

ESCUELA TÉCNICA N.º36 D.E. 15  
“ALMIRANTE GUILLERMO BROWN”

Programación sobre redes

# **ROUTERFOOL PARK**

Docente: Alan Uzal

Grupo: Cativa Rodrigo, Silva  
Benjamín, Vargas Lucas

Curso: 6º 3º

Turno: Mañana

Materia: Programación sobre redes

# ÍNDICE

INTRODUCCIÓN-----	3
OBJETIVOS-----	3
DESCRIPCIÓN GENERAL DEL SISTEMA-----	3
DISEÑO DEL SISTEMA-----	3
MANEJO DE CONCURRENCIA-----	4
FUNCIONAMIENTO-----	4
DIAGRAMA DE CLASES-----	5
ESQUEMA DE TABLAS-----	5
VISITANTES-----	5
JUEGOS-----	5
BANOS-----	5
DIAGRAMA-----	6
DIAGRAMA DE GANTT-----	6

## INTRODUCCIÓN

Este proyecto consiste en la simulación de un parque de atracciones con lógica concurrente. El sistema permite crear visitantes que recorren diferentes juegos y baños del parque, utilizando programación multihilo para reflejar el comportamiento simultáneo de varias personas. Además, se prepara una conexión con base de datos para almacenar información sobre los recursos y visitantes.

## OBJETIVOS

El principal objetivo del proyecto es simular un parque de atracciones con múltiples visitantes en movimiento al mismo tiempo. A partir de ello, se busca usar hilos y mecanismos de concurrencia como semáforos, colas y bloqueos para controlar el acceso a recursos con capacidad limitada

También se propone el registro completo de la actividad del parque en una base de datos relacional, y la construcción de una interfaz gráfica intuitiva que facilite la visualización y control de la simulación

## DESCRIPCIÓN GENERAL DEL SISTEMA

El sistema permite crear tres tipos de visitantes: niños, adultos y packs familiares (grupo de 4 personas). Cada visitante (o grupo) elige aleatoriamente dos juegos y un baño para visitar. Si el recurso está ocupado, deben esperar en una cola hasta que se libere espacio

Los recursos del parque (juegos y baños) tienen una duración de uso y una capacidad limitada, lo cual genera condiciones ideales para el uso de mecanismos de sincronización.

El programa incluye una interfaz gráfica hecha en Tkinter para visualizar los logs en tiempo real, crear visitantes y lanzar simulaciones

## DISEÑO DEL SISTEMA

El sistema está dividido en módulos:

- **recursos.py**: contiene las clases **Juego**, **Baño** y **Visitante**.
  - **Juego** y **Baño** representan los recursos del parque. Ambos tienen nombre (str), capacidad (int), duración de uso (int), un semáforo para controlar el acceso, una cola (queue.Queue) y un lock
  - **Visitante** tiene nombre (str), tipo (str), bandera para mostrar logs (bool) y un posible grupo\_id para packs familiares

- **parque.py**: clase principal [Parque](#), que gestiona los recursos y visitantes, organiza la simulación, separa visitantes en subgrupos según la capacidad del recurso y controla las colas y esperas
- **bd.py**: clase [BaseDatos](#), preparada para insertar visitantes y recursos en la base de datos routerfoot.
- **interfaz.py**: interfaz gráfica con botones para crear visitantes y correr simulaciones en procesos separados, con logs y estado de colas visibles.
- **procesos.py**: módulo auxiliar que permite correr la simulación en otro proceso, para no bloquear la interfaz gráfica

## MANEJO DE CONCURRENCIA

La concurrencia se maneja principalmente con hilos (`threading.Thread`), semaforización (`Semaphore`) y colas (`queue.Queue`). Cada visitante (o grupo) se maneja en un hilo separado, los recursos tienen una capacidad fija, y su semáforo asegura que no ingresen más personas que las permitidas. Las colas de espera están sincronizadas mediante `Lock`, y solo el primer grupo en la cola puede intentar adquirir los permisos del semáforo. El uso de estos mecanismos evita condiciones de carrera y garantiza un uso justo de los recursos

## FUNCIONAMIENTO

El sistema comienza con la inicialización del parque, el cual contiene una lista predefinida de juegos y baños, cada uno con su capacidad y duración de uso. A continuación, se permite la creación de visitantes, ya sea de forma manual (individuales o packs familiares) o automática.

