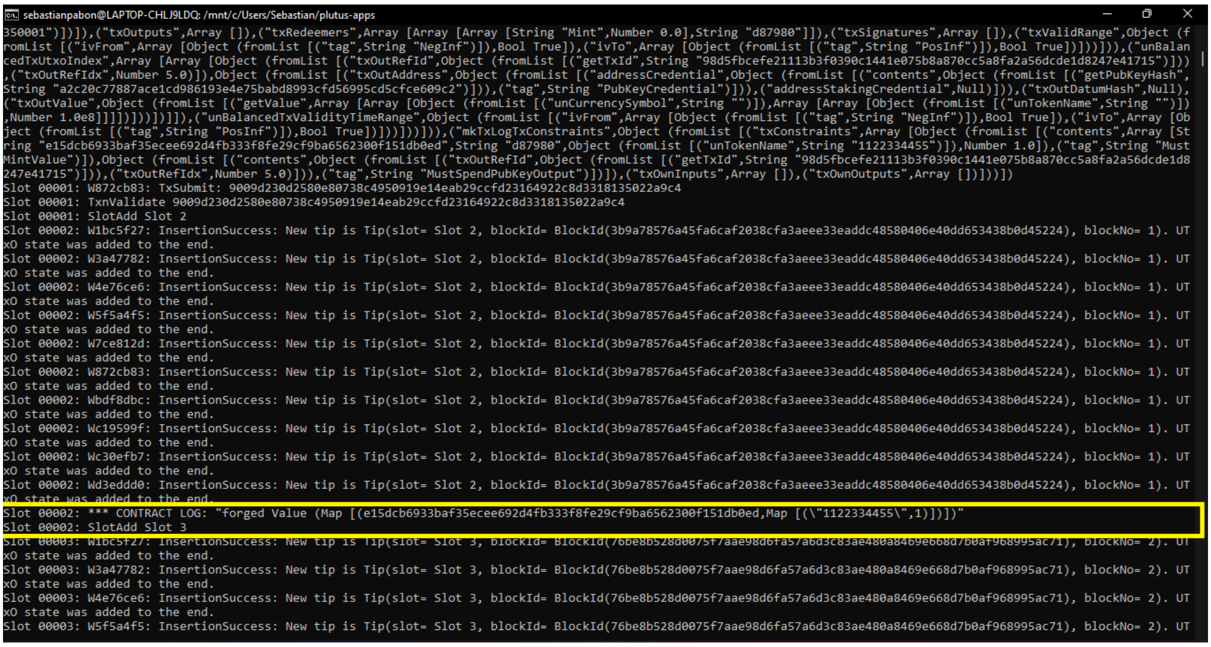
2. the AntiChildLaborReward.hs contract.

1. AntiChildLaborToken.hs

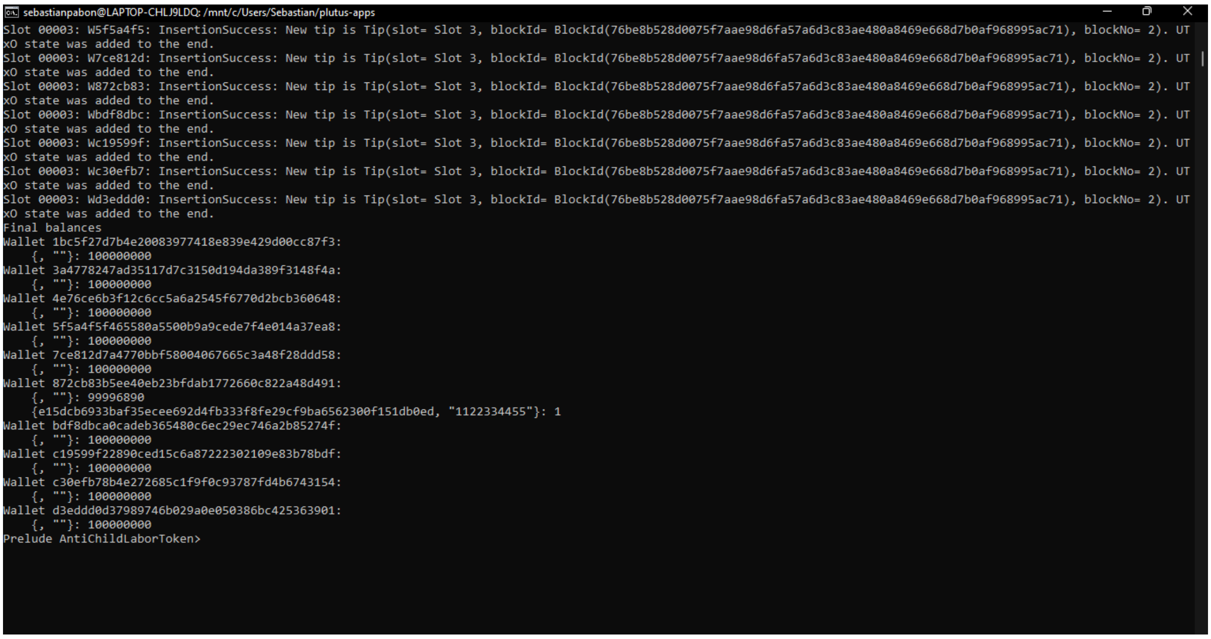
```
{-# LANGUAGE DataKinds           #-}
{-# LANGUAGE DeriveAnyClass      #-}
{-# LANGUAGE DeriveGeneric       #-}
{-# LANGUAGE FlexibleContexts    #-}
{-# LANGUAGE NoImplicitPrelude   #-}
{-# LANGUAGE OverloadedStrings   #-}
{-# LANGUAGE ScopedTypeVariables #-}
{-# LANGUAGE TemplateHaskell     #-}
{-# LANGUAGE TypeApplications    #-}
{-# LANGUAGE TypeFamilies        #-}
{-# LANGUAGE TypeOperators       #-}

module AntiChildLaborToken where

import           Control.Monad          hiding (fmap)
import           Data.Aeson             (FromJSON, ToJSON)
import qualified Data.Map               as Map
import           Data.Text              (Text)
import           Data.Void              (Void)
import           GHC.Generics           (Generic)
import           Plutus.Contract        as Contract
import           Plutus.Trace.Emulator  as Emulator
import qualified PlutusTx
import           PlutusTx.Prelude       hiding (Semigroup(..), unless)
import           Ledger                 hiding (mint, singleton)
import           Ledger.Constraints     as Constraints
import qualified Ledger.Typed.Scripts   as Scripts
import           Ledger.Value           as Value
import           Prelude                (IO, Semigroup (..), Show (..), String)
import           Text.Printf            (printf)
import           Wallet.Emulator.Wallet

{-  When the third-party auditor evaluates the supplier's compliance with the monthly objectives of the
AntiChildLabour campaign and issues a favorable rating, the Company leading the campaign proceeds to minting an
NFT
and deposit it into the supplier's authorized wallet. -}


{-# INLINABLE mkPolicy #-}
mkPolicy :: TxOutRef -> TokenName -> () -> ScriptContext -> Bool
mkPolicy oref tc () ctx = traceIfFalse "UTxO not consumed"   hasUTxO           &&
                          traceIfFalse "wrong amount minted" checkMintedAmount
  where
    info :: TxInfo
    info = scriptContextTxInfo ctx

    hasUTxO :: Bool
    hasUTxO = any (\i -> txInInfoOutRef i == oref) $ txInfoInputs info
```

```haskell
    checkMintedAmount :: Bool
    checkMintedAmount = case flattenValue (txInfoMint info) of
        [(_, tc', amt)] -> tc' == tc && amt == 1
        _               -> False

policy :: TxOutRef -> TokenName -> Scripts.MintingPolicy
policy oref tc = mkMintingPolicyScript $
    $$(PlutusTx.compile [|| \oref' tc' -> Scripts.wrapMintingPolicy $ mkPolicy oref' tc' ||])
    `PlutusTx.applyCode`
    PlutusTx.liftCode oref
    `PlutusTx.applyCode`
    PlutusTx.liftCode tc

curSymbol :: TxOutRef -> TokenName -> CurrencySymbol
curSymbol oref tc = scriptCurrencySymbol $ policy oref tc



data NFTParams = NFTParams
    { npToken   :: !TokenName
    , npAddress :: !Address
    } deriving (Generic, FromJSON, ToJSON, Show)

type NFTSchema = Endpoint "mint" NFTParams

mint :: NFTParams -> Contract w NFTSchema Text ()
mint np = do
    utxos <- utxosAt $ npAddress np
    case Map.keys utxos of
        []       -> Contract.logError @String "no utxo found"
        oref : _ -> do
            let tc      = npToken np
            let val     = Value.singleton (curSymbol oref tc) tc 1
                lookups = Constraints.mintingPolicy (policy oref tc) <> Constraints.unspentOutputs utxos
                tx      = Constraints.mustMintValue val <> Constraints.mustSpendPubKeyOutput oref
            ledgerTx <- submitTxConstraintsWith @Void lookups tx
            void $ awaitTxConfirmed $ getCardanoTxId ledgerTx
            Contract.logInfo @String $ printf "forged %s" (show val)

endpoints :: Contract () NFTSchema Text ()
endpoints = mint' >> endpoints
  where
    mint' = awaitPromise $ endpoint @"mint" mint



{- In the field "tc" the Company introduces the serial number that will be assigned to the NFT.
The serial number of the NFT will be a long integer. -}

test :: IO ()
test = runEmulatorTraceIO $ do
    let tc = "1122334455"
        w1 = knownWallet 1

    h1 <- activateContractWallet w1 endpoints

    callEndpoint @"mint" h1 $ NFTParams
        { npToken   = tc
        , npAddress = mockWalletAddress w1
        }

    void $ Emulator.waitNSlots 1
```

