

TRABAJO PRACTICO INTEGRADOR

SIMULADOR DE ASIGNACIÓN DE MEMORIA Y
PLANIFICACIÓN DE PROCESOS



GRUPO: OPPENHEIMER

- AGUIRRE, JULIAN
- SAUCEDO, GONZALO
- SANCHEZ, GISELA
- RAMIREZ, EDUARDO
- VALERIA, VERON

DOCENTES:

- ING. LILIANA CUENCA PLETSCH
- DR. SERGIO GRAMAO
- ING. ROA JORGE ALEJANDRO
- ING. VIGIL RODRIGO

CATEDRA:

- SISTEMAS OPERATIVOS – CICLO 2023-SEGUNDO CUATRIMESTRE
- INGENIERIA EN SISTEMAS DE INFORMACION

FECHA Y LUGAR : RESISTENCIA , CHACO – 21/11/2023 -

INDICE:

INTRODUCCION:	3
DESARROLLO:	3
CONSIGNA:.....	3
Consideraciones:	4
DIAGRAMA DE FLUJO	4
lector.py	5
main.py	5
ventana.py	5
cargar_archivo.py.....	5
codigo.py	5
EJECUCION DEL SIMULADOR:.....	6
UTILIZACION DE LA HERRAMIENTA “TRELLO”:	9
CONCLUSION:	12

INTRODUCCION:

Este informe presenta la creación de un simulador de asignación de memoria y planificación de procesos como parte de un Trabajo Práctico Integrador (TPI). El simulador permite cargar hasta 10 procesos y utiliza particiones fijas para la asignación de memoria, con categorías predefinidas y una política Best-Fit.

La carga dinámica de procesos, junto con una política de multiprogramación que mantiene un máximo de 5 procesos en ejecución, aseguran la eficiencia del sistema. La planificación de CPU se realiza mediante un algoritmo Round-Robin con un quantum de 2 unidades de tiempo.

La salida del simulador proporciona información detallada sobre el estado del procesador, la tabla de particiones de memoria, el estado de la cola de procesos listos y, al finalizar, un informe estadístico con tiempos de retorno y espera, así como tiempos promedios.

El desarrollo se llevó a cabo en equipos colaborativos con seguimiento a través de Trello. El simulador se implementó en Python y se entrega con el programa ejecutable, código fuente y una guía de uso.

DESARROLLO:

CONSIGNA:

Implementar un simulador de asignación de memoria y planificación de procesos según los siguientes requerimientos. El simulador deberá brindarla posibilidad de cargar procesos por parte del usuario. Para facilitar la implementación se permitirán como máximo 10 procesos y la asignación de memoria se realizará con particiones fijas. El esquema de particiones será el siguiente:

