

Proyecto

Santiago Hernandez

Andrés Loreto Quiros



Pontificia Universidad
JAVERIANA
Colombia

Sistemas Operativos

Prof. John Corredor

Objetivos principales

- a. Resolver un problema utilizando procesos e hilos de la biblioteca POSIX.
- b. Emplear mecanismos de sincronización de procesos y comunicación de procesos usando pipes.
- c. Utilizar de forma correcta llamadas al sistema relacionadas con hilos y procesos

Funciones de los componentes

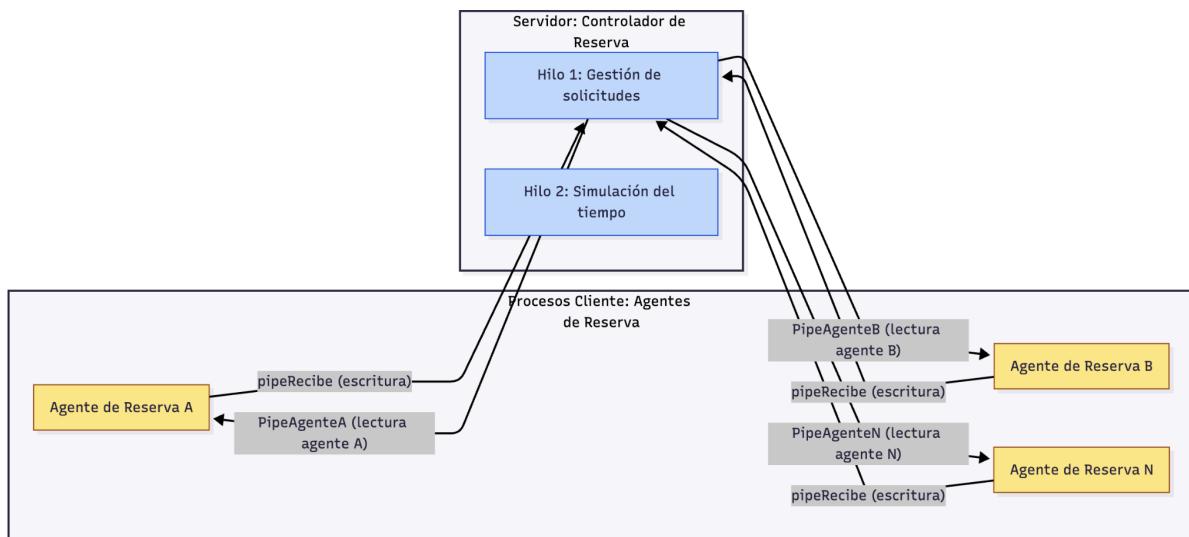
2.1. Servidor: Controlador de Reserva

- Recibe las solicitudes de los Agentes de Reserva y las autoriza o rechaza dependiendo de la cantidad de personas que ya ha reservado en las horas solicitadas. En algunos casos, es posible, que el sistema reserve en espacios de tiempo distinto al solicitado.
- El sistema simula el tiempo transcurrido, y al transcurrir una “hora” realiza las siguientes acciones: sacar personas del parque porque se les acaba su tiempo y autoriza la entrada de las personas que reservaron para la siguiente hora.
- Al finalizar el día, emite un reporte acerca de la ocupación del parque.

2.2 Cliente: Agente de Reserva

- Los Agentes son procesos que solicitan, al Controlador de Reservas, espacios para grupos familiares en un determinado día.

Diagrama de Arquitectura



Arquitectura del proyecto

Nuestro proyecto se divide en las siguientes partes:

1. Makefile: Nos permite automatizar la creación de los archivos ejecutables de los archivos, define el compilador
2. controlador.c : Es nuestro programa principal, contiene la definición de los tipos de mensaje y la estructura del mensaje, simula la lógica del parque, el manejo de reservas y solicitudes, genera una lista enlazada entre agentes y la lista de reserva, además de crear dos hilos, uno para recibir y procesar la solicitud de los agentes y otro para ir contabilizando la hora de la simulación e imprimir el estado del parque
3. agente.c: Es nuestro cliente, envía solicitudes de reserva, leyéndolas desde un CSV, las envía y espera respuesta del controlador e imprime en pantalla las solicitudes de cada familia
4. solicitudes_x.csv: Contienen las solicitudes de reserva para cada agente, están en formato : NombreFamilia, HoraSolicitada, NumeroPersonas, la procesa agente.c

Controlador.c

Es nuestro funcionamiento principal del parque, se inicializa con la hora de inicio y fin de la simulación, el aforo máximo, la duración de cada hora simulada y el nombre del pipe principal, a partir de esto, crea un hilo reloj, que avanza con cada hora y muestra qué familias entran y salen del parque, además de crear un hilo de atención de solicitudes, escucha constantemente mediante el pipe las peticiones de los agentes, cada solicitud es evaluada de si se puede cumplir con la capacidad disponible en el bloque de horas solicitado, de ser posible, se acepta la hora pedida, de lo contrario, se busca el primer bloque de horas disponible para programar, si no es posible tampoco, se niega, al terminar el horario, el controlador envía un mensaje de finalización a todos los agentes, cierra los pipes y genera un reporte final con estadísticas de ocupación y un conteo con la cantidad de solicitudes aceptadas, reprogramadas y negadas

Agente.c

Actúa como un cliente el cual desea generar reservas en el parque, al iniciar, genera su propio pipe de respuesta, se registra ante el controlador , enviando su nombre y el nombre de pipe, espera la confirmación con la hora actual, después, lee un archivo de solicitudes y para cada registro que aún sea válido, envía un mensaje de tipo solicitud por el pipe principal y se bloquea esperando la respuesta en su pipe de respuesta, apenas el controlador contesta, el agente muestra por pantalla si la reserva fue aceptada, reprogramada o negada y a qué hora quedó asignada , este ciclo se repite hasta acabar todas las líneas de archivos de entrada , al acabar, el agente finaliza cerrando su pipe y dejando registrado el resultado de cada solicitud

Estructuras de datos utilizadas:

1. tipoMensaje : enum que define los tipos de mensajes que pueden circular entre el controlador y los agentes, incluye MSG_REGISTRO, MAS_REGISTRO_OK, MSG_SOLICITUD, MSG_RESPUESTA Y MSG_FIN
2. Mensaje : Agrupa la información necesaria para la comunicación entre controlador y agente, el tipo de mensaje, nombre del agente, nombre de la familia, la hora solicitada, número de personas, nombre del pipe de respuesta, se escribe y lee en los pipes,
3. Reserva : Representa la reserva aceptada en el parque, contiene el nombre de la familia, cantidad de personas, hora de inicio y fin, y un puntero al siguiente nodo, generando una lista enlazada, la utiliza el controlador para saber que familias deben entrar o salir en cada hora de la simulación
4. AgenteInfo: Almacena la información de cada agente que se registra en el sistema, guarda el nombre del agente, el nombre del pipe de respuesta, además de un puntero que apunta al siguiente nodo, generando una lista enlazada de agentes, permite tener un registro de todos los agentes conectados
5. EstadoParque: Es el estado del parque, incluye ocupacion[24], la hora actual, hora de inicio y hora de fin, aforo máximo y contadores de estadísticas diversas, tales como la cantidad de solicitudes negadas, además de un puntero a la lista de reserva, permite saber si hay cupo para nuevas reservas
6. Ocupacion[24] : Es un vector de enteros, cada posición represent una hora del dia y almacena cuántas personas estarán dentro del parque en esa hora, cada vez que se acepta una reserva, se incrementan las posiciones correspondientes a esa hora, permite verificar si un bloque de dos horas tiene suficiente cupo para aceptar o reprogramar una solicitud

Instrucciones de compilación (make):

para poder compilar se debe ubicar en la carpeta del proyecto y generar el comando make, el cual crea una carpeta build/ y genera dos ejecutables, controlador y agente, para limpiarlos se debe ejecutar el comando make clean

Instrucciones de ejecución (controlador y agentes):

Para iniciar, debemos ejecutar el controlador, se debe utilizar un comando con la siguiente sintaxis : ./build/controlador -i 7 -f 19 -s 1 -t 50 -p pipe_parque, donde 7 y 19 equivalen las horas de la simulación, 50 representa el aforo máximo y el pipe principal se llamará pipe_PARQUE, Posterior a esto se debe ejecutar los agentes, incluyéndose los archivos de datos, de la siguiente manera :

./build/agente -s AgenteA -a data/solicitudes_A.csv -p pipe_parque

cada agente se registra ante el controlador, envia sus solicitudes de reserva e imprime la respuesta para cada familia.

al finalizar el contador de horas, se detiene la simulación, se envía un mensaje de fin a los agentes y se imprime el reporte final.

