

# UNIVERSIDAD TÉCNICA PARTICULAR DE LOJA



## SISTEMAS OPERATIVOS PROYECTO FINAL

Algoritmos de Planificación: FIFO - FCFS (First come First served)

### Integrantes:

- Gilmar Alexander Jiménez Medina
- Diana Rocio Bermeo Cabrera
- Ariel Cruz Sarango

**Fecha:** 01-12-2021

**Docente:** Ing. Cristian M. Chamba. C.

## Contenido

|  |    |
|--|----|
| TEMA .....                                 | 2  |
| OBEJTIVO.....                              | 2  |
| INTRODUCCION .....                         | 2  |
| MARCO CONCEPTUAL .....                     | 3  |
| PLANIFICACIÓN DE LA CPU .....              | 3  |
| DISCIPLINA DE PLANIFICACIÓN .....          | 3  |
| ALGORITMOS DE PLANIFICACIÓN .....          | 4  |
| TIPOS DE ALGORITMOS DE PLANIFICACIÓN ..... | 4  |
| CARACTERÍSTICAS DEL MÉTODO FCFS.....       | 5  |
| VENTAJAS DE FCFS .....                     | 5  |
| DESVENTAJAS DE FCFS .....                  | 5  |
| FUNCIONAMIENTO DEL ALGORITMO FCFS.....     | 6  |
| DESARROLLO .....                           | 7  |
| NetBeans.....                              | 7  |
| Lenguaje de Programación Java.....         | 8  |
| Programación Orientada a Objetos .....     | 9  |
| Interfaz gráfica en java.....              | 9  |
| CONCLUSIONES.....                          | 10 |
| GLOSARIO.....                              | 10 |
| ANEXOS.....                                | 11 |
| BIBLIOGRAFÍAS .....                        | 12 |

## **TEMA**

Desarrollo del algoritmo de planificación FIFO (FCFS, First–Come–First–Served) en un lenguaje de programación y en el entorno de desarrollo opcional.

## **OBJETIVO**

- Implementar en el entorno de desarrollo gráfico NetBeans, con lenguaje de programación JAVA, el algoritmo de planificación FIFO (FCFS, First–Come–First–Served).

## **INTRODUCCION**

El planificador es un componente funcional muy importante de los sistemas operativos multiusuario y multiproceso, y es esencial en los sistemas operativos de tiempo real. Su función consiste en impartir el tiempo disponible de un microprocesador entre todos los procesos que están disponibles para su ejecución.

El principal objetivo de la planificación es repartir el tiempo del procesador de forma que se optimicen algunos puntos del comportamiento del sistema. Generalmente se fija un conjunto de criterios con los que evaluar las diversas estrategias de planificación. Sería conveniente disponer de una política de planificación que ofrezca un buen servicio a diversos usuarios.

Otros criterios están orientados al sistema, esto es, se centran en el uso efectivo y eficiente del procesador. Un ejemplo puede ser la productividad, es decir, el ritmo con el que los procesos terminan. La productividad es una medida muy válida del rendimiento de un sistema y que sería deseable maximizar.

El presente proyecto surge por el interés de realizar una aplicación que permita controlar y visualizar el desarrollo del algoritmo de planificación FIFO implementado en un lenguaje de programación, fue elaborado durante el primer bimestre del

periodo académico octubre 2021 – febrero 2022, de la asignatura de Sistemas Operativos de la carrera de Ingeniería en Computación. (Williams, 2021)

De forma general el proyecto consistió en implementar en un lenguaje de programación el desarrollo del algoritmo FIFO (FCFS, First–Come–First–Served).

Para llevar a buen término el proyecto, se han delimitado los siguientes bloques de trabajo: *(First Come First Serve(FCFS) Scheduling Algorithm | Studytonight, s. f.)*

- Estudio sobre el algoritmo de planificación FIFO (FCFS, First–Come–First–Served), cuáles son sus características, importancia y como se ejecuta su desarrollo.
- Analizar en que lenguaje de programación podemos implementar el algoritmo de planificación.
- Se procederá a la programación del algoritmo en java y en el entorno de desarrollo NetBeans. Es el objetivo del proyecto.

## MARCO CONCEPTUAL

### PLANIFICACIÓN DE LA CPU

Conjunto de técnicas y algoritmos que se incorporan a los Sistemas Operativos Multiprogramados. Gobierna el orden en que se ejecutan los procesos.

