

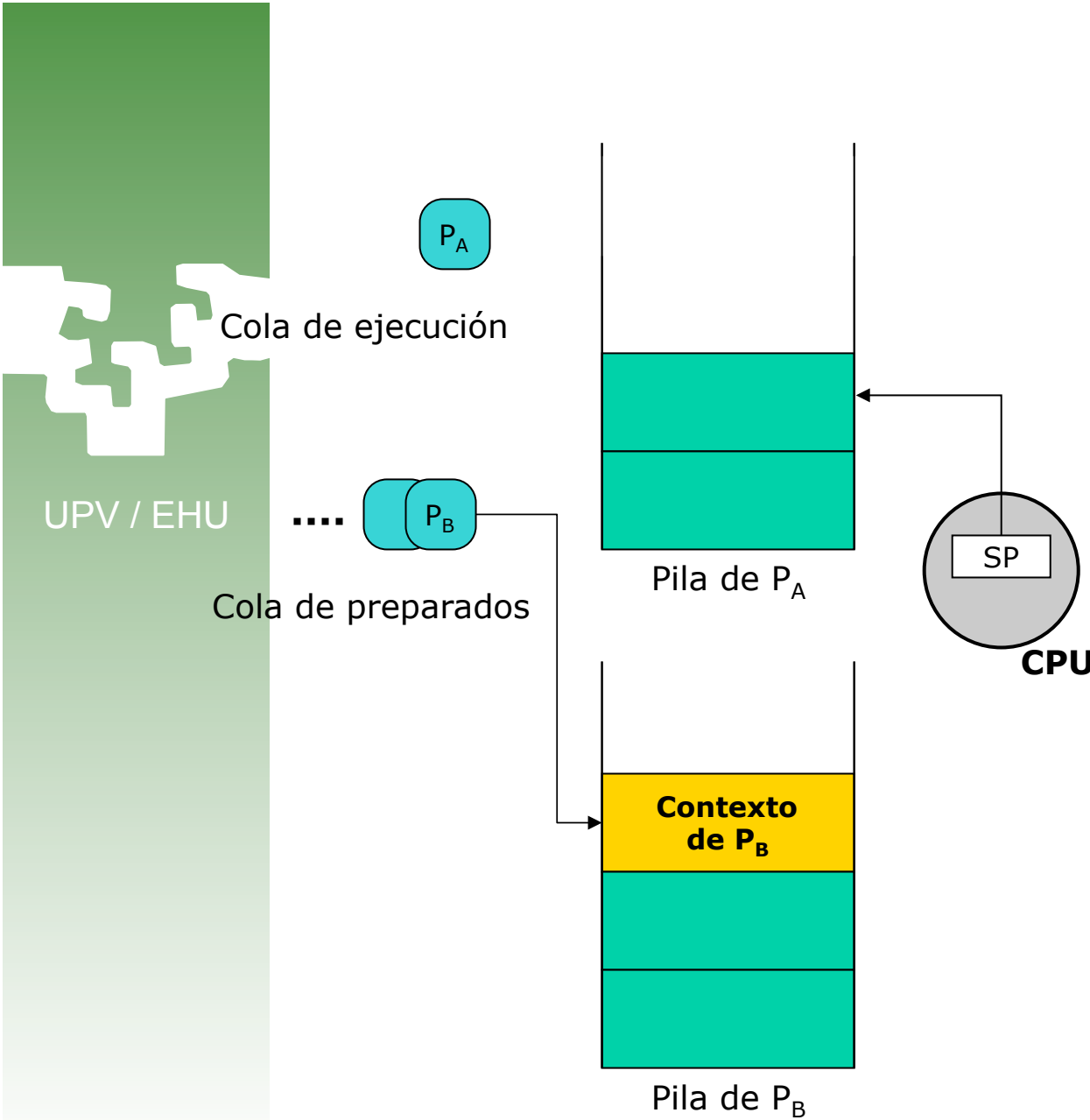
# Programación Concurrente en Linux

## Cambio de contexto

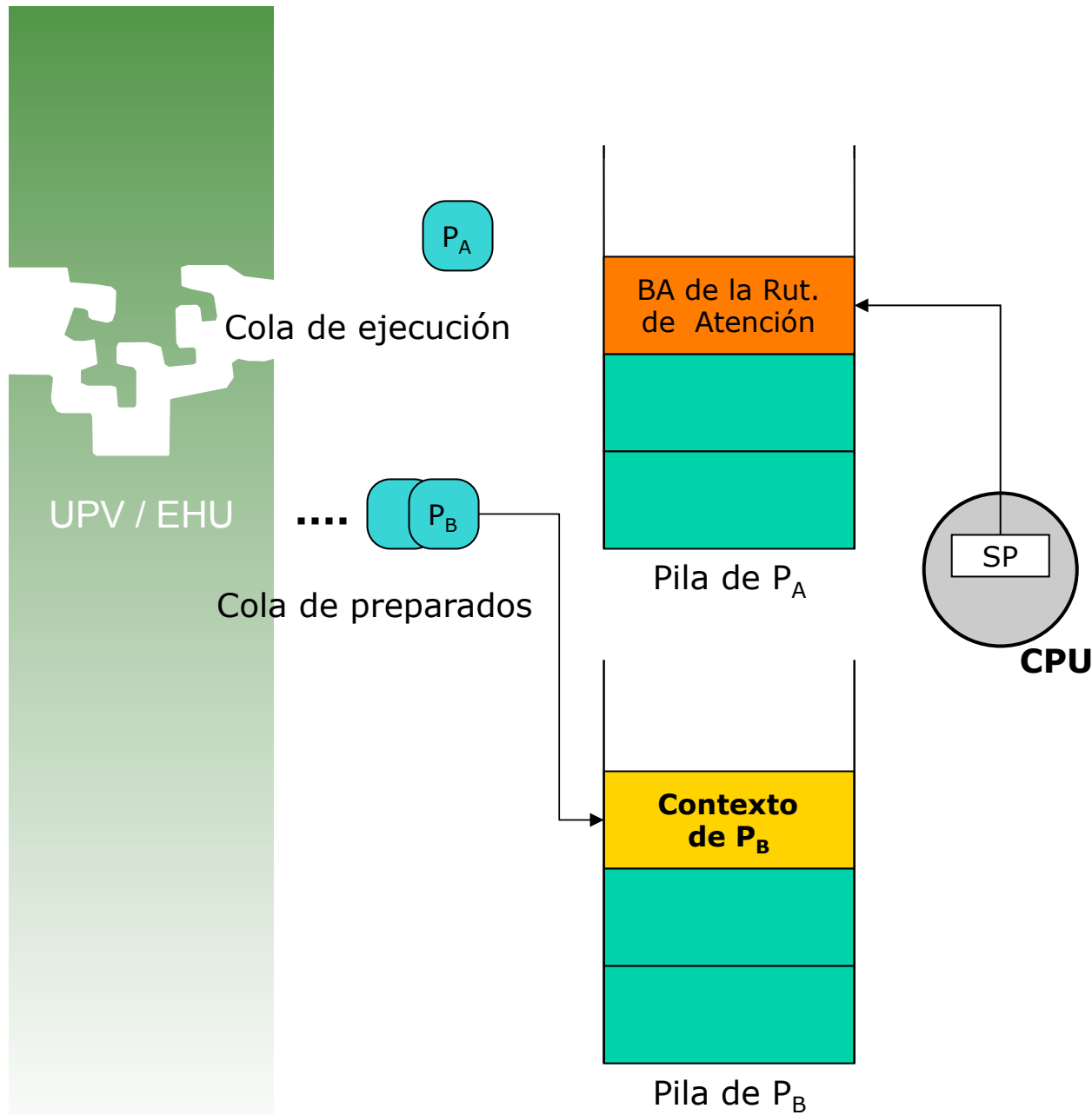


UPV / EHU

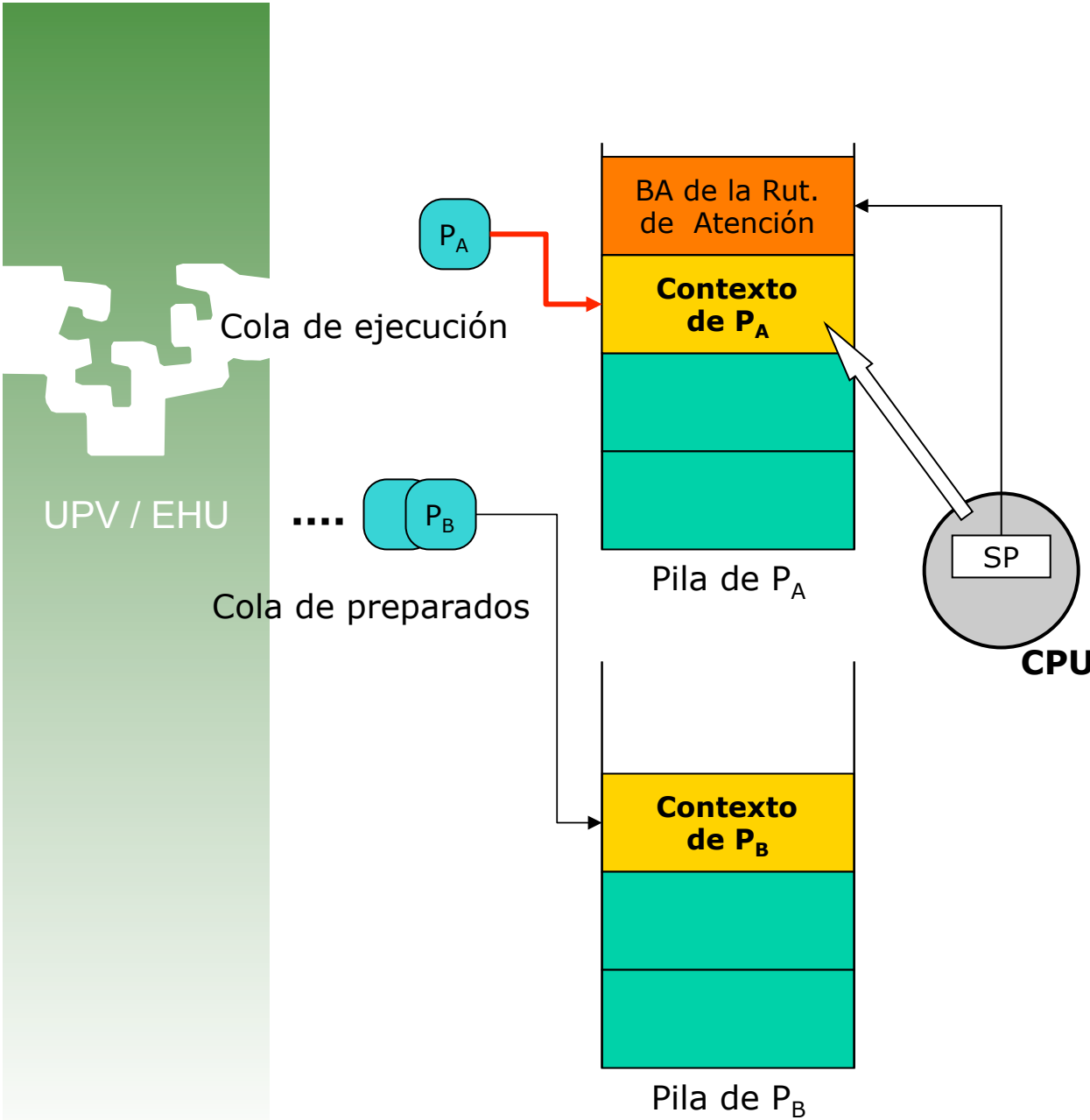
Esta animación describe el cambio en la ejecución de un proceso  $P_A$  a otro  $P_B$  en un sistema multiprogramado, en particular el cambio de contexto y de estado de los procesos.