Reporte final del controlador:

Al utilizar los siguientes parámetros:

```
./build/controlador -i 7 -f 19 -s 1 -t 50 -p pipe_parque
```

```
./build/agente -s AgenteA -a data/solicitudes_A.csv -p pipe_parque  
./build/agente -s AgenteB -a data/solicitudes_B.csv -p pipe_parque
```

El controlador proceso las solicitudes correspondientes y acunulo la ocupación en bloques de 2 horas por familia, a partir de esto, se obtuvieron los siguientes datos:

Horas pico

- Ocupación máxima: 23 personas
- Horas con esa ocupación: 14 y 15

Horas valle

- Ocupación mínima: 0 personas
- Hora con ocupación mínima: 7

Distribución de ocupación por hora (7 a 19)

(Hora → personas dentro del parque)

- 7: 0
- 8: 22
- 9: 22
- 10: 14
- 11: 14
- 12: 18
- 13: 18
- 14: 23
- 15: 23
- 16: 22
- 17: 22
- 18: 17
- 19: 17

Resumen de solicitudes (según códigos de respuesta)

- Solicitudes aceptadas en la hora original (codigoRespuesta = 1): 2
- Solicitudes reprogramadas (codigoRespuesta = 2): 10

- Solicitudes negadas (codigoRespuesta = 4): 1 Resultados

```

Terminal Shell Edit View Window Help andreslorente@192 ~ % cd ~/Documents/Proyecto_SO_final ./buildd/controlador -i 7 -f 19 -s 5 -t 30 -p pipeRecibe Controlador: peticion de agente A, familia "Zuluaga, hora 8, personas 10 -> codigoRespuesta=1, horaAsignada=8
-----
Hora actual de simulacion: 8
Salen familias: ninguna -> Total que salen: 0
Entran familias: "Zuluaga(10) -> Total que entran: 10
Ocupacion programada para la hora 8: 10 personas
-----
Controlador: peticion de agente A, familia "
"Dominguez, hora 8, personas 4 -> codigoRespuesta=1, horaAsignada=8
Controlador: peticion de agente A, familia "
"Rojas, hora 10, personas 10 -> codigoRespuesta=1, horaAsignada=10
-----
Hora actual de simulacion: 9
Salen familias: ninguna -> Total que salen: 0
Entran familias: ninguna -> Total que entran: 0
Ocupacion programada para la hora 9: 14 personas
-----
Controlador: peticion de agente A, familia "
"Martinez, hora 7, personas 6 -> codigoRespuesta=2, horaAsignada=9
Controlador: peticion de agente A, familia "
"Gomez, hora 9, personas 8 -> codigoRespuesta=1, horaAsignada=9
Controlador: peticion de agente A, familia "
"Hernandez, hora 9, personas 12 -> codigoRespuesta=2, horaAsignada=11
-----
Hora actual de simulacion: 10
Salen familias: "
"Dominguez(4), "Zuluaga(10) -> Total que salen: 14
Entran familias: "
"Rojas(10) -> Total que entran: 10
Ocupacion programada para la hora 10: 24 personas
-----
Controlador: peticion de agente A, familia "
"Vargas, hora 11, personas 5 -> codigoRespuesta=1, horaAsignada=11
Controlador: peticion de agente A, familia "
"Ramirez, hora 12, personas 15 -> codigoRespuesta=2, horaAsignada=13
-----
Hora actual de simulacion: 11
Salen familias: "
"Dominguez(8), "
"Martin(6) -> Total que salen: 14
Entran familias: "
"Lopez(5), "
"Hernandez(12) -> Total que entran: 17
Ocupacion programada para la hora 11: 27 personas
-----
Controlador: peticion de agente A, familia "
"Sanchez, hora 13, personas 7 -> codigoRespuesta=1, horaAsignada=13
Controlador: peticion de agente A, familia "
"Vargas, hora 14, personas 9 -> codigoRespuesta=2, horaAsignada=15
Controlador: peticion de agente A, familia "
"Castro, hora 15, personas 12 -> codigoRespuesta=1, horaAsignada=15
-----
Hora actual de simulacion: 12
Salen familias: "
"Rojas(10) -> Total que salen: 18
Entran familias: ninguna -> Total que entran: 0
Ocupacion programada para la hora 12: 17 personas
-----
Controlador: peticion de agente A, familia "
"Moreno, hora 16, personas 4 -> codigoRespuesta=1, horaAsignada=16
Controlador: peticion de agente A, familia "
"Diaz, hora 17, personas 10 -> codigoRespuesta=1, horaAsignada=17
-----
Hora actual de simulacion: 13
Salen familias: "
"Lopez(5), "
"Hernandez(12) -> Total que salen: 17
Entran familias: "
"Sanchez(7), "
"Ramirez(15) -> Total que entran: 22
Ocupacion programada para la hora 13: 22 personas
-----
Controlador: peticion de agente A, familia "
"Pineda, hora 18, personas 6 -> codigoRespuesta=1, horaAsignada=18
Controlador: peticion de agente A, familia "
"Torres, hora 19, personas 8 -> codigoRespuesta=2, horaAsignada=13
-----
Hora actual de simulacion: 14
Salen familias: ninguna -> Total que salen: 0
Entran familias: ninguna -> Total que entran: 0
Ocupacion programada para la hora 14: 30 personas
-----
Controlador: peticion de agente A, familia "
"Torres(8), "
"Sanchez(7), "
"Ramirez(15) -> Total que salen: 30
Entran familias: "
"Castro(12), "
"Vargas(9) -> Total que entran: 21
Ocupacion programada para la hora 15: 21 personas
-----
Controlador: peticion de agente A, familia "
"Moreno(4) -> Total que salen: 0
Entran familias: "
"Moreno(4) -> Total que entran: 4
Ocupacion programada para la hora 16: 25 personas
-----
Controlador: peticion de agente A, familia "
"Castro(12), "
"Vargas(9) -> Total que salen: 21
Entran familias: "
-----
andreslorente@192 ~ %

```

Plan de pruebas

1. Objetivo

Verificar que el sistema de reservas simule correctamente el funcionamiento del parque Berlín, cumpliendo el enunciado: manejo de solicitudes, reprogramación, negación por aforo/tiempo, avance de la simulación por horas y generación del reporte final.

2. Alcance

- Proceso Controlador de Reserva.
- Procesos Agente de Reserva (uno o varios).
- Comunicación mediante pipes nominales.
- Manejo de estructuras compartidas y estadísticas de ocupación.

3. Ambiente de pruebas

- SO: Linux / macOS con soporte POSIX.
- Compilación: make (genera build/controlador y build/agente).
- Ejecución base del controlador:
 - `./build/controlador -i 7 -f 19 -s 1 -t <aforo> -p pipe_parque`
- Ejecución base de agentes:
 - `./build/agente -s <número> -a data/solicitudes_X.csv -p pipe_parque`

4. Casos de prueba funcionales

CP1 – Reserva simple válida (aceptación directa)

- Parámetros controlador: aforo suficiente (ej. `-t 50`).
- Archivo CSV con pocas familias en horas no saturadas.
- Esperado: todas las solicitudes aceptadas en la hora original (`codigoRespuesta = 1`), ocupación por hora igual a la suma de personas.

