

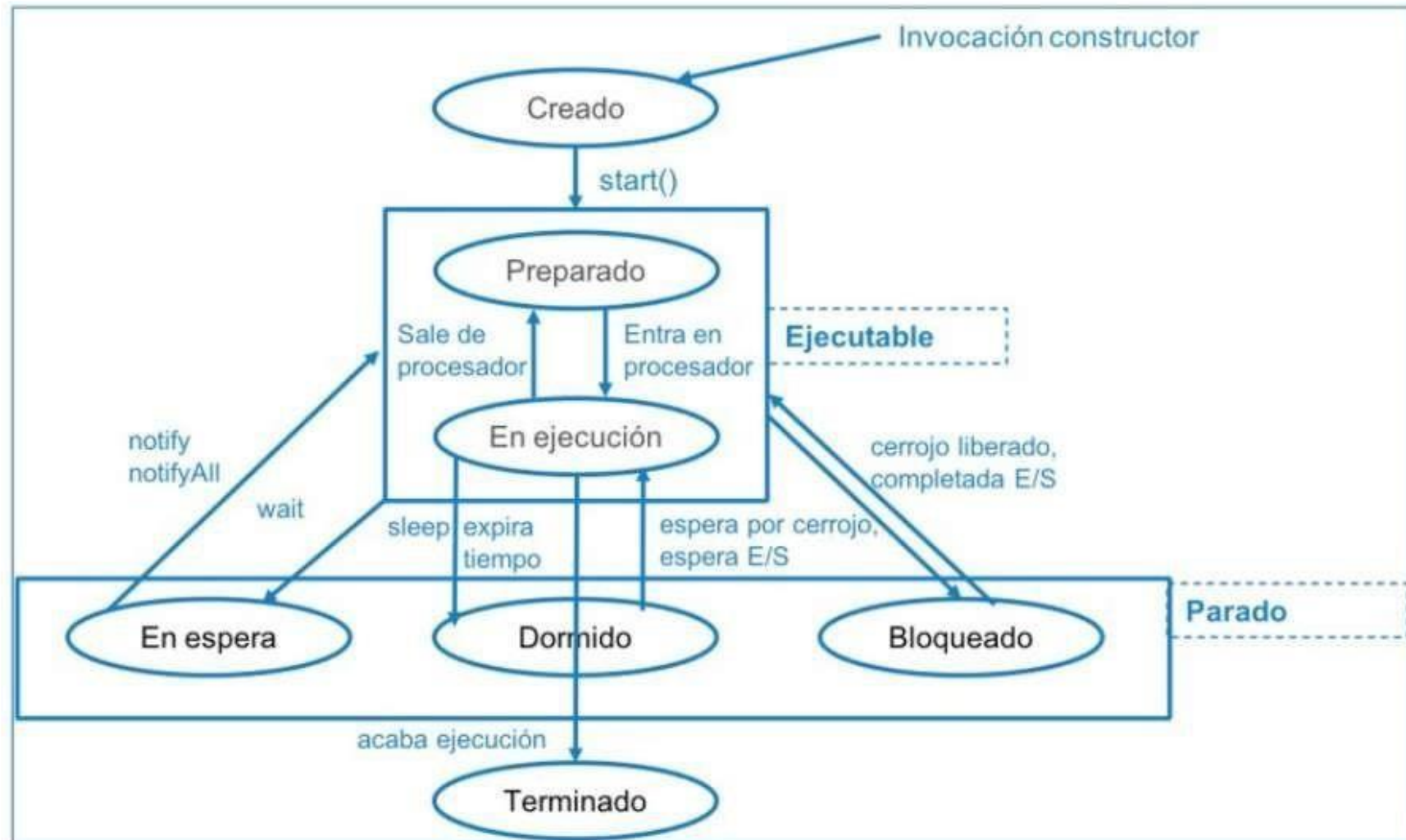
Programación Científica y HPCI

Máster Universitario en Ingeniería Matemática y Computación

Docente: Jesús Cigales

Tema 7

Estados de un hilo en Python



Objetos de Thread	Objetos clases derivadas de Thread
Definición del objetos	Definición de la clase y de objetos
Inicialización con constructor de Thread	Inicialización con constructor de clases derivadas
Asociar una función para su ejecución	Sobrecarga del método run()
Excepciones	Problemas durante la ejecución