Result:


sebastianpabon@LAPTOP-CHLJ9LDQ: /mnt/c/Users/Sebastian/plutus-apps

```
350001")])]),("txOutputs",Array []),("txRedeemers",Array [Array [Array [String "Mint",Number 0.0],String "d87980"]]),("txSignatures",Array []),("txValidRange",Object (f
romList [("ivFrom",Array [Object (fromList [("tag",String "NegInf")]),Bool True]),("ivTo",Array [Object (fromList [("tag",String "PosInf")]),Bool True])]))])),("unBalan
cedTxUtxoIndex",Array [Array [Object (fromList [("txOutRefId",Object (fromList [("getTxId",String "98d5fbcefe21113b3f0390c1441e075b8a870cc5a8fa2a56dcde1d8247e41715")]))
,("txOutRefIdx",Number 5.0)]),Object (fromList [("txOutAddress",Object (fromList [("addressCredential",Object (fromList [("contents",Object (fromList [("getPubKeyHash",
String "a2c20c77887ace1cd986193e4e75babd8993cfd56995cd5cfce609c2")])),("tag",String "PubKeyCredential")])),("addressStakingCredential",Null)])),("txOutDatumHash",Null),
("txOutValue",Object (fromList [("getValue",Array [Array [Object (fromList [("unCurrencySymbol",String "")]),Array [Array [Object (fromList [("unTokenName",String "")])
,Number 1.0e8]]]])]))])]]),("unBalancedTxValidityTimeRange",Object (fromList [("ivFrom",Array [Object (fromList [("tag",String "NegInf")]),Bool True]),("ivTo",Array [Ob
ject (fromList [("tag",String "PosInf")]),Bool True])]))]))])),("mkTxLogTxConstraints",Object (fromList [("txConstraints",Array [Object (fromList [("contents",Array [St
ring "e15dcb6933baf35ecee692d4fb333f8fe29cf9ba6562300f151db0ed",String "d87980",Object (fromList [("unTokenName",String "1122334455")]),Number 1.0]),("tag",String "Must
MintValue")]),Object (fromList [("contents",Object (fromList [("txOutRefId",Object (fromList [("getTxId",String "98d5fbcefe21113b3f0390c1441e075b8a870cc5a8fa2a56dcde1d8
247e41715")])),("txOutRefIdx",Number 5.0)])),("tag",String "MustSpendPubKeyOutput")])]),("txOwnInputs",Array []),("txOwnOutputs",Array [])]))])
Slot 00001: W872cb83: TxSubmit: 9009d230d2580e80738c4950919e14eab29ccfd23164922c8d3318135022a9c4
Slot 00001: TxnValidate 9009d230d2580e80738c4950919e14eab29ccfd23164922c8d3318135022a9c4
Slot 00001: SlotAdd Slot 2
Slot 00002: W1bc5f27: InsertionSuccess: New tip is Tip(slot= Slot 2, blockId= BlockId(3b9a78576a45fa6caf2038cfa3aeee33eaddc48580406e40dd653438b0d45224), blockNo= 1). UT
xO state was added to the end.
Slot 00002: W3a47782: InsertionSuccess: New tip is Tip(slot= Slot 2, blockId= BlockId(3b9a78576a45fa6caf2038cfa3aeee33eaddc48580406e40dd653438b0d45224), blockNo= 1). UT
xO state was added to the end.
Slot 00002: W4e76ce6: InsertionSuccess: New tip is Tip(slot= Slot 2, blockId= BlockId(3b9a78576a45fa6caf2038cfa3aeee33eaddc48580406e40dd653438b0d45224), blockNo= 1). UT
xO state was added to the end.
Slot 00002: W5f5a4f5: InsertionSuccess: New tip is Tip(slot= Slot 2, blockId= BlockId(3b9a78576a45fa6caf2038cfa3aeee33eaddc48580406e40dd653438b0d45224), blockNo= 1). UT
xO state was added to the end.
Slot 00002: W7ce812d: InsertionSuccess: New tip is Tip(slot= Slot 2, blockId= BlockId(3b9a78576a45fa6caf2038cfa3aeee33eaddc48580406e40dd653438b0d45224), blockNo= 1). UT
xO state was added to the end.
Slot 00002: W872cb83: InsertionSuccess: New tip is Tip(slot= Slot 2, blockId= BlockId(3b9a78576a45fa6caf2038cfa3aeee33eaddc48580406e40dd653438b0d45224), blockNo= 1). UT
xO state was added to the end.
Slot 00002: Wbdf8dbc: InsertionSuccess: New tip is Tip(slot= Slot 2, blockId= BlockId(3b9a78576a45fa6caf2038cfa3aeee33eaddc48580406e40dd653438b0d45224), blockNo= 1). UT
xO state was added to the end.
Slot 00002: Wc19599f: InsertionSuccess: New tip is Tip(slot= Slot 2, blockId= BlockId(3b9a78576a45fa6caf2038cfa3aeee33eaddc48580406e40dd653438b0d45224), blockNo= 1). UT
xO state was added to the end.
Slot 00002: Wc30efb7: InsertionSuccess: New tip is Tip(slot= Slot 2, blockId= BlockId(3b9a78576a45fa6caf2038cfa3aeee33eaddc48580406e40dd653438b0d45224), blockNo= 1). UT
xO state was added to the end.
Slot 00002: Wd3eddd0: InsertionSuccess: New tip is Tip(slot= Slot 2, blockId= BlockId(3b9a78576a45fa6caf2038cfa3aeee33eaddc48580406e40dd653438b0d45224), blockNo= 1). UT
xO state was added to the end.
Slot 00002: *** CONTRACT LOG: "forged Value (Map [(e15dcb6933baf35ecee692d4fb333f8fe29cf9ba6562300f151db0ed,Map [(\"1122334455\",1)])])"
Slot 00002: SlotAdd Slot 3
Slot 00003: W1bc5f27: InsertionSuccess: New tip is Tip(slot= Slot 3, blockId= BlockId(76be8b528d0075f7aae98d6fa57a6d3c83ae480a8469e668d7b0af968995ac71), blockNo= 2). UT
xO state was added to the end.
Slot 00003: W3a47782: InsertionSuccess: New tip is Tip(slot= Slot 3, blockId= BlockId(76be8b528d0075f7aae98d6fa57a6d3c83ae480a8469e668d7b0af968995ac71), blockNo= 2). UT
xO state was added to the end.
Slot 00003: W4e76ce6: InsertionSuccess: New tip is Tip(slot= Slot 3, blockId= BlockId(76be8b528d0075f7aae98d6fa57a6d3c83ae480a8469e668d7b0af968995ac71), blockNo= 2). UT
xO state was added to the end.
Slot 00003: W5f5a4f5: InsertionSuccess: New tip is Tip(slot= Slot 3, blockId= BlockId(76be8b528d0075f7aae98d6fa57a6d3c83ae480a8469e668d7b0af968995ac71), blockNo= 2). UT
```