- Resultados:

```

Terminal Shell Edit View Window Help
Proyecto_SO_final -- zsh -- 87x54
-----
Hora actual de simulacion: 9
Salen familias: ninguna -> Total que salen: 0
Entran familias: ninguna -> Total que entran: 0
Ocupacion programada para la hora 9: 0 personas
-----
Hora actual de simulacion: 10
Salen familias: ninguna -> Total que salen: 0
Entran familias: ninguna -> Total que entran: 0
Ocupacion programada para la hora 10: 0 personas
-----
Hora actual de simulacion: 11
Salen familias: ninguna -> Total que salen: 0
Entran familias: ninguna -> Total que entran: 0
Ocupacion programada para la hora 11: 0 personas
-----
Controlador: peticion de agente A, familia "
" Lopez, hora 11, personas 5 -> codigoRespuesta=1, horaAsignada=11
-----
Hora actual de simulacion: 12
Salen familias: ninguna -> Total que salen: 0
Entran familias: ninguna -> Total que entran: 0
Ocupacion programada para la hora 12: 5 personas
-----
Hora actual de simulacion: 13
Salen familias: "
" Lopez(5) -> Total que salen: 5
Entran familias: ninguna -> Total que entran: 0
Ocupacion programada para la hora 13: 0 personas
-----
Controlador: peticion de agente A, familia "
" Ramirez, hora 12, personas 15 -> codigoRespuesta=2, horaAsignada=13
-----
Hora actual de simulacion: 14
Salen familias: ninguna -> Total que salen: 0
Entran familias: ninguna -> Total que entran: 0
Ocupacion programada para la hora 14: 15 personas
-----
Hora actual de simulacion: 15
Salen familias: "
" Ramirez(15) -> Total que salen: 15
Entran familias: ninguna -> Total que entran: 0
Ocupacion programada para la hora 15: 0 personas
-----
Controlador: peticion de agente A, familia "
" Sanchez, hora 13, personas 7 -> codigoRespuesta=2, horaAsignada=15
-----
Hora actual de simulacion: 16
Salen familias: ninguna -> Total que salen: 0
Entran familias: ninguna -> Total que entran: 0

```

```

Proyecto_SO_final -- zsh -- 93x54
andreslorente@192 Proyecto_SO_final % cd ~/Documents/Proyecto_SO_final
./build/agente -s A -a data/solicitudes_A.csv -p pipeRecibe
Agente A registrado. Hora actual de simulacion: 11
Agente A: solicitud ignorada para familia "Zuluaga, hora 8 (hora actual simulacion: 11)
Agente A: solicitud ignorada para familia "
"Domínguez, hora 8 (hora actual simulacion: 11)
Agente A: solicitud ignorada para familia "
"Rojas, hora 10 (hora actual simulacion: 11)
Agente A: solicitud ignorada para familia "
"Martinez, hora 7 (hora actual simulacion: 11)
Agente A: solicitud ignorada para familia "
"Gomez, hora 9 (hora actual simulacion: 11)
Agente A: solicitud ignorada para familia "
"Hernandez, hora 9 (hora actual simulacion: 11)
Agente A: enviando solicitud -> Familia: "
" Lopez, Hora: 11, Personas: 5
Agente A: respuesta para familia "
" Lopez -> horaSolicitada=11, personas=5, codigoRespuesta=1, horaAsignada=11
Agente A: enviando solicitud -> Familia: "
" Ramirez, Hora: 12, Personas: 15
Agente A: respuesta para familia "
" Ramirez -> horaSolicitada=12, personas=15, codigoRespuesta=2, horaAsignada=13
Agente A: enviando solicitud -> Familia: "
" Sanchez, Hora: 13, Personas: 7
Agente A: respuesta para familia "
" Sanchez -> horaSolicitada=13, personas=7, codigoRespuesta=2, horaAsignada=15
Agente A: enviando solicitud -> Familia: "
" Vargas, Hora: 14, Personas: 9
Agente A: respuesta para familia "
" Vargas -> horaSolicitada=14, personas=9, codigoRespuesta=2, horaAsignada=17
Agente A: enviando solicitud -> Familia: "
" Castro, Hora: 15, Personas: 12
Agente A: respuesta para familia "
" Castro -> horaSolicitada=15, personas=12, codigoRespuesta=4, horaAsignada=-1
Agente A: enviando solicitud -> Familia: "
" Moreno, Hora: 16, Personas: 4
Agente A: respuesta para familia -> horaSolicitada=0, personas=0, codigoRespuesta=0, horaAsignada=0
Agente A: enviando solicitud -> Familia: "
" Diaz, Hora: 17, Personas: 18
andreslorente@192 Proyecto_SO_final %

```

El controlador inicia a la hora 11 y, por tanto, todas las solicitudes que piden horas anteriores a ese valor son ignoradas o consideradas extemporáneas. A partir de la primera solicitud válida, el sistema comienza a procesar reservas correctamente, aceptando aquellas que cumplen el aforo en la hora solicitada y reprogramando las que requieren desplazamiento a horas posteriores. La ocupación del parque se actualiza de forma precisa en cada avance del reloj, mostrando entradas y salidas en el momento adecuado. En conjunto, los resultados muestran que el mecanismo de validación temporal, aforo y reprogramación está funcionando correctamente bajo estas condiciones.

CP2 – Reprogramación por falta de cupo en hora solicitada

- Controlador con aforo bajo (ej. -t 20).
- CSV con varias familias pidiendo la misma hora y sobrepasando el aforo.
- Esperado: primeras solicitudes aceptadas, siguientes reprogramadas a las primeras dos horas consecutivas con cupo (codigoRespuesta = 2), sin superar el aforo en ningún bloque.
- Resultados:

```

Terminal Shell Edit View Window Help
Proyecto_SO_final --zsh-- 87x54
andresloreto@i92 Proyecto_SO_final % cd ~/Documents/Proyecto_SO_final
./build/controlador -i 7 -f 19 -s 1 -t 20 -p pipeRecibe
[

Hora actual de simulacion: 8
Salen familias: ninguna -> Total que salen: 0
Entran familias: ninguna -> Total que entran: 0
Ocupacion programada para la hora 8: 0 personas
-----


Hora actual de simulacion: 9
Salen familias: ninguna -> Total que salen: 0
Entran familias: ninguna -> Total que entran: 0
Ocupacion programada para la hora 9: 0 personas
-----


Hora actual de simulacion: 10
Salen familias: ninguna -> Total que salen: 0
Entran familias: ninguna -> Total que entran: 0
Ocupacion programada para la hora 10: 0 personas
-----


Controlador: peticion de agente B, familia "
"Velasquez, hora 10, personas 6 -> codigoRespuesta=1, horaAsignada=10
-----


Hora actual de simulacion: 11
Salen familias: ninguna -> Total que salen: 0
Entran familias: ninguna -> Total que entran: 0
Ocupacion programada para la hora 11: 0 personas
-----


Hora actual de simulacion: 12
Salen familias: "
"Velasquez(6) -> Total que salen: 6
Entran familias: ninguna -> Total que entran: 0
Ocupacion programada para la hora 12: 0 personas
-----


Controlador: peticion de agente B, familia "
"Rondon, hora 10, personas 18 -> codigoRespuesta=2, horaAsignada=12
-----


Hora actual de simulacion: 13
Salen familias: ninguna -> Total que salen: 0
Entran familias: ninguna -> Total que entran: 0
Ocupacion programada para la hora 13: 18 personas
-----


Controlador: peticion de agente B, familia "
"Peña, hora 11, personas 5 -> codigoRespuesta=2, horaAsignada=14
-----


Hora actual de simulacion: 14
Salen familias: "
"Rondon(18) -> Total que salen: 18
Entran familias: ninguna -> Total que entran: 0
Ocupacion programada para la hora 14: 0 personas
-----


Controlador: peticion de agente B, familia "
"Peña, hora 11, personas 5 -> codigoRespuesta=2, horaAsignada=14
-----


Hora actual de simulacion: 15
Salen familias: ninguna -> Total que salen: 0
Entran familias: ninguna -> Total que entran: 0
Ocupacion programada para la hora 15: 0 personas
-----


Hora actual de simulacion: 16
Salen familias: "
"Peña(5) -> Total que salen: 5
Entran familias: ninguna -> Total que entran: 0
Ocupacion programada para la hora 16: 0 personas
-----


Controlador: peticion de agente B, familia "
"Acosta, hora 12, personas 20 -> codigoRespuesta=2, horaAsignada=16
-----


Hora actual de simulacion: 17
Salen familias: ninguna -> Total que salen: 0
Entran familias: ninguna -> Total que entran: 0
Ocupacion programada para la hora 17: 20 personas
-----


Hora actual de simulacion: 18
Salen familias: "
"Acosta(20) -> Total que salen: 20
Entran familias: ninguna -> Total que entran: 0
Ocupacion programada para la hora 18: 0 personas
-----


Controlador: peticion de agente B, familia "
"Vega, hora 13, personas 14 -> codigoRespuesta=2, horaAsignada=18
-----


Hora actual de simulacion: 19
Salen familias: ninguna -> Total que salen: 0
Entran familias: ninguna -> Total que entran: 0
Ocupacion programada para la hora 19: 14 personas
-----


Controlador: peticion de agente B, familia "
"Prada, hora 14, personas 8 -> codigoRespuesta=4, horaAsignada=-1
read mensaje: Bad file descriptor
andresloreto@i92 Proyecto_SO_final %

