# **Table of Contents**

1 cadenaDeBloques_RPava	4
1.1 Capa de Negocio	4
1.1.1 Diagrama de Capa de Negocio diagram	4
1.1.2 Nuevo Bloque	5
1.1.3 Roles Y Actores	5
1.1.3.1 Cliente	6
1.1.3.2 Minero	6
1.1.4 Servicios del Negocio	7
1.1.5 Transacción	7
1.2 Diagramas Estructurales	8
1.2.1 1. Diagrama de paquetes	8
1.2.1.1 1. Diagrama de paquetes diagram	8
1.2.2 2. Diagrama de Clases	9
1.2.2.1 2. Diagrama de Clases diagram	9
1.2.2.2 cadena	9
1.2.2.2.1 3.1 Diagrama detallado cadena diagram	10
1.2.2.2.2 Bloque	10
1.2.2.2.3 CadenaBloques	12
1.2.2.2.4 NodoMinador	13
1.2.2.2.5 PoolMinado	14
1.2.2.2.6 Transaccion	16
1.2.2.2.7 pruebasUnidadBloque	17
1.2.2.2.8 pruebasUnidadCadena	17
1.2.2.2.9 pruebasUnidadNodoMineria	18
1.2.2.2.10 pruebasUnidadPoolMinado	18
1.2.2.2.11 pruebasUnidadTransaccion	18
1.2.2.3 cliente	20
1.2.2.3.1 3.2 Diagrama detallado cliente diagram	20
1.2.2.3.2 Usuario	20
1.2.2.3.3 pruebasUnidadUsuario	21
1.2.2.4 simulador	23
1.2.2.4.1 3.3 Diagrama detallado simulador diagram	23
1.2.2.4.2 Simulador	
1.2.2.4.3 pruebasUnidadSimulador	25
1.2.3 4. Diagrama de Objetos	26
1.2.3.1 4. Diagrama de Objetos diagram	26
1.2.4 5. Diagrama de componentes	27
1.2.4.1 5. Diagrama de componentes diagram	27
1.2.5 6. Diagrama de despliegue	
1.2.5.1 6. Diagrama de despliegue diagram	20
1.2.5.2 Nodes	
1.2.5.2.1 Nodes diagram	28
1.2.5.2.2 Clients	20
1.2.5.2.2.1 Clients diagram	29
1.2.5.2.3 Devices	30
1.2.5.2.3.1 Devices diagram	30
1.2.5.2.4 Servers	31

1.2.5.2.4.1 Servers diagram	31
1.2.5.3 Artifacts	32
1.2.5.3.1 Artifacts diagram	32
1.2.5.4 Topology	33
1.2.5.4.1 Network diagram	33
1.3 Diagramas de comportamiento	34
1.3.1 8. Diagramas de casos de uso	34
1.3.1.1 8. Diagramas de casos de uso diagram	34
1.3.1.2 Actores	34
1.3.1.2.1 Actores diagram	35
1.3.1.3 Primary Use Cases	36
1.3.1.3.1 Primary Use Cases diagram	36
1.3.2 9. Diagramas de Secuencia	37
1.3.2.1 9.1 Diagrama de Secuencia pruebasUnidadUsuario diagram	37
1.3.2.2 9.2 Diagramas de Secuencia pruebasUnidadSimulador diagram	38
1.3.2.3 9.3 Diagramas de Secuencia pruebasUnidadTransaccion diagram	39
1.3.3 10. Diagramas de Comunicación	41
1.3.3.1 10.1 Diagramas de Comunicación Usuario diagram	41
1.3.3.2 10.2 Diagramas de Comunicación pruebasUnidadSimulador diagram	41
1.3.3.3 10.3 Diagramas de Comunicación pruebaUnidadTrasaccion diagram	42
1.4 Extendidos	43
1.4.1 15. Requisitos de requerimientos	43
1.4.1.1 15. Requisitos requerimientos diagram	43
1.4.2 16. Metamodelo	45
1.4.2.1 16. Metamodelo diagram	45

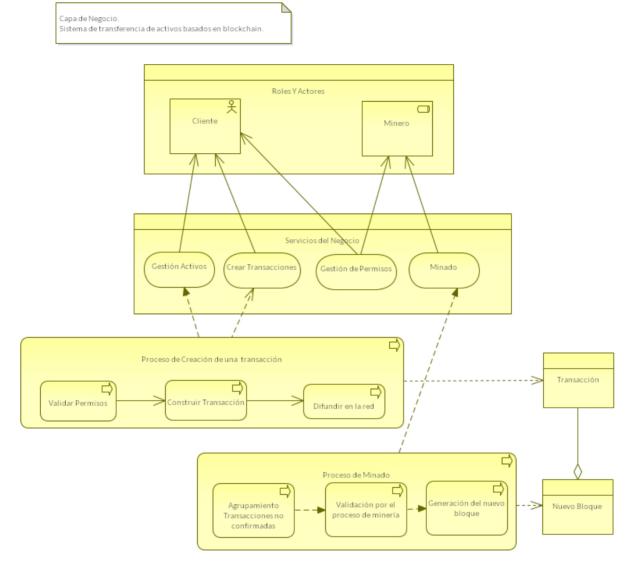


Figure 1: Diagrama de Capa de Negocio

# 1.1.2Nuevo Bloque

BusinessObject «ArchiMate\_BusinessObject» in package 'Capa de Negocio'

Nuevo Bloque Version 1.0 Phase 1.0 Proposed blockchain created on 31/05/2018. Last modified 31/05/2018



# 1.1.3Roles Y Actores

BusinessObject «ArchiMate\_BusinessObject» in package 'Capa de Negocio'

Roles Y Actores
Version 1.0 Phase 1.0 Proposed
blockchain created on 28/05/2018. Last modified 28/05/2018

#### **ELEMENTS OWNED BY Roles Y Actores**

■ Cliente : BusinessActor «ArchiMate BusinessActor»

Es poseedor de los activos de información o valor que se gestiona en la blockchain. Está habilitado para transferir activos a otro cliente dentro del sistema. Si dispone de los privilegios de administrador puede asignar y revocar permisos a otros clientes.

Minero: BusinessRole «ArchiMate\_BusinessRole»

Este actor es responsable de validar las transacciones de los clientes ejecutando el algoritmo de consenso definido. El bloque génesis de la cadena es generado por un usuario con privilegios.

## 1.1.3.1 Cliente

BusinessActor «ArchiMate\_BusinessActor» owned by 'Roles Y Actores', in package 'Capa de Negocio'

Es poseedor de los activos de información o valor que se gestiona en la blockchain. Está habilitado para transferir activos a otro cliente dentro del sistema. Si dispone de los privilegios de administrador puede asignar y revocar permisos a otros clientes.