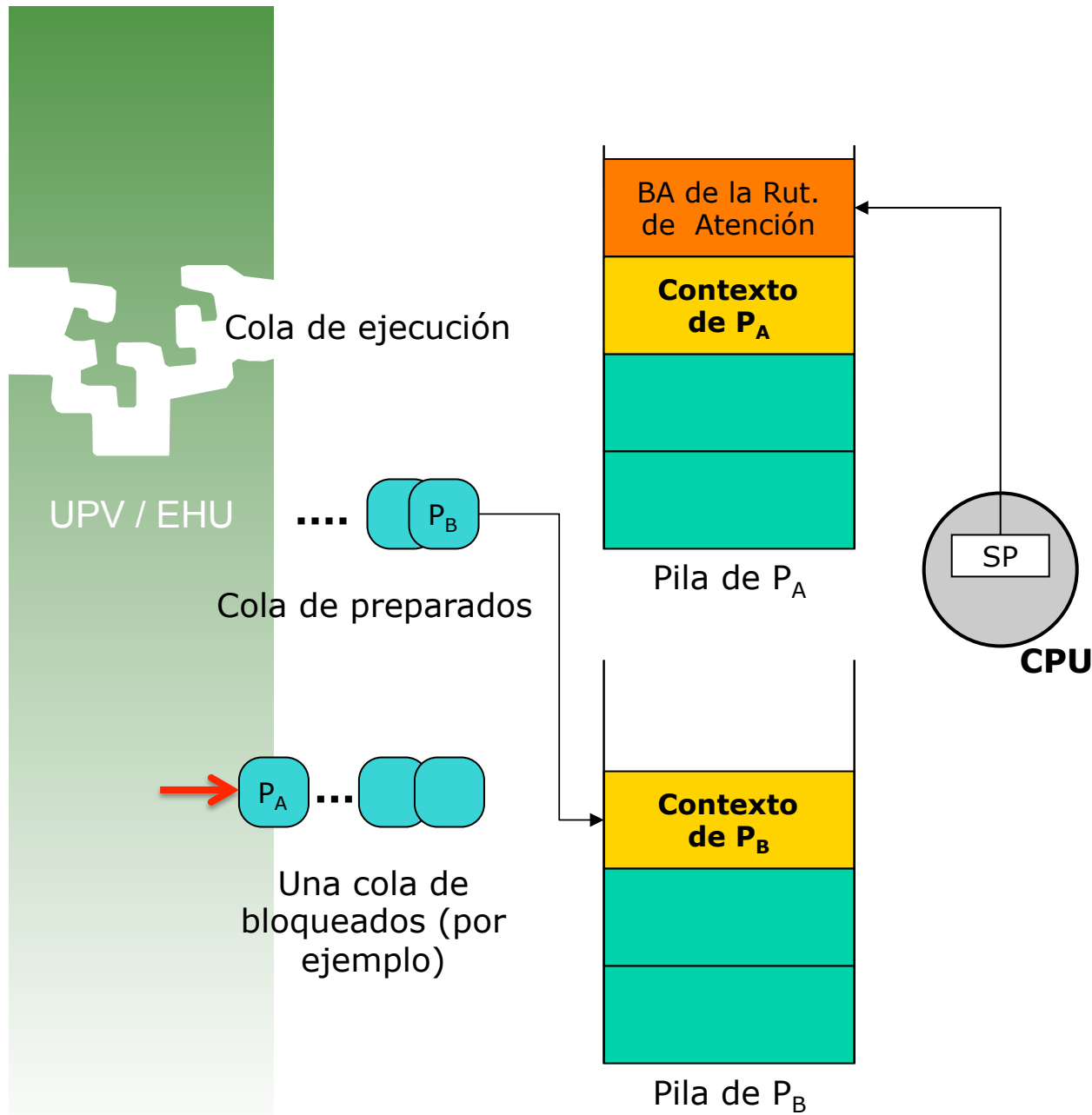
1. Se está ejecutando  $P_A$ .