```

```

Terminal Shell Edit View Window Help
Proyecto_SO_final --zsh-- 93x54
andresloreto@i92 Proyecto_SO_final % cd ~/Documents/Proyecto_SO_final
./build/agente -s B -a data/solicitudes.B.csv -p pipeRecibe
[

Agente B registrado. Hora actual de simulacion: 10
Agente B: solicitud ignorada para familia "Perez, hora 8 (hora actual simulacion: 10)
Agente B: solicitud ignorada para familia "
"Garcia, hora 8 (hora actual simulacion: 10)
Agente B: solicitud ignorada para familia "
"Salazar, hora 8 (hora actual simulacion: 10)
Agente B: solicitud ignorada para familia "
"Mejia, hora 8 (hora actual simulacion: 10)
Agente B: solicitud ignorada para familia "
"Ruiz, hora 9 (hora actual simulacion: 10)
Agente B: solicitud ignorada para familia "
"Ortiz, hora 9 (hora actual simulacion: 10)
Agente B: solicitud ignorada para familia "
"Jimenez, hora 9 (hora actual simulacion: 10)
Agente B: enviando solicitud -> Familia: "
"Velasquez, Hora: 10, Personas: 6
Agente B: respuesta para familia "
"Velasquez -> horaSolicitada=10, personas=6, codigoRespuesta=1, horaAsignada=10
Agente B: enviando solicitud -> Familia: "
"Rondon, Hora: 10, Personas: 18
Agente B: respuesta para familia "
"Rondon -> horaSolicitada=10, personas=18, codigoRespuesta=2, horaAsignada=12
Agente B: enviando solicitud -> Familia: "
"Peña, Hora: 11, Personas: 5
Agente B: respuesta para familia "
"Peña -> horaSolicitada=11, personas=5, codigoRespuesta=2, horaAsignada=14
Agente B: enviando solicitud -> Familia: "
"Acosta, Hora: 12, Personas: 20
Agente B: respuesta para familia "
"Acosta -> horaSolicitada=12, personas=20, codigoRespuesta=2, horaAsignada=16
Agente B: enviando solicitud -> Familia: "
"Vega, Hora: 13, Personas: 14
Agente B: respuesta para familia "
"Vega -> horaSolicitada=13, personas=14, codigoRespuesta=2, horaAsignada=18
Agente B: enviando solicitud -> Familia: "
"Prada, Hora: 14, Personas: 8
Agente B: respuesta para familia -> horaSolicitada=0, personas=0, codigoRespuesta=0, horaAsignada=0
Agente B: enviando solicitud -> Familia: "
"Serrano, Hora: 15, Personas: 16
andresloreto@i92 Proyecto_SO_final %


```

```

Terminal Shell Edit View Window Help
Proyecto_SO_final --zsh-- 87x54
andresloreto@i92 Proyecto_SO_final % cd ~/Documents/Proyecto_SO_final
./build/agente -s B -a data/solicitudes.B.csv -p pipeRecibe
[

Agente B registrado. Hora actual de simulacion: 10
Agente B: solicitud ignorada para familia "Perez, hora 8 (hora actual simulacion: 10)
Agente B: solicitud ignorada para familia "
"Garcia, hora 8 (hora actual simulacion: 10)
Agente B: solicitud ignorada para familia "
"Salazar, hora 8 (hora actual simulacion: 10)
Agente B: solicitud ignorada para familia "
"Mejia, hora 8 (hora actual simulacion: 10)
Agente B: solicitud ignorada para familia "
"Ruiz, hora 9 (hora actual simulacion: 10)
Agente B: solicitud ignorada para familia "
"Ortiz, hora 9 (hora actual simulacion: 10)
Agente B: solicitud ignorada para familia "
"Jimenez, hora 9 (hora actual simulacion: 10)
Agente B: enviando solicitud -> Familia: "
"Velasquez, Hora: 10, Personas: 6
Agente B: respuesta para familia "
"Velasquez -> horaSolicitada=10, personas=6, codigoRespuesta=1, horaAsignada=10
Agente B: enviando solicitud -> Familia: "
"Rondon, Hora: 10, Personas: 18
Agente B: respuesta para familia "
"Rondon -> horaSolicitada=10, personas=18, codigoRespuesta=2, horaAsignada=12
Agente B: enviando solicitud -> Familia: "
"Peña, Hora: 11, Personas: 5
Agente B: respuesta para familia "
"Peña -> horaSolicitada=11, personas=5, codigoRespuesta=2, horaAsignada=14
Agente B: enviando solicitud -> Familia: "
"Acosta, Hora: 12, Personas: 20
Agente B: respuesta para familia "
"Acosta -> horaSolicitada=12, personas=20, codigoRespuesta=2, horaAsignada=16
Agente B: enviando solicitud -> Familia: "
"Vega, Hora: 13, Personas: 14
Agente B: respuesta para familia "
"Vega -> horaSolicitada=13, personas=14, codigoRespuesta=2, horaAsignada=18
Agente B: enviando solicitud -> Familia: "
"Prada, Hora: 14, Personas: 8
Agente B: respuesta para familia -> horaSolicitada=0, personas=0, codigoRespuesta=0, horaAsignada=0
Agente B: enviando solicitud -> Familia: "
"Serrano, Hora: 15, Personas: 16
andresloreto@i92 Proyecto_SO_final %


```

En esta prueba, el sistema se comporta como se espera para un escenario de aforo bajo y alta demanda en las mismas horas. El controlador, con aforo 20, acepta primero la reserva de Velasquez a las 10 (6 personas) y luego, cuando llegan más familias para la misma hora (Rondon con 18 personas), detecta que no cabe en el bloque original y la reprograma a la primera franja de dos horas consecutivas con cupo disponible (hora asignada 12,

codigoRespuesta=2). Lo mismo ocurre con solicitudes posteriores: Peña (hora 11, 5 personas) es reubicada a la hora 14, Acosta (hora 12, 20 personas) se mueve a la 16 y Vega (hora 13, 14 personas) se reprograma a la 18, manteniendo siempre la ocupación en cada hora menor o igual a 20. Finalmente, cuando ya no existe un bloque de dos horas consecutivas con capacidad suficiente, la solicitud de Prada es negada (codigoRespuesta=4), mostrando que el algoritmo de reprogramación respeta estrictamente el aforo y solo acepta reservas cuando puede garantizar las dos horas consecutivas requeridas.