Cliente
Version 1.0 Phase 1.0 Proposed
ncapey created on 16/05/2018. Last modified 28/05/2018

ASSOCIATIONS	
Serving (direction: Source -> Destination) «ArchiMate_Serving»	
Source: Public (BusinessService) Gestión Activos «ArchiMate_BusinessService»	Target: Public (BusinessActor) Cliente «ArchiMate_BusinessActor»
Serving (direction: Source -> Destination) «ArchiMate_Serving»	
Source: Public (BusinessService) Crear Transacciones «ArchiMate_BusinessService»	Target: Public (BusinessActor) Cliente «ArchiMate_BusinessActor»
Serving (direction: Source -> Destination) «ArchiMate_Serving»	
Source: Public (BusinessService) Gestión de Permisos «ArchiMate_BusinessService»	Target: Public (BusinessActor) Cliente «ArchiMate_BusinessActor»

## 1.1.3.2 Minero

BusinessRole «ArchiMate\_BusinessRole» owned by 'Roles Y Actores', in package 'Capa de Negocio'

Este actor es responsable de validar las transacciones de los clientes ejecutando el algoritmo de consenso definido. El bloque génesis de la cadena es generado por un usuario con privilegios.

Minero

Version 1.0 Phase 1.0 Proposed ncapey created on 16/05/2018. Last modified 28/05/2018

#### **CONSTRAINTS**

% Invariant. El usuario dispone de la propiedad de un activo de valor o de información para su transferencia

[ Proposed, Weight is 0. ]

#### **ASSOCIATIONS**

Serving (direction: Source -> Destination) «ArchiMate Serving»

Source: Public (BusinessService) Minado Target: Public (BusinessRole) Minero «ArchiMate\_BusinessService» «ArchiMate\_BusinessRole»

Serving (direction: Source -> Destination) «ArchiMate\_Serving»

Source: Public (BusinessService) Gestión de Permisos Target: Public (BusinessRole) Minero «ArchiMate\_BusinessService» «ArchiMate\_BusinessRole»

# 1.1.4Servicios del Negocio

BusinessObject «ArchiMate\_BusinessObject» in package 'Capa de Negocio'

Servicios del Negocio Version 1.0 Phase 1.0 Proposed blockchain created on 28/05/2018. Last modified 31/05/2018

# 1.1.5Transacción

BusinessObject «ArchiMate\_BusinessObject» in package 'Capa de Negocio'

Registro de la transferencia de valor, el cual contiene hash del origen y del destino y los datos del objeto a transferir

Transacción Version 1.0 Phase 1.0 Proposed blockchain created on 31/05/2018. Last modified 31/05/2018

## CONNECTORS

Access «ArchiMate\_Access» Source -> Destination

From: Proceso de Creación de una transacción: BusinessProcess, Public

To: Transacción : BusinessObject, Public

# 1.2 Diagramas Estructurales

Package in package 'cadenaDeBloques\_RPava'

Diagramas Estructurales
Version 1.0 Phase 1.0 Proposed
ENVY created on 09/07/2018. Last modified 09/07/2018
Alias

# 1.2.11. Diagrama de paquetes

Package in package 'Diagramas Estructurales'

1. Diagrama de paquetes Version Phase 1.0 Proposed ENVY created on 09/07/2018. Last modified 09/07/2018 Alias

# 1.2.1.1 1. Diagrama de paquetes diagram

Class diagram in package '1. Diagrama de paquetes'

1. Diagrama de paquetes Version 1.0 ENVY created on 09/07/2018. Last modified 09/07/2018

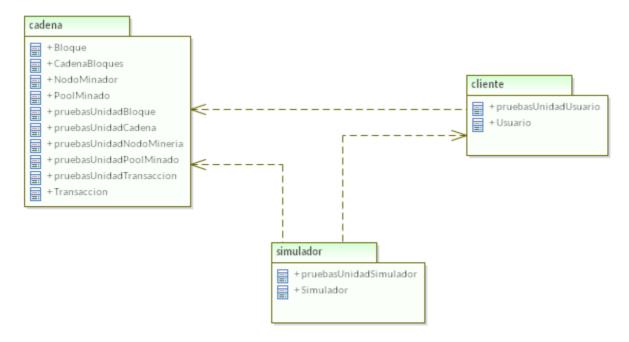


Figure 2: 1. Diagrama de paquetes

# 1.2.22. Diagrama de Clases

Package in package 'Diagramas Estructurales'

2. Diagrama de Clases Version 1.0 Phase 1.0 Proposed ENVY created on 09/07/2018. Last modified 09/07/2018 Alias

# 1.2.2.1 2. Diagrama de Clases diagram

Class diagram in package '2. Diagrama de Clases'

2. Diagrama de Clases Version 1.0 ENVY created on 09/07/2018. Last modified 09/07/2018

El diagrama de clases está compuestos por tres grandes paquetes:

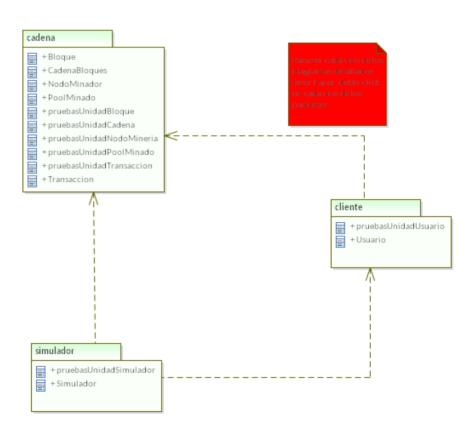


Figure 3: 2. Diagrama de Clases

## 1.2.2.2 cadena

Package in package '2. Diagrama de Clases'

cadena Version 1.0 Phase 1.0 Proposed blockchain created on 17/06/2018. Last modified 09/07/2018 Alias

## 1.2.2.2.1 3.1 Diagrama detallado cadena diagram

Class diagram in package 'cadena'

3.1 Diagrama detallado cadena Version 1.0 blockchain created on 17/06/2018. Last modified 09/07/2018

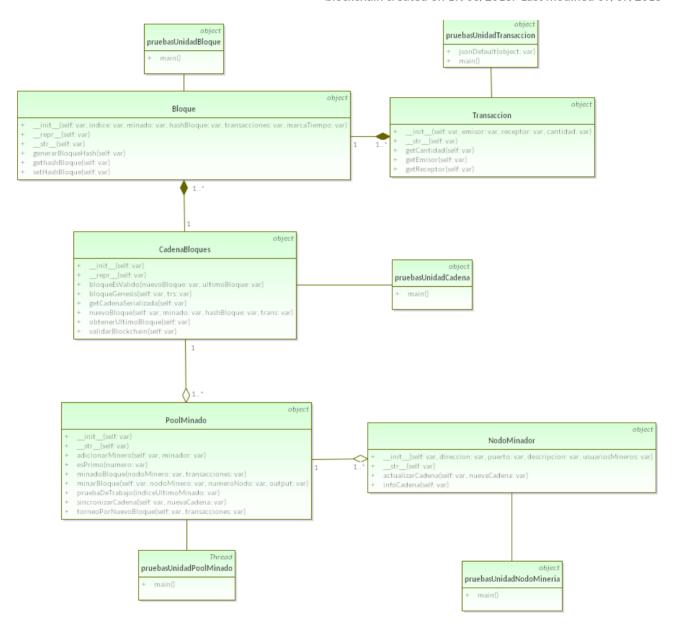


Figure 4: 3.1 Diagrama detallado cadena

## 1.2.2.2.2 Bloque

Class in package 'cadena'

