

Simulador de estados de procesos de un SO

Grupo:5 Taller # 1

Fabian Leonardo Correa Rojas; Diego Alejandro Rodríguez López; Jose Luis Ortega Castillo

Profesor: Jose Osbaldo Rojas Moreno

1. Introducción

El presente documento describe el desarrollo de un simulador que representa el ciclo de vida de los procesos en un sistema operativo, mostrando los estados básicos y sus transiciones.

2. Objetivos

2.1 Objetivo general

Desarrollar un simulador que represente el ciclo de vida de los procesos en un sistema operativo, gestionando sus estados básicos y las transiciones entre ellos.

2.2 Objetivos Específicos

- Implementar una máquina de estados finita para modelar procesos.
- Representar gráficamente los estados y transiciones
- Incluir controles básicos y avanzados de simulación
- Generar reportes con métricas de transición y tiempo en cada estado

3. Tecnologías utilizadas

El simulador fue desarrollado usando las siguientes tecnologías:

- Lenguaje: JavaScript/TypeScript
- Framework: React
- Entorno: Node.js con npm

- Estilos: TailwindCSS
- Componentes UI: Shadcn/UI
- Gráficos/animaciones: Recharts

4. Requisitos del sistema y como ejecutarlo

Para poder usar el simulador se necesitan los siguientes requisitos previos:

- Tener instalado Node.js (versión LTS recomendada).
- npm instalado (incluido en Node.js).
- Navegador moderno (Chrome, Firefox, Edge).

Pasos para ejecutar el simulador:

1. Descargar el proyecto o clonarlo desde el repositorio.
2. Abrir una terminal en la carpeta del proyecto.
3. Instalar las dependencias con:
 - npm install
4. Iniciar el servidor de desarrollo con:
 - npm run dev
5. Abrir en el navegador la dirección que aparece en la terminal (por defecto <http://localhost:5173/>).

5. Explicación Técnica

En un sistema operativo, los procesos no siempre están ejecutándose. Dependiendo de la situación, pueden estar esperando, listos para correr o incluso finalizados. Nuestro simulador representa estos estados de forma visual y animada.

Estados básicos de un proceso: New, Ready, Running, Waiting, Terminated.

Estado	Descripción
New	Proceso recién creado, esperando ser admitido.
Ready	Proceso listo para ejecutarse, esperando CPU.
Running	Proceso que se está ejecutando en la CPU.
Waiting	Proceso bloqueado, esperando un recurso o E/S.
Terminated	Proceso que ya terminó su ejecución.

Transiciones válidas:

- New → Ready
- Ready → Running
- Running → Ready
- Running → Waiting
- Waiting → Ready
- Running → Terminated

⚠ Cualquier otra transición no es permitida y el simulador mostrará un error.

6. Manual de Usuario

Este apartado explica paso a paso cómo utilizar el simulador:

1. Cómo iniciar el simulador

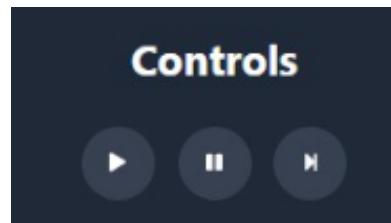
- Abre el programa en tu navegador tras ejecutar los pasos anteriores.
- En la pantalla principal verás los estados: New, Ready, Running, Waiting y Terminated.

2. Crear un proceso

- Presiona el botón New process.
- Aparecerá un cuadro verde en la columna New. Ese es tu proceso.

3. Botones principales(manejo)

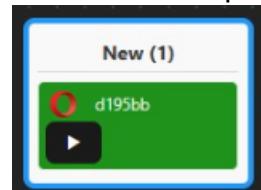
- Start: inicia la simulación
- Pause: detiene la simulación
- Resume: vuelve a correr la simulación justo donde hiba



4. Botones principales(opciones)



- New Process: Permite crear un nuevo proceso en el estado *New*.



- View Reports: Muestra los reportes generados con las métricas de transición y el tiempo que cada proceso permaneció en los diferentes estados.

