



**UNIVERSIDADE FEDERAL DE SANTA CATARINA CAMPUS
TRINDADE
INE-DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA
INE5410 - PROGRAMAÇÃO CONCORRENTE**

Alunos:

Arthur de Lara Machado - 22200348
Felipe Fagundes Pacheco - 22202615
Pedro Costa Casotti - 22200374

Trabalho 2 - IxferaTM
Profs. Giovani Gracioli e Márcio Castro

Florianópolis 2023

Resumo:

A atração IxferaTM em Florianópolis, modelada após o SphereTM de Las Vegas, é um sucesso. Com três experiências para diferentes faixas etárias, o desafio é simular seu funcionamento com um programa em Python. Cada pessoa é representada por uma thread, e mecanismos de sincronização devem ser aplicados sem uso de espera ocupada.

O projeto foi estruturado em três partes principais: Pessoas, responsáveis por todo o comportamento executado por uma pessoa e também consta seu tempo de permanência em uma atração. Ixphere_class, representa a atração IxferaTM, nesta parte se tem a lógica de funcionamento da atração, incorporando semáforos e mutexes para garantir sincronização entre as threads. Main ou Ixphere, nesta parte é onde fica responsável pela execução do programa de simulação da atração IxferaTM.

Soluções e Lógicas adotadas:

A simulação da atração IxferaTM em Florianópolis é composta por três partes essenciais: a classe Ixfera que representa o funcionamento da atração, a classe Pessoa que simula o comportamento de cada visitante, e o código principal (ixphere.py) responsável pela execução da simulação. Essas três partes colaboram para criar uma simulação dinâmica e organizada da atração IxferaTM, garantindo que as regras e restrições estabelecidas sejam respeitadas durante toda a execução do programa. Abaixo será descrito, a lógica de cada parte.

Código da Classe Ixphere:

A classe Ixfera desempenha um papel central na simulação da atração IxferaTM, gerenciando o fluxo de visitantes, o início e pausa das experiências, e garantindo a ordem e sincronização adequadas durante a execução do programa.

Método __init__:

- Inicializa os atributos da instância, como o número de vagas, tempo de permanência, entre outros.

Método full:

- Retorna True se a atração estiver cheia (número máximo de pessoas atingido), e False caso contrário.

Método iniciar_experiencia:

- Inicia a experiência da atração para uma determinada faixa etária se não houver uma experiência em curso.

Método entrar_na_ixfera:

- Permite que uma pessoa entre na atração, aguardando se necessário. Atualiza o tempo de espera da pessoa e inicia a experiência se necessário.

Método sair_da_ixfera:

- Permite que uma pessoa saia da atração, atualizando o número de pessoas presentes e pausando a experiência se todas as pessoas saíram.

Método entrar_na_fila:

- Adiciona uma pessoa à fila, aguardando até ser liberada para entrar na atração.

Método finalizar_simulacao:

- Imprime uma mensagem indicando o fim da simulação.

Código da Pessoa:

A classe Pessoa simula o comportamento de cada visitante da atração. Cada pessoa é uma thread independente, e seus principais métodos são:

- **__init__**: Inicializa os atributos da pessoa, como número, faixa etária, e parâmetros da atração.
- **run**: Implementa o comportamento da thread, simulando a chegada, entrada na fila, entrada na atração, tempo de permanência, e saída da atração.

Código Principal (main):

O código principal coordena a execução da simulação. Ele lê os parâmetros da linha de comando, cria uma instância da classe Ixfera, gera pessoas com faixas etárias aleatórias, inicia suas threads, aguarda a conclusão da simulação, calcula estatísticas e imprime o relatório final.

Dificuldades:

Na elaboração do projeto de simulação da atração IxferaTM, diversos desafios e considerações foram enfrentados para garantir a implementação e execução eficaz do programa. Abaixo, destacamos alguns dos desafios enfrentados durante o desenvolvimento do projeto:

1. **Entrada na atração:** Houve testes que quando iniciavam, as pessoas da mesma faixa etária, conseguiam entrar na mesma atração, porém ao decorrer da execução, se tivessem duas pessoas da mesma faixa etária seguidas na fila, elas não conseguiam entrar junto, fazendo que uma entrasse e a outra esperasse a atração terminar, sendo que eram da mesma faixa etária.
2. **Sincronização e Controle de Acesso:** A gestão simultânea de múltiplas threads representando os visitantes e a atração exigiu uma cuidadosa implementação de mecanismos de sincronização, como locks e semáforos. Esses elementos foram cruciais para evitar conflitos e garantir a consistência dos dados compartilhados entre as threads.
3. **Modelagem da Lógica da Atração:** A tradução das regras específicas da IxferaTM para a lógica do programa demandou uma análise detalhada para assegurar que cada aspecto da atração, desde o controle de filas até o início e pausa das experiências, fosse corretamente modelado no código.
4. **Gerenciamento de Tempo:** A integração adequada do tempo na simulação foi um ponto crítico. Foi necessário garantir que o tempo de permanência das pessoas na atração e o tempo total de simulação fossem precisamente controlados, evitando divergências entre o modelo virtual e a realidade simulada.

Referências:

<https://python.readthedocs.io/en/v2.7.2/library/threading.html>