

- a) Aclaraciones: nuestros nodos se numeran con nombres desde 1 hasta 8 (es decir, N1, N2,...).

i -> el nodo activo atiende 2^i

Entonces, si $i=0$, atiende 1, si $i=1$, atiende 2, si $i=2$, atiende 4.

Tabla finger del nodo 3

Índice	Rango	Sucesor
0	4...4	5
1	5...6	5
2	7...2	7

Tabla finger del nodo 5

Índice	Rango	Sucesor
0	6...6	7
1	7...8	7
2	1...4	3

Tabla finger del nodo 7

Índice	Rango	Sucesor
0	8...8	3
1	1...2	3
2	3...6	3

- b) Algoritmo de búsqueda:

El parámetro de nodoActual e ID serán definidos por el usuario, y luego, nodoActual se iniciará con 3, o 5, o 7.

Además, hemos definido una función buscarDHT en la que asumimos que cada nodo, tiene una tabla DHT que contiene un array o lista que contiene los nodos que cubre dicho nodo, por ejemplo, el nodo N3, cubre a los nodos N8, N1, N2 y N3. Al ejecutar esta función, podemos verificar si el ID de nodo, se encuentra en el parámetro de nodo actual.

Asumo que los nodos activos tienen la función getFingerTable, que es una función que devuelve una matriz de 3 filas y 2 columnas que contiene los datos de la finger table.

```
int nodoActual;  
int tablaFingers[3][2];
```

```
Get(hashID){  
    int ID = hashID % 8;
```

```

//asumo que el hashID es equitativo

//Aquí llamamos a la búsqueda con un nodoActual definido por el usuario, y buscamos el ID
definido arriba
//A que nodo activo corresponde. Una vez hecho esto, del nodo activo, el archivo estará ahí
dentro y
//luego, asumo que la función buscarArchivo en un nodo está implementada, y con un
hashID dado, me devolverá
//el archivo requerido
return Búsqueda(ID, nodoActual).buscarArchivo(hashID);
}

Búsqueda(int ID, nodoActual){
    if(buscarDHT(nodoActual,ID)){
        return nodoActual;
    }else{
        //me fijo en la tabla finger a donde redireccionar
        int nodoSucesor = buscoSucesor(ID,nodoActual);
        return Búsqueda(ID, nodoSucesor);
    }
}

buscoSucesor(int ID, int nodo){
    tablaFingers = nodo.getFingerTable();
    i=0;
    stop = 0;
    while(i< tablaFingers.length() && stop == 0){
        int targetID = tablaFingers[i][0];
        int rango = (targetID + (2^i) -1) % 8;
        //int sucesor = tablaFingers[i][1]
        if(targetID <= rango){
            if(targetID <= ID && ID <= rango){
                sucesor = tablaFingers[i][1];
                stop = 1;
            }
        }else{
            if(targetID <= ID || ID <= rango){
                sucesor = tablaFingers[i][1];
                stop = 1;
            }
        }
        i++;
    }
    return sucesor;
}

buscarDHT(int ID, int nodo){
    int i = 0;
    int array[] = nodo.DHT();
    encuentre = 0;
    while( !encontre && i<array.length() ){

```

```
        if(array[i] == ID){  
            rencontre = 1;  
        }  
        i++;  
    }  
    return rencontre;  
}
```