Process Reports

Select Process: 6d028f

PID	Priority	PC	Registers	Syscalls	Total Transitions
6d028f	High	3	AX: 98, BX: 56, CX: 87, DX: 98		3

Transition History

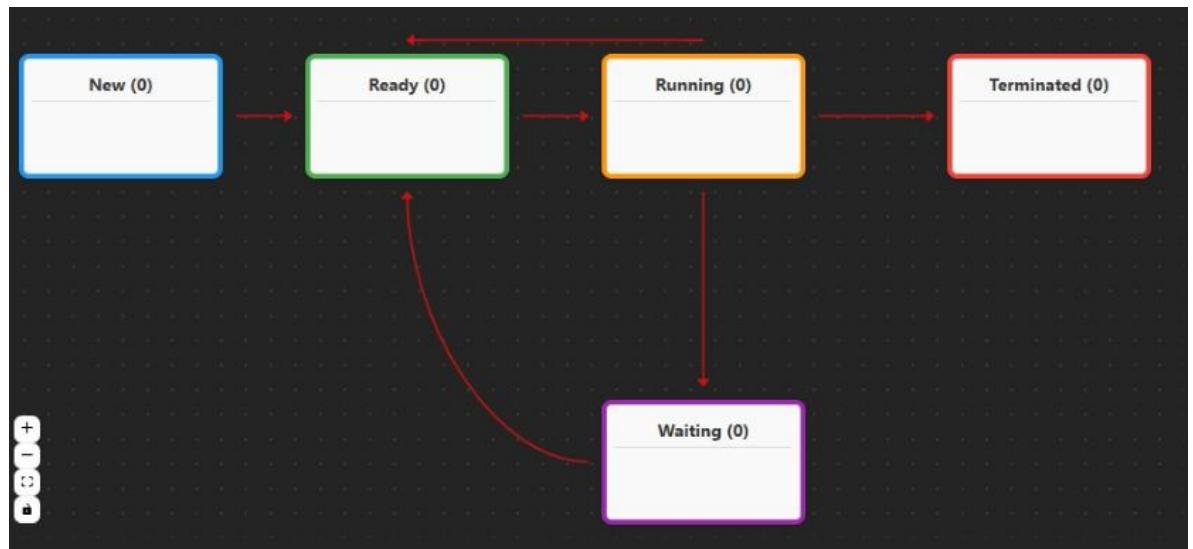
State	Timestamp	Reason
New	8:48:45 a. m.	Process Created
Ready	8:48:48 a. m.	Admitted to system
Running	8:48:52 a. m.	CPU assigned
Terminated	8:48:57 a. m.	Process finished

- Show Technical Details: Despliega información técnica detallada de los procesos, como el PID, historial de transiciones y tiempo en cada estado.

Ready (5)

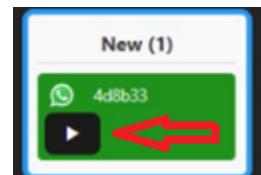
PID: 419651
 Program Counter: 4
 Registers: AX: 61, BX:
 61, CX: 65, DX: 4
 Syscalls: I/O Request
 State Times (ms):
 • New: 3331 ms
 • Ready: 4185 ms
 • Running: 5110 ms
 • Waiting: 6241 ms
 Total Transitions: 4
 Transition Events:
 • New - Process Created - 8:48:45 a. m.
 • Ready - Admitted to system - 8:48:49 a. m.
 • Running - CPU assigned - 8:48:53 a. m.
 • Waiting - I/O requested - 8:48:58 a. m.
 • Ready - I/O

- Clear Simulation: Limpia la simulación actual, eliminando todos los procesos y reiniciando el entorno.

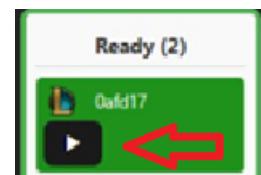


5. Controles avanzados

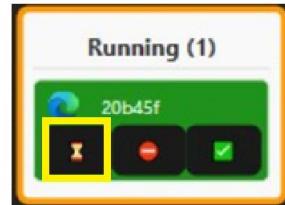
- Admit: Mueve el proceso de New a Ready.



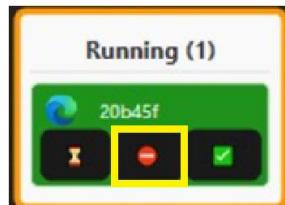
- Assign CPU: Pasa el proceso de Ready a Running.



- Request I/O: Desde Running, envía el proceso a Waiting.



- Release I/O: Devuelve un proceso de Waiting a Ready.



- Terminate: Finaliza un proceso que estaba en Running.



