

Proyecto 2



Baño Unisex

Definición del Problema

Se quiere transformar un baño de hombres de una empresa a un baño unisex por comodidad de las mujeres que trabajan en el lugar. Para ello se deben cumplir ciertas restricciones:

- No pueden haber hombres y mujeres en el baño a la vez.
- No pueden haber más de 3 empleados a la misma vez utilizando el baño.

Algoritmos utilizados

Algoritmo Con semáforos (guiado por el libro)

```
sem_wait(&empty);
    sem_wait(&womanSemaphore);
    m_count++;
    if(m_count == 1){
        sem_wait(&menSemaphore);
    }
    printf("Mujer entra al baño\n");
```

```
sem_post(&full);
    sem_post(&womanSemaphore);
    bathroom();
    sem_wait(&womanSemaphore);
    sem_post(&empty);
    printf("Mujer sale del baño\n");
    m_count--;
    if (m_count == 0){
        sem_post(&menSemaphore);
    }
    sem_post(&womanSemaphore);
    sem_wait(&full);
    pthread_exit(NULL);
```

Solución Alternativa

Algoritmo con Mutex

```
static void* mujer(void *datos) {
    bano_u *bano = (bano_u *) datos;
    pthread_mutex_lock(&bano->mutex);
    while (bano->hombres > 0 || bano->mujeres > 2) {
        printf("Mujer esperando....\n");
        pthread_cond_wait(&bano->lleno, &bano->mutex);
    }
}
```

- Nos resultó menos enredado y fácil de manipular en este tipo de problemas.

```
printf("Mujer entra al baño\n");
    bano->mujeres++;
    bano->total_mujeres++;
    pthread_mutex_unlock(&bano->mutex);
    usar_bano(bano, 1000);
    pthread_mutex_lock(&bano->mutex);
    bano->mujeres--;
    printf("Mujer salio del baño\n");

    pthread_cond_broadcast(&bano->lleno);
    pthread_mutex_unlock(&bano->mutex);
    return NULL;
}
```



Problemas Encontrados.



Gracias.