Bloque Version 1.0 Phase 1.0 Proposed blockchain created on 17/06/2018. Last modified 17/06/2018

Extends object

#### **OUTGOING STRUCTURAL RELATIONSHIPS**

Aggregation from Bloque to Transaccion

[ Direction is 'Source -> Destination'. ]

#### **INCOMING STRUCTURAL RELATIONSHIPS**

Aggregation from CadenaBloques to Bloque

[ Direction is 'Source -> Destination'. ]

#### **ASSOCIATIONS**

Association (direction: Unspecified)

Source: Public (Class) pruebasUnidadBloque Target: Public (Class) Bloque

#### **OPERATIONS**

🎍 \_\_init\_\_ (self : var , indice : var , minado : var , hashBloque : var , transacciones : var , marcaTiempo : var ) : Public

Inicialización del bloque :param indice: Valor entero que representa la posición de cada bloque en la cadena :param minado: Nðmero entero usado en el proceso de minerÃa :param hashBloque: hash del bloque actual en la cadena :param transacciones: Conjunto de transacciones del bloque :param marcaTiempo: Instante de tiempo de creación del nuevo bloque

[ Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False. ]

\_\_repr\_\_ (self : var ) : Public

[ Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False. ]

\_\_str\_\_ (self : var ) : Public

Genera un string con la información del bloque :return: representación en String del bloque

[ Is static False. Is abstract False. Is return array False. Is guery False. Is synchronized False. ]

generarBloqueHash (self : var ) : Public

Genera Hash la información del bloque :return: Hash construido a partir de la información del bloque

Properties:

decorators = @staticmethod

[ Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False. ]

gethashBloque (self : var ) : Public

Obtiene el valor actual del hash del bloque :return: Hash del bloque

Properties:

ea\_guid = {93941509-FCA2-4d45-B5EB-9F78D678FC4B}

[ Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False. ]

🔖 setHashBloque (self : var ) : Public

[ Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False. ]

OPERATIONS

## 1.2.2.2.3 CadenaBloques

Class in package 'cadena'

CadenaBloques
Version 1.0 Phase 1.0 Proposed
blockchain created on 17/06/2018. Last modified 17/06/2018
Extends object

#### **OUTGOING STRUCTURAL RELATIONSHIPS**

Aggregation from CadenaBloques to PoolMinado

[ Direction is 'Source -> Destination'. ]

Aggregation from CadenaBloques to Bloque

[ Direction is 'Source -> Destination'. ]

#### **ASSOCIATIONS**

Association (direction: Unspecified)

Source: Public (Class) CadenaBloques

Target: Public (Class) pruebasUnidadCadena

#### **OPERATIONS**

\_\_init\_\_ (self : var ) : Public

Inicialización de la blockchain :param \_cadena: Estructura de datos tipo array para almacenar la cadena de bloques.

[ Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False. ]

\_\_repr\_\_ (self : var ) : Public

[ Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False. ]

bloqueEsValido (nuevoBloque : var , ultimoBloque : var ) : Public

Verifica bajo cuatro reglas básicas la validez de un Bloque Regla 1: Secuencia ordenada de bloques por valor del Ãndice Regla 2: Secuencia ordenada de bloques por fecha de creación Regla 3: Hash diferente entre el ðltimo bloque de la cadena y el nuevo bloque Regla 4: Bloque correctamente minado

@return True si el bloque es válido, False en otro caso

Properties:

decorators = @staticmethod

[ Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False. ]

bloqueGenesis (self : var , trs : var ) : Public

Crea el primer bloque en la cadena. Es el bloque seminal.

[ Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False. ]

getCadenaSerializada (self : var ) : Public

#### **OPERATIONS**

Serializa la cadena de bloques en un documento JSON :return conjunto de bloques en formato JSON

[ Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False. ]

nuevoBloque (self : var , minado : var , hashBloque : var , trans : var ) : Public

Construye e inserta el siguiente bloque de la cadena :param minado Método de minado del bloque :param hashBloque Hash generado para el bloque :return Bloque nuevo bloque generado

[ Is static False. Is abstract False. Is return array False. Is guery False. Is synchronized False. ]

obtenerUltimoBloque (self : var ) : Public

Devuelve una referencia al  $\tilde{A}^{\circ}$ ltimo nodo de la blockchain @return  $\tilde{A}^{\circ}$ ltimo nodo

[ Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False. ]

validarBlockchain (self : var ) : Public

Verifica la integridad de la cadena de bloques. para este procedimiento se valida que el Ãndice del siguiente bloque sea menor en una unidad del bloque anterior y que el hash de cada bloque sea correcto segðn la polÃtica de minado establecida, nðmeros primos en este caso.

@return True si la cadena esta bien formada, False en otro caso

[ Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False. ]

## 1.2.2.2.4 NodoMinador

Class in package 'cadena'

NodoMinador Version 1.0 Phase 1.0 Proposed blockchain created on 17/06/2018. Last modified 17/06/2018 Extends object

### **INCOMING STRUCTURAL RELATIONSHIPS**

→ Aggregation from PoolMinado to NodoMinador

[ Direction is 'Source -> Destination'. ]

#### ASSOCIATIONS

Association (direction: Unspecified)

Source: Public (Class) pruebasUnidadNodoMineria Target: Public (Class) NodoMinador

#### **OPERATIONS**

🌳 \_\_init\_\_ (self : var , direccion : var , puerto : var , descripcion : var , usuariosMineros : var ) : Public

Creación de un nodo para la minerÃa de blockchain :param \_direccción: Dirección ip del nodo minero :param \_puerto: puerto TCP/UDP donde se publica el servicio :param \_descripcion: información adicional del nodo minador :param \_fechaCreacion: fecha de adición del nodo a la red :param \_cadenaBloques: Recibe una copia de la cadena de bloques :param \_hashRateNodo : Capacidad de resolución de hash por segundo del nodo :param \_usuariosMineros: Conjunto de usuarios participantes en el nodo minero

#### **OPERATIONS**

[ Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False. ]

\_\_str\_\_ (self : var ) : Public

Genera un string con la información de un nodo de minerÃ- a :return: representación en String del nodo

[ Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False. ]

actualizarCadena (self : var , nuevaCadena : var ) : Public

El proceso de minerÃ- a requiere que el nodo actualice la blockchain :param: Cadena de bloques actualizada

[ Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False. ]

infoCadena (self : var ) : Public

Genera un string con la informaci $\tilde{A}^3$ n b $\tilde{A}_i$ sica de la cadena de bloques :return: una cadena con el reporte del total de nodos de la cadena

[ Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False. ]

## 1.2.2.2.5 PoolMinado

Class in package 'cadena'