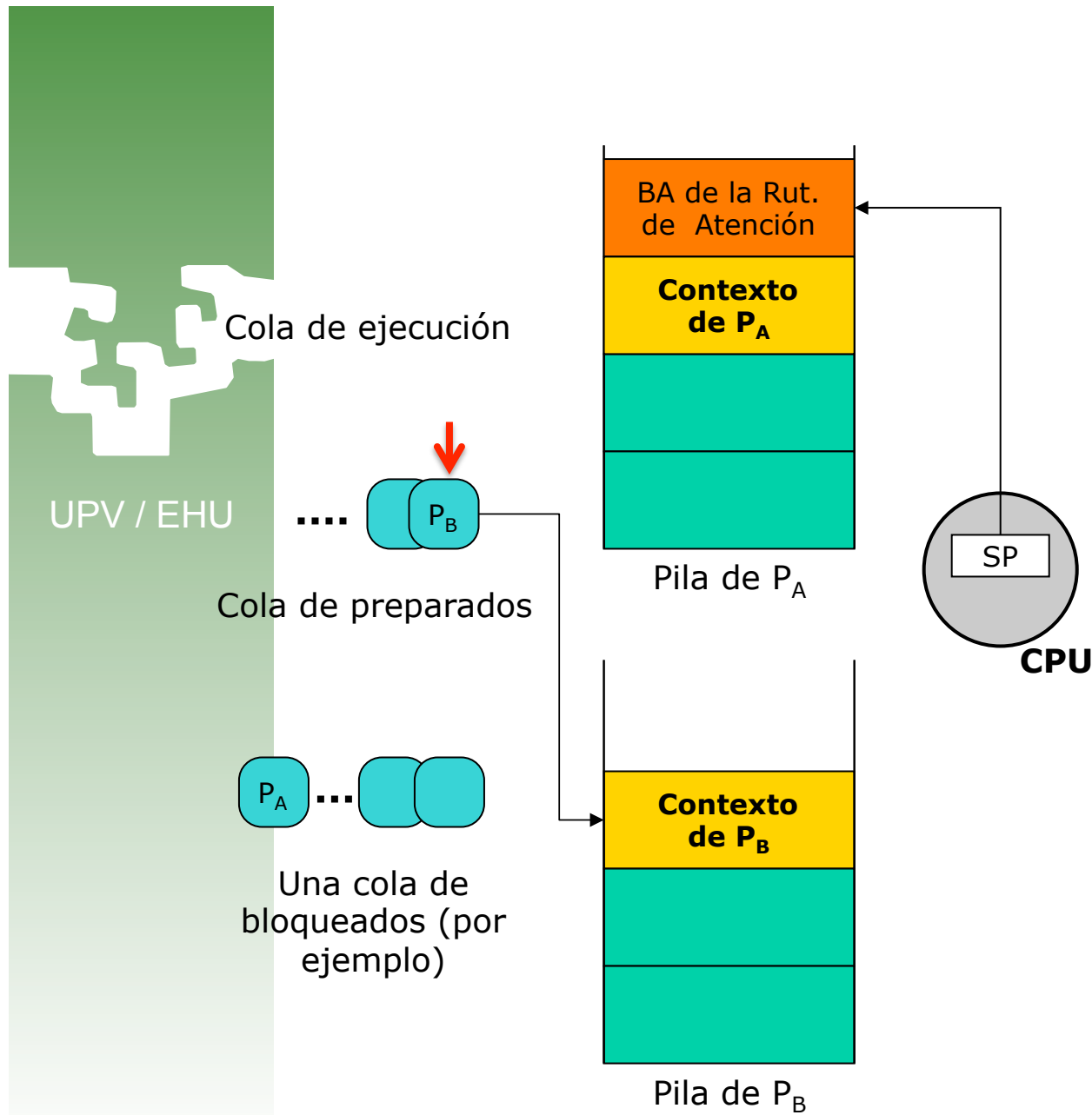
1. Se está ejecutando  $P_A$ .
2. Se produce una interrupción (externa o trap).



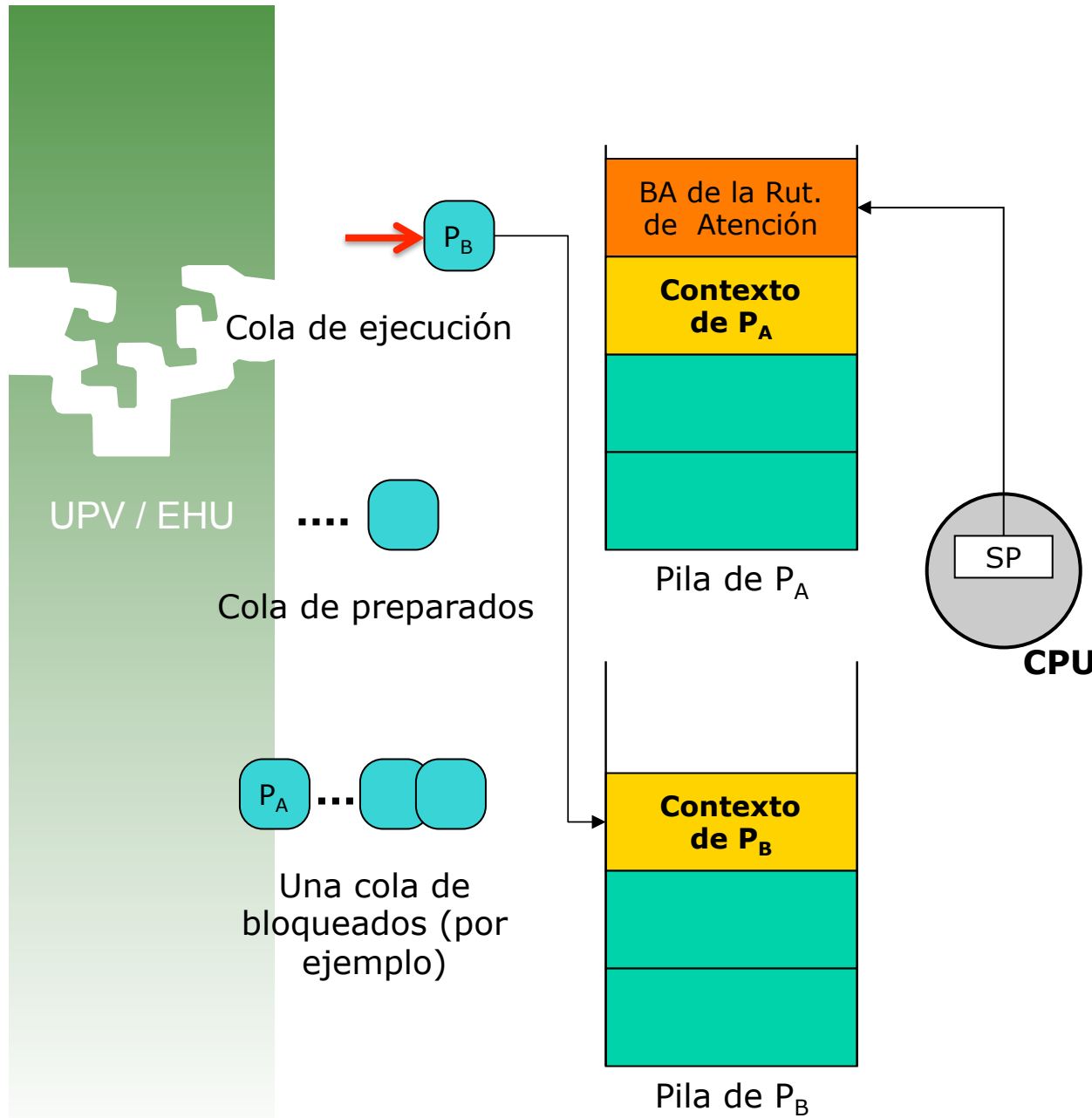
1. Se está ejecutando  $P_A$ .
2. Se produce una interrupción (externa o trap).
3. Se guarda el contexto de  $P_A$



