

# Informe del Proyecto: Sistema de Reservas del Parque Berlín

Juan Esteban Muñoz Díaz

19 de noviembre de 2023

## Índice

<b>1. Introducción</b>	<b>2</b>
<b>2. Descripción del Sistema</b>	<b>3</b>
2.1. Arquitectura General . . . . .	3
2.2. Funciones del Controlador de Reserva . . . . .	3
2.3. Funciones de los Agentes de Reserva . . . . .	3
<b>3. Implementación</b>	<b>4</b>
3.1. Detalles Técnicos . . . . .	4
<b>4. Resultados y Pruebas</b>	<b>4</b>
4.1. Casos de Prueba . . . . .	4
4.2. Comportamiento Observado . . . . .	5
<b>5. Conclusiones</b>	<b>6</b>
5.1. Aprendizaje y Desafíos . . . . .	6
5.2. Posibles Mejoras . . . . .	6
<b>6. Anexos</b>	<b>6</b>

# 1. Introducción

El proyecto desarrollado consiste en la simulación de un sistema de reserva de horas para el Parque Berlín, un escenario ficticio que representa un entorno de parque urbano. Este sistema, diseñado para abordar desafíos reales en la gestión de espacios públicos, utiliza una arquitectura cliente/servidor para gestionar eficientemente el aforo del parque y mejorar la experiencia del usuario.

La implementación de este sistema aborda varios aspectos críticos de la programación de sistemas y la informática aplicada. El principal entre ellos es el desarrollo de un sistema concurrente que utilice procesos y hilos POSIX, reflejando el comportamiento en tiempo real de un sistema operativo moderno. La simulación del sistema de reserva para el Parque Berlín es un caso de estudio que ilustra la aplicación de estos conceptos en un contexto práctico.

El sistema se compone de dos componentes principales: el Controlador de Reserva, que actúa como servidor, y los Agentes de Reserva, que funcionan como clientes. El Controlador de Reserva se encarga de gestionar las solicitudes de reserva, mantener un registro actualizado del aforo del parque, y simular el paso del tiempo para ajustar el estado de las reservas. Por otro lado, los Agentes de Reserva simulan a los usuarios finales que interactúan con el sistema, enviando solicitudes de reserva y recibiendo respuestas del servidor.

Este enfoque cliente/servidor no solo demuestra la implementación técnica de un sistema distribuido, sino que también pone en práctica la comunicación entre procesos a través de pipes nominales. La habilidad para manejar eficientemente la comunicación entre diferentes procesos es un aspecto crucial en la programación de sistemas operativos y es explorada a fondo en este proyecto.

Además, el proyecto incorpora el manejo de señales y archivos en un entorno de múltiples procesos, lo cual es fundamental para la estabilidad y eficiencia del sistema. Estos elementos reflejan desafíos comunes en el desarrollo de software para sistemas operativos y son abordados mediante técnicas de programación avanzadas.

En resumen, el proyecto del sistema de reserva del Parque Berlín no solo representa una aplicación práctica de conceptos clave en sistemas operativos y programación concurrente, sino que también ilustra cómo la tecnología puede ser utilizada para resolver problemas complejos y mejorar la gestión y

operatividad de espacios públicos.

## **2. Descripción del Sistema**

### **2.1. Arquitectura General**

El sistema diseñado para la simulación de reservas en el Parque Berlín emplea una arquitectura cliente/servidor, una estructura común en aplicaciones distribuidas. En esta arquitectura, el Controlador de Reserva actúa como el servidor, centralizando la lógica de negocio y la gestión de datos. Los Agentes de Reserva, por otro lado, funcionan como clientes, interactuando directamente con los usuarios finales (simulados en este caso). Esta división permite una clara separación de responsabilidades y facilita la escalabilidad del sistema.