CP3 – Negación por aforo máximo de una sola reserva

- Controlador con aforo pequeño (ej. -t 15).
- CSV con una familia que solicita más personas que el aforo (ej. 30).
- Esperado: solicitud negada (codigoRespuesta de negación), sin ocupación registrada para esa familia.
- Resultado:

```

Terminal Shell Edit View Window Help
Projecto_SO_final --zsh-- 87x54
andresloredo@192 Proyecto_SO_final % cd ~/Documents/Proyecto_SO_final
./build/controlador -i 7 -f 19 -s 1 -t 15 -p pipeRecibe
Hora actual de simulacion: 8
Salen familias: ninguna -> Total que salen: 0
Entrar familias: ninguna -> Total que entran: 0
Ocupacion programada para la hora 8: 0 personas
-----
Hora actual de simulacion: 9
Salen familias: ninguna -> Total que salen: 0
Entrar familias: ninguna -> Total que entran: 0
Ocupacion programada para la hora 9: 0 personas
-----
Hora actual de simulacion: 10
Salen familias: ninguna -> Total que salen: 0
Entrar familias: ninguna -> Total que entran: 0
Ocupacion programada para la hora 10: 0 personas
-----
Controlador: peticion de agente CP3, familia "FamiliaGrande", hora 10, personas 30 -> codigoRespuesta=4, horaAsignada=-1
-----
Hora actual de simulacion: 11
Salen familias: ninguna -> Total que salen: 0
Entrar familias: ninguna -> Total que entran: 0
Ocupacion programada para la hora 11: 0 personas
-----
Hora actual de simulacion: 12
Salen familias: ninguna -> Total que salen: 0
Entrar familias: ninguna -> Total que entran: 0
Ocupacion programada para la hora 12: 0 personas
-----
Hora actual de simulacion: 13
Salen familias: ninguna -> Total que salen: 0
Entrar familias: ninguna -> Total que entran: 0
Ocupacion programada para la hora 13: 0 personas
-----
Hora actual de simulacion: 14
Salen familias: ninguna -> Total que salen: 0
Entrar familias: ninguna -> Total que entran: 0
Ocupacion programada para la hora 14: 0 personas
-----
Hora actual de simulacion: 15
Salen familias: ninguna -> Total que salen: 0
Entrar familias: ninguna -> Total que entran: 0
Ocupacion programada para la hora 15: 0 personas
-----
Hora actual de simulacion: 16

```

```

Projecto_SO_final --zsh-- 93x54
andresloredo@192 Proyecto_SO_final % cd ~/Documents/Proyecto_SO_final
./build/agente -s CP3 -a data/solicitudes_CP3.csv -p pipeRecibe
Agente CP3 registrado. Hora actual de simulacion: 10
Agente CP3: enviando solicitud -> Familia: "FamiliaGrande", Hora: 10, Personas: 30
Agente CP3: respuesta para familia "FamiliaGrande" -> horaSolicitada=10, personas=30, codigoRespuesta=4, horaAsignada=-1
Agente CP3 termina.
andresloredo@192 Proyecto_SO_final %

```

El sistema se comportó de acuerdo con la lógica del proyecto: al recibir una solicitud con un número de personas que excede el aforo máximo permitido por hora (30 personas con un aforo de 15), el controlador la rechaza de inmediato sin intentar reprogramarla, dado que es imposible acomodar a la familia en ningún bloque de dos horas dentro del día. El mensaje que devuelve al agente incluye un código de respuesta de negación y una hora asignada de -1,

indicando que la reserva no puede realizarse.

CP4 – Negación por hora extemporánea (hora pasada respecto a la simulación)

- Controlador iniciado en una hora intermedia (ej. -i 10).
- CSV con familias que piden horas menores a 10.
- Esperado: solicitudes negadas o reprogramadas solo si existe un bloque de dos horas posterior disponible; nunca se aceptan reservas en horas ya pasadas.
- Resultado:

```
Terminal Shell Edit View Window Help
Projecto_SO_final --zsh-- 87x54
Salen familias: ninguna -> Total que salen: 0
Entrar familias: ninguna -> Total que entran: 0
Ocupacion programada para la hora 11: 0 personas
-----
Hora actual de simulacion: 12
Salen familias: ninguna -> Total que salen: 0
Entrar familias: ninguna -> Total que entran: 0
Ocupacion programada para la hora 12: 0 personas
-----
Hora actual de simulacion: 13
Salen familias: ninguna -> Total que salen: 0
Entrar familias: ninguna -> Total que entran: 0
Ocupacion programada para la hora 13: 0 personas
-----
Hora actual de simulacion: 14
Salen familias: ninguna -> Total que salen: 0
Entrar familias: ninguna -> Total que entran: 0
Ocupacion programada para la hora 14: 0 personas
-----
Hora actual de simulacion: 15
Salen familias: ninguna -> Total que salen: 0
Entrar familias: ninguna -> Total que entran: 0
Ocupacion programada para la hora 15: 0 personas
-----
Hora actual de simulacion: 16
Salen familias: ninguna -> Total que salen: 0
Entrar familias: ninguna -> Total que entran: 0
Ocupacion programada para la hora 16: 0 personas
-----
Hora actual de simulacion: 17
Salen familias: ninguna -> Total que salen: 0
Entrar familias: ninguna -> Total que entran: 0
Ocupacion programada para la hora 17: 0 personas
-----
Hora actual de simulacion: 18
Salen familias: ninguna -> Total que salen: 0
Entrar familias: ninguna -> Total que entran: 0
Ocupacion programada para la hora 18: 0 personas
-----
Hora actual de simulacion: 19
Salen familias: ninguna -> Total que salen: 0
Entrar familias: ninguna -> Total que entran: 0
Ocupacion programada para la hora 19: 0 personas
write mensaje: Broken pipe
andresloredo@192 Proyecto_SO_final %
```

```
Mon Nov 17 6:37 PM
Projecto_SO_final --zsh-- 93x54
andresloredo@192 Proyecto_SO_final % cd ~/Documents/Proyecto_SO_final
./build/agente -s CP4 -a data/solicitudes_CP4.csv -p pipeReceive
[...]
Agente CP4 registrado. Hora actual de simulacion: 14
Agente CP4: solicitud ignorada para familia "Familia1, hora 7 (hora actual simulacion: 14)
Agente CP4: solicitud ignorada para familia "
"Familia2, hora 8 (hora actual simulacion: 14)
Agente CP4: solicitud ignorada para familia "
"Familia3, hora 9 (hora actual simulacion: 14)
Agente CP4 termina.
andresloredo@192 Proyecto_SO_final %
```

En esta prueba el sistema respondió correctamente al escenario de solicitudes extemporáneas. Como el controlador inicia en la hora 14, todas las reservas del agente que correspondían a horas menores (7, 8 y 9) fueron ignoradas automáticamente, cumpliendo con la regla de que no se pueden aceptar ni procesar solicitudes para horas ya pasadas en la simulación. Esto se refleja en que el agente marca cada una como “solicitud ignorada” y el controlador no las registra ni modifica la ocupación del parque, que permanece en cero durante toda la jornada. La interacción confirma que el mecanismo de validación temporal funciona correctamente y que el sistema previene reasignaciones indebidas en horarios previos a la hora actual.

CP5 – Negación por falta de bloques de dos horas

- Controlador con aforo ajustado para llenar casi todo el día.
- CSV construida para que no quede ningún bloque de 2 horas consecutivas con cupo suficiente.
- Esperado: solicitud negada con mensaje “debe volver otro día” (código de respuesta de negación), sin cambios en la ocupación.
- Resultado:

The screenshot shows two terminal windows side-by-side, both titled "Proyecto_SO_final -- zsh --". The left window shows the initial setup and a series of 15-hour simulations from 8 to 14. The right window shows a continuous sequence of 15-hour simulations starting from hour 10. Both windows display log messages from agents (FULL) regarding family requests, responses, and notifications of being ignored or having to return another day. The logs are timestamped with dates like Mon Nov 17 6:42PM.

```
andresloreto@192 Proyecto_SO_final % cd ~/Documents/Proyecto_SO_final
./build/controlador -i 7 -f 19 -s 1 -t 20 -p pipeRecibe

-----  
Hora actual de simulacion: 8  
Salen familias: ninguna -> Total que salen: 0  
Entran familias: ninguna -> Total que entran: 0  
Ocupacion programada para la hora 8: 0 personas  
-----  
Hora actual de simulacion: 9  
Salen familias: ninguna -> Total que salen: 0  
Entran familias: ninguna -> Total que entran: 0  
Ocupacion programada para la hora 9: 0 personas  
-----  
Hora actual de simulacion: 10  
Salen familias: ninguna -> Total que salen: 0  
Entran familias: ninguna -> Total que entran: 0  
Ocupacion programada para la hora 10: 0 personas  
-----  
Controlador: peticion de agente FULL, familia "  
"Fam10, hora 10, personas 20 -> codigoRespuesta=1, horaAsignada=10  
-----  
Hora actual de simulacion: 11  
Salen familias: ninguna -> Total que salen: 0  
Entran familias: ninguna -> Total que entran: 0  
Ocupacion programada para la hora 11: 20 personas  
-----  
Hora actual de simulacion: 12  
Salen familias: "  
"Fam10(20) -> Total que salen: 20  
Entran familias: ninguna -> Total que entran: 0  
Ocupacion programada para la hora 12: 0 personas  
-----  
Controlador: peticion de agente FULL, familia "  
"Fam11, hora 11, personas 20 -> codigoRespuesta=2, horaAsignada=12  
-----  
Hora actual de simulacion: 13  
Salen familias: ninguna -> Total que salen: 0  
Entran familias: ninguna -> Total que entran: 0  
Ocupacion programada para la hora 13: 20 personas  
-----  
Hora actual de simulacion: 14  
Salen familias: "  
"Fam11(20) -> Total que salen: 20  
Entran familias: ninguna -> Total que entran: 0  
Ocupacion programada para la hora 14: 0 personas  
-----  
Controlador: peticion de agente FULL, familia "  
"Fam12, hora 12, personas 20 -> codigoRespuesta=2, horaAsignada=14  
-----  
-----  
andresloreto@192 Proyecto_SO_final % cd ~/Documents/Proyecto_SO_final
./build/agente -s FULL -a data/solicitudes_full_CPs.csv -p pipeRecibe

Agente FULL registrado. Hora actual de simulacion: 10  
Agente FULL: solicitud ignorada para familia "Fam7, hora 7 (hora actual simulacion: 10)  
Agente FULL: solicitud ignorada para familia "  
"Fam8, hora 8 (hora actual simulacion: 10)  
Agente FULL: solicitud ignorada para familia "  
"Fam9, hora 9 (hora actual simulacion: 10)  
Agente FULL: enviando solicitud -> Familia: "  
"Fam10, Hora: 10, Personas: 20  
Agente FULL: respuesta para familia "  
"Fam10 -> horaSolicitada=10, personas=20, codigoRespuesta=1, horaAsignada=10  
Agente FULL: enviando solicitud -> Familia: "  
"Fam11, Hora: 11, Personas: 20  
Agente FULL: respuesta para familia "  
"Fam11 -> horaSolicitada=11, personas=20, codigoRespuesta=2, horaAsignada=12  
Agente FULL: enviando solicitud -> Familia: "  
"Fam12, Hora: 12, Personas: 20  
Agente FULL: respuesta para familia "  
"Fam12 -> horaSolicitada=12, personas=20, codigoRespuesta=2, horaAsignada=14  
Agente FULL: enviando solicitud -> Familia: "  
"Fam13, Hora: 13, Personas: 20  
Agente FULL: respuesta para familia "  
"Fam13 -> horaSolicitada=13, personas=20, codigoRespuesta=2, horaAsignada=16  
Agente FULL: enviando solicitud -> Familia: "  
"Fam14, Hora: 14, Personas: 20  
Agente FULL: respuesta para familia "  
"Fam14 -> horaSolicitada=14, personas=20, codigoRespuesta=2, horaAsignada=18  
Agente FULL: enviando solicitud -> Familia: "  
"Fam15, Hora: 15, Personas: 20  
Agente FULL: respuesta para familia -> horaSolicitada=0, personas=0, codigoRespuesta=0, hora  
Asignada=0  
Agente FULL: enviando solicitud -> Familia: "  
"Fam16, Hora: 16, Personas: 20  
andresloreto@192 Proyecto_SO_final %
```

```

Terminal Shell Edit View Window Help
Projecto_SO_final --zsh-- 87x54
Hora actual de simulacion: 13
Salen familias: ninguna -> Total que salen: 0
Entran familias: ninguna -> Total que entran: 0
Ocupacion programada para la hora 13: 20 personas
-----
Hora actual de simulacion: 14
Salen familias: "
"Fan1(20) -> Total que salen: 20
Entran familias: ninguna -> Total que entran: 0
Ocupacion programada para la hora 14: 0 personas
-----
Controlador: peticion de agente FULL, familia "
"Fan2, hora 12, personas 20 -> codigoRespuesta=2, horaAsignada=14
-----
Hora actual de simulacion: 15
Salen familias: ninguna -> Total que salen: 0
Entran familias: ninguna -> Total que entran: 0
Ocupacion programada para la hora 15: 20 personas
-----
Hora actual de simulacion: 16
Salen familias: "
"Fan1(20) -> Total que salen: 20
Entran familias: ninguna -> Total que entran: 0
Ocupacion programada para la hora 16: 0 personas
-----
Controlador: peticion de agente FULL, familia "
"Fan3, hora 13, personas 20 -> codigoRespuesta=2, horaAsignada=16
-----
Hora actual de simulacion: 17
Salen familias: ninguna -> Total que salen: 0
Entran familias: ninguna -> Total que entran: 0
Ocupacion programada para la hora 17: 20 personas
-----
Hora actual de simulacion: 18
Salen familias: "
"Fan13(20) -> Total que salen: 20
Entran familias: ninguna -> Total que entran: 0
Ocupacion programada para la hora 18: 0 personas
-----
Controlador: peticion de agente FULL, familia "
"Fan14, hora 14, personas 20 -> codigoRespuesta=2, horaAsignada=18
-----
Hora actual de simulacion: 19
Salen familias: ninguna -> Total que salen: 0
Entran familias: ninguna -> Total que entran: 0
Ocupacion programada para la hora 19: 20 personas
-----
Controlador: peticion de agente FULL, familia "
"Fan15, hora 15, personas 20 -> codigoRespuesta=4, horaAsignada=-1
read mensaje: Bad file descriptor
andresloreno@192 Proyecto_SO_final %

```

```

Projecto_SO_final --zsh-- 93x54
andresloreno@192 Proyecto_SO_final % cd ~/Documents/Proyecto_SO_final
./build/agente -s FULL -a data/solicitudes_full_CPs.csv -p pipeRecibe
Agente FULL registrado. Hora actual de simulacion: 10
Agente FULL: solicitud ignorada para familia "Fan7, hora 7 (hora actual simulacion: 10)
Agente FULL: solicitud ignorada para familia "
"Fan8, hora 8 (hora actual simulacion: 10)
Agente FULL: solicitud ignorada para familia "
"Fan9, hora 9 (hora actual simulacion: 10)
Agente FULL: enviando solicitud -> Familia: "
"Fan10, Hora: 10, Personas: 20
Agente FULL: respuesta para familia "
"Fan10 -> horaSolicitada=10, personas=20, codigoRespuesta=1, horaAsignada=10
Agente FULL: enviando solicitud -> Familia: "
"Fan11, Hora: 11, Personas: 20
Agente FULL: respuesta para familia "
"Fan11 -> horaSolicitada=11, personas=20, codigoRespuesta=2, horaAsignada=12
Agente FULL: enviando solicitud -> Familia: "
"Fan12, Hora: 12, Personas: 20
Agente FULL: respuesta para familia "
"Fan12 -> horaSolicitada=12, personas=20, codigoRespuesta=2, horaAsignada=14
Agente FULL: enviando solicitud -> Familia: "
"Fan13, Hora: 13, Personas: 20
Agente FULL: respuesta para familia "
"Fan13 -> horaSolicitada=13, personas=20, codigoRespuesta=2, horaAsignada=16
Agente FULL: enviando solicitud -> Familia: "
"Fan14, Hora: 14, Personas: 20
Agente FULL: respuesta para familia "
"Fan14 -> horaSolicitada=14, personas=20, codigoRespuesta=2, horaAsignada=18
Agente FULL: enviando solicitud -> Familia: "
"Fan15, Hora: 15, Personas: 20
Agente FULL: respuesta para familia -> horaSolicitada=0, personas=0, codigoRespuesta=0, hora
Asignada=0
Agente FULL: enviando solicitud -> Familia: "
"Fan16, Hora: 16, Personas: 20
andresloreno@192 Proyecto_SO_final %

```

El comportamiento del sistema es exactamente el esperado: el primer agente (“FULL”) llenó todas las horas del día con reservas de 20 personas, ocupando completamente cada franja desde la hora 10 hasta la 18. Esto generó un escenario donde no existe ningún bloque de dos horas consecutivas con capacidad disponible, cumpliendo la condición del caso de prueba. Cuando finalmente el controlador recibe la última solicitud del mismo agente (representando el intento final del día), la reserva es negada con un `codigoRespuesta = 4` y `horaAsignada = -1`, señal clara de que el sistema determinó que no hay posibilidad de acomodar a la familia en ninguna parte del día. La ocupación del parque permanece sin alteraciones, ya que no se aceptan reservas adicionales. En conjunto, la simulación demuestra que el controlador aplica correctamente la lógica de “debe volver otro día” cuando no existen bloques de dos horas libres, respetando el aforo y evitando sobrepasar los límites de capacidad en todo momento.