1. Se está ejecutando  $P_A$ .
2. Se produce una interrupción (externa o trap).
3. Se guarda el contexto de  $P_A$ .
4. Cambio de estado de  $P_A$ .

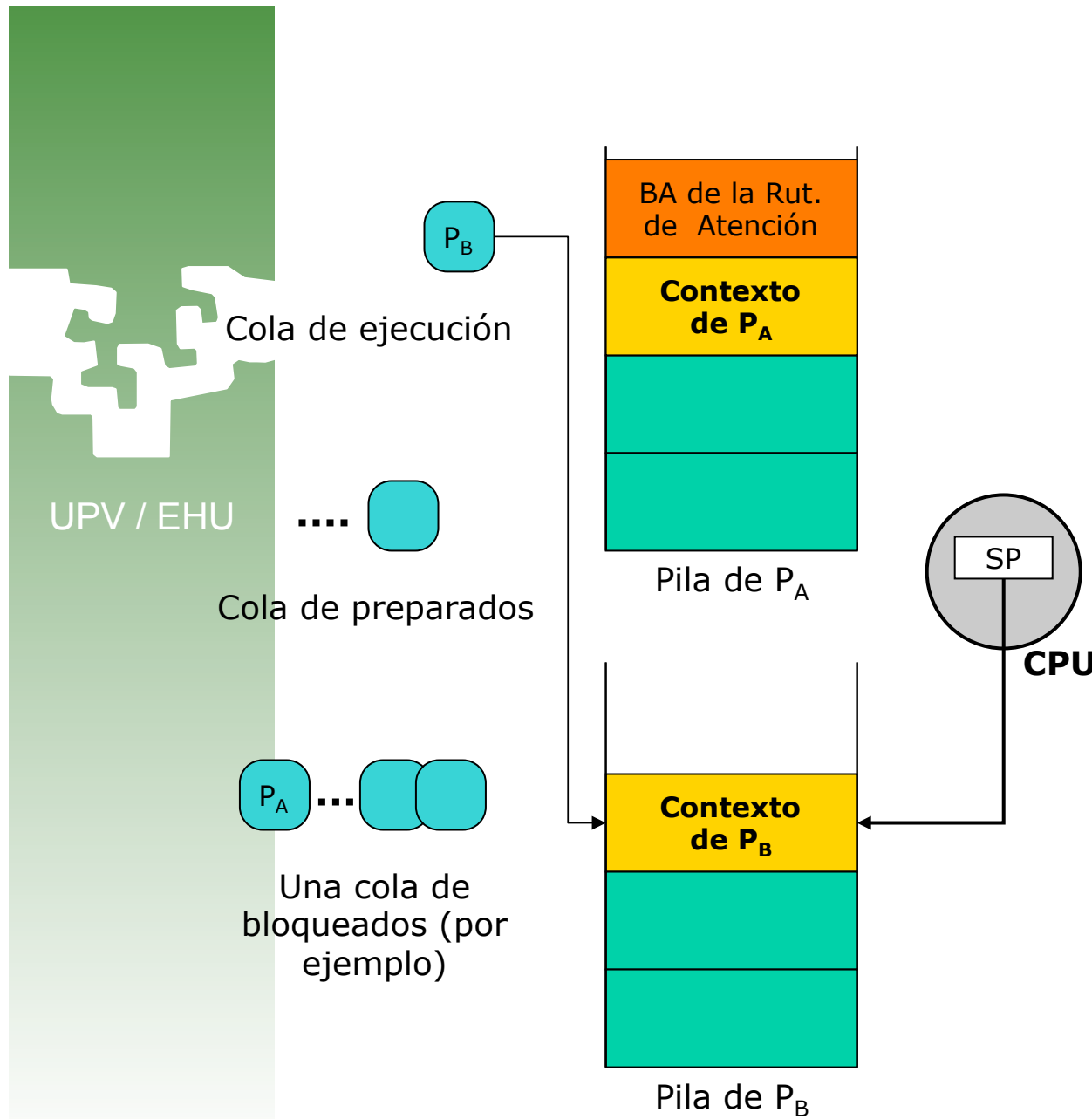


1. Se está ejecutando  $P_A$ .
2. Se produce una interrupción (externa o trap).
3. Se guarda el contexto de  $P_A$ .
4. Cambio de estado de  $P_A$ .
