

Universidad San Carlos de Guatemala

Facultad de Ingeniería

Escuela de Ciencias y sistemas

Estructuras de Datos

Ingenieros:

- Ing. Edgar Ornelis
- Ing. Álvaro Hernández
- Ing. Luis Espino

Auxiliares:

- José Montenegro
- Carlos Castro
- Kevin Martinez



Proyecto Fase 1

**Pixel Print Studio**

Implementación de listas dinámicas

# Índice

<b>Objetivos</b>	<b>3</b>
Objetivo general	3
Objetivos específicos	3
<b>Descripción</b>	<b>3</b>
<b>Simulación</b>	<b>3</b>
<b>Menú de la aplicación</b>	<b>5</b>
<b>Estructuras utilizadas</b>	<b>5</b>
Cola de recepción	5
Lista de ventanillas (simple)	6
Pila de imágenes	6
Lista de clientes atendidos (simplemente enlazada)	7
Cola de impresión	7
Lista de clientes en espera (Lista de listas)	7
<b>Flujo de la aplicación:</b>	<b>8</b>
<b>Ejemplo</b>	<b>8</b>
<b>Reportes</b>	<b>13</b>
Visualización de estructuras:	13
Datos generados	13
<b>Observaciones</b>	<b>14</b>
<b>Entregables:</b>	<b>14</b>
<b>Restricciones</b>	<b>14</b>

# Objetivos

## Objetivo general

- Desarrollar una aplicación que optimice la manipulación de información mediante la aplicación efectiva de los conocimientos adquiridos en el curso de Estructuras de Datos.

## Objetivos específicos

- **Aplicación Práctica de Estructuras de Datos Lineales:**  
Demostrar la comprensión y aplicación de las estructuras de datos lineales a través de una aplicación de simulación que refleje de manera efectiva los procesos de la empresa.
- **Implementación en Fortran:**  
Utilizar el lenguaje de programación Fortran para implementar y gestionar eficientemente las estructuras de datos lineales, aprovechando las características específicas del lenguaje en el desarrollo de la aplicación.
- **Visualización con Graphviz:**  
Integrar la herramienta Graphviz para generar representaciones visuales claras y comprensibles de las estructuras de datos lineales utilizadas en la simulación, facilitando así la comprensión del funcionamiento interno de la aplicación.
- **Definición e Implementación de Algoritmos:**  
Definir algoritmos específicos de búsqueda, recorrido y eliminación adaptados a las necesidades de la simulación, asegurando una manipulación eficiente de la información en todas las etapas del proceso.

## Descripción

La empresa "Pixel Print Studio" se dedica a la impresión de figuras de pixel art en distintos tamaños y tipos de papel. Con el crecimiento constante de clientes, ha surgido la necesidad de optimizar los procesos de recepción y producción. En este sentido, se solicita a un estudiante de ingeniería en sistemas aplicar conocimientos en estructuras de datos para mejorar la eficiencia operativa.

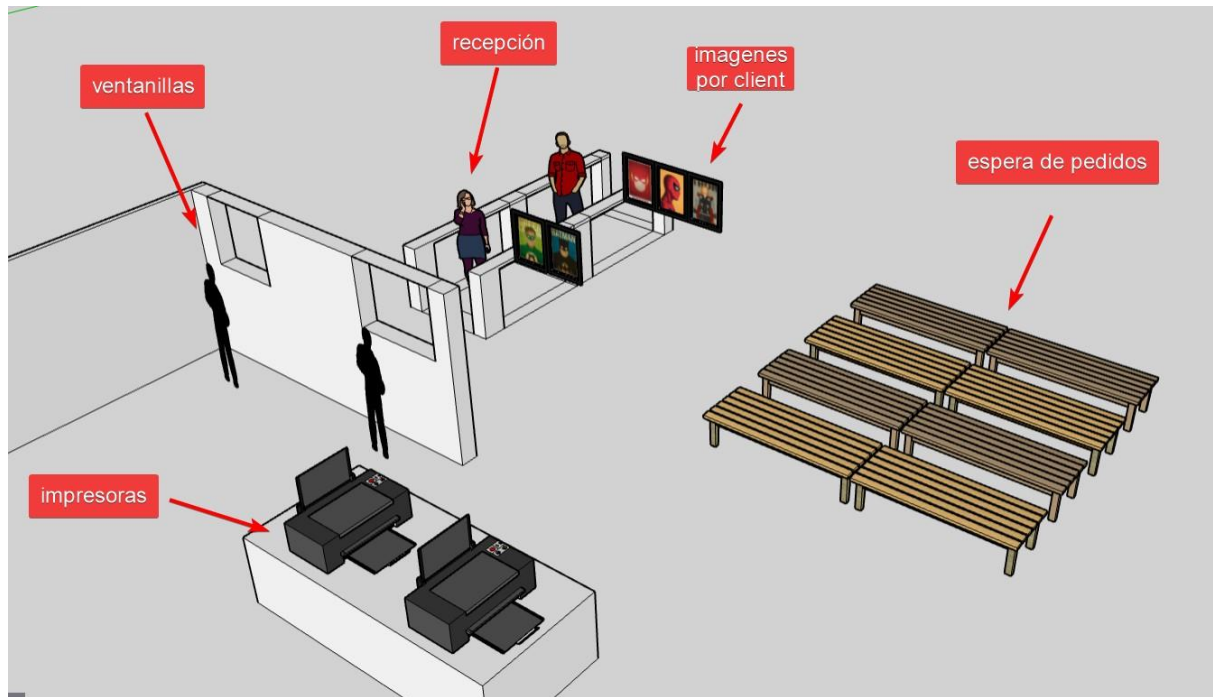
Deberás desarrollar una aplicación que utilice estructuras de datos y algoritmos para simular los diversos procesos en la empresa "Pixel Print Studio". La aplicación deberá representar visualmente las estructuras mediante bibliotecas compatibles, como Graphviz.