### **2.2. Funciones del Controlador de Reserva**

El Controlador de Reserva, como núcleo del sistema, tiene múltiples funciones críticas. Primero, gestiona todas las solicitudes de reserva enviadas por los Agentes de Reserva, incluyendo la asignación de horarios y la confirmación o rechazo de reservas. Además, mantiene un registro actualizado del aforo del parque, asegurando que no se excedan los límites de capacidad. El controlador también simula el paso del tiempo dentro del parque, ajustando el estado de las reservas (activas, pendientes o finalizadas) y generando informes periódicos sobre la utilización del parque y las estadísticas de reserva.

### **2.3. Funciones de los Agentes de Reserva**

Los Agentes de Reserva representan la interfaz del usuario en el sistema. Su función principal es enviar solicitudes de reserva al Controlador de Reserva. Estas solicitudes pueden variar en términos de horarios y número de visitantes. Una vez enviada una solicitud, los agentes manejan las respuestas del servidor, que pueden ser confirmaciones de reserva exitosa o notificaciones de rechazo, en caso de que no haya disponibilidad. Los agentes también pueden solicitar información adicional sobre el estado actual del aforo del

parque o cancelar reservas existentes. Esta flexibilidad en las operaciones permite simular un entorno de reserva realista y dinámico.

## **3. Implementación**

### **3.1. Detalles Técnicos**

La implementación del sistema de reserva para el Parque Berlín abarca varios aspectos técnicos cruciales, reflejando las prácticas avanzadas en la programación de sistemas operativos. Uno de los elementos centrales es la gestión de procesos, donde cada Agente de Reserva se ejecuta como un proceso independiente. Esta aproximación garantiza un aislamiento efectivo de las operaciones y una simulación realista del comportamiento de múltiples usuarios.

En el núcleo del sistema, el Controlador de Reserva, implementado como un proceso separado, maneja todas las interacciones con los agentes. La comunicación entre procesos se realiza mediante pipes nominales, lo que permite un intercambio de datos seguro y eficiente entre el servidor y los clientes. Este método de comunicación es fundamental para mantener la integridad de los datos en un entorno concurrente.

La sincronización de procesos es otro aspecto técnico relevante. Se utilizan mecanismos de sincronización, como semáforos o variables de condición, para controlar el acceso a recursos compartidos, como el registro del aforo del parque. Esta técnica asegura que las actualizaciones al estado del parque se realicen de manera atómica, evitando condiciones de carrera y garantizando la consistencia de los datos.

## **4. Resultados y Pruebas**

### **4.1. Casos de Prueba**

Para evaluar la eficacia y robustez del sistema de reservas del Parque Berlín, se diseñaron varios casos de prueba. Estos casos incluyen situaciones como múltiples solicitudes concurrentes, reservas para horarios de máxima demanda, y manejo de situaciones de aforo completo. También se probaron escenarios de error, como solicitudes de reserva inválidas y la interrupción

inesperada de procesos de agentes. La metodología utilizada para estas pruebas se centró en simular el comportamiento real de los usuarios y las condiciones operativas del parque, garantizando así una evaluación exhaustiva del sistema.

ID del Caso	Descripción	Pasos	Resultado Esperado
CP-01	Solicitud de reserva en horario de baja demanda	1. Iniciar Agente de Reserva. 2. Enviar solicitud de reserva para un horario con baja demanda.	Reserva confirmada.
CP-02	Solicitud de reserva en horario de máxima demanda	1. Iniciar Agente de Reserva. 2. Enviar solicitud de reserva para un horario de máxima demanda.	Reserva confirmada o en lista de espera.
CP-03	Múltiples solicitudes concurrentes	1. Iniciar varios Agentes de Reserva. 2. Enviar solicitudes de reserva simultáneamente.	Todas las reservas son procesadas adecuadamente sin errores.
CP-04	Manejo de aforo completo	1. Llenar el aforo del parque. 2. Enviar solicitud de reserva adicional.	Reserva rechazada debido a aforo completo.
CP-05	Cancelación de reserva	1. Realizar una reserva exitosa. 2. Solicitar cancelación de la reserva.	Reserva cancelada exitosamente.