PoolMinado Version 1.0 Phase 1.0 Proposed blockchain created on 17/06/2018. Last modified 17/06/2018 Extends object

#### **OUTGOING STRUCTURAL RELATIONSHIPS**

Aggregation from PoolMinado to NodoMinador

[ Direction is 'Source -> Destination'. ]

#### **INCOMING STRUCTURAL RELATIONSHIPS**

→ Aggregation from CadenaBloques to PoolMinado

[ Direction is 'Source -> Destination'. ]

#### **ASSOCIATIONS**

Association (direction: Unspecified)

Source: Public (Class) PoolMinado Target: Public (Class) pruebasUnidadPoolMinado

## **OPERATIONS**

\_\_init\_\_ (self : var ) : Public

Inicialización del pool de minerÃa :param \_nodos: conjunto de nodos mineros :param \_hasRate Simula la capacidad de computo del pool de mineria :param \_recompensaPorBloque cantidad que se recompensa a los nodos que resuelvan el problema :param \_dificultad valor usado para simular el incremento de dificultad del proceso de minado

[ Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False. ]

#### **OPFRATIONS**

\_\_str\_\_ (self : var ) : Public

Genera un string con la información del pool de minerÃ- a :return: representación en String del nodo

[ Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False. ]

adicionarMinero (self : var , minador : var ) : Public

Agrega un nodo minero al conjunto de nodos minadores de la blockchain. Si el nodo adicionado aumenta el hashrate del pool ylo iguala o supera el 60% de la dificultad, esta ðltima se incrementa en un 20%

@param minador: Nuevo nodo minero

@return True si el nodo de mineria se puede adicionar, False en otro caso

[ Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False. ]

esPrimo (numero : var ) : Public

Determina si un nð mero es primo

@param numero, Nðmero que necesitamos determinar si es primo

@return True si numero es primo, False en otro caso

Properties:

decorators = @staticmethod

[ Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False. ]

minadoBloque (nodoMinero : var , transacciones : var ) : Public

Realiza el proceso de minado de un nuevo bloque en la cadena

@param nodoMinero que realiza el intento de minar el bloque

@param transacciones: Conjunto de Transacciones del nuevo bloque

Properties:

decorators = @staticmethod

[ Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False. ]

🗣 minarBloque (self : var , nodoMinero : var , numeroNodo : var , output : var ) : Public

Algoritmo que realiza el minado de un bloque para un nodo en especifico.

@param nodoMinero, Nodo minero que realizarÃ; la prueba de trabajo

@param numeroNodo, identificador entero del nodo minero en el pool de minerÃa

@param output,  $Par\tilde{A}_i$ metro de salida donde se conserva la informaci $\tilde{A}^3$ n de la prueba de trabajo ejecutada por el i $\tilde{A}$ ©simo nodo minero.

@return minadoNuevo Valor de minado obtenido para el nuevo bloque

 $[\ \ \text{Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.}\ ]$ 

pruebaDeTrabajo (indiceUltimoMinado : var ) : Public

Algoritmo de minado por prueba de trabajo. En esta versión se implementa una prueba basada en nðmeros primos, en la cual el valor de minado del bloque actual y del ðltimo bloque de la cadena debe ser un nðmero primo.

@param indiceUltimoMinado, Valor generado por la minerÃa para el ðltimo bloque de la cadena

@return minadoNuevo Valor de minado obtenido para el nuevo bloque

Properties:

decorators = @staticmethod

[ Is static False. Is abstract False. Is return array False. Is guery False. Is synchronized False. ]

sincronizarCadena (self : var , nuevaCadena : var ) : Public

#### **OPERATIONS**

Realiza una sincronización de la cadena actualizada a todos los mineros

@param nuevaCadena, Cadena con el nuevo bloque

[ Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False. ]

torneoPorNuevoBloque (self : var , transacciones : var ) : Public

Realiza un torneo para la adjudicaci $\tilde{A}^3$ n del nuevo bloque a un nodo minero. El nodo ganador ser $\tilde{A}_i$  el que realice mayor trabajo.

@param transacciones, Bloque de transacciones que tendrÃ; el nuevo bloque de la cadena

[ Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False. ]

## 1.2.2.2.6 Transaccion

Class in package 'cadena'

Transaccion
Version 1.0 Phase 1.0 Proposed
blockchain created on 17/06/2018. Last modified 17/06/2018
Extends object

#### INCOMING STRUCTURAL RELATIONSHIPS

→ Aggregation from Bloque to Transaccion

[ Direction is 'Source -> Destination'. ]

## ASSOCIATIONS

Association (direction: Unspecified)

Source: Public (Class) Transaccion Target: Public (Class) pruebasUnidadTransaccion

#### **OPERATIONS**

\_\_init\_\_ (self : var , emisor : var , receptor : var , cantidad : var ) : Public

Creación de una transaccióna :param emisor: Hash de quién envÃa :param receptor: Hash de quién recibe :param cantidad: valor de la transacción

[ Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False. ]

\_\_str\_\_ (self : var ) : Public

Genera un string con la información de la transacción :return: representación en String de una transacción

[ Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False. ]

getCantidad (self : var ) : Public

Obtiene el valor de la transacción :return: Valor de la transacción

[ Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False. ]

getEmisor (self : var ) : Public

Obtiene el valor actual del hash del emisor :return: Hash del emisor

#### **OPERATIONS**

[ Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False. ]

getReceptor (self : var ) : Public

Obtiene el valor actual del hash del receptor :return: Hash del receptor

[ Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False. ]

#### 1.2.2.2.7 pruebasUnidadBloque

Class in package 'cadena'

pruebasUnidadBloque Version 1.0 Phase 1.0 Proposed blockchain created on 17/06/2018. Last modified 17/06/2018 Extends object

#### **ASSOCIATIONS**



Association (direction: Unspecified)

Source: Public (Class) pruebasUnidadBloque

Target: Public (Class) Bloque

#### **OPERATIONS**



main (): Public

[ Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False. ]

#### 1.2.2.2.8 pruebasUnidadCadena

Class in package 'cadena'

pruebasUnidadCadena Version 1.0 Phase 1.0 Proposed blockchain created on 17/06/2018. Last modified 17/06/2018 Extends object

## **ASSOCIATIONS**



Association (direction: Unspecified)

Source: Public (Class) CadenaBloques

Target: Public (Class) pruebasUnidadCadena

### **OPERATIONS**



main () : Public

[ Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False. ]

#### 1.2.2.2.9 pruebasUnidadNodoMineria

Class in package 'cadena'

pruebasUnidadNodoMineria Version 1.0 Phase 1.0 Proposed blockchain created on 17/06/2018. Last modified 17/06/2018 Extends object

#### **ASSOCIATIONS**

Association (direction: Unspecified)

Source: Public (Class) pruebasUnidadNodoMineria

Target: Public (Class) NodoMinador

#### **OPERATIONS**



main (): Public

[ Is static False. Is abstract False. Is return array False. Is guery False. Is synchronized False. ]

