





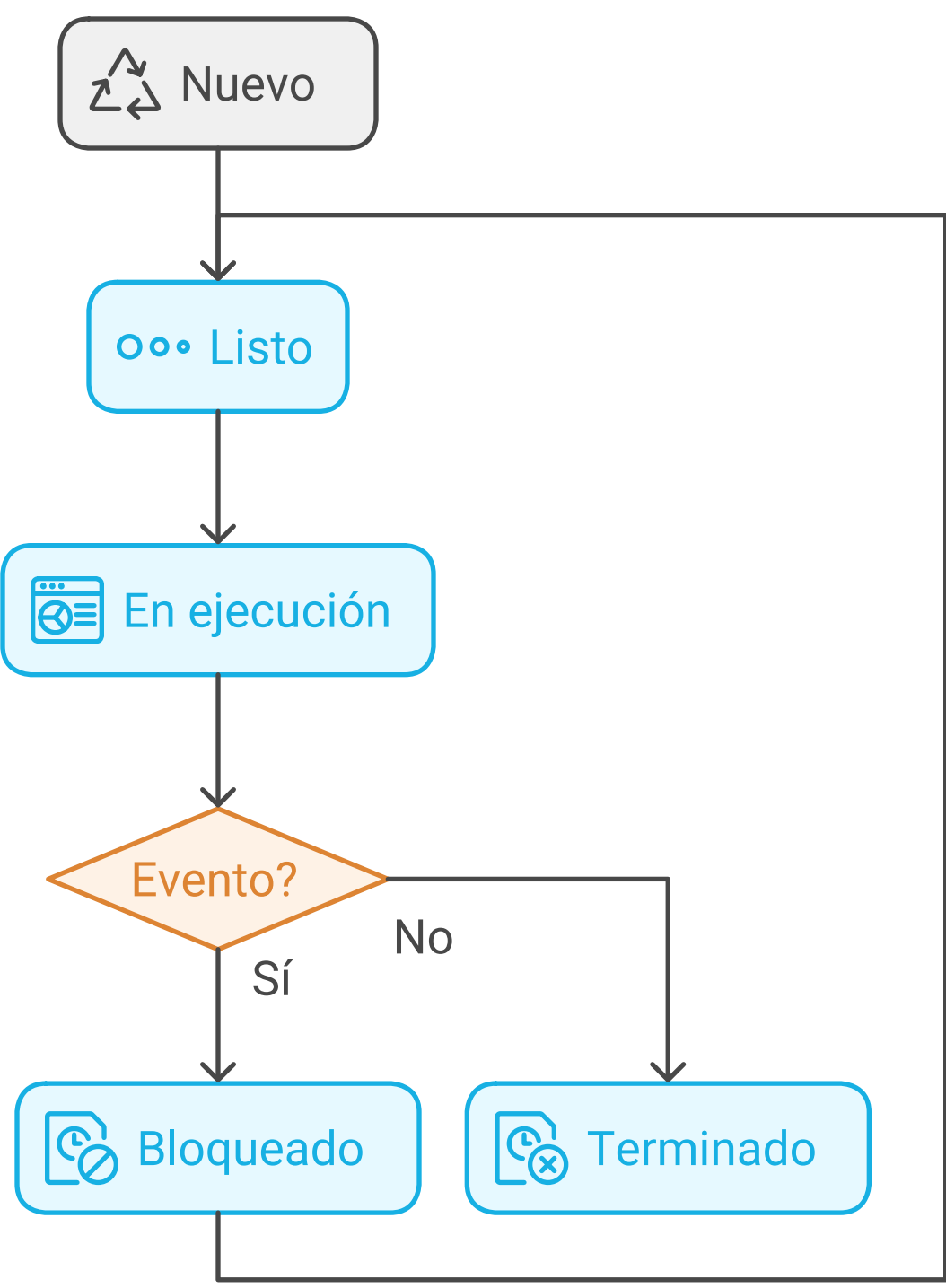
RESUMEN | PROCESOS

-  Programación Concurrente
-  Creación de Procesos
-  Comunicación de procesos
-  Gestión de procesos
-  Sincronización de procesos
-  Programación multiproceso