sebastianpabon@LAPTOP-CHLJ9LDQ: /mnt/c/Users/Sebastian/plutus-apps

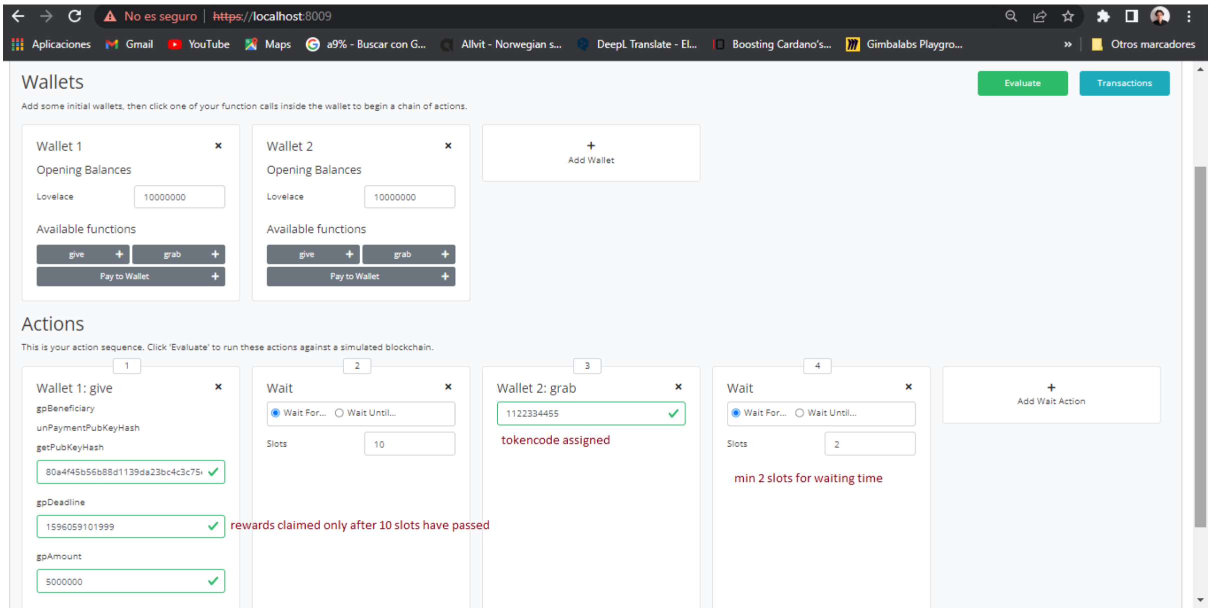
```
Slot 00003: W5f5a4f5: InsertionSuccess: New tip is Tip(slot= Slot 3, blockId= BlockId(76be8b528d0075f7aae98d6fa57a6d3c83ae480a8469e668d7b0af968995ac71), blockNo= 2). UT
xO state was added to the end.
Slot 00003: W7ce812d: InsertionSuccess: New tip is Tip(slot= Slot 3, blockId= BlockId(76be8b528d0075f7aae98d6fa57a6d3c83ae480a8469e668d7b0af968995ac71), blockNo= 2). UT
xO state was added to the end.
Slot 00003: W872cb83: InsertionSuccess: New tip is Tip(slot= Slot 3, blockId= BlockId(76be8b528d0075f7aae98d6fa57a6d3c83ae480a8469e668d7b0af968995ac71), blockNo= 2). UT
xO state was added to the end.
Slot 00003: Wbdf8dbc: InsertionSuccess: New tip is Tip(slot= Slot 3, blockId= BlockId(76be8b528d0075f7aae98d6fa57a6d3c83ae480a8469e668d7b0af968995ac71), blockNo= 2). UT
xO state was added to the end.
Slot 00003: Wc19599f: InsertionSuccess: New tip is Tip(slot= Slot 3, blockId= BlockId(76be8b528d0075f7aae98d6fa57a6d3c83ae480a8469e668d7b0af968995ac71), blockNo= 2). UT
xO state was added to the end.
Slot 00003: Wc30efb7: InsertionSuccess: New tip is Tip(slot= Slot 3, blockId= BlockId(76be8b528d0075f7aae98d6fa57a6d3c83ae480a8469e668d7b0af968995ac71), blockNo= 2). UT
xO state was added to the end.
Slot 00003: Wd3eddd0: InsertionSuccess: New tip is Tip(slot= Slot 3, blockId= BlockId(76be8b528d0075f7aae98d6fa57a6d3c83ae480a8469e668d7b0af968995ac71), blockNo= 2). UT
xO state was added to the end.
Final balances
Wallet 1bc5f27d7b4e20083977418e839e429d00cc87f3:
    {, ""}: 100000000
Wallet 3a4778247ad35117d7c3150d194da389f3148f4a:
    {, ""}: 100000000
Wallet 4e76ce6b3f12c6cc5a6a2545f6770d2bcb360648:
    {, ""}: 100000000
Wallet 5f5a4f5f465580a5500b9a9cede7f4e014a37ea8:
    {, ""}: 100000000
Wallet 7ce812d7a4770bbf58004067665c3a48f28ddd58:
    {, ""}: 100000000
Wallet 872cb83b5ee40eb23bfdab1772660c822a48d491:
    {, ""}: 99996890
    {e15dcb6933baf35ecee692d4fb333f8fe29cf9ba6562300f151db0ed, "1122334455"}: 1
Wallet bdf8dbca0cadeb365480c6ec29ec746a2b85274f:
    {, ""}: 100000000
Wallet c19599f22890ced15c6a87222302109e83b78bdf:
    {, ""}: 100000000
Wallet c30efb78b4e272685c1f9f0c93787fd4b6743154:
    {, ""}: 100000000
Wallet d3eddd0d37989746b029a0e050386bc425363901:
    {, ""}: 100000000
Prelude AntiChildLaborToken>
```

