|  |
| --- |
| **TAD** Tabla de Hash. |
| **Tabla Hash =** { Clave, Valor } ˄ { T = { a1(clave, valor), a2(clave, valor), a3(clave, valor), ..., atamaño(clave, valor) } | tamaño > 0 } ˄ { indiceHash ≥ 0 | indiceHash pertenece a los enteros no negativos (Z) ˄ T[indiceHash] = a(clave, valor) }  **Nota:** La letra ‘**T’** indica el uso de un arreglo para almacenar los elementos de la tabla.  **Invariante:** { Clave ˄ Valor ≠ NIL } ˄ { tamaño es un número primo y un entero positivo mayor a cero (0) } ˄ { 0 ≤ indiceHash ≤ tamaño } ˄ { a es un objeto de tipo Nodo que es compuesto por una clave y valor } |
| **Operaciones primitivas:**   * CrearTablaHash: → TablaHash * InsertarNodo: Nodo x (Clave, Valor) → Booleano * BuscarNodo: Clave → Nodo * EliminarNodo: Clave → Booleano * ConvertirClaveANumero: Clave → Número * FunciónHashPorMétodoDeDivisión: Clave → Número |

**Operaciones principales.**

**Operaciones constructoras.**

|  |
| --- |
| **CrearTablaHash()**  “Crea una tabla de Hash vacía.”  {Pre: TRUE}  {Post: T.tamaño = 0} |

**Operaciones modificadoras.**

|  |
| --- |
| **InsertarNodo(Nodo(Clave, Valor))**  “Crea e inserta un nodo en el respectivo índice hash de la tabla.”  {Pre: TablaHash ≠ NIL}  {Post: TRUE → T[indiceHash] ≠ NIL ˅ T.tamaño > 0 ˄ T[indiceHash] = Nodo(Clave, Valor).  FALSE → T[indiceHash] = NIL ˅ T.tamaño = 0} |

|  |
| --- |
| **EliminarNodo(Clave)**  “Busca y elimina un nodo de la tabla Hash dada una clave.”  {Pre: TablaHash ≠ NIL ˄ T.tamaño ≠ 0}  {Post: TRUE → T.tamaño – 1 ˅ T[indiceHash] = NIL ˅ T[indiceHash].Clave ≠ Clave.  FALSE → T[indiceHash] ≠ NIL ˅ T[indiceHash].Clave = Clave} |

**Operaciones analizadoras.**

|  |
| --- |
| **BuscarNodo(Clave)**  “Busca y retorna un nodo de la tabla Hash dada una clave.”  {Pre: T.tamaño ≠ 0 ˅ T[indiceHash] ≠ NIL}  {Post: retorna el Nodo si Nodo.Clave = Clave.  De lo contrario, retorna NIL} |

|  |
| --- |
| **ConvertirClaveANumero(Clave)**  “Convierte una clave dada en un número entero.”  {Pre: TablaHash ≠ NIL}  {Post: retorna un número entero a partir de la clave ingresada.} |

|  |
| --- |
| **FunciónHashPorMétodoDeDivisión(Clave)**  “Utiliza la función Hash por método de división para retornar un índice de la tabla Hash dada una clave.”  **Nota:** La función Hash por método de división es: a mod b. Donde *a* representa un entero no negativo y *b* el tamaño de la tabla.  {Pre: TablaHash ≠ NIL}  {Post: Retorna un número entero no negativo que representa un índica de la tabla Hash.} |