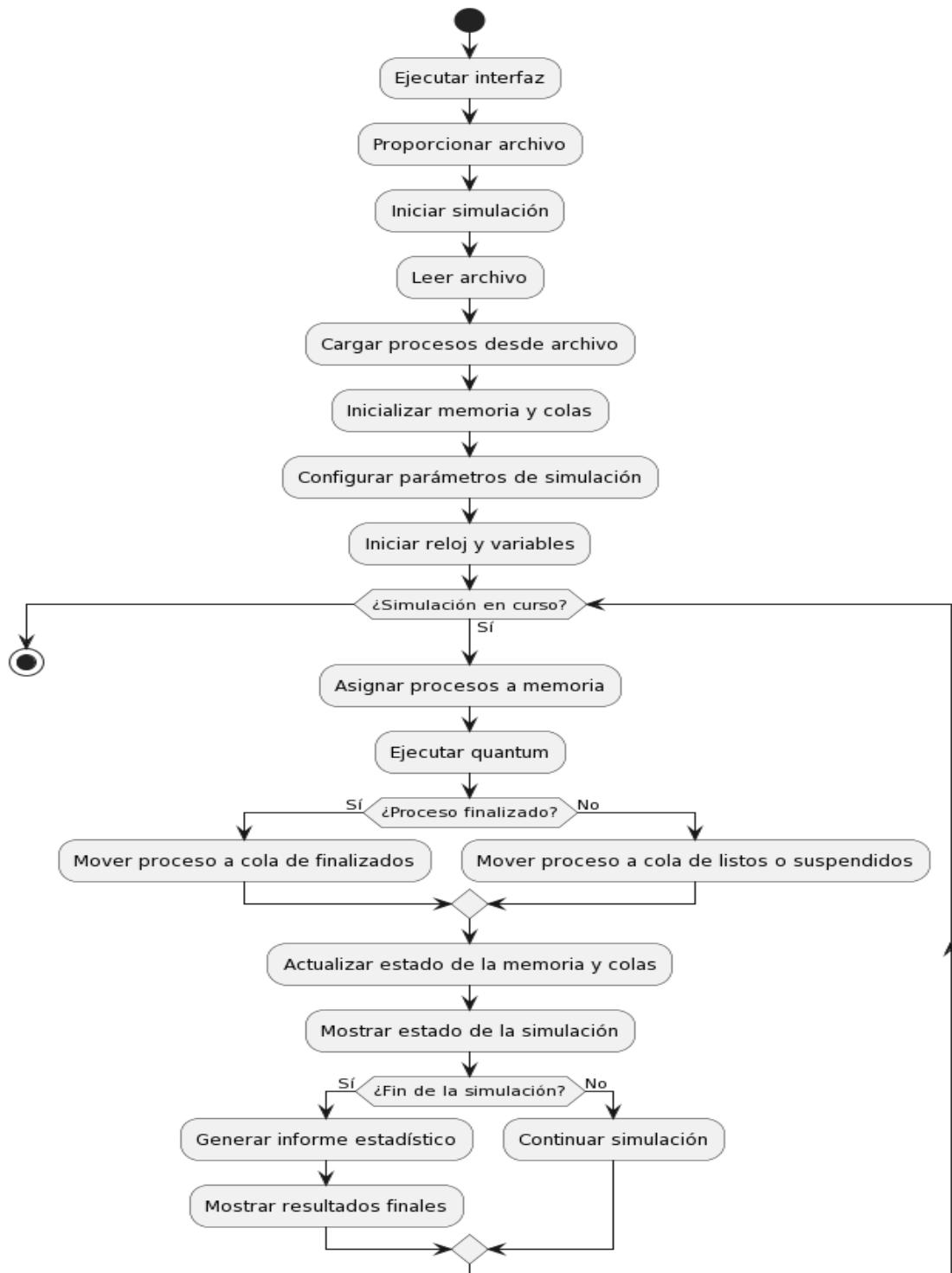
- 100K destinados al Sistema Operativo
- 250K para trabajos los más grandes.
- 120K para trabajos medianos
- 60K para trabajos pequeños. El programa debe permitir ingreso de nuevos procesos cuando sea posible (manteniendo en grado de multiprogramación en 5) La política de asignación de memoria será Best-Fit, por cada proceso se debe ingresar o leer desde un archivo el Id de proceso, tamaño del proceso, tiempo de arribo y tiempo de irrupción. La planificación de CPU será dirigida por un algoritmo Round-Robin con $q=2$. El simulador deberá presentar como salida la siguiente información:
- El estado del procesador (proceso que se encuentra corriendo en ese instante)
- La tabla de particiones de memoria, la cual deberá contener (Id de partición, dirección de comienzo de partición, tamaño de la partición, id de proceso asignado a la partición, fragmentación interna)
- El estado de la cola de procesos listos.

- Al finalizar la simulación se deberá presentar un informe estadístico con, tiempo de retorno y espera para cada proceso y los respectivos tiempos promedios.

Consideraciones:

- Las presentaciones de salida deberán realizarse cada vez que llega un nuevo proceso, se termina un proceso en ejecución.
- No se permiten corridas ininterrumpidas de simulador, desde que se inicia la simulación hasta que termina el último proceso

DIAGRAMA DE FLUJO:



Se decidió separar el código en 5 archivos para proporcionar un desarrollo más modular del simulador:

lector.py

Este archivo contiene funciones para leer datos desde archivos. Soporta archivos CSV y JSON. La función `leer_datos` decide qué tipo de archivo es y llama a la función correspondiente (`csv_reader` o `json_reader`) para procesar y devolver los datos en un formato consistente.

main.py

Este es el archivo principal que ejecuta la simulación. Utiliza la funcionalidad proporcionada en otros archivos, como la carga de archivos (Prompt y `leer_archivo`), y la ejecución de la simulación (Run).