2. AntiChildLaborReward.hs

```haskell
{-# LANGUAGE DataKinds           #-}
{-# LANGUAGE DeriveAnyClass      #-}
{-# LANGUAGE DeriveGeneric       #-}
{-# LANGUAGE FlexibleContexts    #-}
{-# LANGUAGE NoImplicitPrelude   #-}
{-# LANGUAGE OverloadedStrings   #-}
{-# LANGUAGE ScopedTypeVariables #-}
{-# LANGUAGE TemplateHaskell     #-}
{-# LANGUAGE TypeApplications    #-}
{-# LANGUAGE TypeFamilies        #-}
{-# LANGUAGE TypeOperators       #-}

{-# OPTIONS_GHC -fno-warn-unused-imports #-}

module AntiChildLaborRewards where

import           Control.Monad        hiding (fmap)
import           Data.Aeson           (ToJSON, FromJSON)
import           Data.Map             as Map
import           Data.Text            (Text)
import           Data.Void            (Void)
import           GHC.Generics         (Generic)
import           Plutus.Contract
import           PlutusTx             (Data (..))
import qualified PlutusTx
import qualified PlutusTx.Builtins    as Builtins
import           PlutusTx.Prelude     hiding (Semigroup(..), unless)
import           Ledger               hiding (singleton)
import           Ledger.Constraints   (TxConstraints)
import qualified Ledger.Constraints   as Constraints
import qualified Ledger.Typed.Scripts as Scripts
import           Ledger.Ada           as Ada
import           Playground.Contract  (printJson, printSchemas, ensureKnownCurrencies, stage, ToSchema)
import           Playground.TH        (mkKnownCurrencies, mkSchemaDefinitions)
import           Playground.Types     (KnownCurrency (..))
import           Prelude              (IO, Semigroup (..), Show (..), String)
import           Text.Printf          (printf)

{- After the monthly targets met by the supplier have been recognized by the company leading the AntiChildLabour
campaign
by minting an AntiChildLabour Token, the supplier is entitled to claim an economic incentive.

The reward has to be claimed from the wallet authorized by the Company, with the serial number of the token
(called tokenCode),
and after the time limit for performing the campaign activities has passed (after the deadline).

The serial number of the token is an integer. -}

data VestingDatum = VestingDatum
    { beneficiary :: PaymentPubKeyHash
    , deadline    :: POSIXTime
    } deriving Show

PlutusTx.unstableMakeIsData ''VestingDatum

{-# INLINABLE mkValidator #-}
mkValidator :: VestingDatum -> Integer -> ScriptContext -> Bool
mkValidator dat r ctx = traceIfFalse "supplier's signature missing" signedByBeneficiary &&
                         traceIfFalse "deadline not reached" deadlineReached  &&
                         traceIfFalse "wrong tokencode" tokenCode
  where
    info :: TxInfo
    info = scriptContextTxInfo ctx

    signedByBeneficiary :: Bool
    signedByBeneficiary = txSignedBy info $ unPaymentPubKeyHash $ beneficiary dat
```

```
{- We are not looking for a race against time or a sloppily executed awareness campaign, we are looking for
suppliers to take advantage of all the time to develop an internal AntiChildLabour campaign that really raises
awareness of the problem.  Therefore, the minting of the NFT  and payment of the reward will be made after the
deadline, not before the deadline. -}




    deadlineReached :: Bool
    deadlineReached = contains (from $ deadline dat) $ txInfoValidRange info

    {-Introduce here the serial number of the token: the NFT minted by the Company for the supplier
    that accomplishes the monthly goals of the campaign. The serial is of type Integer. -}

    tokenCode :: Bool
    tokenCode = (r == 1122334455)


data Vesting
instance Scripts.ValidatorTypes Vesting where
    type instance DatumType Vesting = VestingDatum
    type instance RedeemerType Vesting = Integer

typedValidator :: Scripts.TypedValidator Vesting
typedValidator = Scripts.mkTypedValidator @Vesting
    $$(PlutusTx.compile [|| mkValidator ||])
    $$(PlutusTx.compile [|| wrap ||])
  where
    wrap = Scripts.wrapValidator @VestingDatum @Integer

validator :: Validator
validator = Scripts.validatorScript typedValidator

valHash :: Ledger.ValidatorHash
valHash = Scripts.validatorHash typedValidator

scrAddress :: Ledger.Address
scrAddress = scriptAddress validator

data GiveParams = GiveParams
    { gpBeneficiary :: !PaymentPubKeyHash
    , gpDeadline    :: !POSIXTime
    , gpAmount      :: !Integer
    } deriving (Generic, ToJSON, FromJSON, ToSchema)




type VestingSchema =
            Endpoint "give" GiveParams
        .\/ Endpoint "grab" Integer

give :: AsContractError e => GiveParams -> Contract w s e ()
give gp = do
    let dat = VestingDatum
                { beneficiary = gpBeneficiary gp
                , deadline    = gpDeadline gp
                }
        tx  = Constraints.mustPayToTheScript dat $ Ada.lovelaceValueOf $ gpAmount gp
    ledgerTx <- submitTxConstraints typedValidator tx
    void $ awaitTxConfirmed $ getCardanoTxId ledgerTx
    logInfo @String $ printf "made a reward of %d lovelace to %s with deadline %s"
        (gpAmount gp)
        (show $ gpBeneficiary gp)
        (show $ gpDeadline gp)

grab :: forall w s e. AsContractError e => Integer -> Contract w s e ()
grab r = do
    now   <- currentTime
    pkh   <- ownPaymentPubKeyHash
    utxos <- Map.filter (isSuitable pkh now) <$> utxosAt scrAddress
    if Map.null utxos
        then logInfo @String $ "no reward available"
        else do
            let orefs   = fst <$> Map.toList utxos
                lookups = Constraints.unspentOutputs utxos  <>
                          Constraints.otherScript validator
                tx :: TxConstraints Void Void
                  tx      = mconcat [Constraints.mustSpendScriptOutput oref $ Redeemer $ Builtins.mkI r | oref <-
orefs] <>
                          Constraints.mustValidateIn (from now)

            ledgerTx <- submitTxConstraintsWith @Void lookups tx
            void $ awaitTxConfirmed $ getCardanoTxId ledgerTx
            logInfo @String $ "collected reward"
  where
    isSuitable :: PaymentPubKeyHash -> POSIXTime -> ChainIndexTxOut -> Bool
    isSuitable pkh now o = case _ciTxOutDatum o of
        Left _          -> False
        Right (Datum e) -> case PlutusTx.fromBuiltinData e of
            Nothing -> False
            Just d  -> beneficiary d == pkh && deadline d <= now

endpoints :: Contract () VestingSchema Text ()
endpoints = awaitPromise (give' `select` grab') >> endpoints
  where
    give' = endpoint @"give" give
    grab' = endpoint @"grab" grab

mkSchemaDefinitions ''VestingSchema

mkKnownCurrencies []
```

