

Manual Técnico Proyecto 01 - Estructura De Datos

[Compatibilidad](#)

[Especificaciones](#)

[Lenguaje Utilizado](#)

[IDE y JDK utilizado](#)

[Estructura Del Proyecto](#)

[Documentación](#)

[Reportes Img](#)

[Reportes Texto](#)

[UDrawingPaper](#)

[Attended List](#)

[Attended List Node](#)

[Client](#)

[Client Window](#)

[Image](#)

[Image Linked List](#)

[Linked List](#)

[Printer](#)

[Printer Queue](#)

[Queue](#)

[Stack List](#)

Waiting List

Window

Compatibilidad

-Windows 7,8,10

-Linux

Especificaciones

En esta sección se mencionará lo que debes conocer de la aplicación.

Lenguaje Utilizado









El lenguaje utilizado para la realización de este proyecto fue Java.

IDE y JDK utilizado

El IDE utilizado fue NetBeans en su versión 8.2, y el JDK utilizado fue 1.8.0

Estructura Del Proyecto

El proyecto estaba dividido en diferentes carpetas, las cuales se describirán a continuación.

 Documentación	9/02/2022 18:15	Carpeta de archivos	
 Reportes Img	13/02/2022 19:02	Carpeta de archivos	
 Reportes Texto	10/02/2022 17:00	Carpeta de archivos	
 UDrawingPaper	2/02/2022 16:24	Carpeta de archivos	
 .gitignore	2/02/2022 16:24	Documento de tex...	1 KB
 ejemplo.txt	8/02/2022 16:54	Documento de tex...	1 KB
 LICENSE	2/02/2022 09:03	Archivo	12 KB
 README.md	2/02/2022 09:03	Archivo MD	1 KB

Documentación

La primera carpeta visible, esta contiene el manual de usuario y el manual técnico, además se adjuntó el enunciado del proyecto.

Reportes Img























Carpeta donde se almacenan los reportes generados con graphviz.

Reportes Texto

Carpeta donde se almacenan los archivos de texto plano con el código generado por graphviz.

UDrawingPaper

Carpeta donde contiene todo el código fuente de la aplicación, en este momento se muestra una imagen de como está estructurado el código.

 AttendedList.java	13/02/2022 17:09	Archivo JAVA	13 KB
 AttendedListNode.java	6/02/2022 21:28	Archivo JAVA	1 KB
 Client.java	10/02/2022 21:25	Archivo JAVA	5 KB
 ClientDAO.java	2/02/2022 16:45	Archivo JAVA	1 KB
 ClientWindow.java	10/02/2022 21:48	Archivo JAVA	11 KB
 ClientWindowNode.java	6/02/2022 19:18	Archivo JAVA	1 KB
 Image.java	5/02/2022 13:57	Archivo JAVA	1 KB
 ImageLinkedList.java	10/02/2022 17:02	Archivo JAVA	4 KB
 ImageNode.java	7/02/2022 16:01	Archivo JAVA	1 KB
 LinkedList.java	8/02/2022 17:09	Archivo JAVA	8 KB
 LinkedListNode.java	5/02/2022 15:55	Archivo JAVA	1 KB
 Printer.java	7/02/2022 16:17	Archivo JAVA	2 KB
 PrinterQueue.java	13/02/2022 16:49	Archivo JAVA	5 KB
 PrinterQueueNode.java	7/02/2022 16:38	Archivo JAVA	1 KB
 Queue.java	8/02/2022 20:51	Archivo JAVA	5 KB
 QueueNode.java	3/02/2022 17:51	Archivo JAVA	1 KB
 StackList.java	8/02/2022 12:57	Archivo JAVA	4 KB
 StackListNode.java	5/02/2022 14:04	Archivo JAVA	1 KB
 UDrawingPaper.java	13/02/2022 19:03	Archivo JAVA	28 KB
 WaitingList.java	10/02/2022 21:49	Archivo JAVA	9 KB
 WaitingNode.java	7/02/2022 16:08	Archivo JAVA	1 KB
 Window.java	5/02/2022 20:19	Archivo JAVA	1 KB

Attended List

Lista enlazada de los clientes atendidos.

Attended List Node

Clase del nodo utilizado en la clase de Attended List, donde en cada nodo recibe un cliente.

Client

Clase donde se define el objeto de tipo cliente con sus atributos, además de generar sus getter y setter.

Client Window

Clase que se encarga del control de la relación ventanilla cliente

Image

Clase donde se define el objeto imagen con sus atributos, además de generar sus getter y setter.

Image Linked List

Clase donde los clientes pueden obtener las imágenes que le corresponden a cada uno.

Linked List

Lista enlazada de ventanillas, en cada nodo recibe un objeto de tipo ventanilla que a su vez tiene una pila, por lo que la podemos llamar Lista de Pilas.

Printer

Clase donde se define el objeto impresora con sus atributos, además de generar sus getter y setter.

Printer Queue

Clase que en cada nodo recibe un objeto imagen, la cual es procesada y mandada a la cola de impresión de las impresoras.

Queue

Cola de recepción de clientes.

Stack List

Clase que define las pilas, donde en cada nodo recibe un objeto de tipo imagen.

Waiting List

Lista doblemente enlazada circular, la cual almacena a los clientes en la lista de espera.

Window

Clase donde se define el objeto Ventanilla con sus atributos, además de generar sus getter y setter.