## Simulación

La simulación abarcará todo el proceso desde que los clientes realizan la solicitud de impresión de figuras de pixel art hasta que reciben sus impresiones. El tiempo de simulación se medirá en pasos, permitiendo avanzar un paso a la vez.

El funcionamiento en general de la empresa es el siguiente:

1. La empresa contará con un número "n" de estaciones de recepción donde se tomarán los pedidos, un valor a definir al inicio.
2. Los clientes ingresan a una cola de recepción al llegar a la empresa y salen de ella cuando una estación esté disponible.
3. La información de cada cliente se almacenará en una lista cuando ingrese a una estación.
4. La duración de la atención en una estación será equivalente a la cantidad de figuras de pixel art que el cliente desea imprimir.
5. La cantidad de clientes que llegan por paso será aleatoria en un rango de 0 a 3.
6. Después de ser atendido en la estación, el cliente pasará a una lista de espera, y sus figuras pasarán a una cola de producción según su tipo (figuras pequeñas y grandes).
7. Existirán estaciones de producción para figuras pequeñas y otra para figuras grandes.
  - a. La producción de una figura pequeña tardará 1 paso.
  - b. La producción de una figura a grande tardará 2 pasos.
8. Cuando una figura se imprima, se añadirá a la lista de figuras de cada cliente en espera.
9. El cliente saldrá de la lista de espera cuando todas sus figuras estén impresas.
10. Al salir de la empresa, se actualizará en el registro del cliente la cantidad de pasos que estuvo esperando para que sus figuras de pixel art fueran impresas.



## Menú de la aplicación

El menú queda a discreción del estudiante, sin embargo, debe poseer al menos las siguientes opciones

1. Parámetros iniciales
  - a. Carga masiva de clientes
  - b. Cantidad de ventanillas
2. Ejecutar paso
3. Estado en memoria de las estructuras
4. Reportes
5. acerca de ←datos del estudiante
6. Salir

## Estructuras utilizadas

### 1. Cola de recepción

En esta estructura es donde se ingresan los clientes al llegar a la empresa, en el menú deberá existir una opción que permita la carga masiva de los clientes que ingresarán a la empresa. La carga masiva se realizará por medio de un archivo con extensión (json) que contendrá la siguiente estructura:

```

1  [
2  {
3      "id": "1",
4      "nombre": "Andres Lopez",
5      "img_g": "3",
6      "img_p": "2"
7  },
8  {
9      "id": "2",
10     "nombre": "Juan Perez",
11     "img_g": "3",
12     "img_p": "0"
13 },
14 {
15     "id": "3",
16     "nombre": "Luiz Higueros",
17     "img_g": "2",
18     "img_p": "1"
19 }
20 ]

```

Los clientes ingresados anteriormente estarán en el estado inicial de la aplicación (Revisar las imágenes de pasos adjuntas), adicionalmente en cada paso se generan aleatoriamente más clientes con las siguientes características:

- Cantidad de clientes aleatorios: entre 0 y 3
- Cantidad de imágenes por cada cliente: entre 0 y 4
- Nombre aleatorio: pueden utilizar vectores de nombres y apellidos para generarlos.