### DISCIPLINA DE PLANIFICACIÓN

- Ser equitativa: debe intentar hacer una planificación justa, esto es, se debe tratar a todos los procesos de la misma forma y no aplazar indefinidamente ningún proceso.
- Ser eficiente: debe maximizar el uso de los recursos tales como intentar que la ocupación de la CPU sea máxima.

- Lograr un tiempo bueno de respuesta.
- Lograr un tiempo de proceso global predecible.
- Elevar al máximo la productividad o el rendimiento, esto es, maximizar el número de trabajos procesados por unidad de tiempo. prioridad.

Estos criterios son dependientes entre sí y es imposible optimizar todos de forma simultánea. (Pozo, 2016)

### ALGORITMOS DE PLANIFICACIÓN

Cuando un proceso es ejecutable, el sistema operativo debe decidir cuál de ellos, debe ejecutarse en primer término. Esa parte del sistema operativo que debe llevar a cabo esa decisión se llama el planificador y el algoritmo que utiliza se llama algoritmo de planificación.

### TIPOS DE ALGORITMOS DE PLANIFICACIÓN

Algunos algoritmos se adaptan mejor a las necesidades de un tipo determinado de planificador. Según que se use una disciplina determinada de planificación por un planificador a largo o corto plazo. Estos algoritmos pueden ser: por prioridad, por colas múltiples, por el primero el más corto, por expropiación basada en prioridades y orden de llegada. (Pozo, 2016)

- **Planificación de servicio por orden de llegada (FCFS, First-Come-First-Served).**

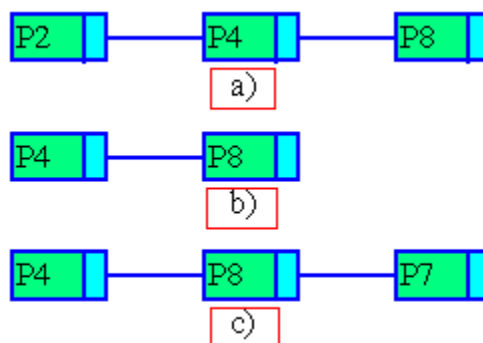


Gráfico 1 Lista de Procesos Preparados en FIFO

Es un algoritmo de programación del sistema operativo que ejecuta automáticamente solicitudes y procesos en cola en orden de llegada.

La política más simple de planificación es la FCFS. A medida que un proceso pasa al estado listo, este es agregado a la cola de listos. Cuando el proceso que actualmente está ejecutando cesa su ejecución entonces el proceso más viejo en la cola es seleccionado para correr. La implementación de esta política es a través de colas FIFO (First-In, First-Out). ¿Cómo actúa? En este tipo de algoritmo, los procesos que solicitan a la CPU primero obtienen la asignación de CPU primero. Esto se gestiona con una cola FIFO. A medida que el proceso ingresa a la cola lista, su PCB se vincula con la cola de la cola y, cuando la CPU queda libre, debe asignarse al proceso al comienzo de la cola.

#### CARACTERÍSTICAS DEL MÉTODO FCFS

Es no apropiativo: cada vez que ejecute un proceso va a respetar su tiempo de duración, el proceso se ejecuta completo. No puede haber interrupción. Compatible con el algoritmo de programación preventiva y no preventiva.

#### VENTAJAS DE FCFS

- ❖ La forma más simple de un algoritmo de programación de CPU.
- ❖ Fácil de programar.
- ❖ El primero en llegar es el primero en ser servido.

#### DESVENTAJAS DE FCFS

- ❖ Es un algoritmo de programación de CPU no preventivo, por lo que una vez que el proceso se ha asignado a la CPU, nunca liberará la CPU hasta que termine de ejecutarse.
- ❖ El tiempo medio de espera es elevado.
- ❖ Los procesos cortos que están al final de la cola tienen que esperar a que finalice el proceso largo al principio.
- ❖ No es una técnica ideal para los sistemas de tiempo compartido.
- ❖ Debido a su simplicidad, FCFS no es muy eficiente.

