



Internet de las cosas

Jorge Becker - A01027251

Gabriel Dichi - A01027301

Reflexión de la actividad 2.3

Mtro. Leonardo Chang

9 de octubre de 2020

TABLA DE CONTENIDOS

ADT de la clase	2
Justificación de las estructuras	3

ADT de la clase

```
class ConexionesComputadora
{
private:
string ip;
string nombre;
stack<Linea>conexiones_entrantes;
queue<Linea>conexiones_salientes;
```

En esta foto podemos observar los atributos de la clase.

```
public:
/*aquí usamos gets para poder acceder a los atributos privados*/
stack<Linea> get_conexiones_entrantes(){return conexiones_entrantes;}
queue<Linea> get_conexiones_salientes(){return conexiones_salientes;}
```

En esta foto se observa cómo usamos gets para poder usar los atributos privados

```
ConexionesComputadora(string ip2, string nombre2, vector<Linea> archivo)
{
    ip = ip2;
    nombre = nombre2;

    for (int i=0; i<archivo.size()-1; i++)
    {
//stack
        if (archivo[i].ip2==ip)
        {
            conexiones_entrantes.push(archivo[i]);
        }

//queue
        if (archivo[i].ip==ip)
        {
            conexiones_salientes.push(archivo[i]);
        }
    }
};

~ConexionesComputadora(){};
};
```

En esta foto podemos observar los métodos para manipular las estructuras de datos que decidimos utilizar para cumplir con el reto.

Justificación de las estructuras

Utilizamos vectores para agrupar cada columna del fichero independientemente y así poder indagar en cada uno de ellos. Utilizamos principalmente los operadores de **stack** y **queue** para almacenar las conexiones entrantes y salientes deseadas y con la función **push()** las fuimos introduciendo al vector . Posteriormente utilizamos la función de **top()** para extraer el dato superior de la pila y así ir respondiendo cada una de las preguntas del reto.

Nuestro programa se realiza en un tiempo lineal ya que todas las estructuras de datos utilizadas así se comportan. No encontramos ningún while ni for anidado (o más complejo) y es por eso que llegamos a la conclusión de que todo nuestro programa corre en una estructura de **$O(n)$** .