5. Scheduler: elige a  $P_B$ .

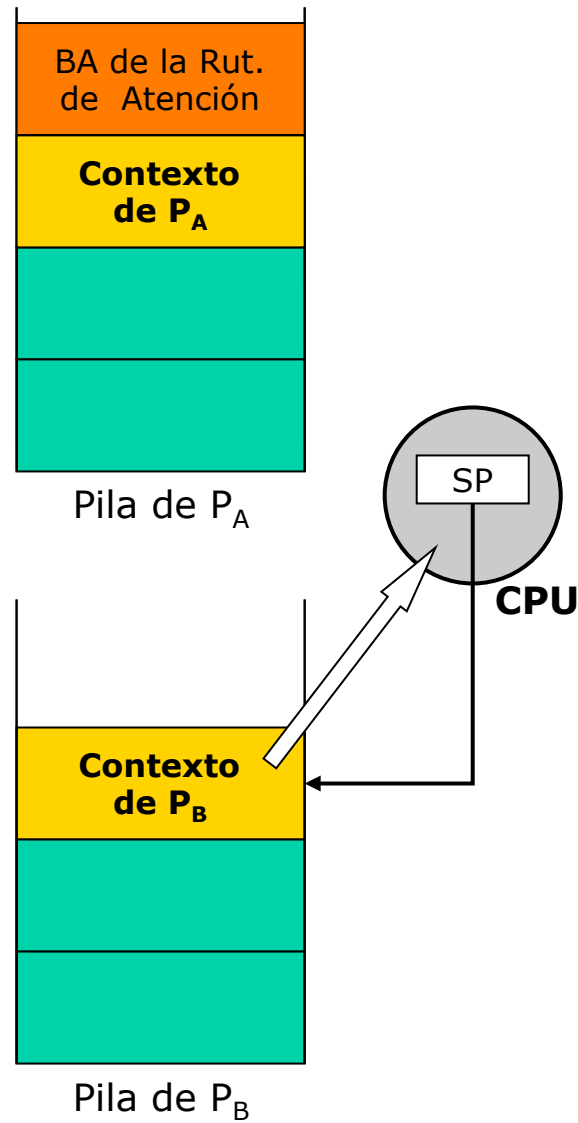
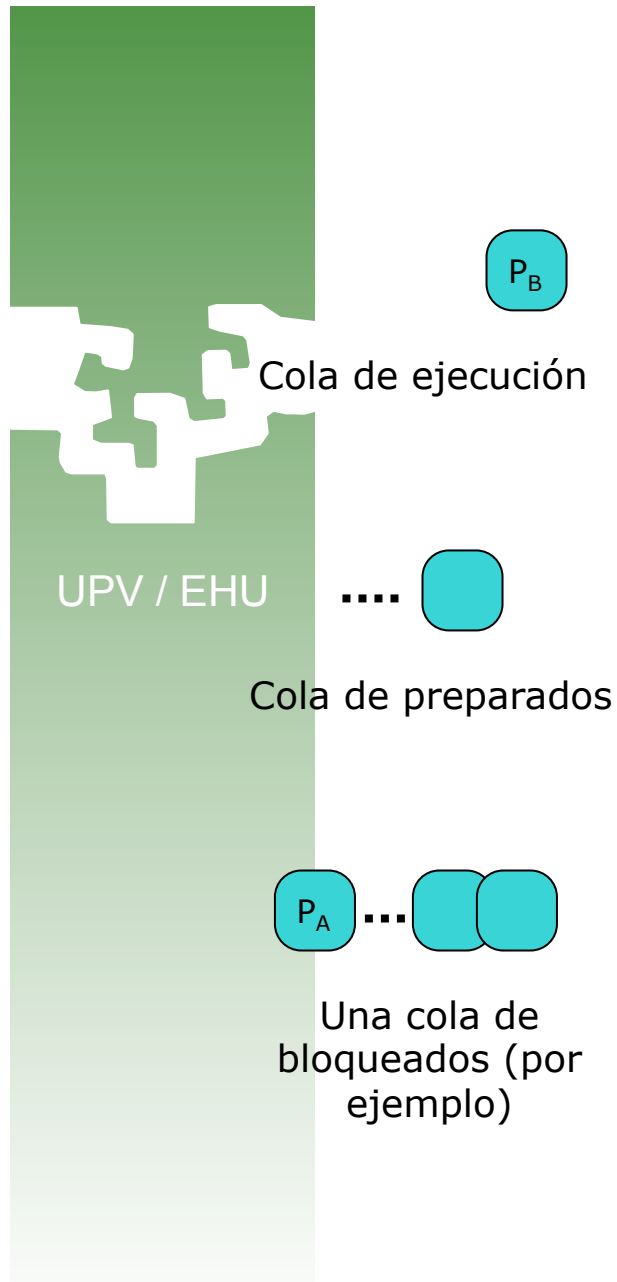


1. Se está ejecutando  $P_A$ .
2. Se produce una interrupción (externa o trap).
3. Se guarda el contexto de  $P_A$ .
4. Cambio de estado de  $P_A$ .
5. Scheduler: elige a  $P_B$ .
6. Cambio de estado de  $P_B$ .

