

TAD LABORATORIO 2

TAD Block



Key
Value

Operaciones primitivas:

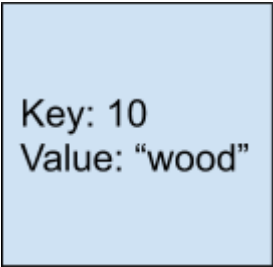
- CrearBlock → Block
- infoKey → Integer key
- infoValue → String value
- insert → Integer key, String value → Block
- deleted → Integer key
- getBlock → Integer key → Block
- changeValue → block * String → block
- changeKey → block * integer → block

crearBlock()

crea un block con datos base

{pre:true}

{post: block=

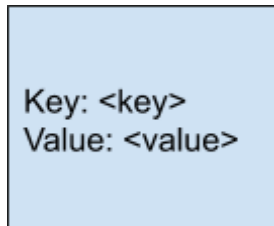


Key: 10
Value: "wood"

}

infoKey()
retorna la key the un block

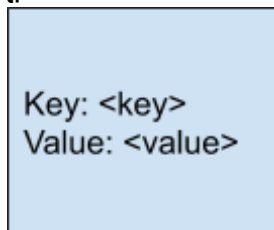
{pre: }



{post:<key>}

infoValue()
retorna el value the un block

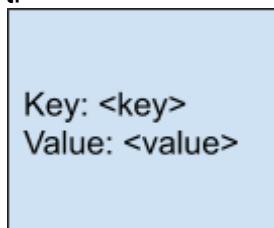
{pre: }



{post:<value>}

getBlock()
retorna un block

{pre: }



{post:<key>, <value>}

changeValue()

cambia el value the un block

{pre: ^ s pertenece a Strings }

Key: <key>
Value: wood

{post: }

Key: <key>
Value: s

changeKey()

cambia la key the un block

{pre: ^ k pertenece a enteros positivos }

Key: 10
Value: <value>

{post: }

Key: k
Value: <value>

TAD Cube

K:<Key>
V:<Value>

Operaciones primitivas:

- CrearCube → Cube
- infoKey → K key
- infoValue → V value
- changeValue → cube* String → cube
- changeKey → cube* integer → cube

crearCube()
crea un Cube con datos base

{pre:true}

{post: block= }

Key: <Key>
Value: <Value>

infoKey()
retorna la key the un cube

{pre: }

Key: <key>
Value: <value>

{post:<key>}

infoValue()
retorna el value the un cube

{pre: }

Key: <key>
Value: <value>

{post:<value>}

changeValue()
cambia el value the un cube

{pre: ^ s pertenece a V}

Key: <key>
Value: wood

{post: }

Key: <key>
Value: s

changeKey()
cambia la key the un cube

{pre: ^ k pertenece a K}

Key: 10
Value: <value>

{post: }

Key: k
Value: <value>