Result:



Wallets

Add some initial wallets, then click one of your function calls inside the wallet to begin a chain of actions.

Wallet 1 ×
Opening Balances
Lovelace 10000000
Available functions

Wallet 2 ×
Opening Balances
Lovelace 10000000
Available functions

Actions

This is your action sequence. Click 'Evaluate' to run these actions against a simulated blockchain.

1
Wallet 1: give ×
gpBeneficiary
unPaymentPubKeyHash
getPubKeyHash
80a4f45b56b88d1139da23bc4c3c75 ✓
gpDeadline
1596059101999 ✓
gpAmount
5000000 ✓

rewards claimed only after 10 slots have passed

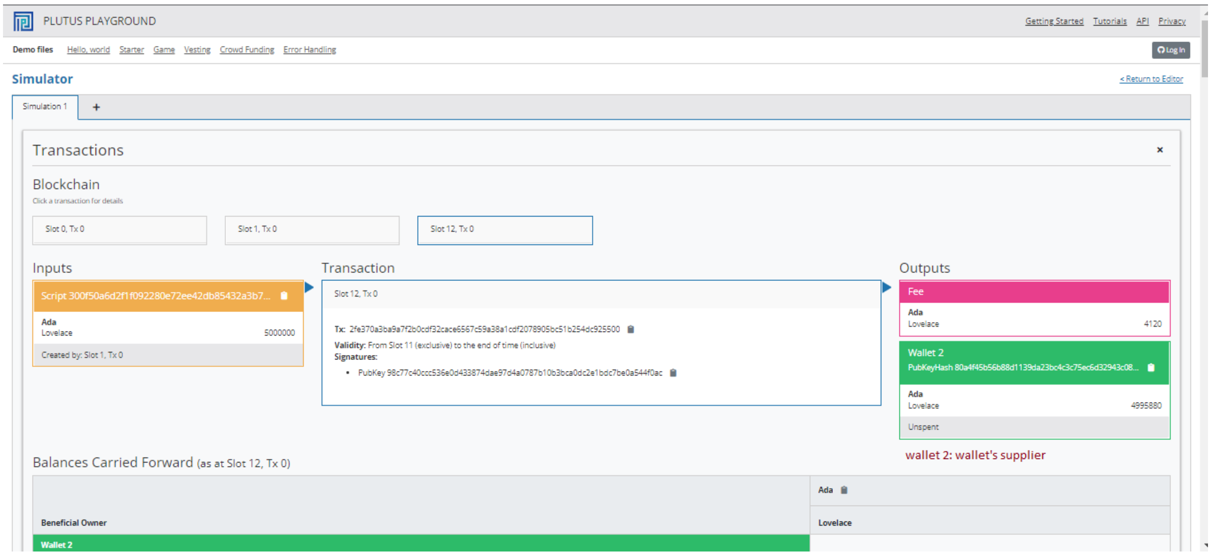
2
Wait ×
● Wait For... ○ Wait Until...
Slots 10

3
Wallet 2: grab ×
1122334455 ✓

tokencode assigned

4
Wait ×
● Wait For... ○ Wait Until...
Slots 2

min 2 slots for waiting time



Simulator

Transactions

Blockchain

Click a transaction for details

Slot 0, Tx 0 | Slot 1, Tx 0 | Slot 12, Tx 0

Inputs

Script 300f50a6d2f1f092280e72ee42db85432a3b7...
Ada
Lovelace 5000000
Created by: Slot 1, Tx 0

Transaction

Slot 12, Tx 0

Tx: 2fe370a3ba9a7f2b0cdf32cace6567c59a38a1cdf2078905bc51b254dc925500
Validity: From Slot 11 (exclusive) to the end of time (inclusive)
Signatures:
- PubKey 98c77c40ccc536e0d433874dae97d4a0787b10b3bca0dc2e1bdc7be0a544f0ac

