

# ESTRUCTURA DE DATOS

## DESAFÍO UNIDAD 2

### SEMANA 6



**Identificación de los estudiantes:**

Integrantes del grupo	
1	Alexis Sáez
2	Leticia Palazuelos
3	Leyla Verdejo

# Índice de Contenidos

## Introducción

### Especificación Escenario 1

#### I.I Especificación Informal:

Nombre:

Operaciones:

#### I.II Especificación Formal:

Nombre:

Sintaxis:

Semántica:

#### I.III Implementación Escenario 1

### II.I Especificación Escenario 2

#### II.I.I Especificación Informal:

#### II.I Especificación Formal:

Nombre:

Sintaxis:

Semántica:

#### Implementación Escenario 2

Descripción:

## Conclusiones

# I.0 Introducción

El siguiente informe describe la solución de la problemática de un servicio de impresión en dos diferentes escenarios. En el primer escenario se describe la especificación formal e informal y las operaciones necesarias para llevar a cabo la implementación. Se requiere una sola red de impresión para lo cual se toma en cuenta una sola cola como estructura. Los documentos se van imprimiendo en cualquiera de los 3 servidores de acuerdo al orden de llegada. El segundo escenario describe la especificación informal y formal del TDA que se había trabajado anteriormente junto con las operaciones necesarias para la funcionalidad en la posterior implementación.

## I.1 Especificación Escenario 1

### I.1.1 Especificación Informal:

**Nombre:**

**TAD ServicioDeImpresion**(Especifica un conjunto de  $n$  elementos donde cada uno representa un documento para imprimir, en cuanto a los documentos, están caracterizados por los siguientes atributos: nombre, número de páginas, números de copias, estado que puede ser [espera o cancelado]).

## Operaciones:

crearServicioDeImpresion()

- Crea una cola que representa la cola de Impresión, devuelve la cola.

elServicioestaVacio(ServicioDeImpresion si)

- Verifica si la cola de impresión está vacía. Retorna un 1 si está vacía y un 0 si no está vacía.

asignarDocumentoAServicioDeImpresion(ServicioDeImpresion si, Documento d)

- Agrega un documento a la cola de impresión. Devuelve la cola de impresión con el documento encolado.

crearDocumento(nombre, num\_paginas, num\_copias)

- Crea un documento con los parámetros especificados y retorna un documento.

recuperarDocumentoDelFrente(ServicioDeImpresion si)

- Retorna el documento que está en la posición frontal de la cola de impresión.

desencolarServicioDeImpresion(ServicioDeImpresion)

- Retorna la cola de impresión y aumenta en 1 el frente de la cola.

cancelarImpresion(Documento d)

- Busca en la cola de impresión el documento que coincide con el nombre del documento pasado por referencia y le cambia el estado a “cancelado” retornando la cola de impresión modificada.

## I.I.II Especificación Formal:

### Nombre:

**TAD ServicioDeImpresion**(Especifica un conjunto de  $n$  elementos donde cada uno representa un documento para imprimir, en cuanto a los documentos, están caracterizados por los siguientes atributos: nombre, número de páginas, números de copias y estado que puede ser [espera o cancelado]).

### Sintaxis:

Operaciones	Retorna
Cola crearServicioDeImpresion()	ServicioDeImpresion
ServicioDeImpresion elServicioestaVacio(ServicioDeImpresion si)	True or False
ServicioDeImpresion asignarDocumentoAServicioDeImpresion(ServicioDeImpresion si, Documento d)	Documento
Documento recuperarDocumentoDelFrente(ServicioDeImpresion si)	Documento
Documento desencolarServicioDeImpresion(ServicioDeImpresion)	ServicioDeImpresion
Documento crearDocumento(nombre, num_paginas,num_copias)	Documento
ServicioDeImpresion cancelarImpresion(Documento d)	ServicioDeImpresion

## Semántica:

$\forall SI \in \text{ServicioDelImpresion}, \forall d1, d2, d3 \in \text{documento}$

`crearServicioDelImpresion() => SI`

`elServicioEstaVacio(asignarDocumentoAServicioDelImpresion(SI, d1), d2) =>`  
`return 1`

