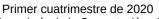
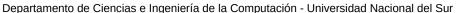


Laboratorio N.º 4

SISTEMAS Distribuidos







Alumnos

Garat Manuel Registro: 103138

Cerdá Gianni Lucas Registro: 110996

Nota: se considera que el anillo empieza a partir del nodo 0 y finaliza en el 7.

Tabla Finger del Nodo 2

Índice	Rango	Sucesor
1	33	4
2	45	4
3	62	6

Tabla Finger del Nodo 4

Índice	Rango	Sucesor
1	55	6
2	67	6
3	04	2

Tabla Finger del Nodo 6

Índice	Rango	Sucesor
1	77	2
2	01	2
3	26	2

```
const int nodos = 8;
const int este_nodo; // nodo actual que quiere encontrar a su sucesor
int entries = 3 // cantidad de entradas en la tabla
struct tabla finger {
    int rangos[][entries];
    int sucesores[entries];
} tabla_finger [nodos];
int Get(int ID, int nodo) {
    if (Pertenece(ID, nodo))
            return nodo;
    else{
            nodo = Buscar_sucesor(ID, nodo);
            return Get(ID, nodo);
    }
}
int Buscar_sucesor(int D, int nodo){
    int seguir = 1, i = 0, retorno = -1;
    while (i < entries && seguir) {
        if (en rango(ID)) {
            retorno = tabla finger[nodo].sucesores[i];
            seguir = 0;
        else i++;
    }
    return retorno;
}
int Pertenece(int ID, int nodo) {
    int seguir = 1, i = 0, retorno = 0;
    while (i < entries && seguir) {
        if (en rango(ID)) {
            if (ID <= tabla finger[nodo].sucesores[i]) {</pre>
                retorno = \overline{1};
                seguir = 0;
            else i++;
        else i++;
    return retorno;
}
char * Buscar archivo(char * nombre) {
    int ID = hash(nombre);
    int nodo = Get(ID%8, este nodo);
    return Obtener_archivo(nodo, ID);
}
```