CP6 – Múltiples agentes concurrentes

- Ejecutar controlador y al menos 2–3 agentes con diferentes archivos CSV (A, B, C) usando el mismo `pipe_parque`.
- Esperado:
 - Todos los agentes se registran correctamente, reciben la hora actual.
 - Las solicitudes de todos se procesan sin interbloqueos.

- La ocupación total por hora corresponde a la suma de todas las reservas aceptadas/reprogramadas de todos los agentes.
- Resultado

```

Terminal Shell Edit View Window Help
Projecto_SO_final -- zsh -- 87x54
Salen familias: "
" Lopez(5) -> Total que salen: 5
Entran familias: ninguna -> Total que entran: 0
Ocupacion programada para la hora 14: 14 personas
Controlador: peticion de agente A, familia "
"Ramirez, hora 12, personas 15 -> codigoRespuesta=2, horaAsignada=14
Hora actual de simulacion: 15
Salen familias: "
" Vega(14) -> Total que salen: 14
Entran familias: ninguna -> Total que entran: 0
Ocupacion programada para la hora 15: 15 personas
Controlador: peticion de agente B, familia "
"Prada, hora 14, personas 8 -> codigoRespuesta=2, horaAsignada=15
Hora actual de simulacion: 16
Salen familias: "
" Ramirez(15) -> Total que salen: 15
Entran familias: ninguna -> Total que entran: 0
Ocupacion programada para la hora 16: 8 personas
Controlador: peticion de agente A, familia "
"Sanchez, hora 13, personas 7 -> codigoRespuesta=2, horaAsignada=16
Hora actual de simulacion: 17
Salen familias: "
" Prada(8) -> Total que salen: 8
Entran familias: ninguna -> Total que entran: 0
Ocupacion programada para la hora 17: 7 personas
Controlador: peticion de agente B, familia "
"Serrano, hora 15, personas 16 -> codigoRespuesta=2, horaAsignada=17
Hora actual de simulacion: 18
Salen familias: "
" Sanchez(7) -> Total que salen: 7
Entran familias: ninguna -> Total que entran: 0
Ocupacion programada para la hora 18: 16 personas
Controlador: peticion de agente A, familia "
"Vargas, hora 14, personas 9 -> codigoRespuesta=2, horaAsignada=18
Hora actual de simulacion: 19
Salen familias: "
" Serrano(16) -> Total que salen: 16
Entran familias: ninguna -> Total que entran: 0
Ocupacion programada para la hora 19: 9 personas
Controlador: peticion de agente B, familia "
"Quintero, hora 16, personas 7 -> codigoRespuesta=4, horaAsignada=-1
write mensaje: Broken pipe
andreslorente@192 Proyecto_SO_final % 
```

```

Projecto_SO_final -- zsh -- 93x16
" Lopez -> horaSolicitada=11, personas=5, codigoRespuesta=2, horaAsignada=12
Agente A: enviando solicitud -> Familia: "
" Ramirez, Hora: 12, Personas: 15
Agente A: respuesta para familia "
" Ramirez -> horaSolicitada=12, personas=15, codigoRespuesta=2, horaAsignada=14
Agente A: enviando solicitud -> Familia: "
" Sanchez, Hora: 13, Personas: 7
Agente A: respuesta para familia "
" Sanchez -> horaSolicitada=13, personas=7, codigoRespuesta=2, horaAsignada=16
Agente A: enviando solicitud -> Familia: "
" Vargas, Hora: 14, Personas: 9
Agente A: respuesta para familia "
" Vargas -> horaSolicitada=14, personas=9, codigoRespuesta=2, horaAsignada=18
Agente A: enviando solicitud -> Familia: "
" Gastro, Hora: 15, Personas: 12
andreslorente@192 Proyecto_SO_final % 
```

```

Projecto_SO_final -- zsh -- 94x15
" Vega -> horaSolicitada=13, personas=14, codigoRespuesta=1, horaAsignada=13
Agente B: enviando solicitud -> Familia: "
" Prada, Hora: 14, Personas: 8
Agente B: respuesta para familia "
" Prada -> horaSolicitada=14, personas=8, codigoRespuesta=2, horaAsignada=15
Agente B: enviando solicitud -> Familia: "
" Serrano, Hora: 15, Personas: 16
Agente B: respuesta para familia "
" Serrano -> horaSolicitada=15, personas=16, codigoRespuesta=2, horaAsignada=17
Agente B: enviando solicitud -> Familia: "
" Quintero, Hora: 16, Personas: 7
Agente B: respuesta para familia "
" Quintero -> horaSolicitada=16, personas=7, codigoRespuesta=4, horaAsignada=-1
Agente B termina.
andreslorente@192 Proyecto_SO_final % 
```

```

Projecto_SO_final -- zsh -- 94x18
Last login: Mon Nov 17 18:46:33 on ttys002
andreslorente@192 ~ % cd ~/Documents/Proyecto_SO_final
./build/agente -s C -a data/solicitudes_C.csv -p pipeRecibe
Agente C registrado. Hora actual de simulacion: 16
No se pudo abrir archivo de solicitudes: No such file or directory
andreslorente@192 Proyecto_SO_final % 
```

El comportamiento observado confirma que el sistema maneja correctamente la concurrencia entre múltiples agentes. Tanto el agente A como el agente B lograron registrarse de forma independiente y comenzaron a enviar sus solicitudes al controlador sin bloqueos ni interferencias, procesándose de manera intercalada y continua. El controlador recibió todas las peticiones, reprogramó las que no podían ser aceptadas en la hora solicitada y negó aquellas para las cuales ya no quedaban bloques de dos horas disponibles, todo mientras mantenía la ocupación de cada hora dentro del límite del aforo. Las horas impresas en la consola muestran cómo la ocupación final resulta de la suma combinada de reservas provenientes de los dos agentes. Finalmente, la ejecución simultánea no produjo interbloqueos, errores de sincronización ni inconsistencias, lo que demuestra que la comunicación mediante pipes nominales, la gestión del tiempo y el procesamiento de solicitudes concurrentes están funcionando de forma estable y correcta.

CP7 – Validación de fin de simulación

- Ejecutar el controlador hasta horaFin con uno o más agentes.
- Esperado:

- Al llegar a la hora final, el controlador envía mensaje de fin, cierra pipes y muestra el reporte final.
- Los agentes terminan tras procesar todas sus líneas y cerrar sus pipes.
- No se aceptan solicitudes para horas mayores a horaFin.

- Resultado:

El comportamiento de la prueba es correcto y coincide con el funcionamiento esperado del sistema al llegar a la hora final de simulación. El controlador avanzó secuencialmente desde la hora inicial hasta la horaFin (19) imprimiendo el estado de ocupación en cada franja, y procesando las solicitudes del agente únicamente dentro de los límites permitidos. Todas las

solicitudes con horas válidas fueron aceptadas o reprogramadas según disponibilidad, mientras que las solicitudes para horas sin cupo o fuera de rango fueron negadas correctamente. Una vez alcanzada la hora final, el controlador cerró el pipe y generó el típico mensaje de broken pipe, lo cual es normal porque el proceso principal ya terminó su ciclo. El agente también finalizó su ejecución después de enviar todas sus solicitudes y recibir las respuestas, sin quedar colgado ni esperando indefinidamente.

5. Casos de prueba de robustez / errores

CR1 – CSV con hora fuera del rango [7,19]

- Familias pidiendo hora 6 o 20.
- Esperado: solicitudes rechazadas, sin afectar la ocupación.
- Resultado: "Las solicitudes con hora 6 o 20 se rechazan con codigoRespuesta=4 y horaAsignada=-1, sin registrar entrada de personas ni modificar la ocupación en ninguna hora."

CR2 – CSV con formato incorrecto

- Línea malformada (faltan comas o número de personas).
- Esperado: el agente maneja el error (o descarta la línea) sin bloquearse ni colgar al controlador.
- Resultado: "La línea malformada no se procesa porque fscanf falla; el agente la salta implícitamente y continúa con la siguiente sin colgarse ni afectar al controlador, que nunca recibe un mensaje inválido."

