

# Explicación Semáforos

Ejercicios

# EJERCICIO 1

---

Se tiene  $C$  chicos y hay una bolsa con caramelos que nunca se vacía. Cada chico saca caramelos de a uno, lo come y lo cuenta en una variable compartida para saber cuantos caramelos se han tomado de la bolsa.

**Var**

```
sem sum = 1  
int cant = 0
```

**Proceso Chico[i:1..C]**

While (true)

```
{ -- tomar caramelo  
  -- comer caramelo  
  P(sum)  
  cant = cant + 1  
  V(sum)  
}
```

**End;**

Debo proteger el  
tomar el caramelo?

**Var**

```
sem sum = 1  
int cant = 0
```

**Proceso Chico[i:1..C]**

While (true)

```
{ P(sum)  
  cant = cant + 1  
  -- tomar caramelo  
  -- comer caramelo  
  V(sum)  
}
```

**End;**

Maximiza la  
conurrencia?



# EJERCICIO 1

---

Se tiene  $C$  chicos y hay una bolsa con caramelos que nunca se vacía. Cada chico saca caramelos de a uno, lo come y lo cuenta en una variable compartida para saber cuantos caramelos se han tomado de la bolsa.

**Var**

sem sum = 1

int cant = 0

**Proceso Chico[i:1..C]**

While (true)

{ P(sum)

*-- tomar caramelo*

  cant = cant + 1

  V(sum)

*-- comer caramelo*

}

**End;**



# EJERCICIO 2

Se tiene  $C$  chicos y hay una bolsa de caramelos con  $N$  caramelos. Cada chico saca caramelos de a uno, lo come y lo cuenta en una variable compartida para saber cuantos caramelos se han tomado de la bolsa.

**Var**

sem sum = 1

int cant = 0

**Proceso Chico[i:1..C]**

P(sum)

While (cant < N)

{ -- tomar caramelo

V(Sum)

-- comer caramelo

P(sum)

cant = cant + 1

V(Sum)

P(sum)

}

V(sum)

**End;**

**Var**

sem sum = 1

int cant = 0

**Proceso Chico[i:1..C]**

While (cant < N)

{ P(sum)

-- tomar caramelo

cant = cant + 1

V(Sum)

-- comer caramelo

}

**End;**

**Var**

sem sum = 1

int cant = 0

**Proceso Chico[i:1..C]**

P(sum)

While (cant < N)

{ -- tomar caramelo

cant = cant + 1

V(Sum)

-- comer caramelo

P(sum)

}

V(sum)

**End;**

# EJERCICIO 3

Modificar el ejercicio anterior. Hasta que no han llegado todos los chicos no puede comenzar a comer. Hay un coordinador que (una vez que todos llegaron), repetidamente selecciona un chico aleatoriamente y lo deja pasar a tomar un caramelo.

```
sem ganador[1..C] = 0;  
sem inicio = 0;  
sem barrera = 1;  
sem listo = 0  
int cant = 0, llegaron = 0;  
bool fin = false
```

## Proceso Chico[id:1..C]

```
{ P(barrera)  
  llegaron++;  
  if (llegaron == C) V(inicio)  
  V(barrera)  
  P(ganador[id])  
  While (not fin)  
  { -- tomar caramelo  
    V(listo)  
    -- comer caramelo  
    P(ganador[id])  
  }  
}
```

**End;**

## Coordinador::

```
{ int i, num, cant = 0  
  
  P(inicio)  
  For i = 1..N  
    num = (rand() mod C) + 1;  
    V(ganador[num])  
    P(listo)  
  fin = true  
  For i = 1..C  
    V(ganador[num])  
}
```