

# PRIMERA PRÁCTICA EVALUABLE: CRUCE

## SEMÁFOROS

Usamos un total de nueve semáforos, el primero de ellos `d.semaforos[0]` es necesario para que funcione la propia biblioteca.

El segundo semáforo, `d.semaforos[1]`, lo inicializamos a 1. Este semáforo controla el semáforo C1 para evitar que los coches que aparecen por la carretera vertical no puedan cruzar la línea de detención cuando C1 está en rojo o amarillo.

Operaciones del semáforo:

```
ciclosem{  
    W(d.semaforos[1])  
    pon color amarillo  
    pon color rojo  
    pon color verde  
    S(d.semaforos[1])  
}
```

```
esCoche{  
    si pos.x==33 Y pos.y=6 entonces  
        W(d.semaforos[1])  
    avanza el coche  
    si pos.y<0 entonces  
        S(d.semaforos[1])  
}
```

El tercer semáforo, `d.semaforos[2]`, lo inicializamos a 0. Este semáforo controla el semáforo C2 para evitar que los coches que aparecen por la carretera horizontal no puedan cruzar la línea de detención cuando C2 está en rojo o amarillo.

Operaciones del semáforo:

```
ciclosem{  
    S(d.semaforos[2])  
    pon color verde  
    W(d.semaforos[2])  
    pon color amarillo  
    pon color rojo  
}
```

```
esCoche{  
    si pos.x==13 Y pos.y=10 entonces  
        W(d.semaforos[2])  
    avanza el coche  
    si pos.y<0 entonces  
        S(d.semaforos[2])  
}
```

El cuarto semáforo, `d.semaforos[3]`, lo inicializamos a 0. Este semáforo controla el semáforo P1 para evitar que los peatones no puedan cruzar por el paso de cebra horizontal cuando P1 está en rojo.

Operaciones del semáforo:

```
ciclosem{  
    pon color verde  
    S(d.semaforos[3])  
    W(d.semaforos[3])  
    pon color rojo  
}
```

```
esPeaton{  
    si pos.x==30 Y pos.y<=15 Y pos.y>=13 entonces  
        W(d.semaforos[3])  
    avanza el peatón  
    si sale del paso de cebra entonces  
        S(d.semaforos[3])  
}
```

Roberto Merchán González i0909939

Sergio Sánchez García i0961594

El quinto semáforo, d.semaforos[4], lo inicializamos a 1. Este semáforo controla el semáforo P2 para evitar que los peatones no puedan cruzar por el paso de cebra vertical cuando P2 está en rojo.

Operaciones del semáforo:

```
ciclosem{  
    W(d.semaforos[4])  
    pon color rojo  
    pon colo verde  
    S(d.semaforos[4])  
}
```

```
esPeaton{  
    si pos.y==11 Y pos.x<=28 Y pos.x>=22 entonces  
        W(d.semaforos[4])  
    avanza el peatón  
    si sale del paso de cebra entonces  
        S(d.semaforos[4])  
}
```

El sexto semáforo, d.semaforos[5], lo inicializamos al número de procesos máximos introducidos por la línea de ordenes menos dos debido al proceso padre y al proceso encargado de controlar el ciclo semafórico. Este es el encargado de controlar el número de procesos existentes en el mapa.

Operaciones del semáforo:

```
esCoche{  
    W(d.semaforos[5])  
    fork()  
    si pos.y<0 entonces  
        S(d.semaforos[5])  
}
```

```
esPeaton{  
    W(d.semaforos[5])  
    fork()  
    si pos.y<0 entonces  
        S(d.semaforos[5])  
}
```

El séptimo semáforo, d.semaforos[6], lo inicializamos a 0. Este semáforo se encarga de controlar que el cruce de los coches sea una sección crítica para que solo pueda haber un coche en esta zona. También controla el tiempo que debe durar el color amarillo en los semáforos de los coches.

Operaciones del semáforo:

```
ciclosem{  
    pon color amarillo C1  
    W0(d.semaforos[6])  
    pon color rojo C1  
    pon color verde C2  
    pon color amarillo C2  
    W0(d.semaforos[6])  
    pon color rojo C2  
    pon color verde C1  
}
```

```
esCoche{  
    si pos.x==33 Y pos.y=6 entonces  
        S(d.semaforos[6])  
    si pos.x==13 Y pos.y=10 entonces  
        S(d.semaforos[6])  
    avanza el coche  
    si pos.y<0 entonces  
        W(d.semaforos[6])  
}
```

Roberto Merchán González i0909939

Sergio Sánchez García i0961594

El octavo semáforo, d.semaforos[7], lo inicializamos a 1. Este se encarga de que no pueda nacer un nuevo peatón mientras otro este en la zona crítica en la que pueden nacer.

Operaciones del semáforo:

```
esPeaton{
    W(d.semaforos[7])
    avanza el peatón
    si NO esta en zona crítica entonces
        S(d.semaforos[7])
}
```

Zona crítica: (pos\_ac.x==0 && pos\_ac.y>10) || (pos\_ac.x<40 && pos\_ac.y==16)

El noveno semáforo, d.semaforos[8], lo inicializamos a 0. Se encarga de que los coches no puedan avanzar hasta que el paso de cebra este libre.

Operaciones del semáforo:

```
esCoche{
    si pos.x==33 Y pos.y=6 entonces
        W0(d.semaforos[8])
    si pos.x==13 Y pos.y=10 entonces
        W0(d.semaforos[8])
}
```

```
esPeaton{
    si pos.y==11 Y pos.x<=28 Y pos.x>=22 entonces
        entraCruce2 <- VERDADERO
        S(d.semaforos[8])
    si pos.x==30 Y pos.y<=15 Y pos.y>=13 entonces
        S(d.semaforos[8])
        EntraCruce1 <- VERDADERO
    avanza el peatón
    si NO esta en paso de peatón horizontal Y entraCruce2
        S(d.semaforos[8])
    si NO esta en paso de peatón vertical Y entraCruce1
        S(d.semaforos[8])
}
```

Roberto Merchán González i0909939

Sergio Sánchez García i0961594

## MENSAJES

### COCHES

Al comienzo del programa se mandan los mensajes necesarios para cubrir la parte del mapa correspondiente a las carreteras.

Operaciones de mensajes:

```
esCoche{
    si nacimientoV Y NO entraCruce entonces
        receive(100+pos.y+6)
    si NO nacimientoV Y NO entraCruce entonces
        receive(200+pos.x+8)
    avanza el coche
    si nacimientoV Y pos.y > 7 && pos.y <= 12 entonces
        send(100+pos.y-1)
    si NO nacimientoV Y pos.x > 5 && pos.x <= 21 entonces
        send(200+pos.x-2)
}
```

### PEATONES

Al comienzo del programa se mandan los mensajes necesarios para cubrir todo el mapa.

Operaciones de mensajes:

```
esPeaton{
    receive(90000+pos.x*100+pos.y)
    pos_ant <- pos_ac
    pos_ac <- pos
    avanza el peatón
    send(90000+pos_ant.x*100+pos_ant.y)
    si pos.y<0 entonces
        send(90000+pos_ac.x*100+pos_ac.y)
}
```

Cada vez, antes de avanzar el peatón, se almacena en la posición anterior la posición actual y en la posición actual la posición a la que va a avanzar. A la función encargada de que avance el peatón se le pasa la posición a la que debe avanzar y en esta misma variable se almacena la posición siguiente.