Pasos básicos:

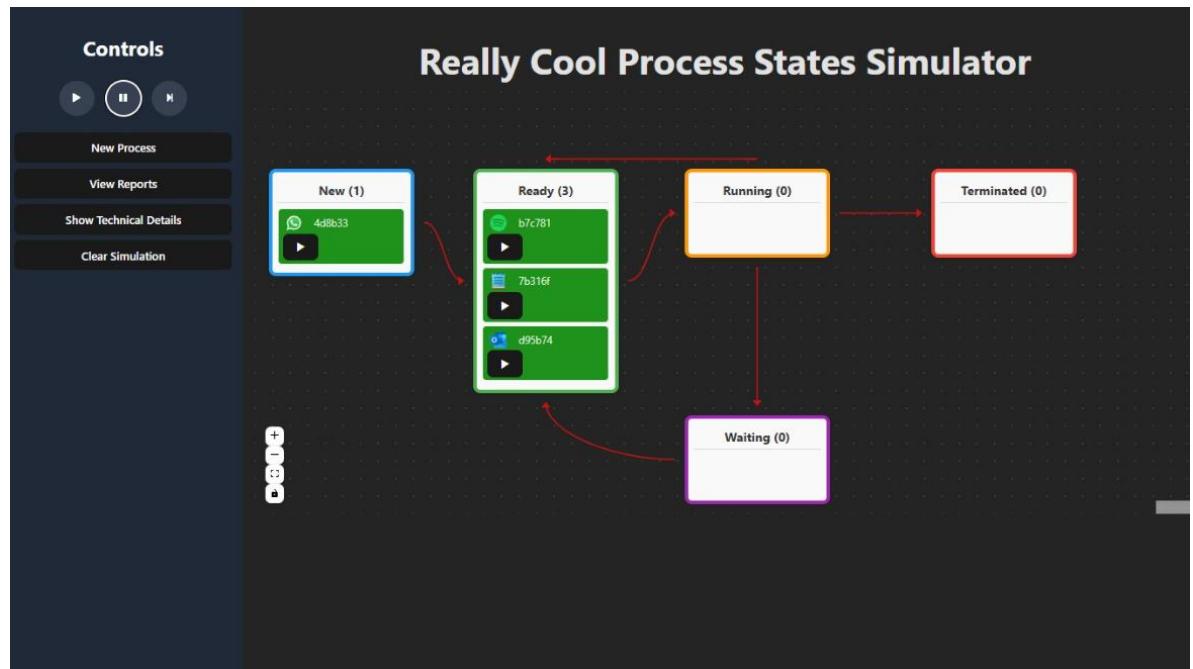
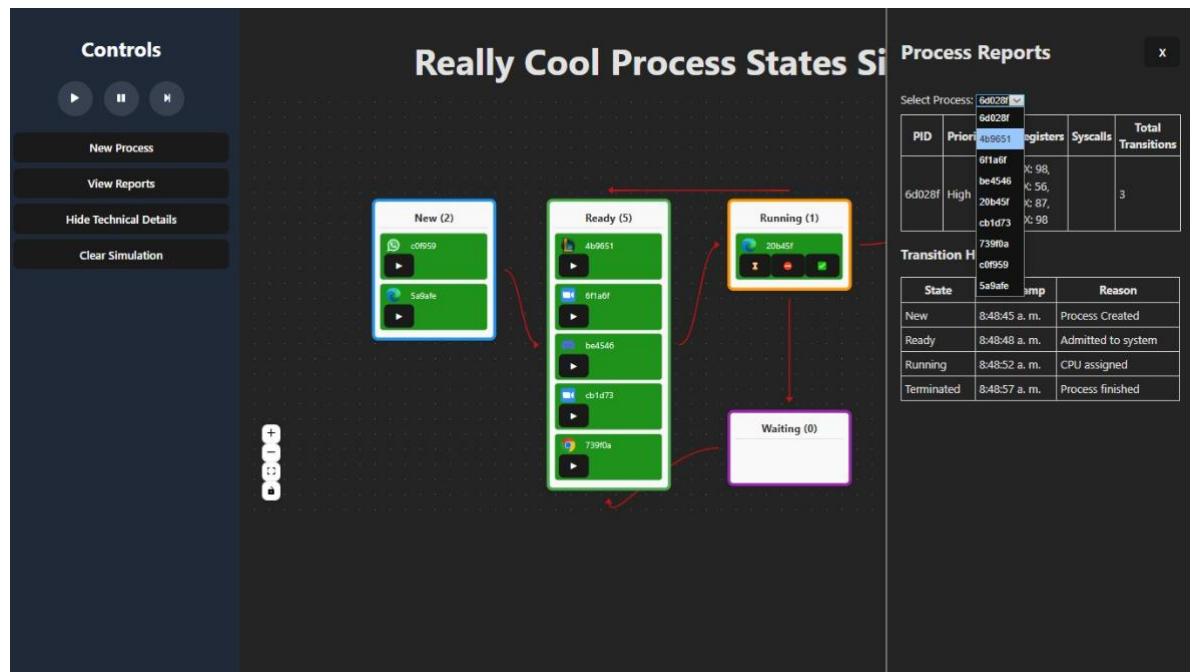
- Crear un nuevo proceso (estado New).
- Pulsar “Admit” para pasarlo a Ready.
- Pulsar “Assign CPU” para moverlo a Running.
- Usar “Request I/O” para pasarlo a Waiting.
- Usar “Release I/O” para devolverlo a Ready.
- Pulsar “Terminate” para finalizarlo en Terminated.