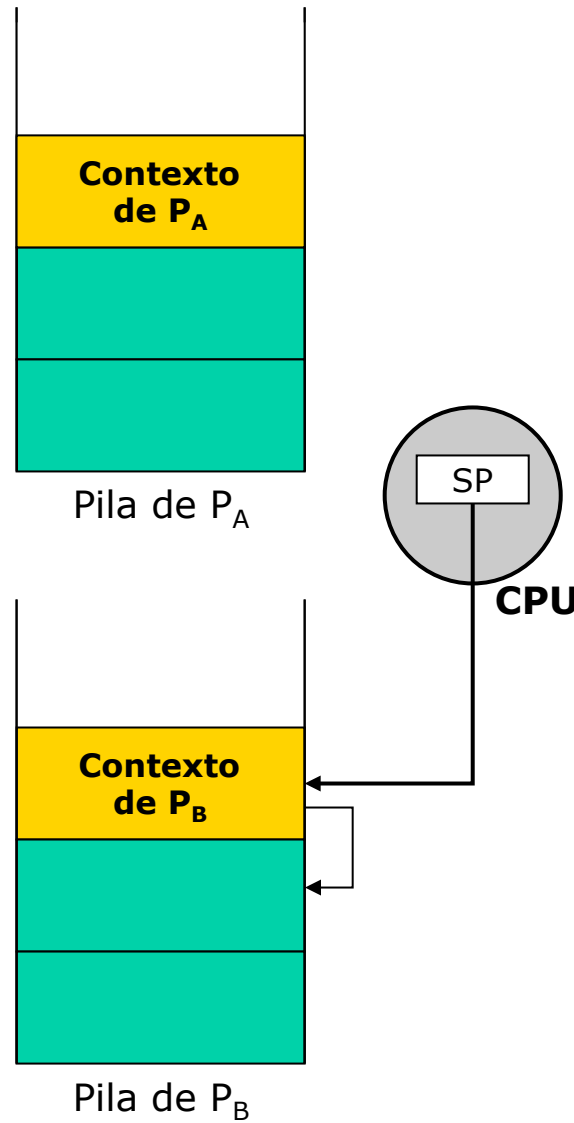
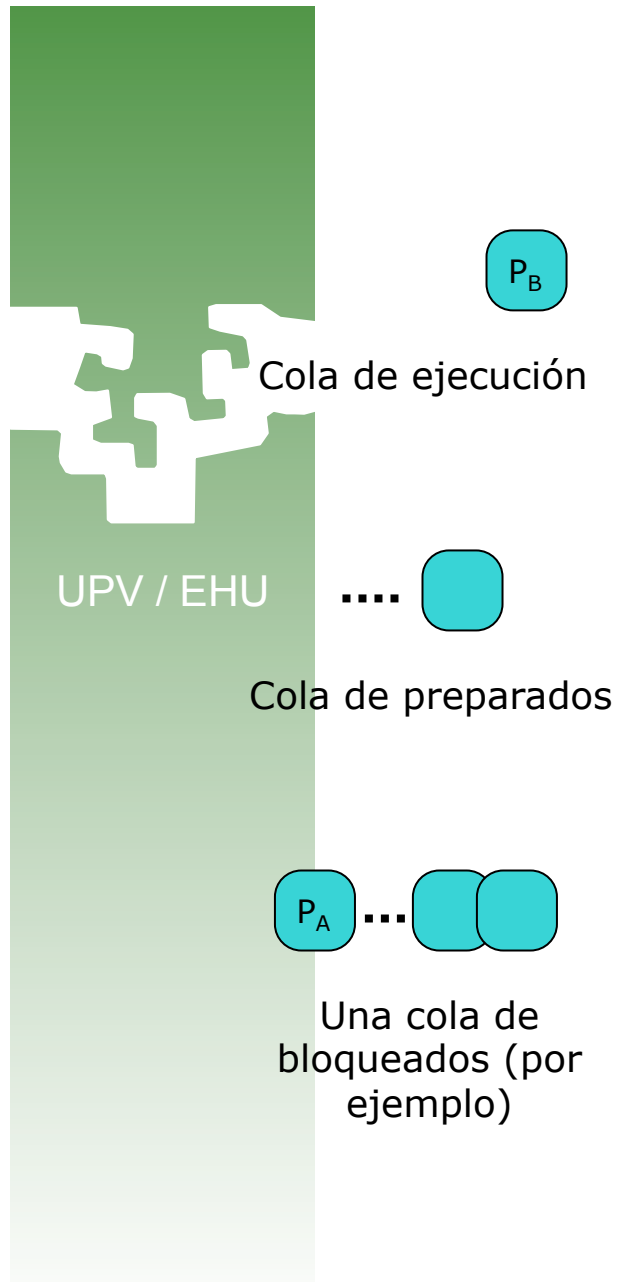




