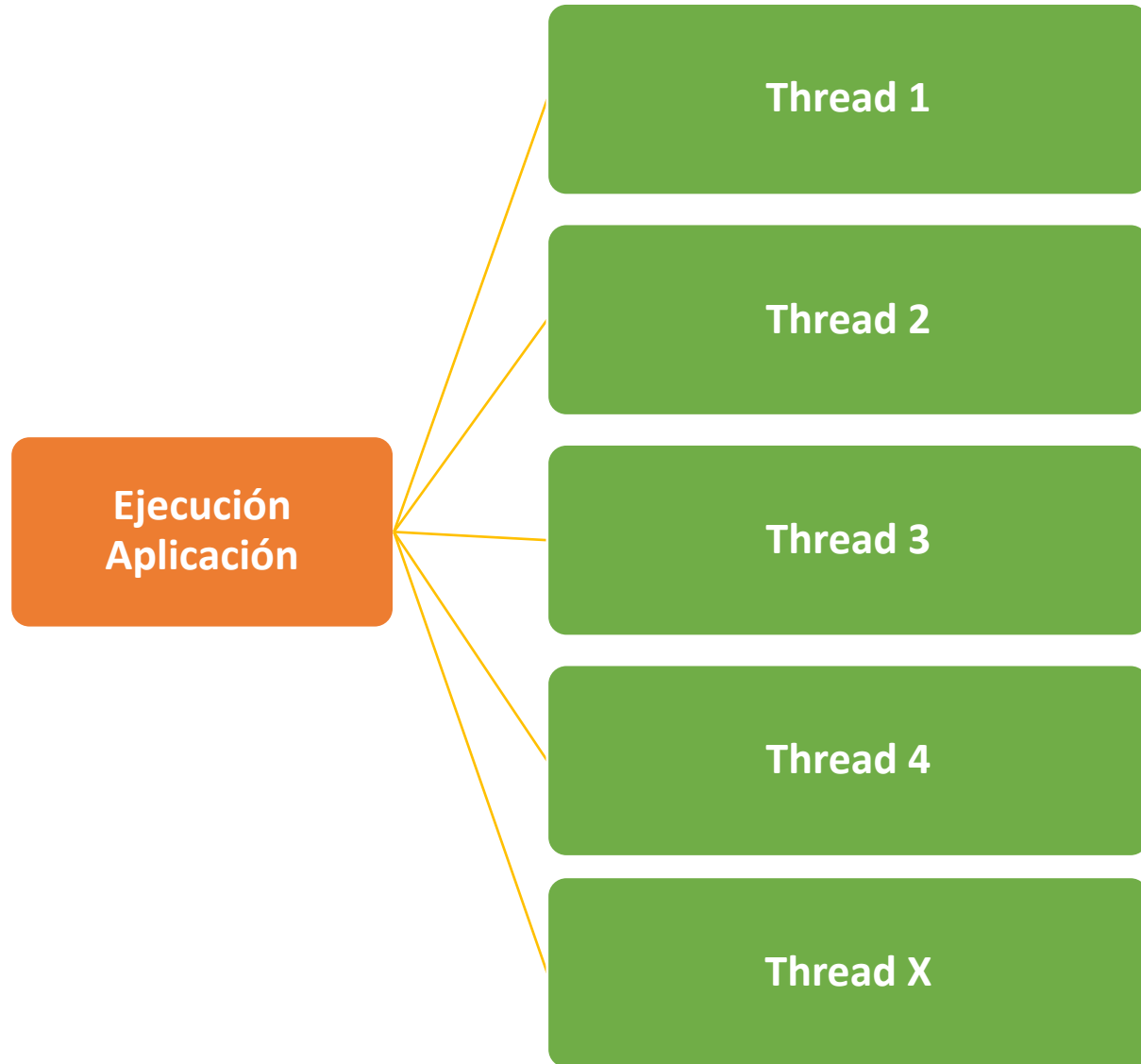