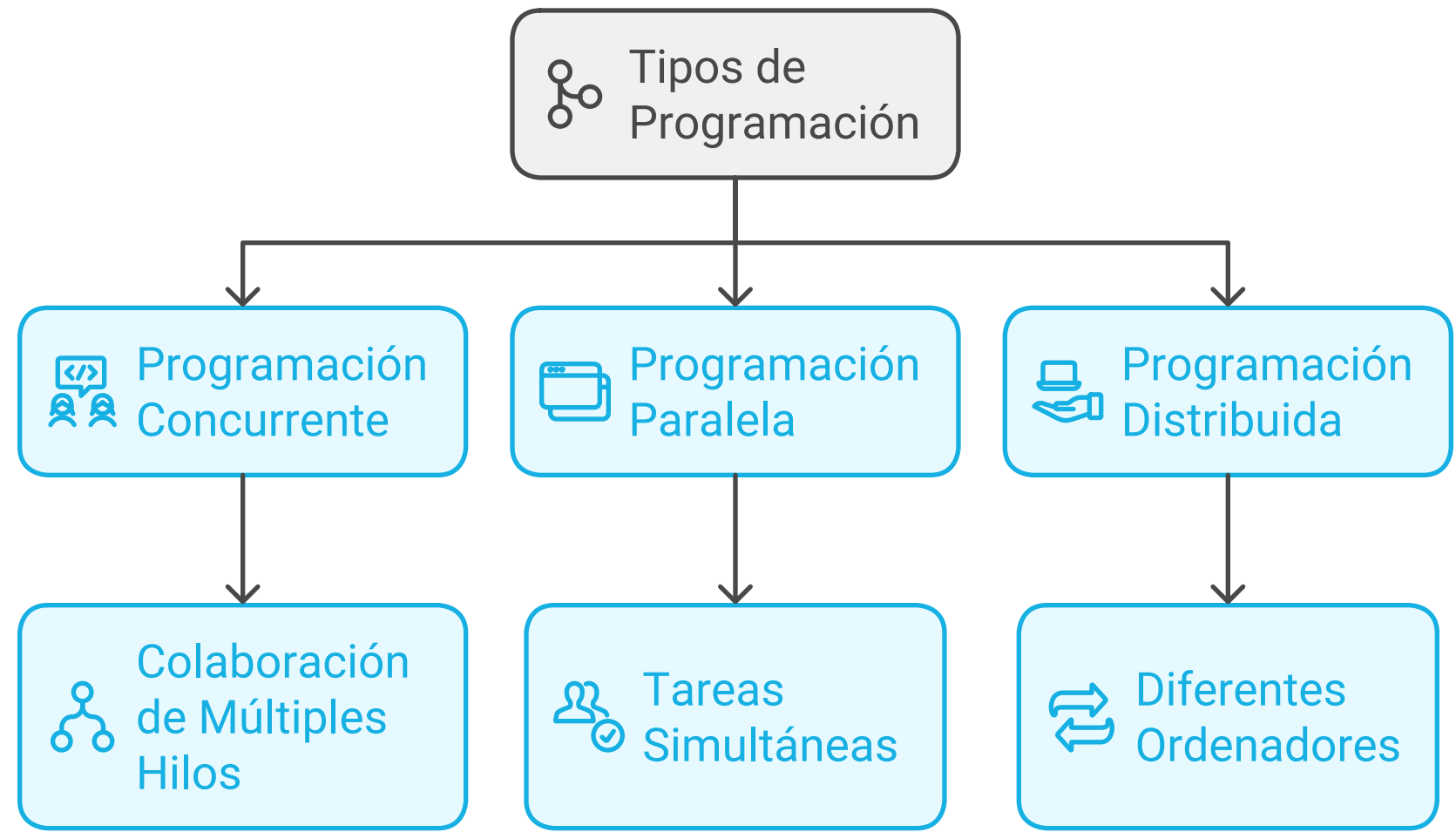
Conceptos básicos

EjecutablesUn ejecutable es un archivo con la estructura necesaria para que el sistema operativo pueda iniciar el programa que hay dentro. En Windows, los ejecutables suelen ser archivos con la extensión .EXE, pero también existen archivos como .COM, .BAT y librerías como .DLL. En Linux, ejemplos incluyen .bin, .run, .py, y .sh. Para que un fichero sea ejecutable en Linux, hay que darle permiso de ejecución.

ProcesosCuando un programa se ejecuta, se crea un proceso que queda bajo el control del sistema operativo. Un proceso puede estar en varios estados durante su ciclo de vida:



Cada proceso se representa con un PCB (Process Control Block) que almacena información como el estado del proceso, registros de CPU, información de planificación y gestión de memoria, entre otros.Programación concurrente, paralela y distribuida



Creación de Procesos en Java


En Java, es posible crear procesos usando la clase ProcessBuilder. Un ejemplo de cómo ejecutar un proceso del sistema operativo:

```
java


public class LanzadorProcesos {
    public void ejecutar(String ruta){
        ProcessBuilder pb;
        try {
            pb = new ProcessBuilder(ruta);
            pb.start();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    public static void main(String[] args) {
        String ruta = "C:\\Program Files\\Notepad++\\notepad++.exe";
        LanzadorProcesos lp = new LanzadorProcesos();
        lp.ejecutar(ruta);
        System.out.println("Finalizado");
    }
}
```


Descripción de ProcessBuilder y ProcessLa clase ProcessBuilder permite ejecutar programas en un proceso separado. El método start() crea un nuevo proceso, retornando una instancia de Process. Los procesos devuelven su estado con exitValue. El estado "0" indica éxito.



¿Cómo ejecutar un programa en un proceso separado?