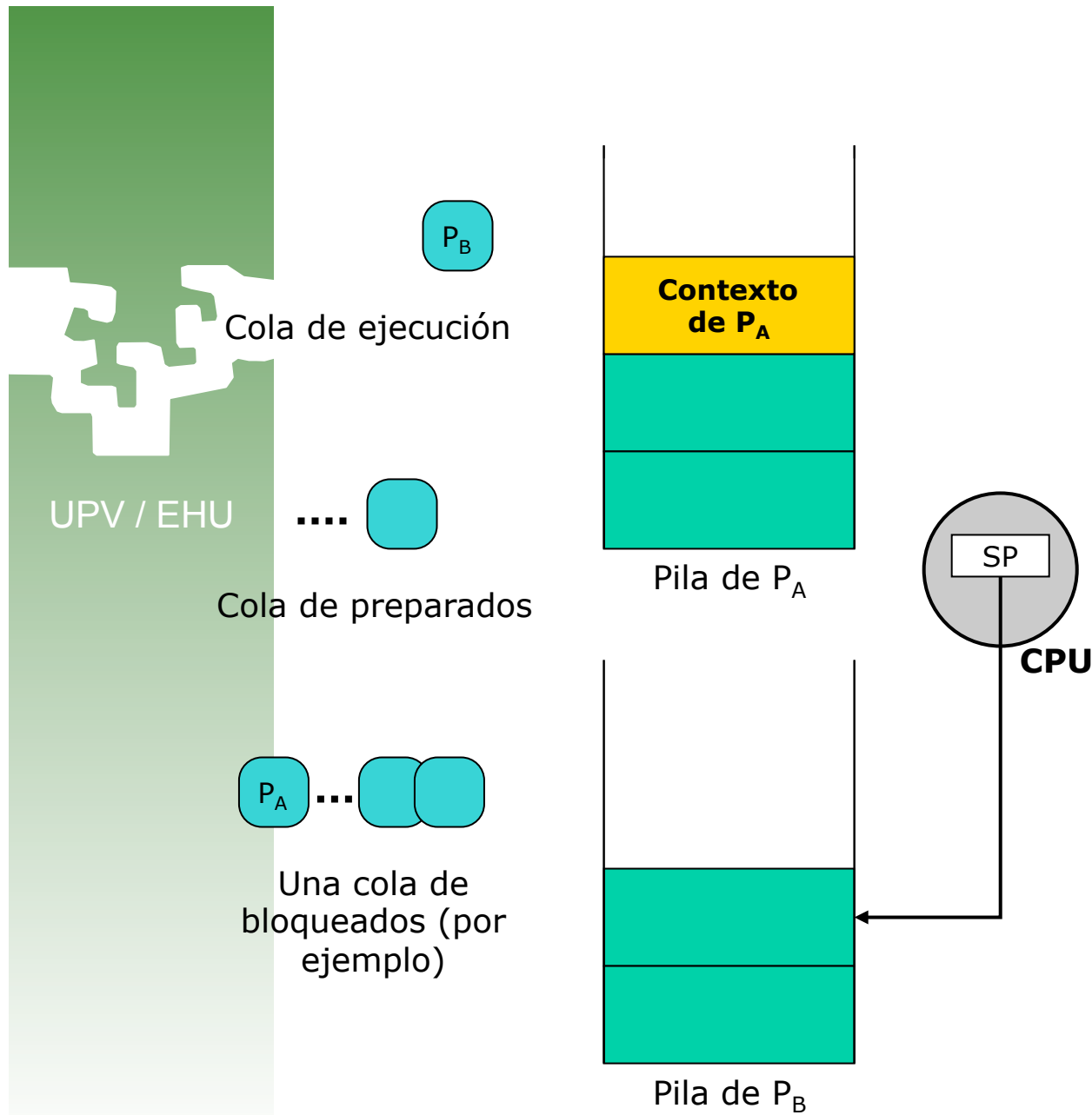
1. Se está ejecutando  $P_A$ .
2. Se produce una interrupción (externa o trap).
3. Se guarda el contexto de  $P_A$ .
4. Cambio de estado de  $P_A$ .
5. Scheduler: elige a  $P_B$ .
6. Cambio de estado de  $P_B$ .
7. Manipulación del Stack Pointer para apuntar al bloque de activación de  $P_B$ .



1. Se está ejecutando  $P_A$ .
2. Se produce una interrupción (externa o trap).
3. Se guarda el contexto de  $P_A$ .
4. Cambio de estado de  $P_A$ .
5. Scheduler: elige a  $P_B$ .
6. Cambio de estado de  $P_B$ .
7. Manipulación del Stack Pointer para retornar al bloque de activación de  $P_B$ .
8. Se carga el contexto de  $P_B$  en la CPU.



1. Se está ejecutando  $P_A$ .
2. Se produce una interrupción (externa o trap).
3. Se guarda el contexto de  $P_A$ .
4. Cambio de estado de  $P_A$ .
5. Scheduler: elige a  $P_B$ .
6. Cambio de estado de  $P_B$ .
7. Manipulación del Stack Pointer para retornar al bloque de activación de  $P_B$ .
8. Se carga el contexto de  $P_B$  en la CPU.
9. Retorno de la interrupción.



1. Se está ejecutando  $P_A$ .
2. Se produce una interrupción (externa o trap).
3. Se guarda el contexto de  $P_A$ .
4. Cambio de estado de  $P_A$ .
5. Scheduler: elige a  $P_B$ .
6. Cambio de estado de  $P_B$ .
7. Manipulación del Stack Pointer para retornar al bloque de activación de  $P_B$ .
8. Se carga el contexto de  $P_B$  en la CPU.
9. Retorno de la interrupción.
10. Se ejecuta  $P_B$ .