#### 1.2.2.2.10 pruebasUnidadPoolMinado

Class in package 'cadena'

pruebasUnidadPoolMinado Version 1.0 Phase 1.0 Proposed blockchain created on 17/06/2018. Last modified 17/06/2018 **Extends Thread** 

#### **ASSOCIATIONS**



Association (direction: Unspecified)

Source: Public (Class) PoolMinado

Target: Public (Class) pruebasUnidadPoolMinado

## **OPERATIONS**



main (): Public

[ Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False. ]

#### 1.2.2.2.11 pruebasUnidadTransaccion

Class in package 'cadena'

pruebasUnidadTransaccion Version 1.0 Phase 1.0 Proposed blockchain created on 17/06/2018. Last modified 17/06/2018 Extends object

### **ASSOCIATIONS**



Association (direction: Unspecified)

## **ASSOCIATIONS**

Target: Public (Class) pruebasUnidadTransaccion Source: Public (Class) Transaccion

#### **OPERATIONS**

jsonDefault (object : var ) : Public

Properties:

decorators = @staticmethod

[ Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False. ]

main (): Public

[ Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False. ]

## 1.2.2.3 cliente

Package in package '2. Diagrama de Clases'

cliente
Version 1.0 Phase 1.0 Proposed
blockchain created on 17/06/2018. Last modified 09/07/2018
Alias

## 1.2.2.3.1 3.2 Diagrama detallado cliente diagram

Class diagram in package 'cliente'

3.2 Diagrama detallado cliente Version 1.0 blockchain created on 17/06/2018. Last modified 09/07/2018

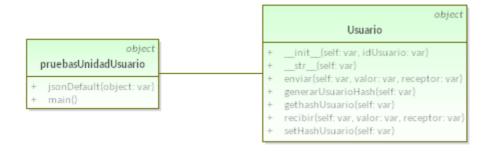


Figure 5: 3.2 Diagrama detallado cliente

#### 1.2.2.3.2 Usuario

Class in package 'cliente'

Usuario Version 1.0 Phase 1.0 Proposed blockchain created on 17/06/2018. Last modified 17/06/2018 Extends object





#### **OPERATIONS**

Genera un string con la información del usuario :return: representación en String del usuario

[ Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False. ]

enviar (self : var , valor : var , receptor : var ) : Public

:param receptor: hash del usuario receptor de la transferencia :param valor: Cantidad que se desea transferir :return: La nueva transacci $\tilde{A}^3$ n que ser $\tilde{A}_i$  enviada a validaci $\tilde{A}^3$ n

[ Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False. ]

generarUsuarioHash (self : var ) : Public

Genera Hash la información del bloque :return: Hash construido a partir de la información del usuario

[ Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False. ]

gethashUsuario (self : var ) : Public

Obtiene el valor actual del hash del usuario :return: Hash del bloque

[ Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False. ]

recibir (self : var , valor : var , receptor : var ) : Public

:param receptor: hash del usuario receptor de la transferencia :param valor: Cantidad que se ha recibido

[ Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False. ]

setHashUsuario (self : var ) : Public

[ Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False. ]

## 1.2.2.3.3 pruebasUnidadUsuario

Class in package 'cliente'

pruebasUnidadUsuario Version 1.0 Phase 1.0 Proposed blockchain created on 17/06/2018. Last modified 17/06/2018 Extends object

### **ASSOCIATIONS**

Association (direction: Unspecified)

Source: Public (Class) pruebasUnidadUsuario Target: Public (Class) Usuario

### **OPERATIONS**

👂 jsonDefault (object : var ) : Public

Properties:

decorators = @staticmethod

 $[\ \ \text{Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.}\ ]$ 

main (): Public

# OPERATIONS [ Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False. ]

## 1.2.2.4 simulador

Package in package '2. Diagrama de Clases'

simulador Version 1.0 Phase 1.0 Proposed ENVY created on 09/07/2018. Last modified 09/07/2018 Alias

## 1.2.2.4.1 3.3 Diagrama detallado simulador diagram

Class diagram in package 'simulador'

3.3 Diagrama detallado simulador Version 1.0 ENVY created on 09/07/2018. Last modified 09/07/2018

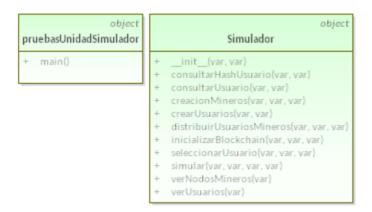


Figure 6: 3.3 Diagrama detallado simulador

### 1.2.2.4.2 **Simulador**

Class in package 'simulador'

Simulador
Version 1.0 Phase 1.0 Proposed
ENVY created on 09/07/2018. Last modified 09/07/2018
Extends object

# 

#### **OPERATIONS**

ea\_guid = {6DF6F9B1-B625-40dc-9600-93CD25CCB976}

[ Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False. ]

onsultarUsuario (self : var , hashUsuario : var ) : Public

Consulta la informaci $\tilde{A}^3$ n de un usuario dado su hash :param hashUsuario LLave del diccionario :return usuario que corresponde con la llave dada

Properties:

ea guid = {0C9F8C53-0611-4260-8A77-9CA712EBE8CC}

[ Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False. ]

🗣 creacionMineros (self : var , cantidadMineros : var , usuariosNodo : var ) : Public

Adiciona los nodos mineros a la blockchain. Cada nodo minero tendrá asociado un grupo de usuarios, quienés recibirán recompensa por su esfuerzo de minado. :param cantidadMineros, Total de nodos Mineros para la cadena :param usuariosNodo, Distribución de usuarios mineros para cada nodo Minero

[ Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False. ]

crearUsuarios (self : var , cantidadUsuarios : var ) : Public

Inicializa el conjunto de usuarios de la cadena de bloques :param cantidad Usuarios: Conjunto inicial de Usuarios en la Simulaci $\tilde{A}^3$ n

Properties:

ea\_guid = {C4974C29-34C2-4af6-9DBC-C4ACDB022CA7}

[ Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False. ]

👂 distribuirUsuariosMineros (self : var , maxUsuarios : var , maxMineros : var ) : Public

Permite distribuir de manera uniforme los usuarios a los nodos mineros :param maxMineros: Conjunto de Nodos mineros de la blockchain :param maxUsuarios: Total de usuarios en el sistema :return arreglo con los usuarios distribuidos para cada nodo minero

[ Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False. ]

inicializarBlockchain (self : var , maxUsuarios : var , cantidadMineros : var ) : Public

Creación del bloque gé nesis de la cadena y asignación de nodos mineros :param cantidadMineros, Total de nodos Mineros para la cadena :param maxUsuarios: Total de usuarios en el sistema

Properties:

ea\_guid = {5D0CE9D1-D370-42fc-905D-5D37B6B5A7F8}

 $[\ \ \text{Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.}\ ]$ 

👂 seleccionarUsuario (self : var , cantidadPosibleReceptores : var , diferentea : var ) : Public

print("Torneo " + str(i) + " " + str(segundos) + " microseconds");

[ Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False. ]

📦 simular (self : var , maxTransaccionesPorBloque : var , totalBloquesCaena : var , cantidadUsuarios : var ) : Public

