```
#define NONE 3
int mesa[2] = {NONE, NONE};
#define TABACO 0
#define CERILLAS 1
#define PAPEL 2
```

```
mutex_t m;
cond_t llena, vacia;
```

```
void* fumador(void* arg)
{
    int ing1, ing2;
    int ing0 = (int) arg;
    ing1 = (ing0 + 1) % 3;
    ing2 = (ing0 + 2) % 3;
    while (1) {
        lock(&m);
        while ((mesa[0] != ing1 || mesa[0] != ing2) &&
               (mesa[1] != ing1 || mesa[1] != ing2))
            cond_wait(&llena, &m);

        mesa[0] = NONE;
        mesa[1] = NONE;
        cond_signal(&vacia);
        unlock(&m);
        fumar();
    }
}
```

Sec crítica

← Concurrentemente

```
void* agente(void* arg)
{
    int ing1, ing2;

    while (1) {
        ing1 = rand() % 3;
        do {
            ing2 = rand() % 3;
        } while (ing2 == ing1);
        lock(&m)
        while (mesa[0] != NONE || mesa[1] != NONE)
            cond_wait(&vacia, &m);

        mesa[0] = ing1;
        mesa[1] = ing2;
        cond_broadcast(&lleno);
        unlock(&m);
    }
}
```

→ concurrentemente