ventana.py

Este archivo contiene una interfaz gráfica básica creada con la biblioteca Tkinter de Python. La interfaz permite al usuario seleccionar un archivo CSV mediante un cuadro de diálogo. También muestra una imagen y proporciona botones para abrir el cuadro de diálogo y para iniciar la simulación. La simulación se inicia ejecutando el archivo `main.py` con el archivo CSV seleccionado.

cargar archivo.py

En este archivo se encuentra una serie de funciones para validar diferentes tipos de entrada, como tamaños de archivos y números positivos.

codigo.py

Este archivo contiene la lógica principal de la simulación. Define clases como `BloqueParticion`, `BloqueProceso`, y `BloqueMemoria`, que se utilizan para representar particiones de memoria, procesos y la memoria principal, respectivamente. También incluye funciones para leer datos desde un archivo, generar tablas, mostrar el estado del sistema y ejecutar la simulación.

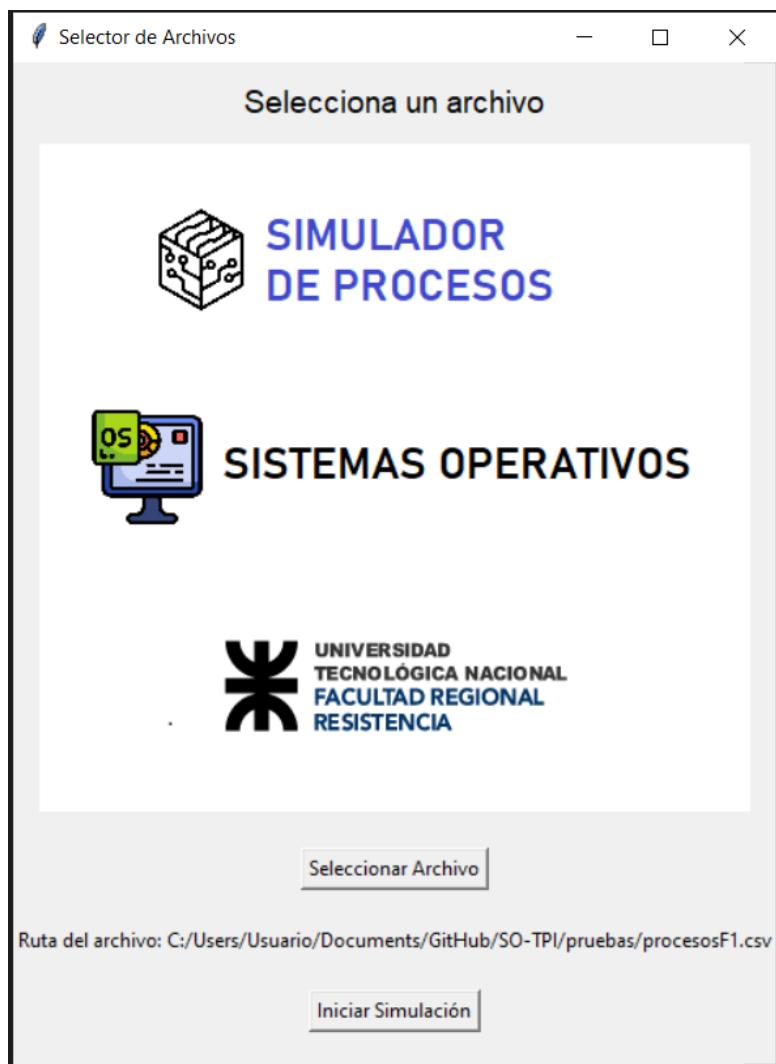
La simulación en Run se basa en un algoritmo de planificación Round-Robin con un quantum de tiempo definido. Muestra el estado del sistema en cada paso, incluyendo información sobre la memoria, colas de procesos y procesos finalizados. Al final de la simulación, se genera un informe estadístico con tiempos de retorno y espera para cada proceso, así como los tiempos promedios.

EJECUCION DEL SIMULADOR:

Para ejecutar el simulador, se debe acceder a través de un archivo ejecutable, que se encuentra en la carpeta /ejecutable, cuyo nombre es "SimuladorOppenheimer.exe"

Este equipo > Documentos > GitHub > SO-TPI > ejecutable			
	Nombre	Fecha de modificación	Tipo
Facultad	SimuladorOppenheimer	20/11/2023 10:57	Aplicación

El mismo posee una interfaz que permite cargar un archivo .json o .csv con procesos.