Activar hilo start()

```

import threading
import time
import logging

def tareaHilo(tiempo):
    print ("Comienzo de la ejecución del %s \n" % threading.currentThread().name)
    i=5
    while i:
        time.sleep(tiempo)
        print ("Ejecutándose %s \n" % threading.currentThread().name)
        i = i - 1
    print ("%s terminado " % threading.currentThread().name)

if __name__ == "__main__":
    hilol = threading.Thread(name="Hilo Uno", target=tareaHilo, args=(3,))

```

```

import time
import threading

class HiloEjemplo (threading.Thread):
    def __init__(self, id, nombre, tiempo):
        threading.Thread.__init__(self)
        self.id = id
        self.name = nombre
        self.tiempo = tiempo

    def run(self):
        print ("Comienzo de %s \n" % self.name)
        i=5
        while i:
            time.sleep(self.tiempo)
            print ("Ejecutándose %s \n" % self.name)
            i = i - 1
        print ("%s terminado " % self.name)

if __name__ == "__main__":
    hilol = HiloEjemplo(1, "Hilo Uno", 5)

```

```

import threading
import time
import logging

def tareaHilo(tiempo):
    print ("Comienzo de la ejecución del %s \n" %threading.currentThread().name)
    i=5
    while i:
        -time.sleep(tiempo)
        -print ("Ejecutándose %s \n" %threading.currentThread().name)
        -i = i - 1
    print ("%s terminado " %threading.currentThread().name)

if __name__ == "__main__":
    hilo1 = threading.Thread(name="Hilo Uno", target=tareaHilo, args=(3,))
    hilo2 = threading.Thread(name="Hilo Dos", target=tareaHilo, args=(2,))
    hilo3 = threading.Thread(name="Hilo Tres",target=tareaHilo, args=(5,))
    hilo1.start()
    hilo2.start()
    hilo3.start()

    hilo1.join()
    hilo2.join()
    hilo3.join()

```

■ Ejecución:

Comienzo de la ejecución del Hilo Uno

Comienzo de la ejecución del Hilo Dos

Comienzo de la ejecución del Hilo Tres

Ejecutándose Hilo Dos

Ejecutándose Hilo Uno
Ejecutándose Hilo Dos
Ejecutándose Hilo Tres
Ejecutándose Hilo Uno
Ejecutándose Hilo Dos
Ejecutándose Hilo Dos
Ejecutándose Hilo Uno
Ejecutándose Hilo Tres
Ejecutándose Hilo Dos
Hilo Dos terminado
Ejecutándose Hilo Uno
Ejecutándose Hilo Tres
Ejecutándose Hilo Uno
Hilo Uno terminado
Ejecutándose Hilo Tres
Ejecutándose Hilo Tres
Hilo Tres terminado


```

import time
import threading

class HiloEjemplo (threading.Thread):
    def __init__(self, id, nombre, tiempo):
        threading.Thread.__init__(self)
        self.id = id
        self.name = nombre
        self.tiempo = tiempo

    def run(self):
        print ("Comienzo de %s \n" %self.name)
        i=5
        while i:
            time.sleep(self.tiempo)
            print ("Ejecutándose %s \n" %self.name)
            i = i - 1
        print ("%s terminado " %self.name)

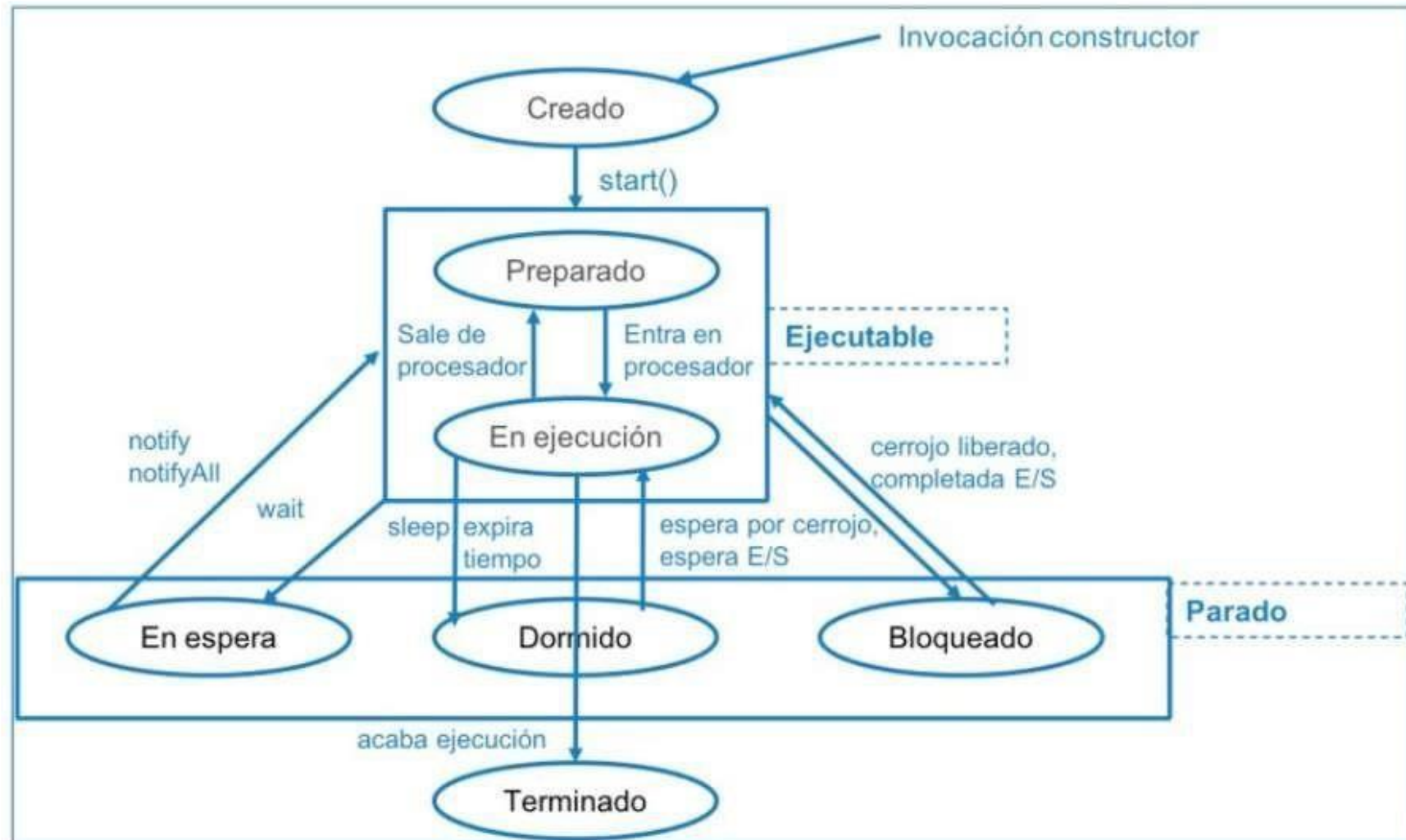
if __name__ == '__main__':
    hilo1 = HiloEjemplo(1, "Hilo Uno", 5)
    hilo2 = HiloEjemplo(2, "Hilo Dos", 2)
    hilo3 = HiloEjemplo(3, "Hilo Tres", 3)

    hilo1.start()
    hilo2.start()
    hilo3.start()

    hilo1.join()
    hilo2.join()
    hilo3.join()

```

Estados de un hilo en Python



Métodos de sincronización

► Conflictos de lectura/escritura

- un hilo consulta el estado de un objeto a la vez que otro intenta modificarlo.
- No en todas las ocasiones (condiciones de carrera).

► Conflictos de escritura/escritura

- Dos hilos tratan de modificar el estado de un objeto simultáneamente.
- Las acciones no son atómicas

- Cerrojos o bloqueos sobre recursos.
- Bloqueos o cerrojos recursivos o reentrantes
- Semáforos
- Condiciones
- Eventos
- Barreras



www.unir.net