

# **Manual de Usuario**

Simulador de Planificación de Procesos

Versión 2.0

Sistema Mejorado con Validación Robusta

Proyecto Académico - Sistemas Operativos

# Introducción

El **Simulador de Planificación de Procesos v2.0** es una aplicación de consola que permite simular y comparar el comportamiento de cuatro algoritmos clásicos de planificación de CPU. Esta versión incluye mejoras significativas en validación de entradas, manejo de errores y experiencia de usuario.

## Características Principales

- **Cuatro algoritmos implementados: FCFS, SJF, Round Robin y Prioridad**
- **Dos modos de entrada: Manual (con validación exhaustiva) y desde archivo CSV**
- **Validación robusta: Todas las entradas son validadas en tiempo real**
- **Manejo inteligente de errores: El sistema permite reintentar sin cerrar la aplicación**
- **Interfaz amigable: Mensajes claros y limpieza de pantalla automática**
- **Ciclo continuo: Ejecutar múltiples simulaciones sin reiniciar el programa**
- **Métricas completas: Espera, retorno, respuesta y utilización de CPU**

## Algoritmos Implementados

Algoritmo	Tipo	Ventaja	Desventaja
<b>FCFS</b> (First Come First Served)	No expropiativo	Simple, justo en orden	Efecto convoy
<b>SJF</b> (Shortest Job First)	No expropiativo	Minimiza tiempo de espera	Inanición en procesos largos
<b>Round Robin</b>	Expropiativo	Equitativo, previene inanición	Overhead por cambios de contexto
<b>Prioridad</b>	No expropiativo	Prioriza tareas críticas	Riesgo de inanición

# Inicio Rápido

## Paso 1: Iniciar el Simulador

### Windows:

```
simulador.exe
```

### Linux/Mac:

```
./simulador
```

**Tip:** El programa muestra un menú claro con dos opciones principales: Entrada Manual o Carga desde Archivo.

## Paso 2: Seleccionar Modo de Entrada

```
SIMULADOR DE PLANIFICACION

Cual va a ser la entrada de datos:
1. Manual
2. Archivo
Opcion: _
```

# Modo 1: Entrada Manual de Datos

## Validaciones Aplicadas

Campo	Tipo	Validación	Ejemplo Válido
Nombre/ID	String	No vacío, permite espacios	Proceso_A , P1
Tiempo Llegada	Entero	$\geq 0$	0 , 5 , 10
Tiempo Ráfaga	Entero	$> 0$	1 , 10 , 20
Prioridad	Entero	$\geq 0$	1 , 3 , 5

### Ejemplo Completo:

```
Cuantos procesos vas a registrar?: 3

--- Proceso 1 ---
Nombre/ID: P1
Tiempo de Llegada: 0
Tiempo de Rafaga: 10
Prioridad: 3

--- Proceso 2 ---
Nombre/ID: P2
Tiempo de Llegada: 2
Tiempo de Rafaga: 5
Prioridad: 1
```

## Modo 2: Carga desde Archivo CSV

### Formato del Archivo

Característica	Especificación
Tipo de archivo	Texto plano <code>.csv</code> o <code>.txt</code>
Separador	Coma ( , )
Encabezados	NO incluir
Líneas vacías	Permitidas (se ignoran)

### Estructura: ID,TiempoLlegada,TiempoRafaga,Prioridad

#### Ejemplo de archivo válido:

```
P1,0,10,3
P2,2,5,1
P3,4,8,2
P4,6,3,4
P5,8,7,2
```

**Ventaja del modo archivo:** Si falla la carga, el programa NO se cierra, sino que vuelve al menú para reintentar.

# Visualización de Resultados

## Tabla de Resultados Detallada

ID	Llegada	Ejecucion	Prioridad	Inicio	Fin	Espera	Ret
P1	0	10	3	0	10	0	
P2	2	5	1	10	15	8	
P3	4	8	2	15	23	11	

## Métricas Explicadas

- Espera:** Tiempo que esperó en cola (Retorno - Ejecución)
- Retorno:** Tiempo total en el sistema (Fin - Llegada)
- Respuesta:** Tiempo hasta primera ejecución (Inicio - Llegada)
- Utilización CPU:** Porcentaje del tiempo que la CPU estuvo ocupada