Una vez seleccionado el archivo, se muestran los datos cargados:

```
~~~~~<DATOS INGRESADOS>~~~~~
```

Nuevos Procesos				
ID	Tamaño Proceso(KB)	Tiempo Arribo	Tiempo Irrupción	Estado
248	150	0	5	Nuevo
7628	120	0	1	Nuevo
123	20	0	4	Nuevo
942	50	1	3	Nuevo
4824	30	2	4	Nuevo
28942	250	3	5	Nuevo
392	200	3	10	Nuevo
6124	30	5	2	Nuevo
1283	10	5	3	Nuevo


```
>>>Presione Enter para continuar o CTRL+C para terminar la ejecución.
```

A medida que van transcurriendo los instantes en memoria, se van mostrando los cambios de estado en los procesos:

```
# Datos:  

» Tiempo Transcurrido: 0  

» Quantum: 2  

» Pasa a ejecución: 248
```

Memoria				
ID	Dirección	Tamaño Partición(KB)	Tamaño Proceso(KB)	Frag. Interna(KB)
123	0x22e83ddede0	60	20	40
7628	0x22e840dff20	120	120	0
248	0x22e840dfbf0	250	150	100

Cola de Nuevos				
ID	Tamaño Proceso(KB)	Tiempo Arribo	Tiempo Irrupción	Estado
942	50	1	3	Nuevo
4824	30	2	4	Nuevo
28942	250	3	5	Nuevo
392	200	3	10	Nuevo
6124	30	5	2	Nuevo
1283	10	5	3	Nuevo

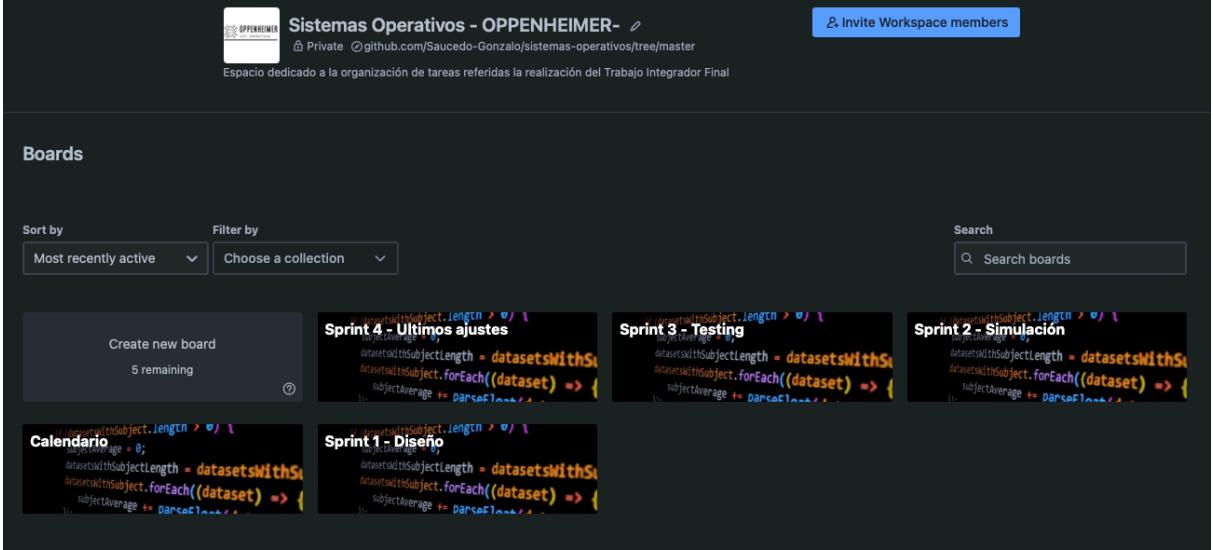
Cola de Listos				
ID	Tamaño Proceso(KB)	Tiempo Arribo	Tiempo Irrupción	Estado
248	150	0	5	Ejecutando
7628	120	0	1	Listo
123	20	0	4	Listo

Datos:
» Tiempo Transcurrido: 15
» Quantum: 1
» Finalización del proceso: 123
+
Memoria
+
ID Dirección Tamaño Partición(KB) Tamaño Proceso(KB) Frag. Interna(KB)
+
942 0x22e840dff20 120 50 70
28942 0x22e840dfbf0 250 250 0
+
+
Cola de Nuevos
+
ID Tamaño Proceso(KB) Tiempo Arribo Tiempo Irrupción Estado
+
392 200 3 10 Nuevo
6124 30 5 2 Nuevo
1283 10 5 3 Nuevo
+
+
Cola de Listos
+
ID Tamaño Proceso(KB) Tiempo Arribo Tiempo Irrupción Estado
+
942 50 1 1 Listo
248 150 0 1 Suspendido
4824 30 2 2 Suspendido
28942 250 3 3 Listo
+
+
Procesos Finalizados
+
ID Tamaño Proceso(KB) Tiempo Arribo Tiempo Irrupción Estado
+
7628 120 0 1 Finalizado
123 20 0 4 Finalizado
+

