

* David Alexander Cardenas Almendares 20121003387
* Cristian Alexander Martínez Ochoa 20131015700
* Ariel Isaí Turcios García 20131006640

INTEGRANTES:

**Proyecto Sistemas Operativos I**

**Simulador de Gestor de Procesos**

**PERIODO 1 2020**



Manual Técnico

**Contenido**

1. Objetivos ………………………………………………………………………………….. 3

1.1 Objetivos específicos …………………………………………………………. 3

1. Alcance …………………………………………………………………………………… 3
2. Requerimientos técnicos ……………………………………………………….. 4

3.1 Software ………………………………………………………………………….. 4

3.2 Hardware ………………………………………………………………………… 4

1. Herramientas utilizadas para el desarrollo …………………………… 5
2. Instalación ……………………………………………………………………………. 5
3. Configuración ………………………………………………………………………… 6
4. Diccionario de clases y métodos ………………………………………………………………. 6 -

**1. Objetivos.**

Se ha creado dicho documento con el propósito de mostrar cómo fue diseñada la aplicación,

Y al mismo tiempo dar referencias de como interactuar con el programa.

Todo esto para futuras actualizaciones o para darle mantenimiento por otro programador.

Se especifica su creación, proceso de instalación, código fuente etc…

**1.1 Objetivos específicos.**

1. Comprende los pasos a seguir para gestionar los procesos por parte del

sistema operativo.

2. Desarrollar un gestor de procesos en base al modelo de los 5 estados.

**2. Alcance.**

Este documento va dirigido a un programador con conocimientos medios de lenguaje java

además de los conocimientos de la clase de sistemas operativos uno relacionado con los

con el modelo de los 5 estados.

**3. Requerimientos técnicos.**

**Software:**

* Java, JDK
* IDE NetBeans o Eclipse
* Computadora con Windows 7 o superior

**Hardware:**

* Una computadora completa (bocinas no necesarias)

**3.1 Requerimientos mínimos de hardware.**

* Procesador: Intel Inside 1.5 Ghz
* Memoria RAM (mínimo): 512 mb
* Disco duro: 10gb

**3.2 Requerimientos mínimos de software**

* Privilegios de administrador: si
* Sistema operativo: Windows 7 o superior

**4. Herramientas utilizadas para el desarrollo.**

* IDE NetBeans 8.2
* Documentación de java en línea
* Editor de texto
* Documentación de la clase acerca de los 5 estados

**5. Instalación.**

* No requiere de una instalación ya que se es un archivo ejecutable



**Pantalla principal de la aplicación**

**6. Configuración.**

No hay una configuración general, la aplicación en si viene configurada y los comandos son intuitivos

para el usuario.

**7. Diccionario de clases y métodos**

Se utilizó una única clase llamada **Procesar.java** (es un Jframe, para manejo de interfaces graficas en java) esta contiene todas las variables, así como también los métodos utilizados en el desarrollo del proyecto.

**import** java.awt.Color;

**import** javax.swing.JOptionPane;

**import static** java.lang.Integer.parseInt;

**import** java.util.logging.Level;

**import** java.util.logging.Logger;

**import** javax.swing.table.DefaultTableModel;

**import** Atxy2k.CustomTextField.RestrictedTextField;

\*\* Se deben de importar todas estas bibliotecas para un correcto funcionamiento de la aplicación.

**public class** Procesar **extends** javax.swing.JFrame {

int Contador;//Contador del total de procesos que se van ingresando

int NProceso;//Carga el número de procesos ejecutándose

int IdProceso;//Carga el ID del proceso

int Rafaga=0;//Carga la ráfaga en ejecución

int Quantum=0;//Carga el Intervalo - Periodo (quantum) en ejecución

int ResiduoRafaga=0;//Carga el residuo en ejecución

int TiempoProceso=0;//Carga el tiempo que se dura procesando

int ValorBarra;//Carga el progreso de la Barra

int CantidadProcesos;//Número de procesos terminados

.

.

.

}

/\*\*

\* Creates new form Procesar

\*/

public Procesar() {

initComponents();

jTIngreso.setBackground(Color.CYAN);

jTIngreso.setForeground(Color.blue);

jTFinal.setBackground(Color.green);

//jTFinal.setBackground(Color.red);

jTFCapturaQuantum.grabFocus();

RestrictedTextField r = new RestrictedTextField(jTFCapturaID);

r.setOnlyNums(true);

r.setLimit(4);

RestrictedTextField r2 = new RestrictedTextField(jTFCapturaRafaga);

r2.setOnlyNums(true);

r2.setLimit(3);

}