Outputs

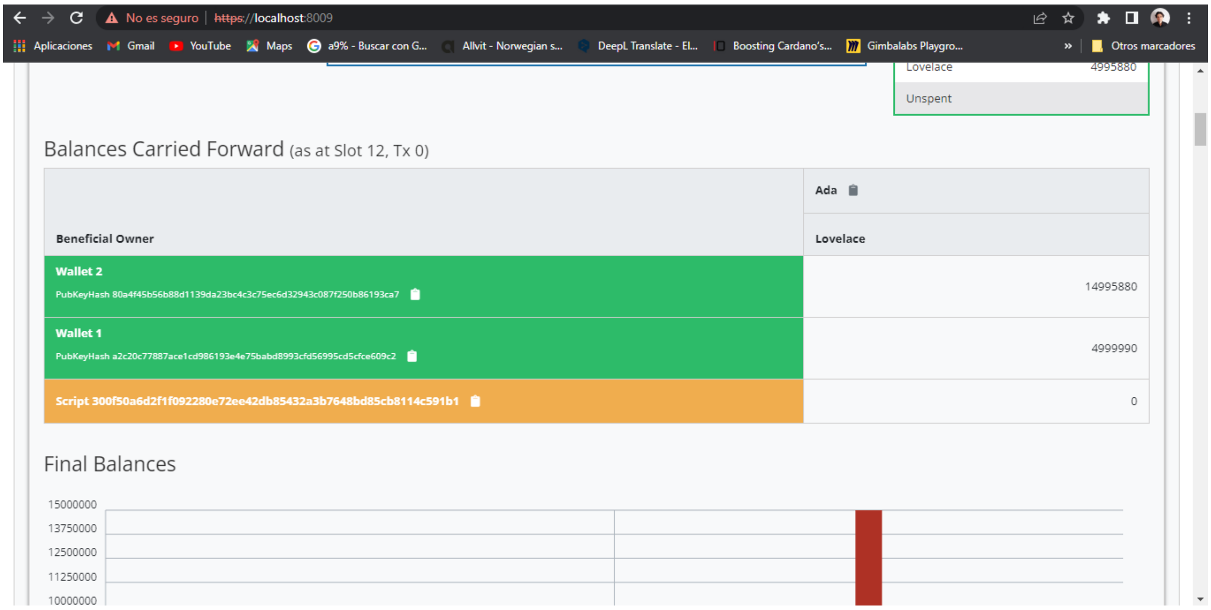
Fee
Ada
Lovelace 4120

Wallet 2
PubKeyHash 80a4f45b56b88d1139da23bc4c3c75ec6d32943c08...
Ada
Lovelace 4995880
Unspent

wallet 2: wallet's supplier

Balances Carried Forward (as at Slot 12, Tx 0)

	Ada
Beneficial Owner	Lovelace
Wallet 2	

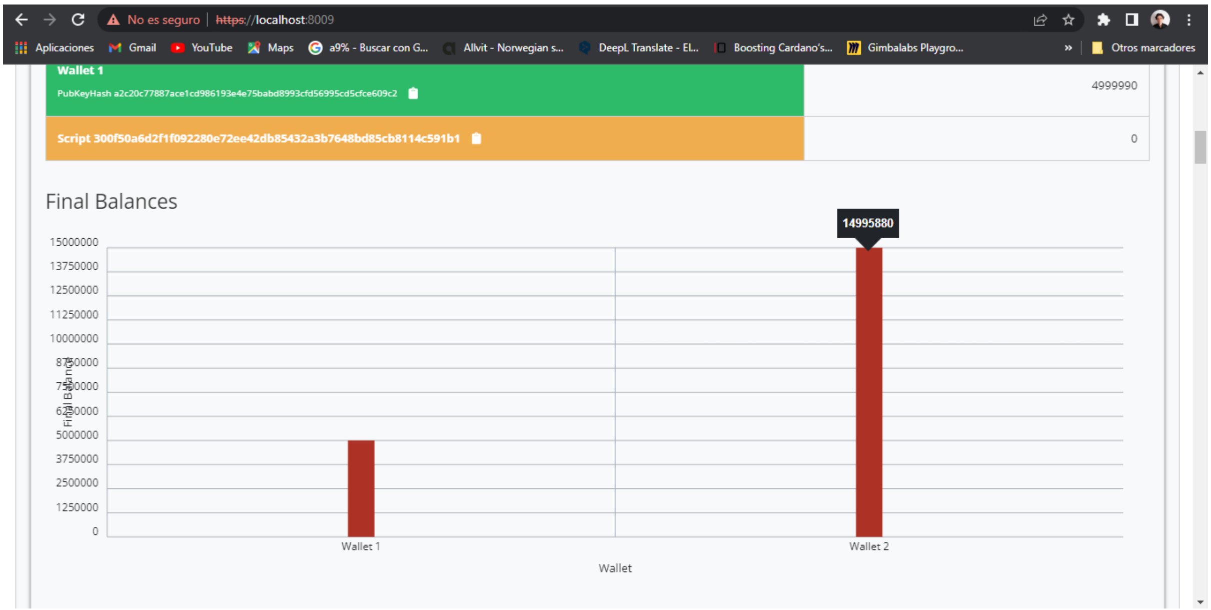


Lovelace 4995880
Unspent

Balances Carried Forward (as at Slot 12, Tx 0)

	Ada
Beneficial Owner	**Lovelace**
Wallet 2 PubKeyHash 80a4f45b56b88d1139da23bc4c3c75ec6d32943c087f250b86193ca7	14995880
Wallet 1 PubKeyHash a2c20c77887ace1cd986193e4e75babd8993cfd56995cd5cfce609c2	4999990
Script 300f50a6d2f1f092280e72ee42db85432a3b7648bd85cb8114c591b1	0

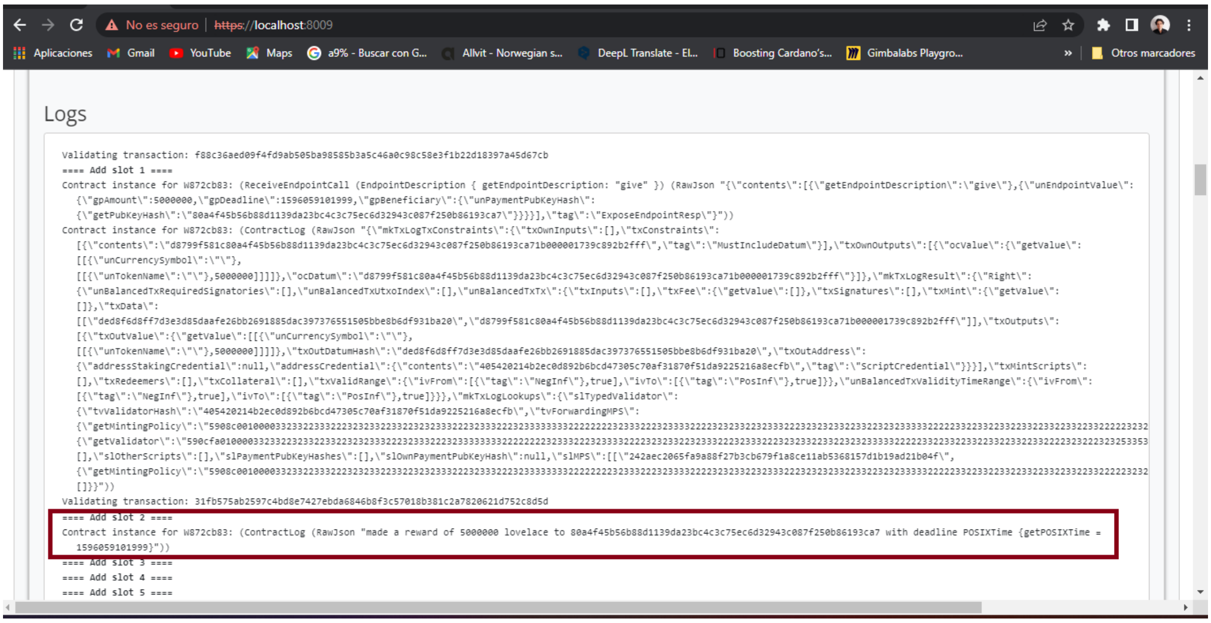
Final Balances



Wallet 1 PubKeyHash a2c20c77887ace1cd986193e4e75babd8993cfd56995cd5cfce609c2	4999990
Script 300f50a6d2f1f092280e72ee42db85432a3b7648bd85cb8114c591b1	0