Al finalizar todos los procesos, se muestra un informe estadístico.

<INFORME ESTADÍSTICO>																																											
-->PROMEDIOS																																											
<table border="1"> <thead> <tr> <th>RESPUESTA</th><th>ESPERA</th><th>RETORNO</th><th></th></tr> </thead> <tbody> <tr> <td>19.11</td><td>6.22</td><td>10.33</td><td></td></tr> </tbody> </table>				RESPUESTA	ESPERA	RETORNO		19.11	6.22	10.33																																	
RESPUESTA	ESPERA	RETORNO																																									
19.11	6.22	10.33																																									
-->RESUMEN DE BLOQUES DE PROCESOS																																											
<table border="1"> <thead> <tr> <th>ID</th><th>RESPUESTA</th><th>ESPERA</th><th>RETORNO</th></tr> </thead> <tbody> <tr> <td>7628</td><td>3</td><td>2</td><td>3</td></tr> <tr> <td>123</td><td>15</td><td>7</td><td>11</td></tr> <tr> <td>942</td><td>15</td><td>10</td><td>13</td></tr> <tr> <td>248</td><td>17</td><td>7</td><td>12</td></tr> <tr> <td>4824</td><td>17</td><td>4</td><td>8</td></tr> <tr> <td>6124</td><td>20</td><td>7</td><td>9</td></tr> <tr> <td>28942</td><td>25</td><td>5</td><td>10</td></tr> <tr> <td>1283</td><td>26</td><td>9</td><td>12</td></tr> <tr> <td>392</td><td>34</td><td>5</td><td>15</td></tr> </tbody> </table>				ID	RESPUESTA	ESPERA	RETORNO	7628	3	2	3	123	15	7	11	942	15	10	13	248	17	7	12	4824	17	4	8	6124	20	7	9	28942	25	5	10	1283	26	9	12	392	34	5	15
ID	RESPUESTA	ESPERA	RETORNO																																								
7628	3	2	3																																								
123	15	7	11																																								
942	15	10	13																																								
248	17	7	12																																								
4824	17	4	8																																								
6124	20	7	9																																								
28942	25	5	10																																								