`elServicioEstaVacio(crearServicioDelImpresion()) => return 0`

`asignarDocumentoAServicioDelImpresion(asignarDocumentoAServicioDelImpresion`  
`(SI, d1), d2) => asignarDocumentoAServicioDelImpresion(`  
`asignarDocumentoAServicioDelImpresion(SI, d2), d1)`

`recuperarDocumentoDelFrente(asignarDocumentoAServicioDelImpresion(SI, d1),`  
`d2) => asignarDocumentoAServicioDelImpresion(`  
`asignarDocumentoAServicioDelImpresion(SI, d2), d1) [frente]`

`CancelarImpresion(AgregarColasALista(AgregarColasALista(AgregarColasALista(`  
`L, c1) , c2), c3), d1), d1) => si d1  $\in$  c1  $\cup$  c2 entonces d1->estatus = cancelado`

`desencolarServicioDelImpresion(crearServicioDelImpresion()) => return`  
`ServicioDelImpresion()`

`desencolarServicioDelImpresion(asignarDocumentoAServicioDelImpresion(`  
`asignarDocumentoAServicioDelImpresion(SI, d2), d1)) =>`  
`asignarDocumentoAServicioDelImpresion(SI, d2), d1) -> frente+1; return`  
`asignarDocumentoAServicioDelImpresion(SI, d2), d1)`

## I.I.III Implementación Escenario 1

### Descripción:

La estructura implementada es de un solo servicio de Impresión, en el cual entran varios documentos y son destinados a cualquier servidor en el orden de llegada.

Comando para ejecutar el programa:

```
gcc main.c documento.c utils.c servicio_impresion.c -o app.o && ./app.o
```

## II.I Especificación Escenario 2

### II.I.I Especificación Informal:

#### Nombre:

**TAD Colas**(Especifica un conjunto de secuencias de  $n$  elementos donde cada uno representa un documento para imprimir, en cuanto a los documentos, están caracterizados por los siguientes atributos: nombre, número de páginas, números de copias, estado que puede ser [espera o cancelado]).

#### Operaciones:

ListaColasVacias()

- se crea una lista de colas vacías, devuelve la lista de colas.

AgregarColaALista(ListaDeColas, cola)

- agrega una cola a la lista, devuelve la lista de colas.

SumaDeDocumentos(cola)

- retorna la sumatoria de los documentos con su cantidad de páginas multiplicado por la cantidad de copias.

ObtenerIndiceMejorCola(ListaDeColas)

- recorre lista de colas, compara la cantidad de impresiones que debe hacer un servidor de impresoras en base a la suma de los documentos por la cantidad de páginas y la cantidad de copias. Retorna la primera lista con 0 páginas o la que tenga menor número de ellas.

AsignarDocumentoAServidorImpresion(ListaDeColas, documento)

- consulta con ObtenerIndiceMejorCola() e ingresa el documento en la mejor lista correspondiente a cierto servidor, retorna ListadeColas.

CancelarImpresion(ListaDeColas, documento)

- recorre el conjunto de colas y busca el documento a ser cancelado, lo elimina de la lista de impresión y luego redistribuye los documentos y devuelve ListadeColas.

Redistribuir(ListaDeColas)

- Toma los documentos en orden copiando las 3 colas y las redistribuye para cada cola de impresión. Devuelve la ListaDeColas.



## II.1.11 Especificación Formal:

### Nombre:

**TAD Colas**(Especifica un conjunto de secuencias de  $n$  elementos donde cada uno representa un documento para imprimir, en cuanto a los documentos, están caracterizados por los siguientes atributos: nombre, número de páginas, números de copias, estado que puede ser [espera o cancelado])

### Sintaxis:

Operaciones	Retorna
*ListadeColas	ListadeColas
*AgregarColaALista(ListaDeColas, cola)	ListadeColas
SumadeDocumentos(cola)	int
ObtenerIndiceMejorCola(ListadeColas)	cola
AsignarDocumentoAServidorImpresion(ListaDeColas, documento)	ListadeColas
CancelarImpresion(ListadeColas, documento)	ListadeColas
Redistribuir(ListadeColas)	ListadeColas

## Semántica:

$\forall L \in \text{ListadeColas}, \forall c1, c2, c3 \in \text{cola}, \forall d1, d2, d3 \in \text{documento}$

