

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)  $\rightarrow_L$  totalCreado < 3000}
  requiere {( $\forall i : \mathbb{Z}$ ) ((0  $\leq i < |bloque.transacciones|$ )  $\wedge$   $\neg$ esCreativa(bloque.transacciones[i])
 $\rightarrow_L$  (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])
 $\rightarrow_L$  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])
 $\rightarrow_L$ 
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