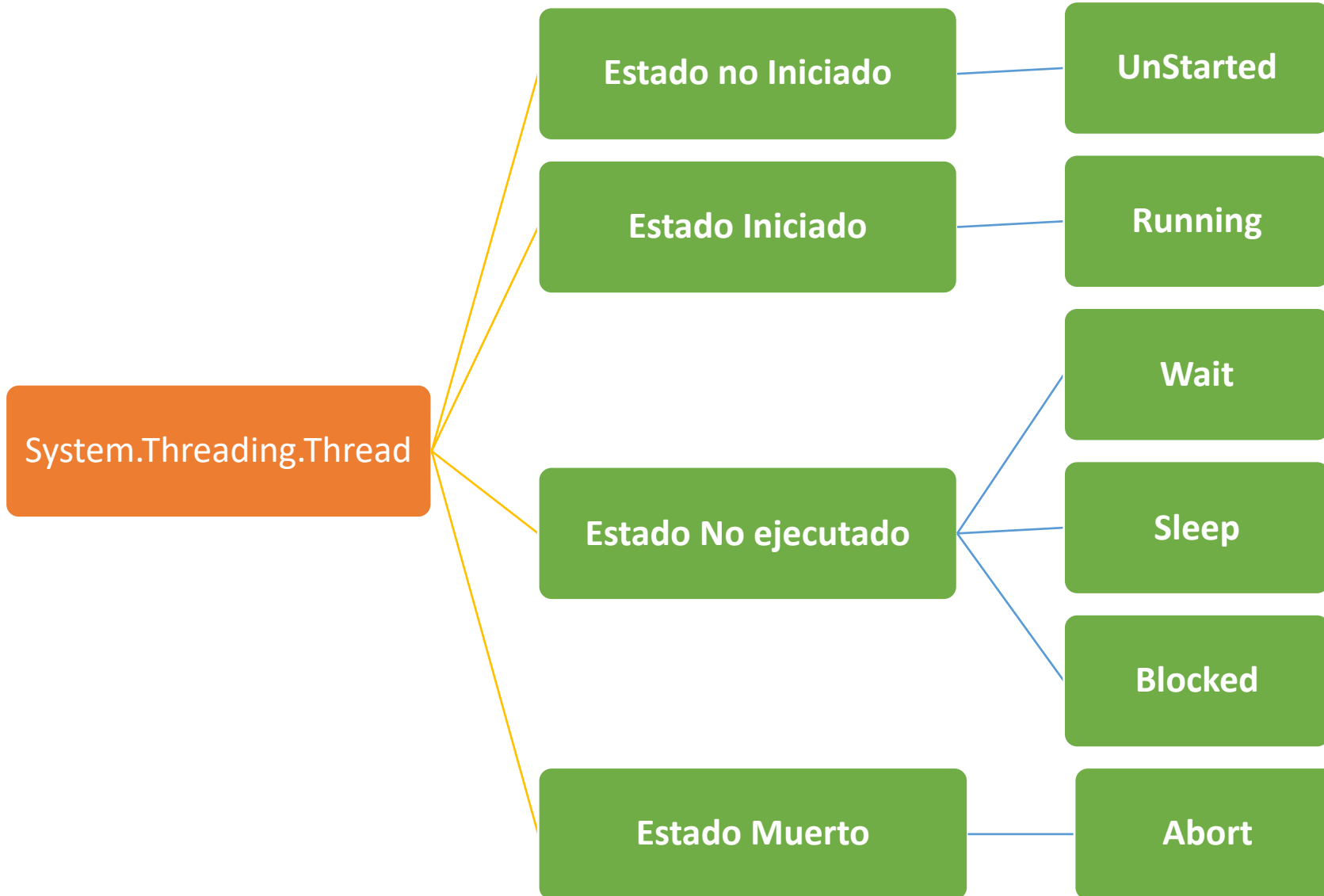