**Usar ProcessBuilder**  
Permite ejecutar programas en un proceso separado.



**Usar Runtime.exec()**  
Método más antiguo, menos flexible.

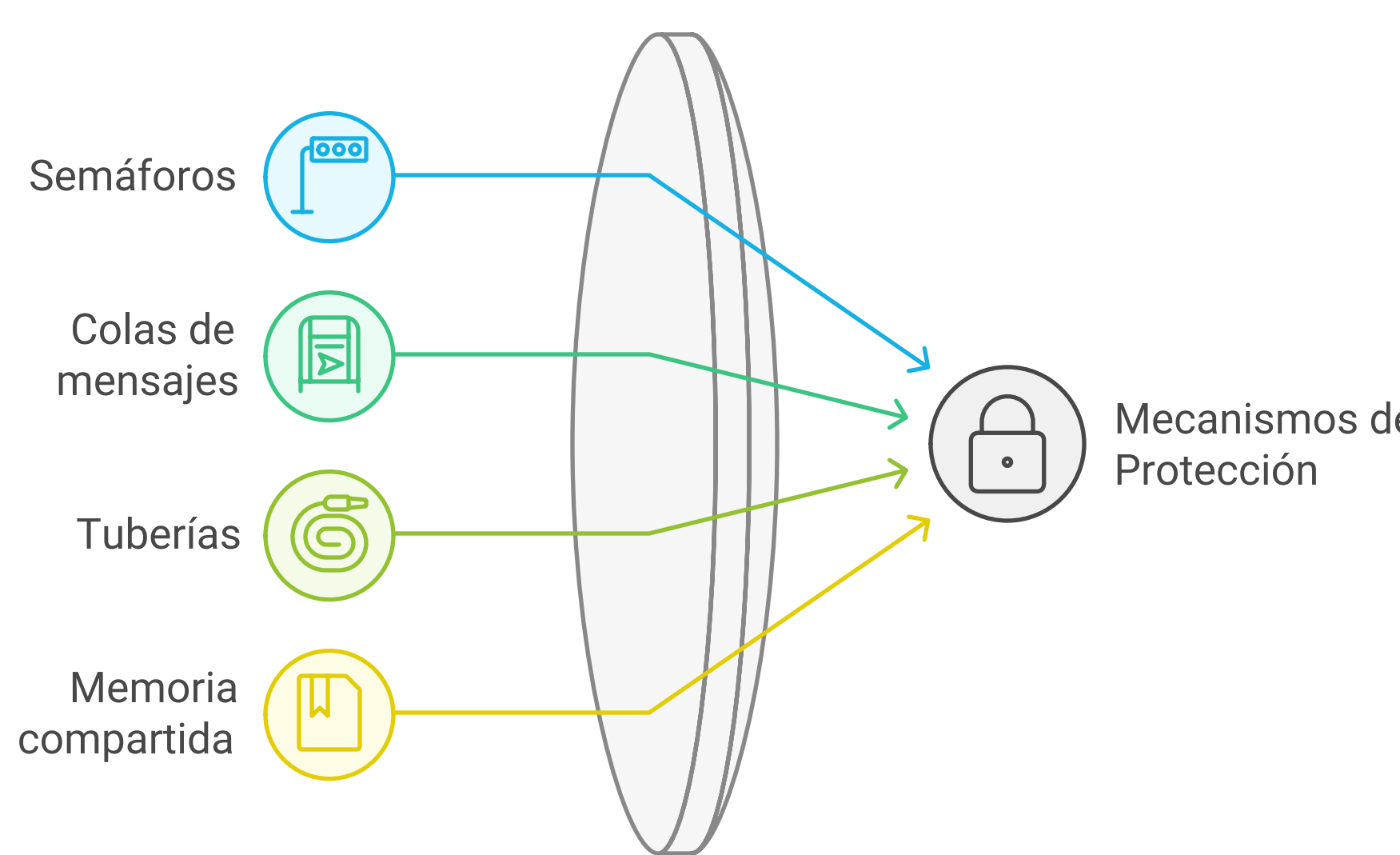
Comunicación de procesos

Las operaciones multiproceso requieren comunicación entre procesos, normalmente utilizando ficheros o redirigiendo las salidas estándar (stdout) y de error (stderr) a archivos log.Gestión de procesos

En Windows, la gestión de procesos se realiza mediante el Administrador de tareas. En UNIX, se utilizan comandos en la terminal para gestionar procesos, permitiendo arrancarlos, detenerlos, reanudarlos, y modificar su prioridad.Sincronización de procesos

Cuando varios procesos acceden a una misma sección de código, se pueden producir conflictos. Para proteger estas secciones críticas, se usan mecanismos como:

Protegiendo Secciones Críticas



En Java, se utiliza la palabra clave synchronized para proteger métodos o bloques de código, garantizando que solo un subproceso accede a ellos a la vez.Programación multiproceso

La programación concurrente alterna la ejecución de procesos para proporcionar multiprogramación. El sistema operativo gestiona esta alternancia. Para procesos cooperativos, se deben diseñar mecanismos de comunicación y sincronización adecuados. Ejemplo de uso de threads en Java:

```
public interface Runnable {
    public void run();
}
```

Las clases que actúan de forma concurrente deben implementar Runnable o extender de Thread, e implementar el método run para definir el comportamiento concurrente.