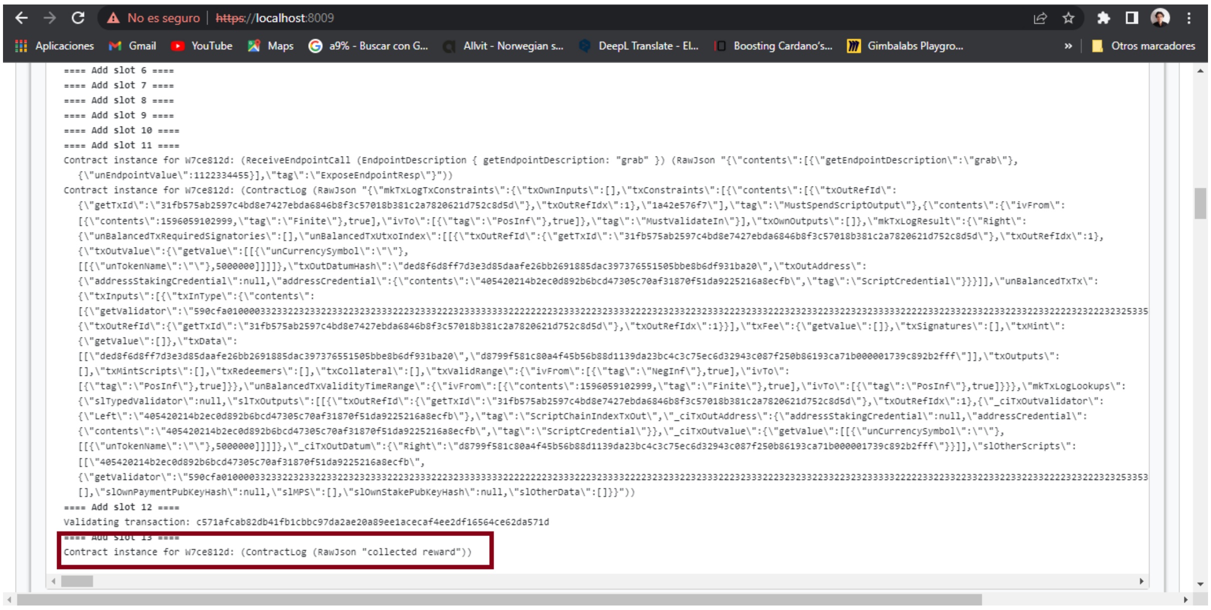
Final Balances

Students: Sebastian Pabon, Raphael Rotondari



Logs

```
Validating transaction: f88c36aed09f4fd9ab505ba98585b3a5c46a0c98c58e3f1b22d18397a45d67cb
==== Add slot 1 ====
Contract instance for W872cb83: (ReceiveEndpointCall (EndpointDescription { getEndpointDescription: "give" }) (RawJson "{\"contents\":[{\"getEndpointDescription\":\"give\"},{\"unEndpointValue\":
  {\"gpAmount\":5000000,\"gpDeadline\":1596059101999,\"gpBeneficiary\":{\"unPaymentPubKeyHash\":
  {\"getPubKeyHash\":\"80a4f45b56b88d1139da23bc4c3c75ec6d32943c087f250b86193ca7\"}}}}],\"tag\":\"ExposeEndpointResp\"}"))
Contract instance for W872cb83: (ContractLog (RawJson "{\"mkTxLogTxConstraints\":{\"txOwnInputs\":[],\"txConstraints\":
  [{\"contents\":\"d8799f581c80a4f45b56b88d1139da23bc4c3c75ec6d32943c087f250b86193ca71b000001739c892b2fff\",\"tag\":\"MustIncludeDatum\"}],\"txOwnOutputs\":[{\"ocValue\":{\"getValue\":
  [[{\"unCurrencySymbol\":\"\"},
  [[{\"unTokenName\":\"\"},5000000]]]]},\"ocDatum\":\"d8799f581c80a4f45b56b88d1139da23bc4c3c75ec6d32943c087f250b86193ca71b000001739c892b2fff\"}]},\"mkTxLogResult\":{\"Right\":
  {\"unBalancedTxRequiredSignatories\":[],\"unBalancedTxUtxoIndex\":[],\"unBalancedTxTx\":{\"txInputs\":[],\"txFee\":{\"getValue\":[]},\"txSignatures\":[],\"txMint\":{\"getValue\":
  []},\"txData\":
  [[\"ded8f6d8ff7d3e3d85daafe26bb2691885dac397376551505bbe8b6df931ba20\",\"d8799f581c80a4f45b56b88d1139da23bc4c3c75ec6d32943c087f250b86193ca71b000001739c892b2fff\"]],\"txOutputs\":
  [{\"txOutValue\":{\"getValue\":[[{\"unCurrencySymbol\":\"\"},
  [[{\"unTokenName\":\"\"},5000000]]]]},\"txOutDatumHash\":\"ded8f6d8ff7d3e3d85daafe26bb2691885dac397376551505bbe8b6df931ba20\",\"txOutAddress\":
  {\"addressStakingCredential\":null,\"addressCredential\":{\"contents\":\"405420214b2ec0d892b6bcd47305c70af31870f51da9225216a8ecfb\",\"tag\":\"ScriptCredential\"}}}],\"txMintScripts\":
  [],\"txRedeemers\":[],\"txCollateral\":[],\"txValidRange\":{\"ivFrom\":[{\"tag\":\"NegInf\"},true],\"ivTo\":[{\"tag\":\"PosInf\"},true]}},\"unBalancedTxValidityTimeRange\":{\"ivFrom\":
  [{\"tag\":\"NegInf\"},true],\"ivTo\":[{\"tag\":\"PosInf\"},true]}}},\"mkTxLogLookups\":{\"slTypedValidator\":
  {\"tvValidatorHash\":\"405420214b2ec0d892b6bcd47305c70af31870f51da9225216a8ecfb\",\"tvForwardingMPS\":
  {\"getMintingPolicy\":\"5908c0010000332332233322232323322333223232333222323332223233333333222222223233322232333322232323322323332223232323322332232323333222223322332233223322332233223322222223232
  {\"getValidator\":\"590cfa0100003323322323322332232323332223233322232333333333222222223233322232333322232323322323332223233322232323322332232323333222223322332233223322332233222223232222323253353
  [],\"slOtherScripts\":[],\"slPaymentPubKeyHashes\":[],\"slOwnPaymentPubKeyHash\":null,\"slMPS\":[[\"242aec2065fa9a88f27b3cb679f1a8ce11ab5368157d1b19ad21b04f\",
  {\"getMintingPolicy\":\"5908c0010000332332233322232323322333223232333222323332223233333333222222223233322232333322232323322323332223232323322332232323333222223322332233223322332233223322222223232
  []}}"))
Validating transaction: 31fb575ab2597c4bd8e7427ebda6846b8f3c57018b381c2a7820621d752c8d5d
==== Add slot 2 ====
Contract instance for W872cb83: (ContractLog (RawJson "made a reward of 5000000 lovelace to 80a4f45b56b88d1139da23bc4c3c75ec6d32943c087f250b86193ca7 with deadline POSIXTime {getPOSIXTime =
  1596059101999}"))
==== Add slot 3 ====
==== Add slot 4 ====
==== Add slot 5 ====
```



```
==== Add slot 6 ====
==== Add slot 7 ====
==== Add slot 8 ====
==== Add slot 9 ====
==== Add slot 10 ====
==== Add slot 11 ====
Contract instance for W7ce812d: (ReceiveEndpointCall (EndpointDescription { getEndpointDescription: "grab" }) (RawJson "{\"contents\":[{\"getEndpointDescription\":\"grab\"},