Realiza una simulación de evolución de la blockchain con los parámetros dados. :param cantidadUsuarios, Total de usuarios activos en la cadena :param cantidadMineros, Total de nodos Mineros para la cadena :param usuariosNodo, Distribución de usuarios mineros para cada nodo Minero

Properties:

ea\_guid = {3CF6CA39-BA5C-4115-A9F3-BE0A7FC40F00}

#### **OPERATIONS**

[ Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False. ]

🗣 verNodosMineros (self : var ) : Public

Consulta el total de nodos mineros registrados en la cadena de bloques :return pool de mineros de la blockchain

Properties:

ea\_guid = {A73B3E02-C1F3-480a-8755-F203EAC14FBC}

[ Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False. ]

verUsuarios (self : var ) : Public

Consulta el total de usuarios registrados en un momento dado en la cadena de bloques :return RepresentaciÃ<sup>3</sup>n en string del conjunto de Usuarios en la Simulación

Properties:

ea\_guid = {419E85C7-3FDF-4cb5-BF1A-E3AE0B5A3495}

[ Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False. ]

#### 1.2.2.4.3 pruebasUnidadSimulador

Class in package 'simulador'

pruebasUnidadSimulador Version 1.0 Phase 1.0 Proposed ENVY created on 09/07/2018. Last modified 09/07/2018 Extends object

#### **OPERATIONS**

main (): Public

[ Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False. ]

# 1.2.34. Diagrama de Objetos

Package in package 'Diagramas Estructurales'

4. Diagrama de Objetos Version 1.0 Phase 1.0 Proposed ENVY created on 09/07/2018. Last modified 09/07/2018 Alias

# 1.2.3.1 4. Diagrama de Objetos diagram

Object diagram in package '4. Diagrama de Objetos'

4. Diagrama de Objetos Version 1.0 ENVY created on 09/07/2018. Last modified 09/07/2018

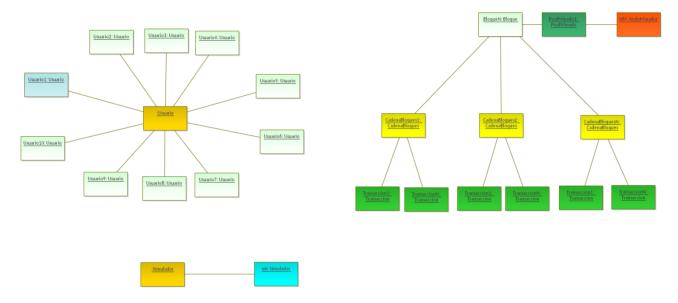


Figure 7: 4. Diagrama de Objetos

# 1.2.45. Diagrama de componentes

Package in package 'Diagramas Estructurales'

5. Diagrama de componentes Version 1.0 Phase 1.0 Proposed ENVY created on 09/07/2018. Last modified 09/07/2018 Alias

# 1.2.4.1 5. Diagrama de componentes diagram

Component diagram in package '5. Diagrama de componentes'

5. Diagrama de componentes Version 1.0 ENVY created on 09/07/2018. Last modified 10/07/2018

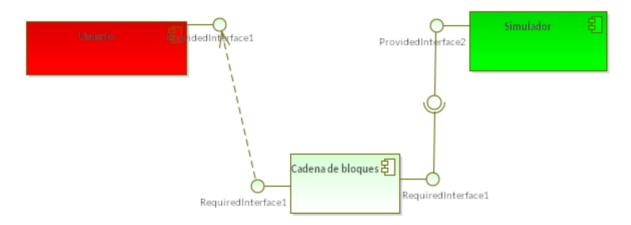


Figure 8: 5. Diagrama de componentes

# 1.2.56. Diagrama de despliegue

Package in package 'Diagramas Estructurales'

6. Diagrama de despliegue Version Phase 1.0 Proposed ENVY created on 10/07/2018. Last modified 10/07/2018 Alias

# 1.2.5.1 6. Diagrama de despliegue diagram

Deployment diagram in package '6. Diagrama de despliegue'

6. Diagrama de despliegue Version 1.0 ENVY created on 10/07/2018. Last modified 10/07/2018

Nodes The Deployment model describes how and where the system will The Nodes package contains models of all be deployed. the processors, devices and hardware +Clients components that are required to be +Devices Physical machines, devices and processors are reflected as nodes, deployed or utilized in the final + Servers and the internal construction can be depicted by embedding implementation of the modeled system. additional nodes or artifacts. Artifacts, such as executables, are allocated to nodes to model the Artifacts system's run-time configuration. The allocation is guided by the use The Artifacts package contains models of all of deployment specifications. software and related components that will The physical locations, deployment of artifacts and connectivity be deployed onto the Nodes. between nodes of the final deployed system is depicted in the Topology package Topology The Topology package models the Read about Deployment Modeling connectivity, networks, physical locations and spatial characteristics of the system to View Further Examples be implemented.

Figure 9: 6. Diagrama de despliegue

## 1.2.5.2 **Nodes**

Package in package '6. Diagrama de despliegue'

Nodes

Version 1.0 Phase 1.0 Proposed

ENVY created on 10/07/2018. Last modified 10/07/2018

Alias

## 1.2.5.2.1 Nodes diagram

Deployment diagram in package 'Nodes'

Nodes Version 1.0

blockchain created on 10/07/2018. Last modified 10/07/2018

Figure 10: Nodes

## 1.2.5.2.2 Clients

Package in package 'Nodes'

Clients

Version 1.0 Phase 1.0 Proposed

ENVY created on 10/07/2018. Last modified 10/07/2018

Alias

# 1.2.5.2.2.1 Clients diagram

Deployment diagram in package 'Clients'

Clients
Version 1.0
blockchain created on 10/07/2018. Last modified 10/07/2018

Figure 11: Clients

## 1.2.5.2.3 **Devices**

Package in package 'Nodes'

Devices

Version 1.0 Phase 1.0 Proposed

ENVY created on 10/07/2018. Last modified 10/07/2018

Alias

# 1.2.5.2.3.1 Devices diagram

Deployment diagram in package 'Devices'

Devices
Version 1.0
blockchain created on 10/07/2018. Last modified 10/07/2018

Figure 12: Devices

## 1.2.5.2.4 Servers

Package in package 'Nodes'

Servers

Version 1.0 Phase 1.0 Proposed
ENVY created on 10/07/2018. Last modified 10/07/2018

Alias

# 1.2.5.2.4.1 Servers diagram

Deployment diagram in package 'Servers'

Servers

Version 1.0
blockchain created on 10/07/2018. Last modified 10/07/2018

Figure 13: Servers

# 1.2.5.3 Artifacts

Package in package '6. Diagrama de despliegue'

Artifacts
Version 1.0 Phase 1.0 Proposed
ENVY created on 10/07/2018. Last modified 10/07/2018
Alias

# 1.2.5.3.1 Artifacts diagram

Deployment diagram in package 'Artifacts'