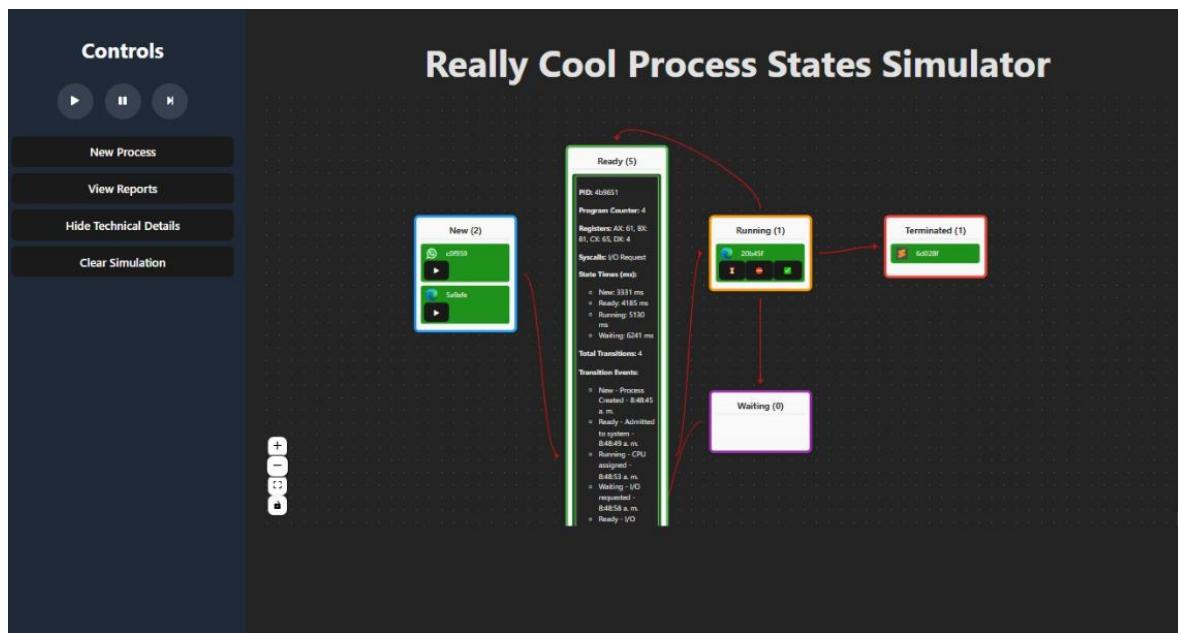
Controles avanzados:, auto-transiciones, mostrar/ocultar detalles técnicos.

7. Casos de Prueba

- Caso 1: Proceso sin E/S → New → Ready → Running → Terminated.
- Caso 2: Proceso con E/S → New → Ready → Running → Waiting → Ready → Running → Terminated.

- Caso 3: Transición inválida (ejemplo: New → Terminated) debe generar alerta.





8. Conclusiones

El simulador permite comprender de manera visual el ciclo de vida de los procesos. Se implementaron los estados y transiciones principales, así como un control de simulación que facilita el aprendizaje práctico. Como mejora futura se propone integrar un planificador real.

9. Referencias

- Silberschatz, A., Galvin, P. B., & Gagne, G. (2018). Operating System Concepts.
- Tanenbaum, A. S., & Bos, H. (2015). Modern Operating Systems.
- Documentación de React: <https://react.dev/>