1283	26	9	12																																								
392	34	5	15																																								

UTILIZACION DE LA HERRAMIENTA “TRELLO”:



The screenshot shows the Trello workspace interface. At the top, there's a header with the workspace name "Sistemas Operativos - OPPENHEIMER-", a "Private" status, and a GitHub link. Below the header, a message says "Espacio dedicado a la organización de tareas referidas la realización del Trabajo Integrador Final". The main area is titled "Boards" and shows several boards arranged in a grid:

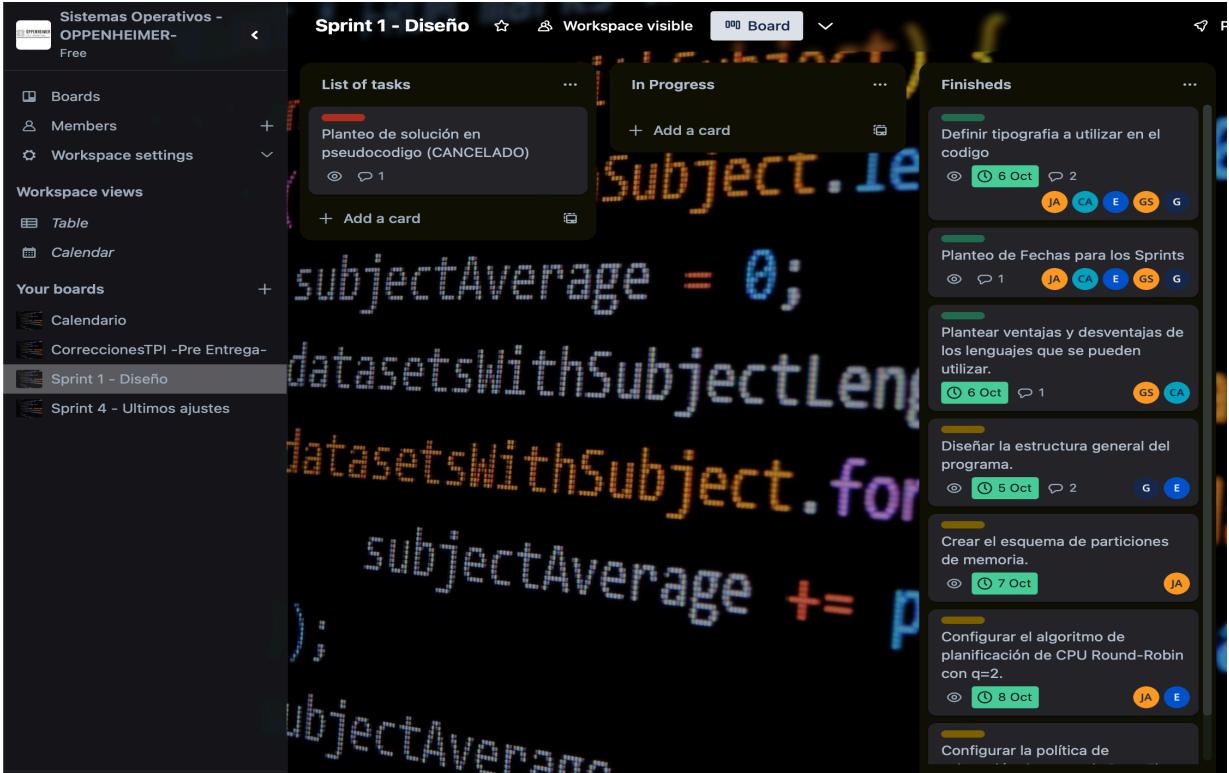
- Create new board**: A dark board with "5 remaining" items.
- Sprint 4 - Ultimos ajustes**: A board with code snippets and tasks.
- Sprint 3 - Testing**: A board with code snippets and tasks.
- Sprint 2 - Simulación**: A board with code snippets and tasks.
- Calendario**: A board with code snippets and tasks.
- Sprint 1 - Diseño**: A board with code snippets and tasks.

On the right side of the interface, there are filters for "Sort by" (Most recently active), "Filter by" (Choose a collection), and a search bar labeled "Search boards".

Cada tarea fue asignada a un integrante del grupo los cuales son:

- **JA**→ Julian Aguirre
- **CA**→ Valeria Veron
- **E**→ Eduardo Ramirez
- **GS**→ Gisela Sanchez
- **G**→ Gonzalo Saucedo

Sprint 1- Diseño-



The screenshot shows the "Sprint 1 - Diseño" board in Trello. The left sidebar shows the workspace navigation with the current board selected. The main board view has three columns: "List of tasks", "In Progress", and "Finisheds".

List of tasks:

- Planteo de solución en pseudocódigo (CANCELADO)

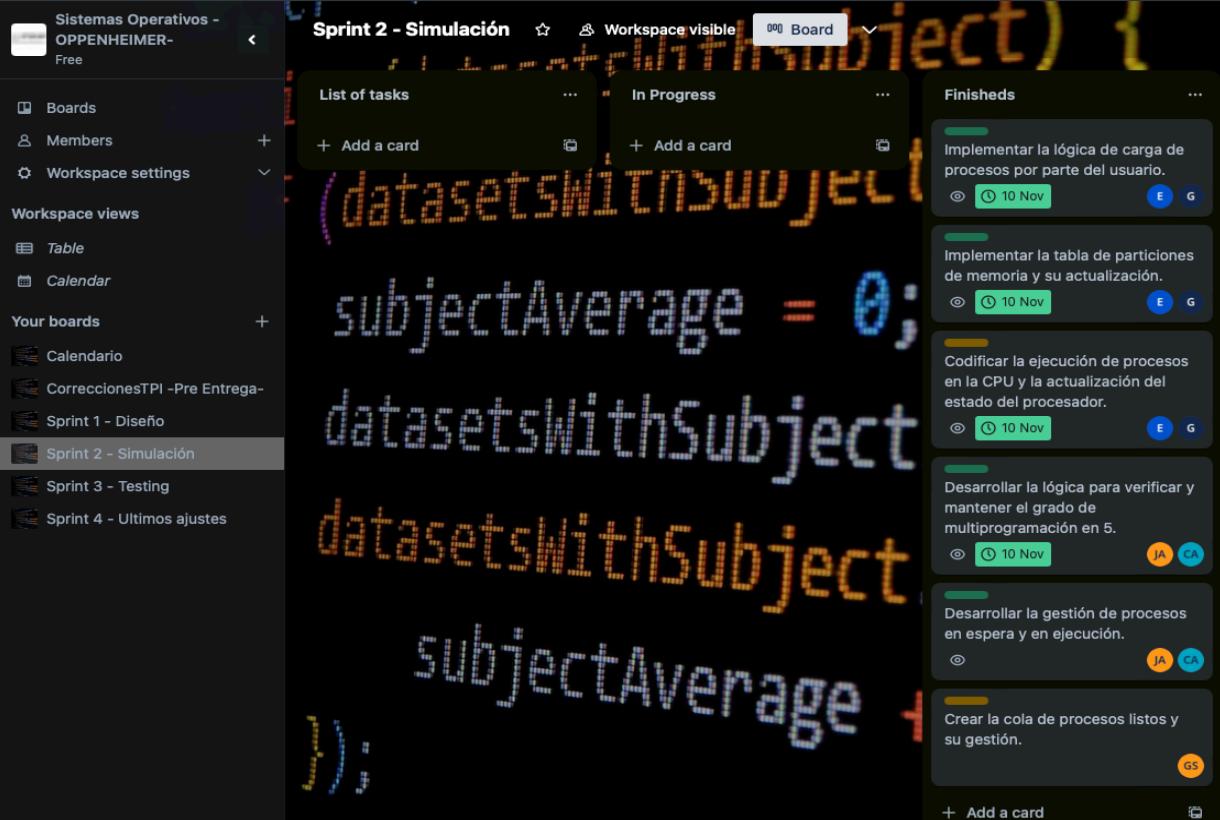
In Progress:

- + Add a card

Finisheds:

- Definir tipografía a utilizar en el código (6 Oct, JA, CA, E, GS, G)
