**Tarea Integradora I**

Gloria Vanesa Vicuña - A00369332

Ricardo Medina Sterling - A00369009

Alejandro Osejo Ochoa - A00372469

Computación y Estructuras Discretas I

Universidad Icesi

2023

**DESIGN**

1. TAD design of the data structures used.

|  |
| --- |
| **HashTable** |
| **HashTable= {Size = <size>, Table = <lista(size)>** |
| {inv: <Invariante del TAD>} |
| Primitive Operations   * Hash: <key> → <hash> * Put: <key>, <value> → Table * Get: <key> → value * Remove: <key> → value * Size:    → size |

|  |
| --- |
| hash(key)  “Allows you to retrieve the Hash Code with the given key”  {pre:(key.hashCode() mod size) < 0}  {hash = (key.hashCode() mod size) + size} |

|  |
| --- |
| put(key,value)  “Inserts a key-value pair into the hash table.”  {pre: (table.key.hash.code() mod size) ≠ ∅ }  {post: key.hash, new HashNode(key, value)} |

|  |
| --- |
| get(key)  “Returns the value associated with the given key.”  {pre: (table.key.hash.code() mod size) ≠ ∅ }  {post: table.get(hash).getValue} |

|  |
| --- |
| remove (key)  “Removes the key-value pair associated with the given key.”  {pre: (table.key.hash.code() mod size) ≠ ∅ }  {post: table.get(hash).getValue  ∧  hash(k) ∉ table} |

|  |
| --- |
| **MaxPriorityQueue** |
| **MaxPriorityQueue = {Size = <0>, heap= <HeapNode(capacity)>** |
| {inv: <Invariante del TAD>} |
| Operaciones Primitivas   * GetMax :     → <max> * ExtracMax:    → <max> * MaxInset: <key>, <value>  → <heap> * Parent: <i> →  <parent> * Swap: <i>, <j> →  <j>, <i> * MaxHeapify: <i> → <maxHeapify> |

|  |
| --- |
| getMax()  “Returns the element with the highest priority in the priority queue.”  {pre: (heap[0].getValue) ≠ ∅ }  {post:heap[0].getValue } |