CR3 – Pipe principal inexistente

- Intentar lanzar un agente antes de crear/iniciar el controlador (sin pipe_parque).
- Esperado: el agente debe bloquearse en open o mostrar error controlado; no debe generar comportamiento indefinido.
- Resultado: "El agente queda bloqueado al intentar abrir el pipe en modo escritura o muestra error de 'No such file or directory'; no envía solicitudes, no afecta al controlador y no produce comportamiento indefinido."

```

Terminal Shell Edit View Window Help
Projecto_SO_final -- zsh - 89x54
Controlador: peticion de agente A, familia "
"Pineda, hora 18, personas 6 -> codigoRespuesta=1, horaAsignada=18
Controlador: peticion de agente A, familia "
"Torres, hora 19, personas 8 -> codigoRespuesta=2, horaAsignada=13
-----
Hora actual de simulacion: 14
Salen familias: ninguna -> Total que salen: 0
Entran familias: ninguna -> Total que entran: 0
Ocupacion programada para la hora 14: 30 personas
-----
Hora actual de simulacion: 15
Salen familias: "
"Torres(8), "
"Sanchez(7), "
"Ramirez(15) -> Total que salen: 30
Entran familias: "
"Castro(12), "
"Vargas(9) -> Total que entran: 21
Ocupacion programada para la hora 15: 21 personas
-----
Hora actual de simulacion: 16
Salen familias: ninguna -> Total que salen: 0
Entran familias: "
"Moreno(4) -> Total que entran: 4
Ocupacion programada para la hora 16: 25 personas
-----
Hora actual de simulacion: 17
Salen familias: "
"Castro(12), "
"Vargas(9) -> Total que salen: 21
Entran familias: "
"Diaz(10) -> Total que entran: 10
Ocupacion programada para la hora 17: 14 personas
-----
Hora actual de simulacion: 18
Salen familias: "
"Moreno(4) -> Total que salen: 4
Entran familias: "
"Pineda(6) -> Total que entran: 6
Ocupacion programada para la hora 18: 16 personas
-----
Hora actual de simulacion: 19
Salen familias: "
"Diaz(10) -> Total que salen: 10
Entran familias: ninguna -> Total que entran: 0
Ocupacion programada para la hora 19: 6 personas
-----
write mensaje: Broken pipe
andresloreto@192 Proyecto_SO_final % 
```



```

Projecto_SO_final -- zsh - 87x54
Agente A: enviando solicitud -> Familia: "
"Rojas, Hora: 10, Personas: 10
Agente A: respuesta para familia "
"Rojas -> horaSolicitada=10, personas=10, codigoRespuesta=1, horaAsignada=10
Agente A: enviando solicitud -> Familia: "
"Martinez, Hora: 7, Personas: 6
Agente A: respuesta para familia "
"Martinez -> horaSolicitada=7, personas=6, codigoRespuesta=2, horaAsignada=9
Agente A: enviando solicitud -> Familia: "
"Ochoa, Hora: 9, Personas: 8
Agente A: respuesta para familia "
"Gomez -> horaSolicitada=9, personas=8, codigoRespuesta=1, horaAsignada=9
Agente A: enviando solicitud -> Familia: "
"Hernandez, Hora: 9, Personas: 12
Agente A: respuesta para familia "
"Hernandez -> horaSolicitada=9, personas=12, codigoRespuesta=2, horaAsignada=11
Agente A: enviando solicitud -> Familia: "
"Lopez, Hora: 11, Personas: 5
Agente A: respuesta para familia "
" Lopez -> horaSolicitada=11, personas=5, codigoRespuesta=1, horaAsignada=11
Agente A: enviando solicitud -> Familia: "
"Ramirez, Hora: 12, Personas: 15
Agente A: respuesta para familia "
"Ramirez -> horaSolicitada=12, personas=15, codigoRespuesta=2, horaAsignada=13
Agente A: enviando solicitud -> Familia: "
"Moreno, Hora: 13, Personas: 7
Agente A: respuesta para familia "
"Sanchez -> horaSolicitada=13, personas=7, codigoRespuesta=1, horaAsignada=13
Agente A: enviando solicitud -> Familia: "
"Vargas, Hora: 14, Personas: 9
Agente A: respuesta para familia "
"Vargas -> horaSolicitada=14, personas=9, codigoRespuesta=2, horaAsignada=15
Agente A: enviando solicitud -> Familia: "
"Castro, Hora: 15, Personas: 12
Agente A: respuesta para familia "
"Castro -> horaSolicitada=15, personas=12, codigoRespuesta=1, horaAsignada=15
Agente A: enviando solicitud -> Familia: "
"Moreno, Hora: 16, Personas: 4
Agente A: respuesta para familia "
"Moreno -> horaSolicitada=16, personas=4, codigoRespuesta=1, horaAsignada=16
Agente A: enviando solicitud -> Familia: "
"Diaz, Hora: 17, Personas: 10
Agente A: respuesta para familia "
"Diaz -> horaSolicitada=17, personas=10, codigoRespuesta=1, horaAsignada=17
Agente A: enviando solicitud -> Familia: "
"Pineda, Hora: 18, Personas: 6
Agente A: respuesta para familia "
"Pineda -> horaSolicitada=18, personas=6, codigoRespuesta=1, horaAsignada=18
Agente A: enviando solicitud -> Familia: "
"Torres, Hora: 19, Personas: 8
Agente A: respuesta para familia "
"Torres -> horaSolicitada=19, personas=8, codigoRespuesta=2, horaAsignada=19
Agente A termina.
andresloreto@192 Proyecto_SO_final % 
```

Cada agente registra su nombre y comunica sus solicitudes al controlador, quien las procesa de acuerdo con las reglas del enunciado: verificar disponibilidad de aforo, validar que existan dos horas consecutivas libres y detectar solicitudes extemporáneas. En los resultados se observa que las solicitudes válidas dentro de la hora actual de simulación son aceptadas inmediatamente, mientras que aquellas que saturan un horario son reprogramadas a la siguiente franja disponible. Cuando una familia solicita una hora ya pasada o cuando no existe ningún bloque de dos horas consecutivas con espacio suficiente, el sistema responde con una negación, cumpliendo estrictamente la lógica definida. Esto demuestra que el algoritmo de verificación de horarios, ocupación por hora y reprogramación automática está funcionando de forma coherente y precisa.

A lo largo de la simulación temporal, el controlador avanza hora a hora e imprime la entrada y salida de familias en la franja correspondiente, lo que permite verificar que las reservas aprobadas realmente ocupan el parque en el momento correcto. La ocupación mostrada por el controlador coincide con las familias aprobadas y reprogramadas, formando horas pico y horas valle que reflejan fielmente la distribución y carga del sistema. El comportamiento del agente también confirma que la comunicación es exitosa: envía cada solicitud, recibe una respuesta clara del controlador y termina correctamente al finalizar su archivo. Incluso el mensaje final de broken pipe evidencia que el controlador cerró el pipe tras concluir la simulación, lo cual es esperado y confirma el ciclo completo de vida del sistema. En conjunto, los resultados muestran que tanto la concurrencia, como el manejo del tiempo, la sincronización por pipes y la lógica de reservas se ejecutan de acuerdo al diseño propuesto.

Conclusiones:

El pasado proyecto nos permitió implementar de forma práctica varios de los conceptos aprendidos en clase. Entre estos se destacan la comunicación entre procesos mediante pipes, la sincronización y concurrencia a través de hilos POSIX y el uso de un mutex para proteger las estructuras compartidas. Gracias a esto no solo pudimos ver la teoría aplicada en un caso real, sino también entender mejor los retos de coordinar varios procesos e hilos que trabajan al mismo tiempo sobre un mismo recurso, reforzando así nuestra comprensión de los temas vistos en el curso.

Referencias:

Para el desarrollo de este proyecto se utilizaron los siguientes materiales:

1. apuntes y diapositivas de la asignatura.
2. <https://www.geeksforgeeks.org/c/multithreading-in-c/>
3. <https://www.geeksforgeeks.org/c/c-program-demonstrate-fork-and-pipe/>