## FUNCIONAMIENTO DEL ALGORITMO FCFS

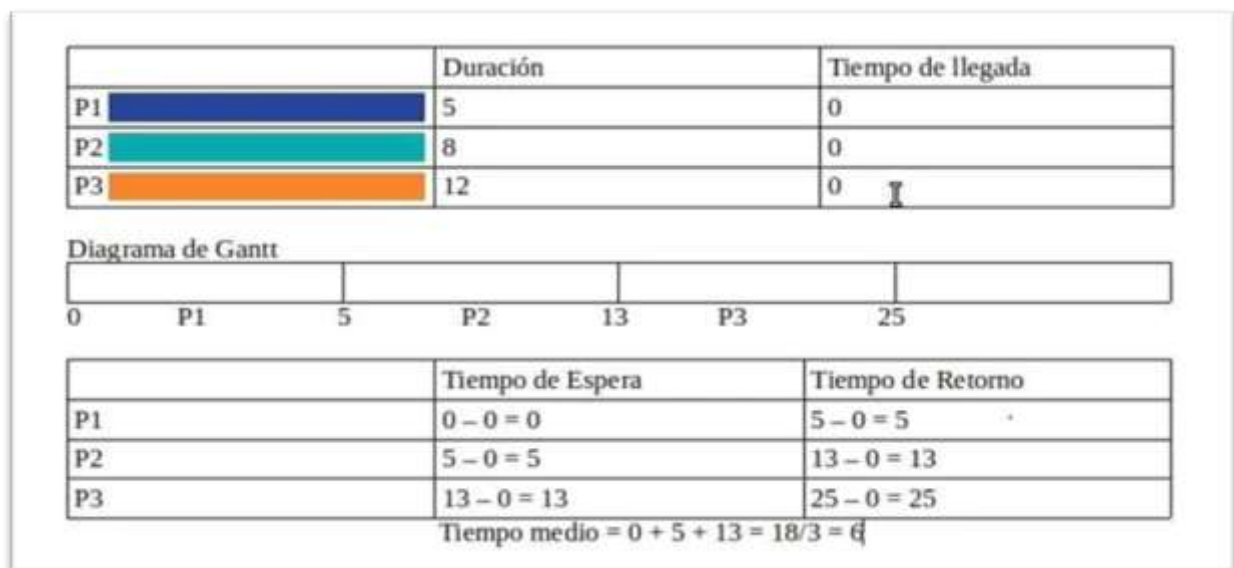


Gráfico 2 Ejemplo tomado de la web de como actual el algoritmo First-Come-First-Served con los mismos tiempos de llegada.

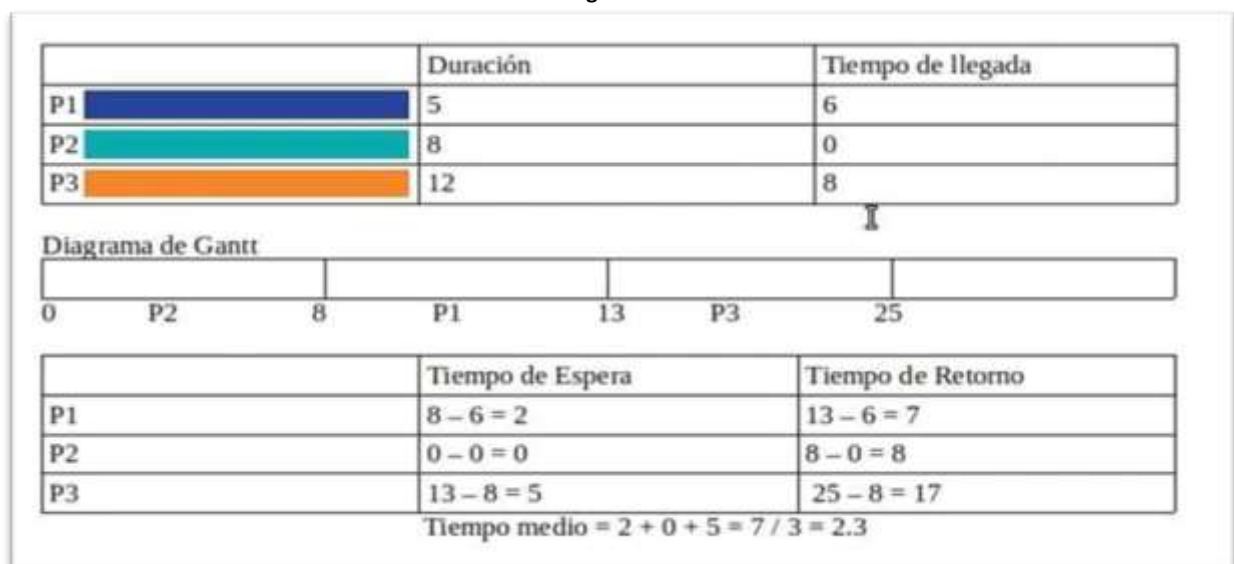


Gráfico 3 Ejemplo tomado de la web de como actual el algoritmo First-Come-First-Served con diferentes tiempos de llegada.