- Planteo de Fechas para los Sprints (6 Oct, JA, CA, E, GS, G)
- Plantear ventajas y desventajas de los lenguajes que se pueden utilizar. (6 Oct, GS, CA)
- Diseñar la estructura general del programa. (5 Oct, G, E)
- Crear el esquema de particiones de memoria. (7 Oct, JA)
- Configurar el algoritmo de planificación de CPU Round-Robin con q=2. (8 Oct, JA, E)
- Configurar la política de...

Sprint 2-Simulacion-

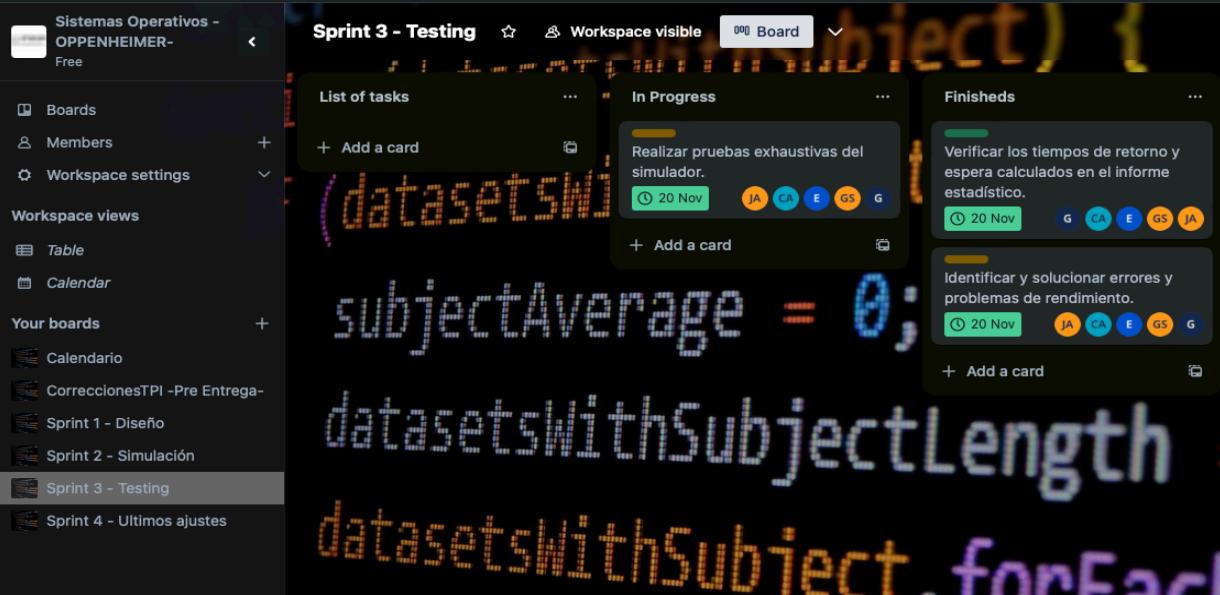


The screenshot shows a Trello board titled "Sprint 2 - Simulación". The board has three main sections: "List of tasks", "In Progress", and "Finished".

- In Progress:**
 - + Add a card
- Finished:**
 - Implementar la lógica de carga de procesos por parte del usuario. (10 Nov) (E, G)
 - Implementar la tabla de particiones de memoria y su actualización. (10 Nov) (E, G)
 - Codificar la ejecución de procesos en la CPU y la actualización del estado del procesador. (10 Nov) (E, G)
 - Desarrollar la lógica para verificar y mantener el grado de multiprogramación en 5. (10 Nov) (JA, CA)
 - Desarrollar la gestión de procesos en espera y en ejecución. (20 Nov) (JA, CA)
 - Crear la cola de procesos listos y su gestión. (20 Nov) (GS)

At the bottom right of the board, there is a button "+ Add a card" and a trash bin icon.

Sprint 3-Testing -



The screenshot shows a Trello board titled "Sprint 3 - Testing". The board has three main sections: "List of tasks", "In Progress", and "Finished".

- In Progress:**
 - + Add a card
- Finished:**
 - Realizar pruebas exhaustivas del simulador. (20 Nov) (JA, CA, E, GS, G)
 - Verificar los tiempos de retorno y espera calculados en el informe estadístico. (20 Nov) (G, CA, E, GS, JA)
 - Identificar y solucionar errores y problemas de rendimiento. (20 Nov) (JA, CA, E, GS, G)

At the bottom right of the board, there is a button "+ Add a card" and a trash bin icon.