## 2. Lista de ventanillas (simple)

Se implementará una lista simplemente enlazada para el número de ventanillas que estarán en el proceso de simulación, cada ventanilla contará con una pila que se utilizará para recibir las imágenes que un cliente desea imprimir.

### 2.1. Pila de imágenes

El funcionamiento de la pila de cada ventanilla es el siguiente:

- En cada paso que el cliente está en ventanilla se inserta una imagen a la pila, con todas las especificaciones de la impresión.
- La pila se vacía cuando el cliente sale de la ventanilla.

### 3. Lista de clientes atendidos (simplemente enlazada)

Se guardará la información de todos los clientes que son atendidos por todas las ventanillas habilitadas durante el proceso de simulación.

En cada nodo de la lista se almacenará la siguiente información:

- Nombre del cliente
- Ventanilla que lo atendió
- Número de imágenes impresas
- Cantidad total de pasos en el sistema

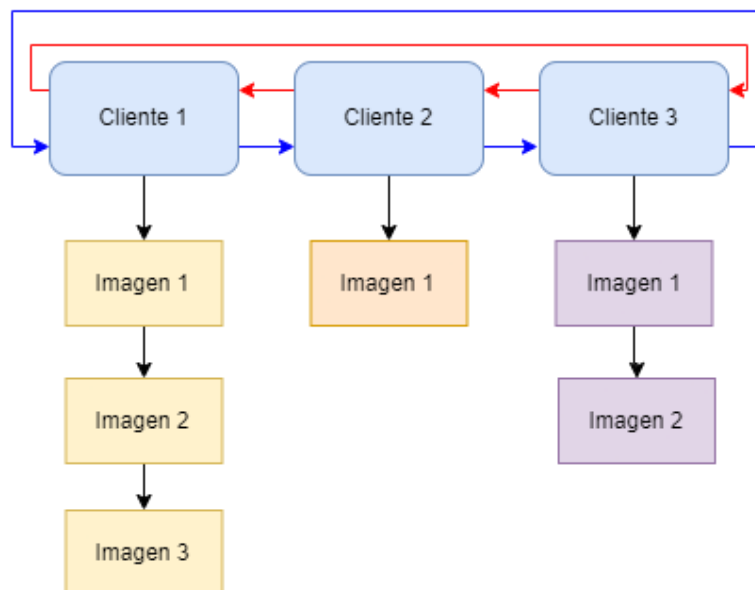
### 4. Cola de impresión

Cada impresora maneja una cola de impresión, al momento que un cliente termina de realizar su orden de impresión, las imágenes son enviadas desde la pila de la ventanilla hacia las distintas colas de acuerdo a las especificaciones solicitadas (imagen pequeña o grande), se debe implementar un mecanismo de clasificación de imágenes que permitirá que en ningún momento una imagen sea enviada a la impresora incorrecta, de tal forma que se cumplan las especificaciones solicitadas por el cliente.

### 5. Lista de clientes en espera (Lista de listas)

Luego de terminar el proceso en ventanilla, cada uno de los clientes se almacenará en una única lista de espera, en la que uno de los clientes (nodos) contará con una lista de imágenes impresas, estas estarán mezcladas.

Después de que la imagen se genere, esta buscará a que cliente pertenece, y se almacenará en la lista de imágenes del cliente sin importar el tipo de impresión. Una vez que la lista de imágenes del cliente esté completa, se eliminará al cliente de la lista de espera.