En este tipo de algoritmo, los procesos pueden tener el mismo tiempo de llegada o pueden llegar en diferentes tiempos de llegada u orden. Cuando tenemos el mismo tiempo de llegada, el primer proceso en ejecutar es como se presenta en la lista de evaluación de procesos, es decir se ejecutan como lleguen, si tenemos distinto tiempos de llegada la prioridad del proceso o el proceso que se ejecute primero es del que tenga el menor tiempo de llegada. (Planificación de procesos - Algoritmo FIFO - FCFS (First come First served), 2019)

- **Para calcular el tiempo de espera:** Momento de ejecución del proceso menos el tiempo de llegada.
- **Para calcular el tiempo de retorno:** Momento de respuesta del proceso menos el tiempo de llegada.

El tiempo de respuesta es el periodo de tiempo transcurrido desde que se emite una solicitud hasta que la respuesta aparece en la salida.

- **Para calcular el tiempo medio de espera o de retorno:** Suma de los tiempos sea ya de espera o de retorno dividido para el número de procesos.

## DESARROLLO

Nuestro proyecto consiste en escribir el código en una clase de java, en el entorno de desarrollo NetBeans, después elaborar ventanas para la interfaz grafica del algoritmo de planificación, seguidamente mediante la programación orientada a objetos relacionar estos componentes para mostrar al usuario final un tipo de aplicación que simule la ejecución de un algoritmo de planificación FIFO.

Para la realización de este proyecto utilizaremos:

### NetBeans

Entorno de desarrollo integrado libre, orientado principalmente al desarrollo de aplicaciones Java. La plataforma NetBeans permite el desarrollo de aplicaciones estructuradas mediante un conjunto de componentes denominados “módulos”. Cada uno de estos módulos sería un archivo Java conteniendo un conjunto de clases que interactuarán con las APIs de NetBeans.

Esta plataforma incluye servicios para el control del interfaz de usuario, la configuración, el almacenamiento, las ventanas, etc. (NetBeans, s. f.)