$\text{CrearListadeColas}() \Rightarrow L$

$\text{AgregarColasALista}(\text{AgregarColasALista}(L, c1), c2) \Rightarrow$   
 $\text{AgregarColasALista}(\text{AgregarColasALista}(L, c2), c1)$

$\text{AgregarColasALista}(\text{AgregarColasALista}(\text{AgregarColasALista}(L, c1), c2), c3) \Rightarrow$   
 $\text{AgregarColasALista}(\text{AgregarColasALista}(\text{AgregarColasALista}(L, c3), c2),$   
 $c1)$

$\text{ObtenerIndiceMejorCola}(\text{AgregarColasALista}(\text{AgregarColasALista}(\text{AgregarColasA}$   
 $\text{Lista}(L, c1), c2), c3)) \Rightarrow \forall c \in \text{ListaColas}[i] \text{ entonces } \text{ObtenerIndiceMejorCola}()$   
 $\text{retornando el } c1 \text{ si } c1 < c2 < c3$

$\text{SumadeDocumentos}(c) \Rightarrow \forall dn \in c \text{ realizará } \text{SumadeDocumentos}(c) \text{ haciendo } \Sigma$   
 $(d1 * npaginasd1 * ncopiasd1) + (d2 * npaginasd2 * ncopiasd2) + (d3 *$   
 $npaginasd3 * ncopiasd3)$

$\text{AsignarDocumentoAServidorImpresion}((\text{AgregarColasALista}(\text{AgregarColasALista}(\text{AgregarColasALista}(L, c1), c2), c3)), d1) \Rightarrow$   
 $\text{ObtenerIndiceMejorCola}(\text{AgregarColasALista}(\text{AgregarColasALista}(\text{AgregarColasA}$   
 $\text{Lista}(L, c1), c2), c3)) \text{ entonces agregar } d1$

$\text{CancelarImpresion}(\text{AgregarColasALista}(\text{AgregarColasALista}(\text{AgregarColasALista}(L, c1), c2), c3), d1) \Rightarrow \text{si } d1 \in c1 \cup c2 \cup c3 \text{ entonces eliminar de } d, c \text{ luego}$   
 $\text{Redistribuir}(\text{AgregarColasALista}(\text{AgregarColasALista}(\text{AgregarColasALista}(L, c1),$   
 $c2), c3))$

$\text{Redistribuir}(\text{AgregarColasALista}(\text{AgregarColasALista}(\text{AgregarColasALista}(L, c1),$   
 $c2), c3)) \Rightarrow$   
 $L2 = \text{AgregarColasALista}(\text{AgregarColasALista}(\text{AgregarColasALista}(L, c1), c2), c3)$   
 $, \forall c \in L2 \text{ entonces } \text{AsignarDocumentoAServidorImpresion}(L, d)$

## I.II.III Implementación Escenario 2

### Descripción:

Comando para ejecución:

```
gcc main.c documento.c utils.c cola.c listaColas.c -o app.o && ./app.o
```

## I.I.IV Conclusiones

Al realizar la especificación del TDA se pudo identificar el tipo de estructura que se iba a utilizar para la posterior implementación y las funciones que se iban a crear. Se lograron identificar 7 funciones a través de las cuales se crean las colas, se verifica la cola de impresión más corta para poder asignar el documento ésta. La especificación jugó un rol de blueprint, en el cual se pudo organizar la funcionalidad del requerimiento. A través de el TDA y la implementación de este, se logró describir una solución informática para el problema de distribución de impresión de la notaría.

En la siguiente tabla Tabla.1 se muestran las operaciones definidas a través del TDA para problema de distribución de impresión de la notaría.

Tabla.1

Operaciones	Retorna
CrearListadeColas	-> ListadeColas
AgregarColaALista(ListaDeColas, cola)	-> ListadeColas
SumadeDocumentos(cola)	-> int
ObtenerIndiceMejorCola(ListadeColas)	-> cola
AsignarDocumentoAServidorImpresion(ListaDeColas, documento)	-> ListadeColas
CancelarImpresion(ListadeColas, documento)	-> ListadeColas

Redistribuir(ListadeColas)	-> ListadeColas
----------------------------	-----------------