Cuadro 1: Casos de Prueba para el Sistema de Reservas del Parque Berlín

## 4.2. Comportamiento Observado

Durante las pruebas, el sistema demostró una gestión eficiente de múltiples solicitudes concurrentes. Se observó que el Controlador de Reserva ma-

nejaba adecuadamente el aforo del parque, asignando horarios disponibles y rechazando solicitudes cuando se alcanzaba el límite de capacidad. En los casos de prueba de error, el sistema mostró una adecuada resistencia, manejando excepciones y asegurando la continuidad del servicio. Sin embargo, se identificaron áreas de mejora en términos de tiempo de respuesta y manejo de situaciones de alta concurrencia.

## **5. Conclusiones**

### **5.1. Aprendizaje y Desafíos**

El desarrollo de este proyecto proporcionó una valiosa experiencia en la programación de sistemas operativos y la gestión de procesos concurrentes. Los principales desafíos enfrentados incluyeron la implementación eficiente de la comunicación entre procesos y la sincronización de múltiples hilos de ejecución. A través de este proyecto, se adquirió una comprensión más profunda de los conceptos de sistemas operativos y una experiencia práctica en la resolución de problemas complejos de programación concurrente.

### **5.2. Posibles Mejoras**

Como posibles mejoras para el sistema, se sugiere la optimización del manejo de la concurrencia, especialmente bajo cargas de trabajo elevadas. Otras extensiones podrían incluir una interfaz de usuario más interactiva para los Agentes de Reserva y la implementación de un módulo de análisis de datos en tiempo real para el Controlador de Reserva, mejorando así la toma de decisiones y la gestión operativa del parque.

## **6. Anexos**

```

-----
Reserva:
Familia: Zuluaga
Hora Inicio:10
Cantidad personas: 8
-----

ENTRANDO A HILO

Fifo: /tmp/Zuluaga
Fifo: /tmp/Zuluaga
Enviando Respuesta: RESERVA ACEPTADA
-----

Reserva:
Familia: Rojas
Hora Inicio:10
Cantidad personas: 10
-----

ENTRANDO A HILO

Fifo: /tmp/Rojas
Fifo: /tmp/Rojas
Enviando Respuesta: RESERVA ACEPTADA
-> Es posible la reserva

SALIENDO DEL HILO
-> Es posible la reserva

SALIENDO DEL HILO
->HORA ACTUAL<=: 10 hrs
->HORA ACTUAL<=: 11 hrs
->HORA ACTUAL<=: 12 hrs
->HORA ACTUAL<=: 13 hrs
->HORA ACTUAL<=: 14 hrs
->HORA ACTUAL<=: 15 hrs
->HORA ACTUAL<=: 16 hrs
->HORA ACTUAL<=: 17 hrs
->HORA ACTUAL<=: 18 hrs
->HORA ACTUAL<=: 19 hrs
-----

GENERANDO REPORTE
-----
Agregando en: 10 -> 8
Agregando en: 11 -> 8
Agregando en: 10 -> 10
Agregando en: 11 -> 10
Reporte de Ocupación del Parque:
Horas Pico: 10 11
Horas con Menor Cantidad de Personas: 8 9 12 13 14 15 16 17 18
Número de Solicitudes Negadas: 0
Número de Solicitudes Aceptadas: 2
Número de Solicitudes Reprogramadas: 0
A Juaes ~/sop/p2/Simulacion-sistema-de-reservas git-[v main]- >>