Una vez creados los visitantes, se lanza la simulación del parque. En este punto, cada visitante (o grupo) se gestiona de forma concurrente mediante hilos separados. Luego, cada uno de ellos selecciona aleatoriamente dos juegos y un baño que desea visitar durante su recorrido.

A medida que intentan ingresar a un recurso, primero se verifica si hay espacio disponible según el semáforo correspondiente. Si el recurso está ocupado, entonces los visitantes deben esperar su turno en una cola sincronizada. Esta cola está protegida con locks para garantizar un acceso ordenado y evitar interferencias entre hilos.

Cuando el visitante o grupo logra ingresar, permanece en el recurso durante el tiempo establecido por su duración. Al finalizar, libera los permisos del semáforo, permitiendo el ingreso de otros visitantes en espera.

Finalmente, tras completar su paso por los dos juegos y el baño asignado, el visitante abandona el parque y se registra su salida mediante mensajes en los logs.

## DIAGRAMA DE CLASES

## ESQUEMA DE TABLAS

### VISITANTES

Campo	Tipo	Clave	Descripción
id	INT	PK, AI	Identificador único del visitante
nombre	VARCHAR(100)		Nombre del visitante
mostrar_logs	BOOLEAN		Si imprime sus movimientos (True/False)

### JUEGOS

Campo	Tipo	Clave	Descripción
id	INT	PK, AI	ID único del juego
nombre	VARCHAR(100)		Nombre del juego
capacidad	INT		Cuántos visitantes simultáneamente
duracion	INT		Duración en segundos

### BANOS

Campo	Tipo	Clave	Descripción
id	INT	PK, AI	ID único del baño
nombre	VARCHAR(100)		Nombre del baño
capacidad	INT		Capacidad simultánea de visitantes
duracion	INT		Duración en segundos

DIAGRAMA

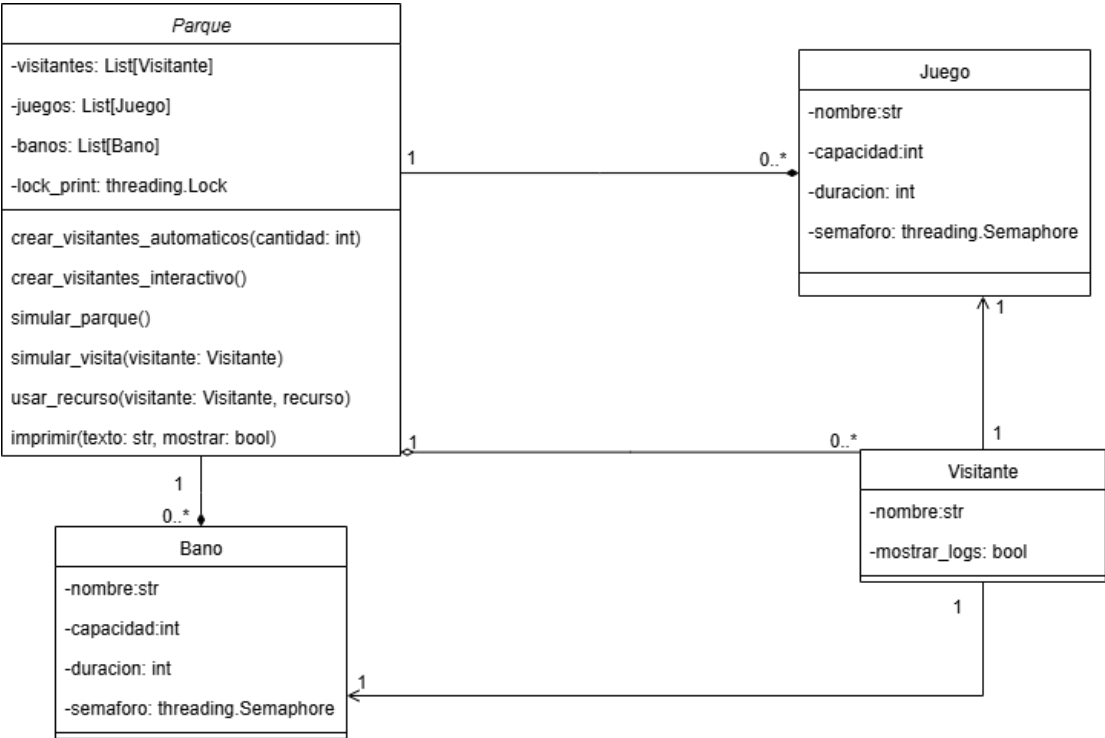


DIAGRAMA DE GANTT