Artifacts
Version 1.0
blockchain created on 10/07/2018. Last modified 10/07/2018

Figure 14: Artifacts

# **1.2.5.4** Topology

Package in package '6. Diagrama de despliegue'

Topology Version 1.0 Phase 1.0 Proposed ENVY created on 10/07/2018. Last modified 10/07/2018 Alias

# 1.2.5.4.1 Network diagram

Deployment diagram in package 'Topology'

Network

Version 1.0
blockchain created on 10/07/2018. Last modified 10/07/2018

Figure 15: Network

# 1.3 Diagramas de comportamiento

Package in package 'cadenaDeBloques\_RPava'

Diagramas de comportamiento Version 1.0 Phase 1.0 Proposed ENVY created on 10/07/2018. Last modified 10/07/2018 Alias

# 1.3.18. Diagramas de casos de uso

Package in package 'Diagramas de comportamiento'

8. Diagramas de casos de uso Version Phase 1.0 Proposed ENVY created on 10/07/2018. Last modified 10/07/2018 Alias

## 1.3.1.1 8. Diagramas de casos de uso diagram

Use Case diagram in package '8. Diagramas de casos de uso'

8. Diagramas de casos de uso Version 1.0 blockchain created on 31/05/2018. Last modified 10/07/2018



Figure 16: 8. Diagramas de casos de uso

## **1.3.1.2** Actores

Package in package '8. Diagramas de casos de uso'

Actores

Version 1.0 Phase 1.0 Proposed
blockchain created on 31/05/2018. Last modified 10/07/2018

Alias

# 1.3.1.2.1 Actores diagram

Use Case diagram in package 'Actores'

Actores
Version 1.0
blockchain created on 31/05/2018. Last modified 10/07/2018





Figure 17: Actores

# 1.3.1.3 Primary Use Cases

Package in package '8. Diagramas de casos de uso'

Primary Use Cases Version 1.0 Phase 1.0 Proposed blockchain created on 31/05/2018. Last modified 10/07/2018 Alias

## 1.3.1.3.1 Primary Use Cases diagram

Use Case diagram in package 'Primary Use Cases'

Primary Use Cases

Version 1.0
blockchain created on 31/05/2018. Last modified 10/07/2018

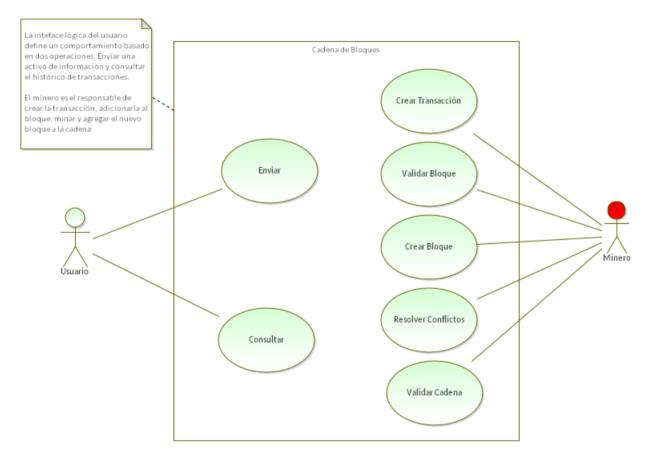


Figure 18: Primary Use Cases

# 1.3.29. Diagramas de Secuencia

Package in package 'Diagramas de comportamiento'

9. Diagramas de Secuencia Version 1.0 Phase 1.0 Proposed ENVY created on 10/07/2018. Last modified 10/07/2018 Alias

# 1.3.2.1 9.1 Diagrama de Secuencia pruebasUnidadUsuario diagram

Interaction diagram in package '9. Diagramas de Secuencia'

9.1 Diagrama de Secuencia pruebasUnidadUsuario Version 1.0 ENVY created on 10/07/2018. Last modified 10/07/2018

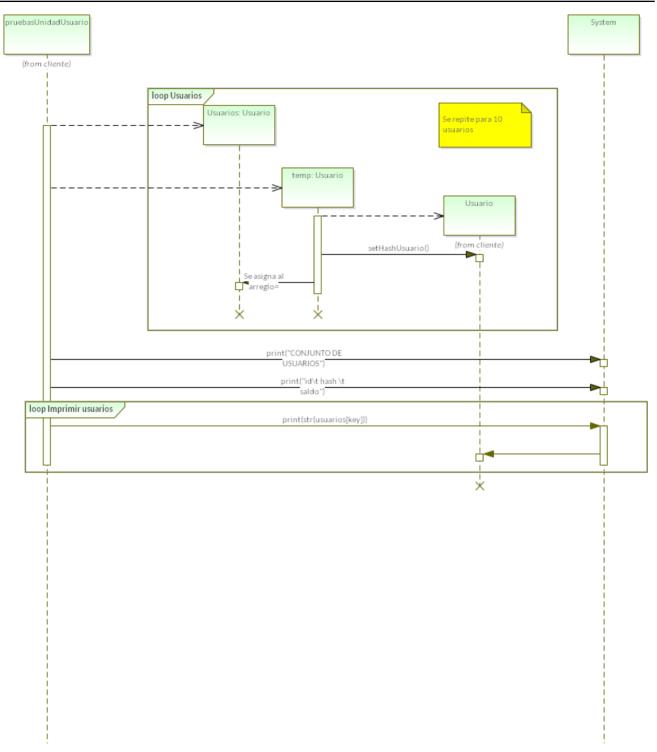


Figure 19: 9.1 Diagrama de Secuencia pruebasUnidadUsuario

# **1.3.2.2 9.2 Diagramas de Secuencia pruebasUnidadSimulador diagram**

Interaction diagram in package '9. Diagramas de Secuencia'

9.2 Diagramas de Secuencia pruebasUnidadSimulador Version 1.0 ENVY created on 10/07/2018. Last modified 10/07/2018

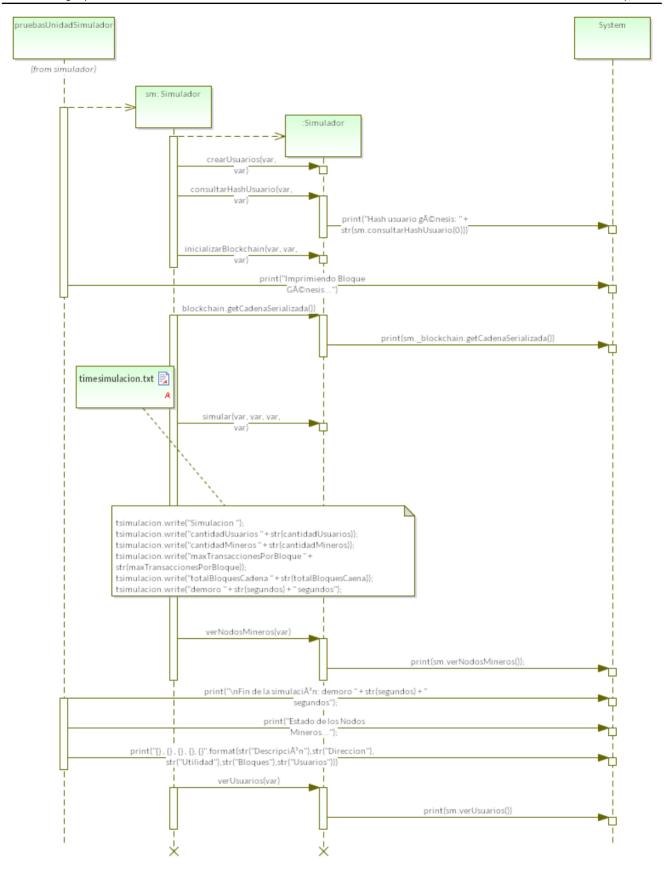


Figure 20: 9.2 Diagramas de Secuencia pruebasUnidadSimulador

# **1.3.2.3 9.3 Diagramas de Secuencia pruebasUnidadTransaccion diagram**

Interaction diagram in package '9. Diagramas de Secuencia'