  {\"unEndpointValue\":1122334455}],\"tag\":\"ExposeEndpointResp\"}"))
Contract instance for W7ce812d: (ContractLog (RawJson "{\"mkTxLogTxConstraints\":{\"txOwnInputs\":[],\"txConstraints\":[{\"contents\":[{\"txOutRefId\":
  {\"getTxId\":\"31fb575ab2597c4bd8e7427ebda6846b8f3c57018b381c2a7820621d752c8d5d\"},\"txOutRefIdx\":1},\"1a42e576f7\"],\"tag\":\"MustSpendScriptOutput\"},{\"contents\":{\"ivFrom\":
  [{\"contents\":1596059102999,\"tag\":\"Finite\"},true],\"ivTo\":[{\"tag\":\"PosInf\"},true]},\"tag\":\"MustValidateIn\"}],\"txOwnOutputs\":[]},\"mkTxLogResult\":{\"Right\":
  {\"unBalancedTxRequiredSignatories\":[],\"unBalancedTxUtxoIndex\":[[{\"txOutRefId\":{\"getTxId\":\"31fb575ab2597c4bd8e7427ebda6846b8f3c57018b381c2a7820621d752c8d5d\"},\"txOutRefIdx\":1},
  {\"txOutValue\":{\"getValue\":[[{\"unCurrencySymbol\":\"\"},
  [[{\"unTokenName\":\"\"},5000000]]]]},\"txOutDatumHash\":\"ded8f6d8ff7d3e3d85daafe26bb2691885dac397376551505bbe8b6df931ba20\",\"txOutAddress\":
  {\"addressStakingCredential\":null,\"addressCredential\":{\"contents\":\"405420214b2ec0d892b6bcd47305c70af31870f51da9225216a8ecfb\",\"tag\":\"ScriptCredential\"}}}]],\"unBalancedTxTx\":
  {\"txInputs\":[{\"txInType\":{\"contents\":
  [{\"getValidator\":\"590cfa0100003323322323322332232323332223233322232333333333222222223233322232333322232323322323332223233322232323322332232323333222223322332233223322332233222223232222323253353
  {\"txOutRefId\":{\"getTxId\":\"31fb575ab2597c4bd8e7427ebda6846b8f3c57018b381c2a7820621d752c8d5d\"},\"txOutRefIdx\":1}}],\"txFee\":{\"getValue\":[]},\"txSignatures\":[],\"txMint\":
  {\"getValue\":[]},\"txData\":
  [[\"ded8f6d8ff7d3e3d85daafe26bb2691885dac397376551505bbe8b6df931ba20\",\"d8799f581c80a4f45b56b88d1139da23bc4c3c75ec6d32943c087f250b86193ca71b000001739c892b2fff\"]],\"txOutputs\":
  [],\"txMintScripts\":[],\"txRedeemers\":[],\"txCollateral\":[],\"txValidRange\":{\"ivFrom\":[{\"tag\":\"NegInf\"},true],\"ivTo\":
  [{\"tag\":\"PosInf\"},true]}},\"unBalancedTxValidityTimeRange\":{\"ivFrom\":[{\"contents\":1596059102999,\"tag\":\"Finite\"},true],\"ivTo\":[{\"tag\":\"PosInf\"},true]}}},\"mkTxLogLookups\":
  {\"slTypedValidator\":null,\"slTxOutputs\":[[{\"txOutRefId\":{\"getTxId\":\"31fb575ab2597c4bd8e7427ebda6846b8f3c57018b381c2a7820621d752c8d5d\"},\"txOutRefIdx\":1},{\"_ciTxOutValidator\":
  {\"Left\":\"405420214b2ec0d892b6bcd47305c70af31870f51da9225216a8ecfb\"},\"tag\":\"ScriptChainIndexTxOut\",\"_ciTxOutAddress\":{\"addressStakingCredential\":null,\"addressCredential\":
  {\"contents\":\"405420214b2ec0d892b6bcd47305c70af31870f51da9225216a8ecfb\",\"tag\":\"ScriptCredential\"}},\"_ciTxOutValue\":{\"getValue\":[[{\"unCurrencySymbol\":\"\"},
  [[{\"unTokenName\":\"\"},5000000]]]]},\"_ciTxOutDatum\":{\"Right\":\"d8799f581c80a4f45b56b88d1139da23bc4c3c75ec6d32943c087f250b86193ca71b000001739c892b2fff\"}}]],\"slOtherScripts\":
  [[\"405420214b2ec0d892b6bcd47305c70af31870f51da9225216a8ecfb\",
  {\"getValidator\":\"590cfa0100003323322323322332232323332223233322232333333333222222223233322232333322232323322323332223233322232323322332232323333222223322332233223322332233222223232222323253353
  [],\"slOwnPaymentPubKeyHash\":null,\"slMPS\":[],\"slOwnStakePubKeyHash\":null,\"slOtherData\":[]}}"))
==== Add slot 12 ====
Validating transaction: c571afcab82db41fb1cbbc97da2ae20a89ee1acecaf4ee2df16564ce62da571d
==== Add slot 13 ====
Contract instance for W7ce812d: (ContractLog (RawJson "collected reward"))
```

Bibliography:

Eliminating and Preventing Child Labour: Checkpoints for Companies. International Labour Organization. 2016. https://www.ilo.org/wcmsp5/groups/public/@dgreports/@dcomm/@publ/documents/publication/wcms_456960.pdf

Report CSR RISK CHECK: COLOMBIA Coffee and coffee substitutes. MVO NEDERLAND & INTERNATIONAL CSR. 06 May 2022. mvorisicochecker.nl