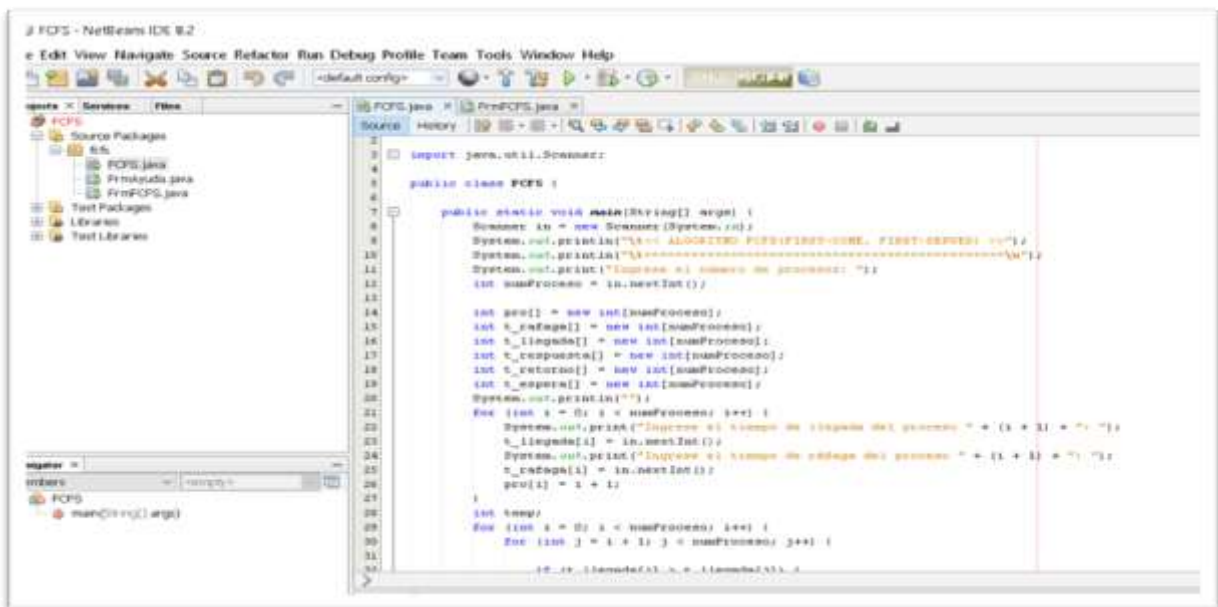


Gráfico 4 Entorno de Desarrollo NetBeans.

## Lenguaje de Programación Java

Java es un lenguaje de programación y una plataforma informática comercializada por primera vez en 1995 por Sun Microsystems. Java es rápido, seguro y fiable.

Java Runtime Environment (JRE) es lo que se obtiene al descargar el software de Java y es la parte de tiempo de ejecución del software de Java, que es todo lo que necesita para ejecutarlo en el explorador web. (NetBeans, s. f.)



Gráfico 5 Código escrito en lenguaje JAVA

## Programación Orientada a Objetos

La Programación Orientada a Objetos (POO) es un paradigma de programación, es decir, un modelo o un estilo de programación que nos da unas guías sobre cómo trabajar con él. Se basa en el concepto de clases y objetos.

La Programación Orientada a Objetos es actualmente el paradigma que más se utiliza para diseñar aplicaciones y programas informáticos. Son muchas sus ventajas, principalmente cuando necesitas resolver desafíos de programación complejos. Permite una mejor estructura de datos y reutilización del código, lo que facilita el ahorro de tiempo a largo plazo.

```
private void btnAyudaActionPerformed(java.awt.event.ActionEvent evt) (  
    //boton ayuda llamará al formulario ayuda  
    FrmAyuda frm = new FrmAyuda();  
    frm.setVisible(true);  
)
```

Gráfico 6 Código de programación orientada a objetos

## Interfaz gráfica en java

Java Swing es una herramienta de interfaz gráfica de usuario (GUI) ligera que incluye un amplio conjunto de widgets. Incluye paquete que le permite crear componentes de GUI para sus aplicaciones Java, y es independiente de la plataforma. AWT (Abstract Windows Toolkit) es la parte de Java que se emplea para construir interfaces gráficas de usuario.