9.3 Diagramas de Secuencia pruebasUnidadTransaccion Version 1.0 ENVY created on 10/07/2018. Last modified 10/07/2018

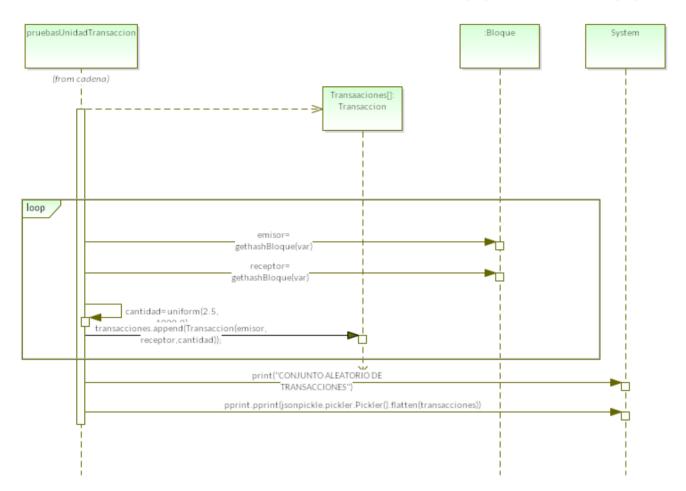


Figure 21: 9.3 Diagramas de Secuencia pruebasUnidadTransaccion

# 1.3.310. Diagramas de Comunicación

Package in package 'Diagramas de comportamiento'

10. Diagramas de Comunicación Version 1.0 Phase 1.0 Proposed ENVY created on 10/07/2018. Last modified 10/07/2018 Alias

# 1.3.3.1 10.1 Diagramas de Comunicación Usuario diagram

Communication diagram in package '10. Diagramas de Comunicación'

10.1 Diagramas de Comunicación Usuario Version 1.0 ENVY created on 10/07/2018. Last modified 10/07/2018

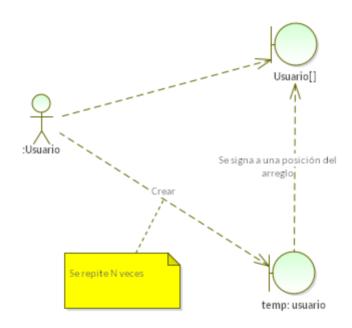


Figure 22: 10.1 Diagramas de Comunicación Usuario

# 1.3.3.2 10.2 Diagramas de Comunicación pruebasUnidadSimulador diagram

Communication diagram in package '10. Diagramas de Comunicación'

10.2 Diagramas de Comunicación pruebasUnidadSimulador Version 1.0 ENVY created on 10/07/2018. Last modified 10/07/2018

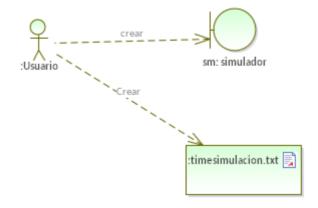


Figure 23: 10.2 Diagramas de Comunicación pruebasUnidadSimulador

# **1.3.3.3 10.3** Diagramas de Comunicación pruebaUnidadTrasaccion diagram

Communication diagram in package '10. Diagramas de Comunicación'

10.3 Diagramas de Comunicación pruebaUnidadTrasaccion Version 1.0 ENVY created on 10/07/2018. Last modified 10/07/2018

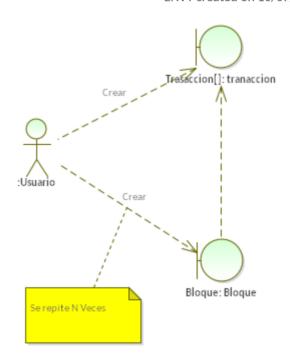


Figure 24: 10.3 Diagramas de Comunicación pruebaUnidadTrasaccion

# 1.4 Extendidos

Package in package 'cadenaDeBloques\_RPava'

Extendidos

Version 1.0 Phase 1.0 Proposed

ENVY created on 10/07/2018. Last modified 10/07/2018

Alias

# 1.4.115. Requisitos de requerimientos

Package in package 'Extendidos'

15. Requisitos de requerimientos Version 1.0 Phase 1.0 Proposed ENVY created on 10/07/2018. Last modified 10/07/2018 Alias

# 1.4.1.1 15. Requisitos requerimientos diagram

Requirements diagram in package '15. Requisitos de requerimientos'

15. Requisitos requerimientos

Version 1.0

ENVY created on 10/07/2018. Last modified 10/07/2018

Functional Requirements describe the features, behavior, business rules and general functionality that the proposed system must support. R1. Validar bloque en en una cadena de bloques, que evite transacciones duplicadas y el doble gasto Checklist ✓ Atomic ✓ Attainable R2. Validar a traves un proceso de mineria mediante el algoritmo  $\checkmark$ Cohesive de prueba de trabajo  $\checkmark$ Complete «trace»  $\checkmark$ Current **V** Independent R3. El usuario debe terner billeterra para enviar y recibir  $\checkmark$ Modifiable transacciones «trace»  $\checkmark$ Traceable  $\checkmark$ Unambiguous ✓ Verifiable R4. El metodo de distribuciones de los tokenes es a traves de recomenpensa por mineria «trace» The Non-Functional Requirements describe various operational parameters that define the environment in which the system will exist, including performance levels, architectural, security, regulatory, implementational and other operational requirements.

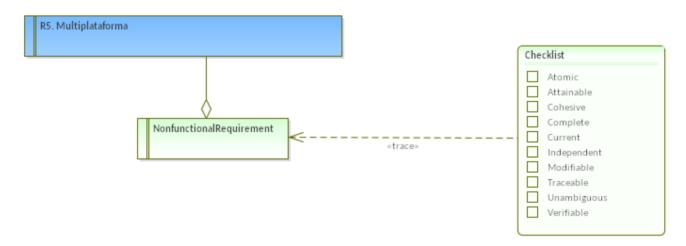


Figure 25: 15. Requisitos requerimientos