```

```
^ Juaesm ~/sop/p2/Simulacion-sistema-de-reservas git-[v main]- >> ./agent -s Reserva1 -a reserva.txt -p /tmp/pruebas
Realizando conexión
Hora Actual: 9
Reserva 2: Zuluaga, 8, 10
Fifo: /tmp/Zuluaga
Respuesta: RESERVA ACEPTADA
Reserva 4: Rojas, 10, 10
Fifo: /tmp/Rojas
Respuesta: RESERVA ACEPTADA
^ Juaesm ~/sop/p2/Simulacion-sistema-de-reservas git-[v main]- >> |
```



```

SALIENDO DEL HILO
-----
Reserva:
Familia: Rojas
Hora Inicio:10
Cantidad personas: 10
-----

ENTRANDO A HILO

Fifo: /tmp/Rojas
Fifo: /tmp/Rojas
Enviando Respuesta: RESERVA ACEPTADA
-> Es posible la reserva

SALIENDO DEL HILO
-----
Reserva:
Familia: Castro
Hora Inicio:20
Cantidad personas: 10
-----

ENTRANDO A HILO
Nuevo Horario: 12
Fifo: /tmp/Castro
Fifo: /tmp/Castro
Enviando Respuesta: RESERVA ACEPTADA A LAS 12
-> Es posible la reserva

SALIENDO DEL HILO
->HORA ACTUAL<=: 10 hrs
->HORA ACTUAL<=: 11 hrs
->HORA ACTUAL<=: 12 hrs
->HORA ACTUAL<=: 13 hrs
->HORA ACTUAL<=: 14 hrs
->HORA ACTUAL<=: 15 hrs
-----
GENERANDO REPORTE
-----
Agregando en: 10 -> 8
Agregando en: 11 -> 8
Agregando en: 10 -> 10
Agregando en: 11 -> 10
Agregando en: 12 -> 10
Agregando en: 13 -> 10
Reporte de Ocupación del Parque:
Horas Pico: 10 11
Horas con Menor Cantidad de Personas: 8 9 14
Número de Solicitudes Negadas: 0
Número de Solicitudes Aceptadas: 2
Número de Solicitudes Reprogramadas: 1
^ Juaesm ~/sop/p2/Simulacion-sistema-de-reservas git-[v main]- >>

```

```
^ Juaesm ~/sop/p2/Simulacion-sistema-de-reservas git-[v main]- >> ./agent -s Reserva1 -a reserva.txt -p /tmp/pruebas
Realizando conexión
Hora Actual: 9
Reserva 2: Zuluaga, 8, 10
Fifo: /tmp/Zuluaga
Respuesta: RESERVA ACEPTADA
Reserva 4: Rojas, 10, 10
Fifo: /tmp/Rojas
Respuesta: RESERVA ACEPTADA
Reserva 5: Castro, 10, 20
Fifo: /tmp/Castro
Respuesta: RESERVA ACEPTADA A LAS 12
^ Juaesm ~/sop/p2/Simulacion-sistema-de-reservas git-[v main]- >> |
```

```
^ Juaesm ~/sop/p2/Simulacion-sistema-de-reservas git-[v main]- >> ./agent -s Reserva1 -a reserva.txt -p /tmp/pruebas
Realizando conexión
Hora Actual: 8
Reserva 2: Zuluaga, 8, 10
Fifo: /tmp/Zuluaga
Respuesta: RESERVA ACEPTADA
Reserva 4: Rojas, 10, 10
Fifo: /tmp/Rojas
Respuesta: RESERVA ACEPTADA
Reserva 5: Castro, 23, 23
Fifo: /tmp/Castro
Respuesta: RESERVA NEGADA
^ Juaesm ~/sop/p2/Simulacion-sistema-de-reservas git-[v main]- >> |
```

```
^ Juaesm ~/sop/p2/Simulacion-sistema-de-reservas git-[v main]- >> ./agent -s Reserva1 -a reserva.txt -p /tmp/pruebas
Realizando conexión
Hora Actual: 9
Reserva 2: Zuluaga, 8, 10
Fifo: /tmp/Zuluaga
Respuesta: RESERVA ACEPTADA
Reserva 4: Rojas, 10, 10
Fifo: /tmp/Rojas
Respuesta: RESERVA ACEPTADA
Reserva 5: Castro, 23, 23
Fifo: /tmp/Castro
Respuesta: RESERVA NEGADA
^ Juaesm ~/sop/p2/Simulacion-sistema-de-reservas git-[v main]- >> |
```