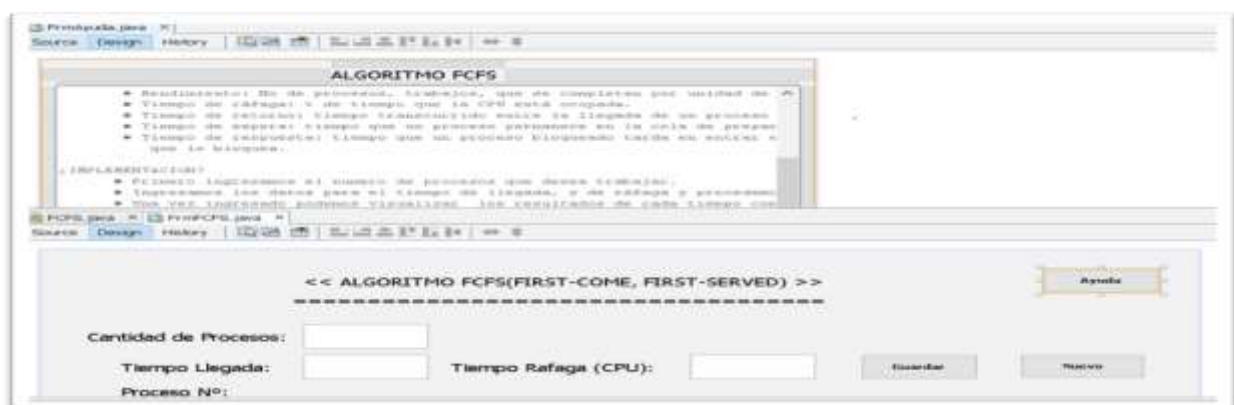


Gráfico 7 Creación de Interfaz Gráfica - Ventanas

## CONCLUSIONES

- La programación orientada a objetos permite la reutilización del código para futuras tareas o proyectos relacionados con la programación, asemejándolo al mundo real con el uso de las ventanas o JFrame que brinda el entorno de desarrollo NetBeans.
- Si bien un esquema FCFS reduce al mínimo la sobrecarga dependiente, el rendimiento dividido por los últimos procesos en llegar (o por procesos cortos llegados en un momento inconveniente) resulta injusto.
- Los algoritmos de planificación se van adecuando según el tiempo de ejecución que tenga el CPU, donde algunos son más rápidos que otros.
- La ventaja de este algoritmo es su fácil ejecución, sin embargo, no es válido para entornos interactivos ya que un proceso de mucho cálculo de CPU hace agrandar el tiempo de espera de los demás procesos.
- En general, los algoritmos de planificación se basan en cómo se adecuan los procesos dependiendo de que tan grandes sea su tiempo de ejecución, todos se acomodan en una cola o con interrupciones de por medio.

## GLOSARIO

- **Swing:** Subsistema de Java encargado de la creación de interfaces gráficas portables al 100% entre distintos sistemas.
- **PCB:** Bloque de control de proceso.
- **GUI:** Interfaz gráfica de usuario, programa informático que actúa de interfaz de usuario.