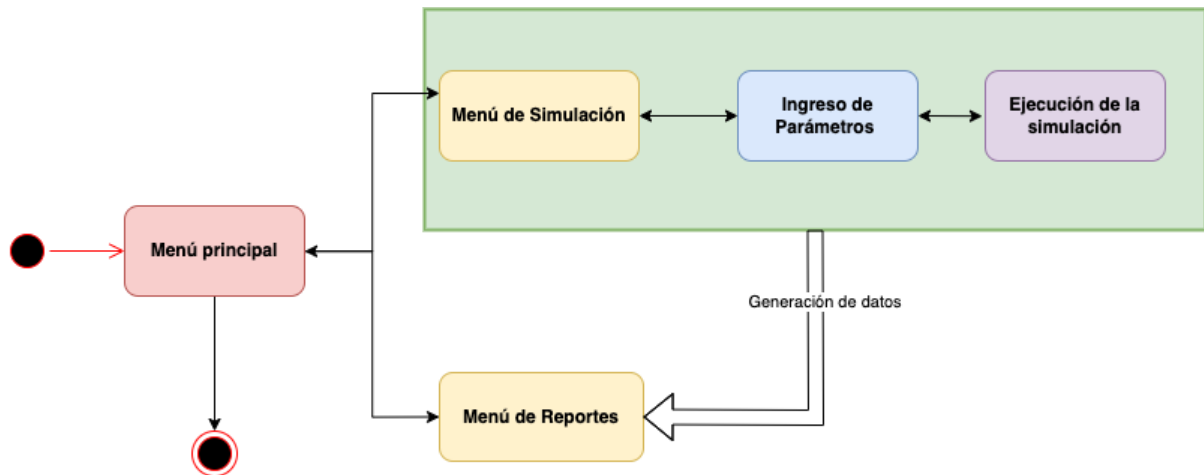


La lista de clientes debe ser una lista circular doblemente enlazada, la lista de imágenes de cada cliente no tiene restricción de implementación.

## Flujo de la aplicación:

### General

Describe el orden de las acciones en la aplicación al momento de su ejecución.

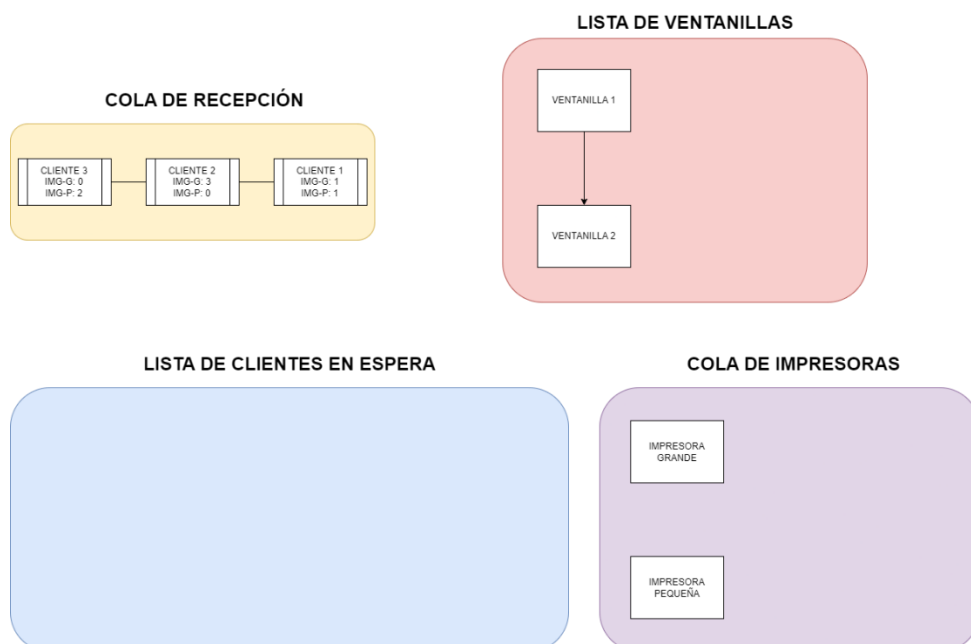


## Ejemplo

Nota: Las imágenes son una representación del flujo de la aplicación, las estructuras pueden cambiar dependiendo de lo que se solicitó en el enunciado.

### Estado Inicial

Luego de ingresar en el menú la cantidad de ventanillas y realizar una carga masiva de clientes las estructuras de datos deberían de verse así