# Solución de Problemas

## Error: Entrada Inválida (Modo Manual)

**Mensaje:** "Entrada invalida. Ingresas un numero entero mayor a 0."

### Causas comunes:

- Escribió letras en lugar de números
- Escribió números negativos o cero donde no está permitido
- Escribió decimales (solo se aceptan enteros)

**Solución:** El sistema permite reintentar inmediatamente. Escriba solo números enteros respetando los rangos.

## Error: Archivo No Encontrado

**Mensaje:** "ERROR: No se pudo abrir el archivo 'procesos.csv'"

### Verificación rápida:

```
# Windows
dir procesos_prueba.csv

# Linux/Mac
ls -l procesos_prueba.csv
```

## Error: Formato CSV Incorrecto

Incorrecto	Correcto
P1,0,diez,2	P1,0,10,2
P2,abc,5,1	P2,2,5,1
P3,0,10	P3,0,10,3
P4;0;10;2	P4,0,10,2



# Casos de Uso Recomendados

## Caso 1: Comparación de Algoritmos

**Objetivo:** Comparar los 4 algoritmos con los mismos datos

**Archivo:** `comparacion.csv`

```
P1,0,10,3  
P2,2,4,1  
P3,4,6,2  
P4,6,8,4  
P5,8,2,1
```

**Procedimiento:**

1. Ejecutar con FCFS → Anotar métricas
2. Presionar Enter → Volver al inicio
3. Cargar mismo archivo → Seleccionar SJF
4. Repetir con Round Robin (quantum 4)
5. Repetir con Prioridad
6. Comparar resultados

## Caso 2: Efecto del Quantum en Round Robin

**Objetivo:** Ver cómo el quantum afecta el rendimiento

**Archivo:** `test_quantum.csv`

```
P1,0,20,1  
P2,0,15,1  
P3,0,10,1
```

**Pruebe con:**

- Quantum = 2 (muchos cambios de contexto)
- Quantum = 4 (balanceado)
- Quantum = 8 (pocos cambios de contexto)

# Mejores Prácticas

## 1. Validación en Tiempo Real

El sistema valida cada entrada inmediatamente. Si se equivoca, puede corregir de inmediato sin empezar de cero.

## 2. Limpieza de Pantalla Automática

La pantalla se limpia entre secciones para mantener una interfaz clara y fácil de seguir.

## 3. Recuperación de Errores

Si un archivo CSV falla, el programa vuelve automáticamente al menú principal. No es necesario reiniciar.

## 4. Organización de Archivos

### Estructura Recomendada:

```
cloanr/  
├─ simulador.exe  
├─ procesos_prueba.csv  
├─ datos/  
│   ├─ fcfs/  
│   ├─ sjf/  
│   └─ rr/  
└─ prioridad/  
    └─ resultados/
```

## 5. Nomenclatura Descriptiva

Buenas Prácticas	Evitar
fcfs_10procesos_mixto.csv	datos.csv
sjf_trabajos_cortos.csv	test1.csv
rr_quantum4_20procesos.csv	archivo.csv

# Referencia Rápida

## Comandos de Compilación

```
# Compilar desde UI/  
  
g++ -std=c++17 interface.cpp -o ../simulador.exe  
  
# Volver a la carpeta raíz  
cd ..
```

## Flujo de Uso Normal

1. **Iniciar:** `simulador.exe`
2. **Elegir modo:** 1 (Manual) o 2 (Archivo)
3. **Ingresar datos**
4. **Seleccionar algoritmo:** 1-4
5. **Ver resultados**
6. **Continuar:** Enter (volver) o "salir" (terminar)

## Validaciones Clave

Campo	Regla	Ejemplo Válido
Cantidad procesos	> 0	3, 10, 20
Tiempo Llegada	>= 0	0, 5, 10
Tiempo Ráfaga	> 0	1, 10, 20
Quantum	> 0	2, 4, 8



# Soporte Técnico

## Información del Sistema

Atributo	Valor
Proyecto	Simulador de Planificación de Procesos
Versión	2.0
Lenguaje	C++17
Plataforma	Windows/Linux/Mac

## Documentación Adicional

- `README_PERSISTENCIA.md` - Detalles técnicos de implementación
- `ENTREGABLE_TEST_SUITE.md` - Suite de pruebas automatizadas
- `QUICK_REFERENCE.md` - Referencia rápida de comandos

---

### Manual de Usuario - Simulador de Planificación de Procesos v2.0

Desarrollado como proyecto académico de Sistemas Operativos

Última actualización: Enero 2026