# ANEXOS

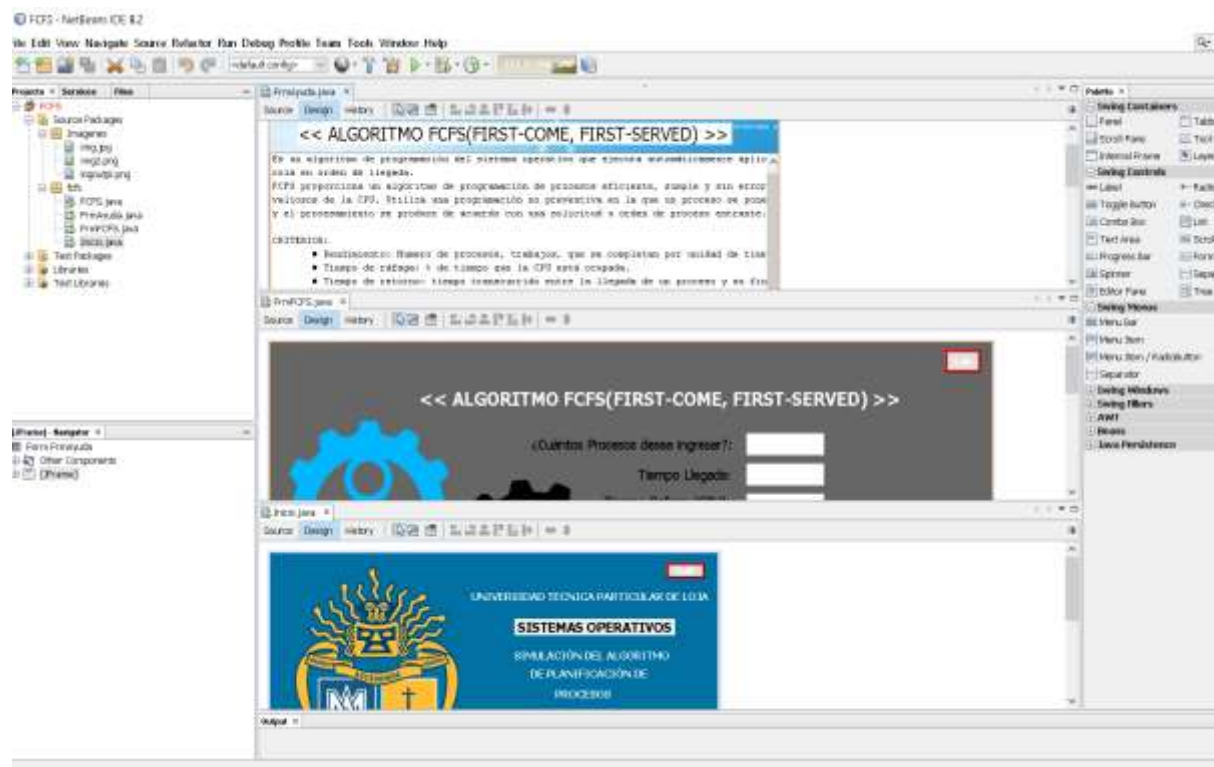


Gráfico 8 Diseño de las tres ventanas de interfaz gráfica del proyecto

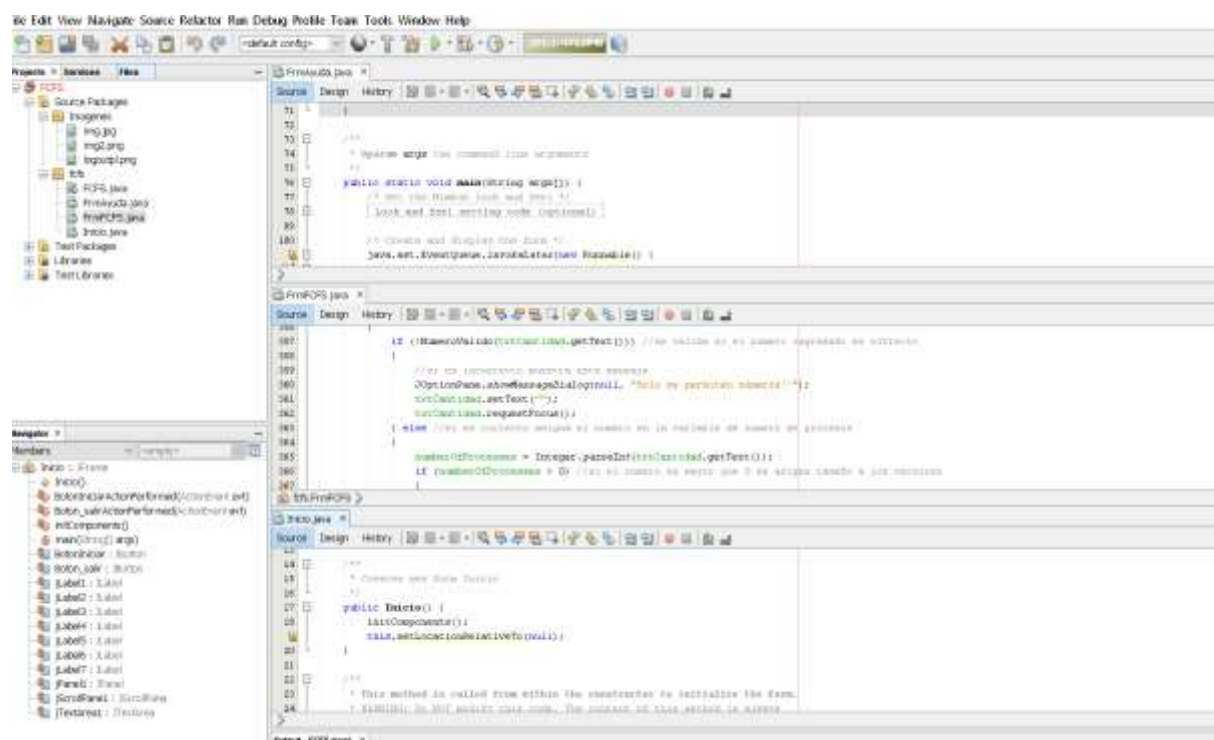


Gráfico 9 Código para el desarrollo del algoritmo y la relacion de las tres ventanas gráficas

## BIBLIOGRAFÍAS

Williams, L. (2021, 6 octubre). *FCFS Scheduling Algorithm: What is, Example Program*. Guru99.

<https://www.guru99.com/fcfs-scheduling.html>

*First Come First Serve(FCFS) Scheduling Algorithm | Studytonight*. (s. f.). Studytonight.

<https://www.studytonight.com/operating-system/first-come-first-serve>

*Planificación de procesos - Algoritmo FIFO - FCFS (First come First served)*. (2019, 14 junio).

[Vídeo]. YouTube. <https://www.youtube.com/watch?v=3R2i3Mnmzes>

Pozo, D. (2016, 14 septiembre). *Procesos: Algoritmos de Planificación*. Ciencia de la Computación.

<https://danielpozoblog.wordpress.com/2016/09/14/procesos-algoritmos-de-planificacion/>

NetBeans, A. (s. f.). *NetBeans Java Language Infrastructure Tutorial*. ApacheNetBeans.

<https://netbeans.apache.org/tutorials/60/nbm-copyfq.html>