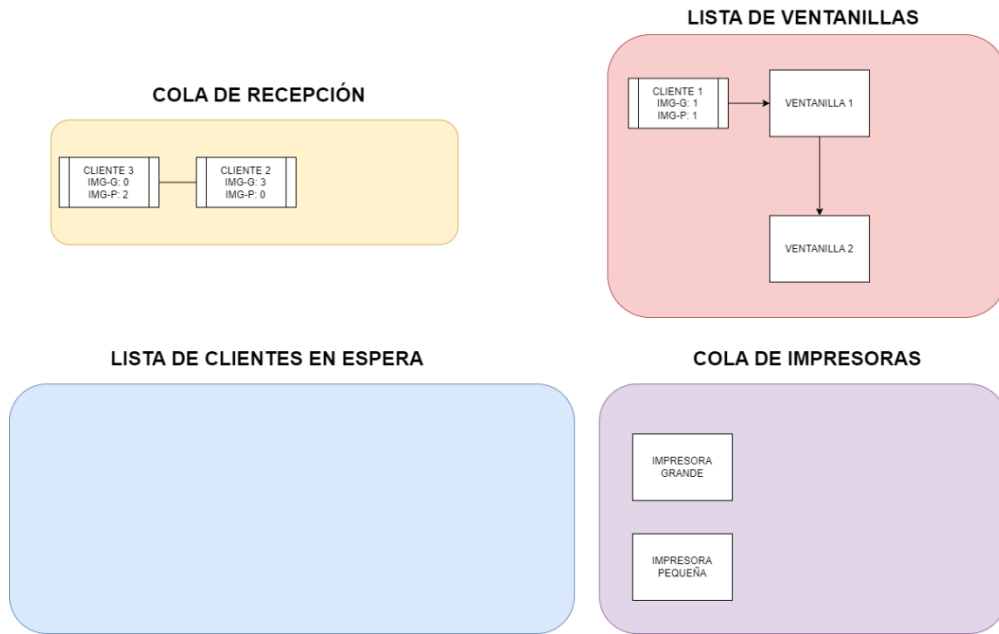




### Paso 1

Al realizar la ejecución del primer paso ocurrirán las siguientes acciones:

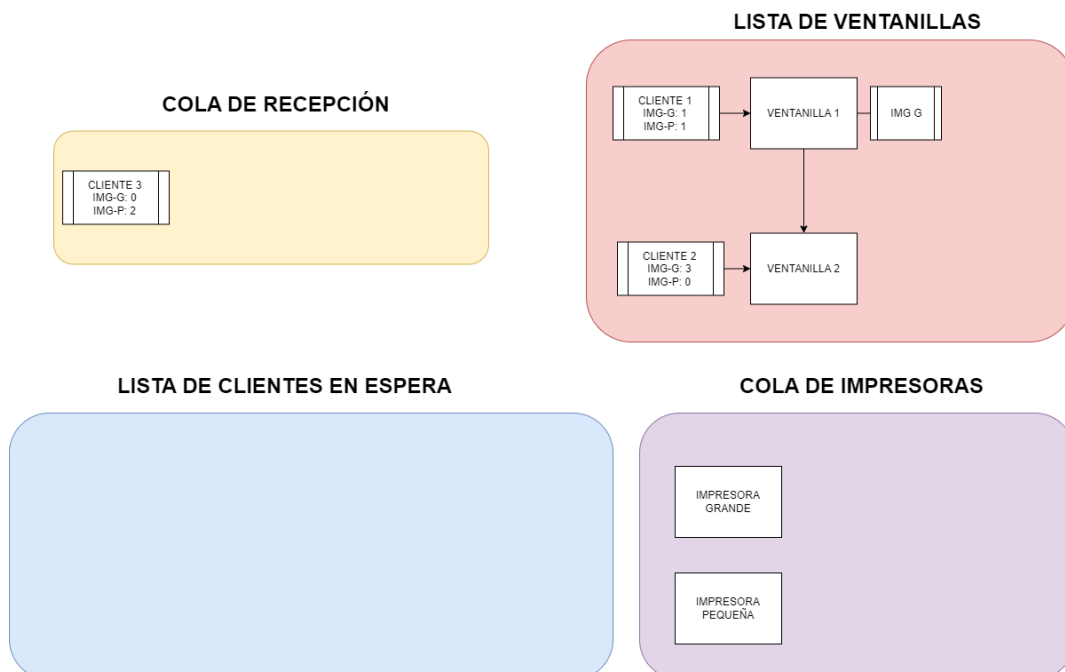
- El cliente 1 ingresa a la ventanilla 1 (la primera disponible).
- En la cola de recepción pueden ingresar clientes de forma aleatoria como se indicó en el apartado 1, por motivos de ejemplo esto no sucede.
- En el resto de las estructuras por el momento aún no suceden cambios.



### Paso 2

Al realizar la ejecución del segundo paso ocurrirán las siguientes acciones:

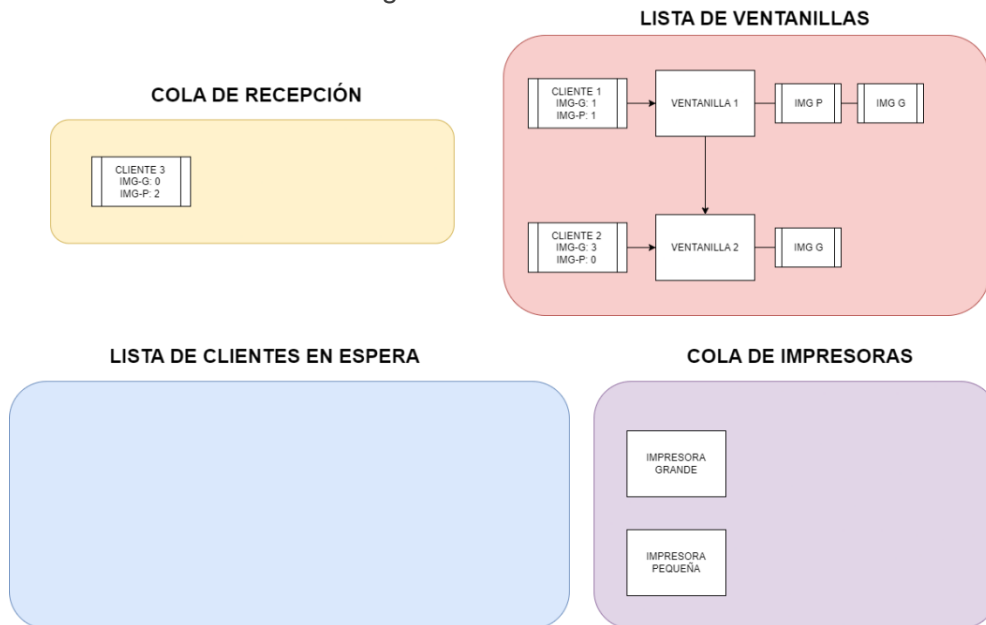
- El cliente 2 ingresa a la ventanilla 2 (la primera disponible).
- La ventanilla 1 recibe una imagen del cliente 1



### Paso 3

Al realizar la ejecución del tercer paso ocurrirán las siguientes acciones:

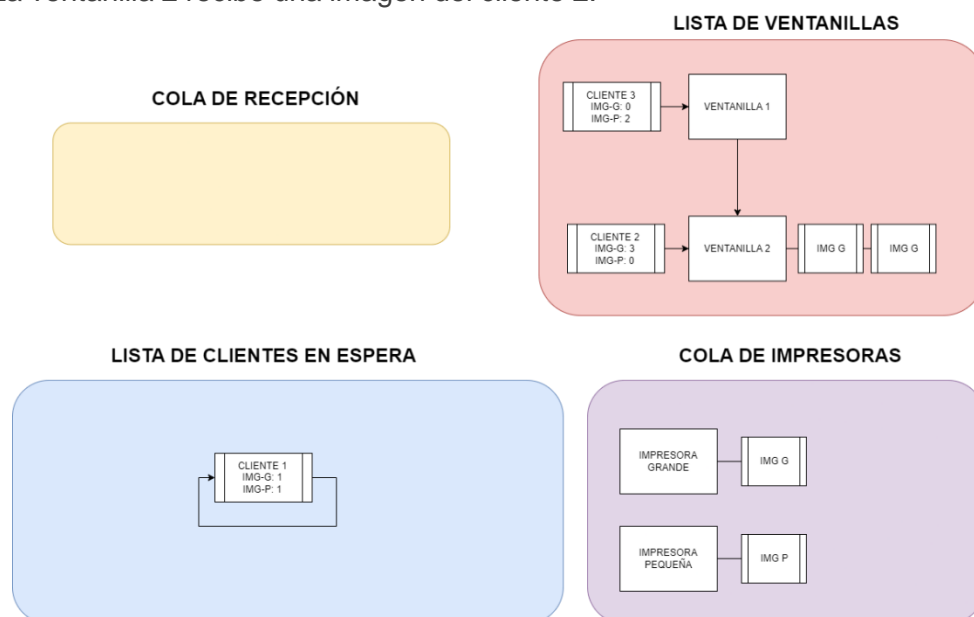
- La ventanilla 1 recibe una imagen del cliente 1
- La ventanilla 2 recibe una imagen del cliente 2



### Paso 4

Al realizar la ejecución del cuarto paso ocurrirán las siguientes acciones:

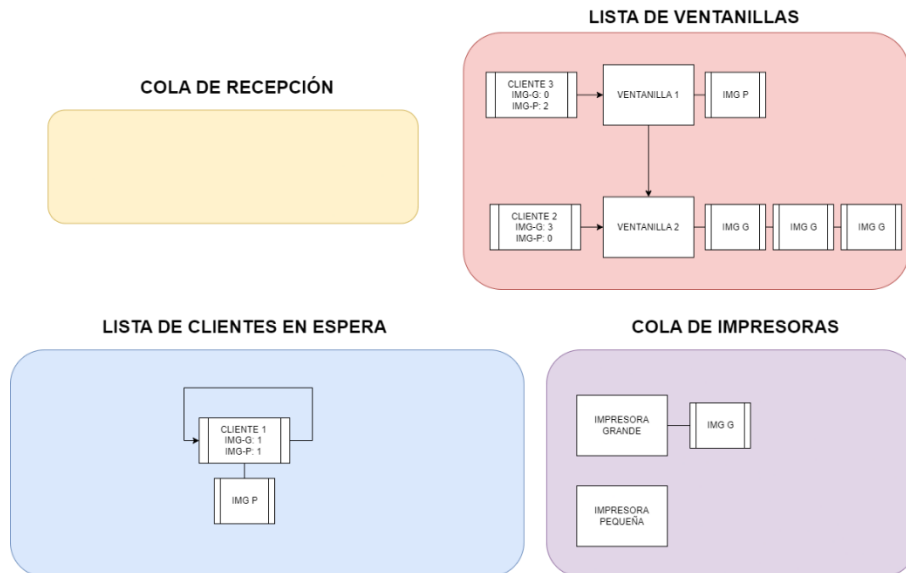
- El cliente 1 es atendido e ingresa a la lista de espera.
- La ventanilla 1 envía las imágenes del cliente 1 a sus respectivas colas de impresión.
- El cliente 3 ingresa a la ventanilla 1 (la primera disponible).
- La ventanilla 2 recibe una imagen del cliente 2.



### Paso 5

Al realizar la ejecución del quinto paso ocurrirán las siguientes acciones:

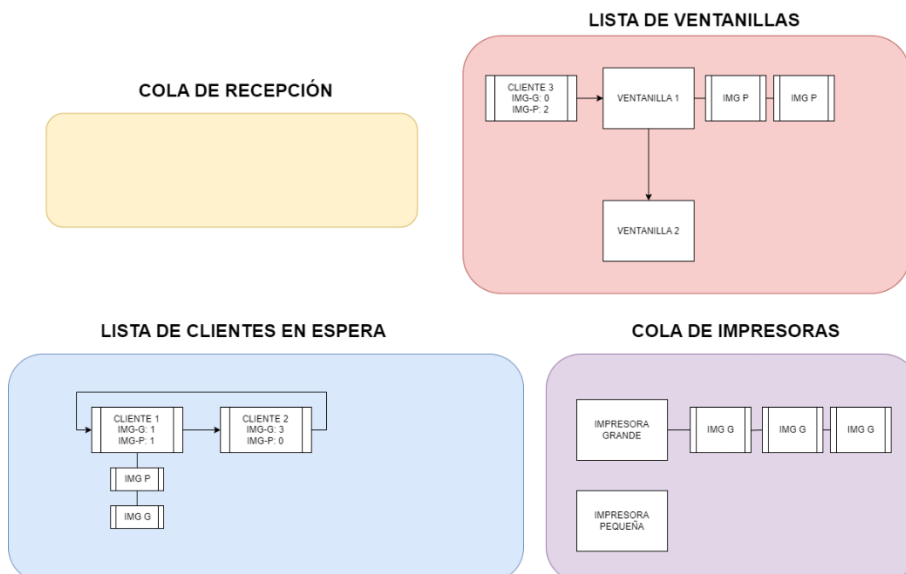
- La ventanilla 1 recibe una imagen del cliente 3.
- La ventanilla 2 recibe una imagen del cliente 2.
- Se completa la impresión de una imagen en pequeña y se le entrega al cliente que la solicitó. (tiempo: 1 paso)



### Paso 6

Al realizar la ejecución del sexto paso ocurrirán las siguientes acciones:

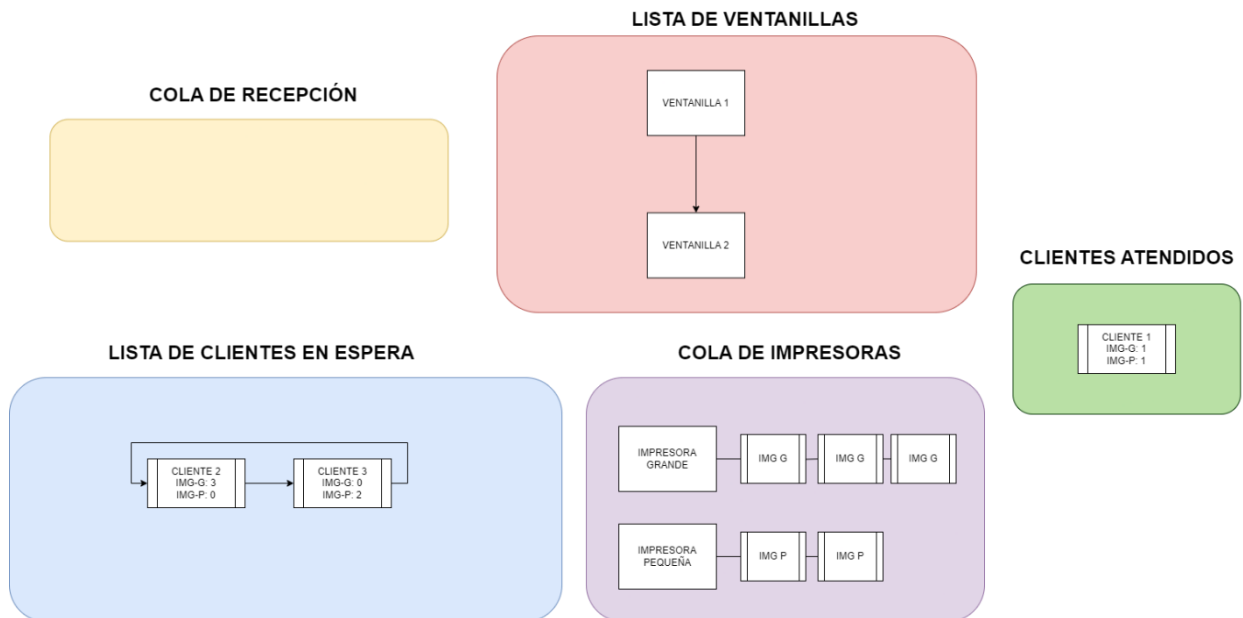
- El cliente 2 es atendido e ingresa a la lista de espera.
- La ventanilla 2 envía las imágenes del cliente 2 a sus respectivas colas de impresión.
- La ventanilla 1 recibe una imagen del cliente 3.
- Se completa la impresión de una imagen grande y se le entrega al cliente que la solicitó. (tiempo: 2 pasos)



## Paso 7

Al realizar la ejecución del séptimo paso ocurrirán las siguientes acciones:

- El cliente 3 es atendido e ingresa a la lista de espera.
- La ventanilla 2 envía las imágenes del cliente 3 a sus respectivas colas de impresión.
- El cliente 1 ya posee todas sus imágenes impresas y sale de la empresa registrando el tiempo total dentro de ella (cantidad de pasos)



**Nota:** La aplicación debe mostrar en consola las operaciones que se realizan en cada paso de la ejecución del programa.

-----PASO 1-----  
-----

EL CLIENTE 1 INGRESA A LA VENTANILLA 1

-----PASO 2-----  
-----

EL CLIENTE 2 INGRESA A LA VENTANILLA 2  
LA VENTANILLA 1 RECIBIÓ UNA IMAGEN

# Reportes

## Visualización de estructuras:

Se debe incluir un módulo en el cual se podrá visualizar el estado de las estructuras implementadas en cualquier momento de la ejecución de la aplicación. Los reportes serán generados utilizando la herramienta **Graphviz**

## Datos generados

1. Top 5 de clientes con mayor cantidad de imágenes grandes.
2. Top 5 de clientes con menor cantidad de imágenes pequeñas.
3. Información del cliente que más pasos estuvo en el sistema.
4. Datos de un cliente en específico, se debe incluir la información del cliente, así como el detalle de todas las imágenes entregadas para impresión

## Observaciones

- Lenguaje de Programación: FORTRAN
- Sistema Operativo: Elección del estudiante.
- Parser de JSON: Elección del estudiante.
- El IDE a utilizar queda a discreción del estudiante.
- La aplicación será compilada y ejecutada al momento de la calificación
- El estudiante debe tener un repositorio privado en GitHub con el nombre EDD\_PROYECTO \_#carné y agregar a su tutor como colaborador al repositorio del proyecto (cada tutor les hará llegar su usuario).
- Fecha de Entrega: miércoles 28 de febrero, a las 23:59 horas.
- Entrega únicamente en la plataforma UEDI en un archivo comprimido (rar|zip).
- Las copias tendrán nota de 0 puntos y serán reportadas al catedrático y a la escuela de sistemas.

## Entregables:

- Manual de Usuario
- Manual Técnico
- Link a repositorio con el código fuente

## Restricciones

- Las estructuras deben de ser desarrolladas por los estudiantes sin el uso de ninguna librería o estructura predefinida en el lenguaje a utilizar.
- No se permite la modificación de código durante la calificación, únicamente se calificará sobre el commit que el estudiante elija **siempre y cuando esté dentro del horario de entrega establecido.**