1. Especificación

1.1. observadores

TAD Berretacoin
obs blockchain: seq⟨Bloque⟩
obs saldos: dict⟨idUsuario: $\mathbb{Z}$, dinero: $\mathbb{Z}$⟩
obs totalCreado: $\mathbb{Z}$

1.2. agregarBloque

`proc agregarBloque` (inout cadena: Berretacoin, in bloque: Bloque)
`requiere` $\{bloque.id = ultimoBloque(cadena.blockchain).id + 1\}$
`requiere` $\{bloqueValido(bloque)\}$
`requiere` $\{tieneTransaccionDeCreacion(bloque) \longrightarrow_L totalCreado < 3000\}$
`requiere` $\{(\forall i : \mathbb{Z})\ ((0 \leq i < |bloque.transacciones|) \wedge \neg esCreativa(bloque.transacciones[i])$
$\longrightarrow_L (montoValido(bloque.transacciones[i])))\}$
`requiere` $\{cadena = C0\}$
`asegura` $\{cadena.blockchain = C0.blockchain + bloque\}$
`asegura` $\{(\forall i : \mathbb{Z})\ ((0 \leq i < |bloque.transacciones|) \wedge esCreativa(bloque.transacciones[i])$
$\longrightarrow_L cadena.saldos[getIdVendedor(bloque.transacciones[i])] =$
$C0.saldos[getIdVendedor(bloque.transacciones[i])] + getMonto(bloque.transacciones[i]))\}$
`asegura` $\{(\forall i : \mathbb{Z})\ ((0 \leq i < |bloque.transacciones|) \wedge esCreativa(bloque.transacciones[i])$
$\longrightarrow_L$
$cadena.saldos[getIdVendedor(bloque.transacciones[i])] =$
$C0.saldos[getIdVendedor(bloque.transacciones[i])] + getMonto(bloque.transacciones[i])$
$\wedge$
$cadena.saldos[getIdComprador(bloque.transacciones[i])] =$
$C0.saldos[getIdComprador(bloque.transacciones[i])] - getMonto(bloque.transacciones[i]))\}$

auxiliares

`aux ultimoBloque` (blockchain: seq⟨Bloque⟩) : Bloque $= blockchain[|blockchain| - 1]$;
`pred montoValido` (transaccion: Transaccion, cadena: Berretacoin) {
$transaccion.monto \leq cadena.saldos[transaccion.idComprador]$
}
`aux getIdVendedor` (transaccion: Transaccion) : $\mathbb{Z}$ $= transaccion.idVendedor$;
`aux getIdComprador` (transaccion: Transaccion) : $\mathbb{Z}$ $= transaccion.idComprador$;
`aux getMonto` (transaccion: Transaccion) : $\mathbb{Z}$ $= transaccion.monto$;

1.3. maximosTenedores

`proc maximosTenedores` (in cadena: Berretacoin) : seq⟨$\mathbb{Z}$⟩
`asegura` $\{(\forall u : \mathbb{Z})\ ((0 \leq u < |cadena.saldos|) \wedge$
$(cadena.saldos[u].dinero = valorMaximo(cadena.saldos)) \longrightarrow_L$
$concat(res, \langle u \rangle))\}$

auxiliares

`aux valorMaximo` (diccionario: dict⟨k: $\mathbb{Z}$, v: $\mathbb{Z}$⟩) : $\mathbb{Z}$ $= (\forall i : \mathbb{Z})\ ((1 \leq i < |diccionario|)($
if $diccionario[i].v > diccionario[i-1].v$ then $res = diccionario[i].v$ else $res = diccionario[i-1].v$ fi)) ;