

TRABAJO TEMA 2

SERVIDOR DE IMPRESIÓN

- Definición de estructuras auxiliares

```
struct trabajo {  
    int identificador;  
    String nombre_fichero;  
    int num_copias;  
    int identificador_usuario;  
    int estado;  
};
```

Esta estructura almacenará diversa información sobre los trabajos, los cuales son mandados a la cola de impresión para su posterior tratamiento.

- **identificador:** campo que almacena un identificador único para cada estructura de tipo trabajo.
- **nombre_fichero:** campo que contiene el nombre del fichero a imprimir.
- **num_copias:** campo que indica el número de copias a realizar del fichero.
- **identificador_usuario:** campo que almacena el identificador del usuario que mando imprimir el fichero, si su valor es 0 se trata del administrador del sistema.
- **estado:** campo que muestra el estado en el que se encuentra el fichero y que puede tomar los siguientes valores:
 - > 1: el fichero esta encolado
 - > 0: el fichero se está imprimiendo
 - > -1: el fichero se ha cancelado

```
struct cola {  
    vector<struct rabajo> lista_trabajos;  
}
```

Esta estructura representa la cola de impresión, en la que se almacenarán los distintos trabajos que lleguen para imprimir.

imprimir(lista de trabajo, identificador de usuario)

ALGORITMO: imprimir

ENTRADA: lista de trabajo, identificador de usuario

SALIDA: lista de indicadores

VARIABLES: i: entera

trabajos: lista de trabajo

id_usuario: entero

identificadores: vector de enteros

INICIO

PARA i=1 HASTA trabajos.tamaño() CON INCREMENTO +1 HACER

asignar_Identificador(trabajos(i),i);

añadir_Identificador(identificadores,i);

añadir_aCola(trabajos(i));

FIN_PARA

DEVOLVER identificadores;

FIN

consultar(identificador de usuario)

ALGORITMO: consultar

ENTRADA: identificador de usuario

SALIDA: lista de trabajo

VARIABLES: i: entera

trabajos: lista de trabajo

cola_impresion: cola

iden: entero

INICIO

PARA i = 1 HASTA cola_impresion.tamaño() CON INCREMENTO +1 HACER

SI cola_impresion(i)->id_usuario == iden ENTONCES

añadir_trabajo(trabajos, cola_impresion(i));

FIN_PARA

DEVOLVER trabajos;

FIN

cancelar(lista de trabajo, identificar de usuario)

ALGORITMO: cancelar

ENTRADA: lista de trabajo, identificador de usuario

SALIDA: éxito o fallo

VARIABLES: i: entera

trabajos: lista de trabajo

cola_impresion: cola

iden: entero

INICIO

PARA i=1 HASTA trabajos.tamaño() CON INCREMENTO +1 HACER

SI buscar_trabajo(cola_impresion, trabajos(i)) == true ENTONCES
cola(trabajos(i)) -> estado = -1

SINO

DEVOLVER fallo;

FIN_PARA

DEVOLVER éxito;

FIN

resetear(identificador de usuario)

ALGORITMO: resetear

ENTRADA: identificador de usuario

SALIDA: éxito o fallo

VARIABLES: i: entera

cola_impresion: cola

iden: entero

INICIO

SI iden == 0 ENTONCES

PARA i=1 HASTA cola_impresion.tamaño() CON INCREMENTO +1 HACER

eliminar(cola_impresion(i));

FIN_PARA

DEVOLVER éxito;

SINO

DEVOLVER fallo;

FIN