Sprint 4 - Últimos ajustes -



The screenshot shows a Trello board titled "Sprint 4 - Ultimos ajustes". The board has three columns: "List of tasks", "In Progress", and "Finished".

- List of tasks:**
 - Realizar entrega final vía campus virtual (Due 21 Nov)
- In Progress:**
 - Crear documentación detallada para la guía (Readme) de ejecución. (Due 21 Nov)
 - Comprimir la carpeta, en donde se encuentra el código fuente. (Due 21 Nov)
- Finisheds:**
 - + Add a card

CONCLUSION:

En conclusión, la realización de este Trabajo Práctico Integrador ha sido fundamental para consolidar los conocimientos teóricos adquiridos en la materia y aplicarlos de manera práctica en el desarrollo de un simulador. A través de este proyecto, se abordaron aspectos cruciales de la planificación a corto plazo y la gestión de memoria con particiones fijas en un entorno de un solo procesador.

El diseño integral del simulador permitió no solo comprender los conceptos teóricos asociados, sino también llevarlos a la práctica mediante la implementación de algoritmos como Best-Fit para la asignación de memoria y Round-Robin para la planificación de CPU. La capacidad de cargar procesos, mantener un grado de multiprogramación específico y proporcionar información detallada sobre el estado del procesador, la memoria y la cola de procesos, contribuyó significativamente a la comprensión del ciclo de vida completo de un proceso en un sistema operativo.

Además, la presentación de un informe estadístico al finalizar la simulación permitió evaluar el rendimiento del sistema, analizando tiempos de retorno y espera para cada proceso, así como los respectivos tiempos promedios. Este enfoque no solo reforzó los conocimientos sobre la teoría de sistemas operativos, sino que también proporcionó una experiencia valiosa en la implementación práctica de estos conceptos.