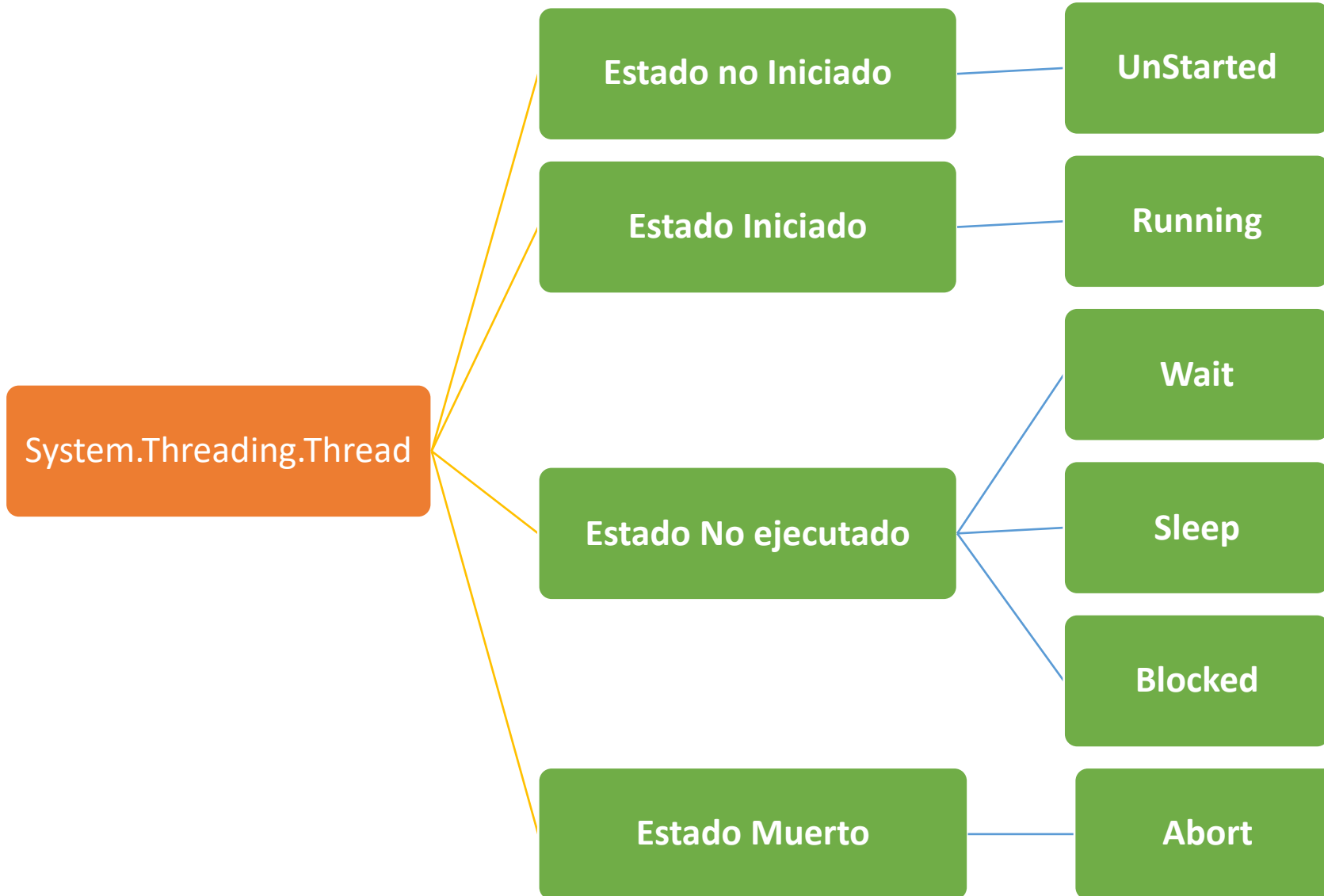
Introducción a Threads



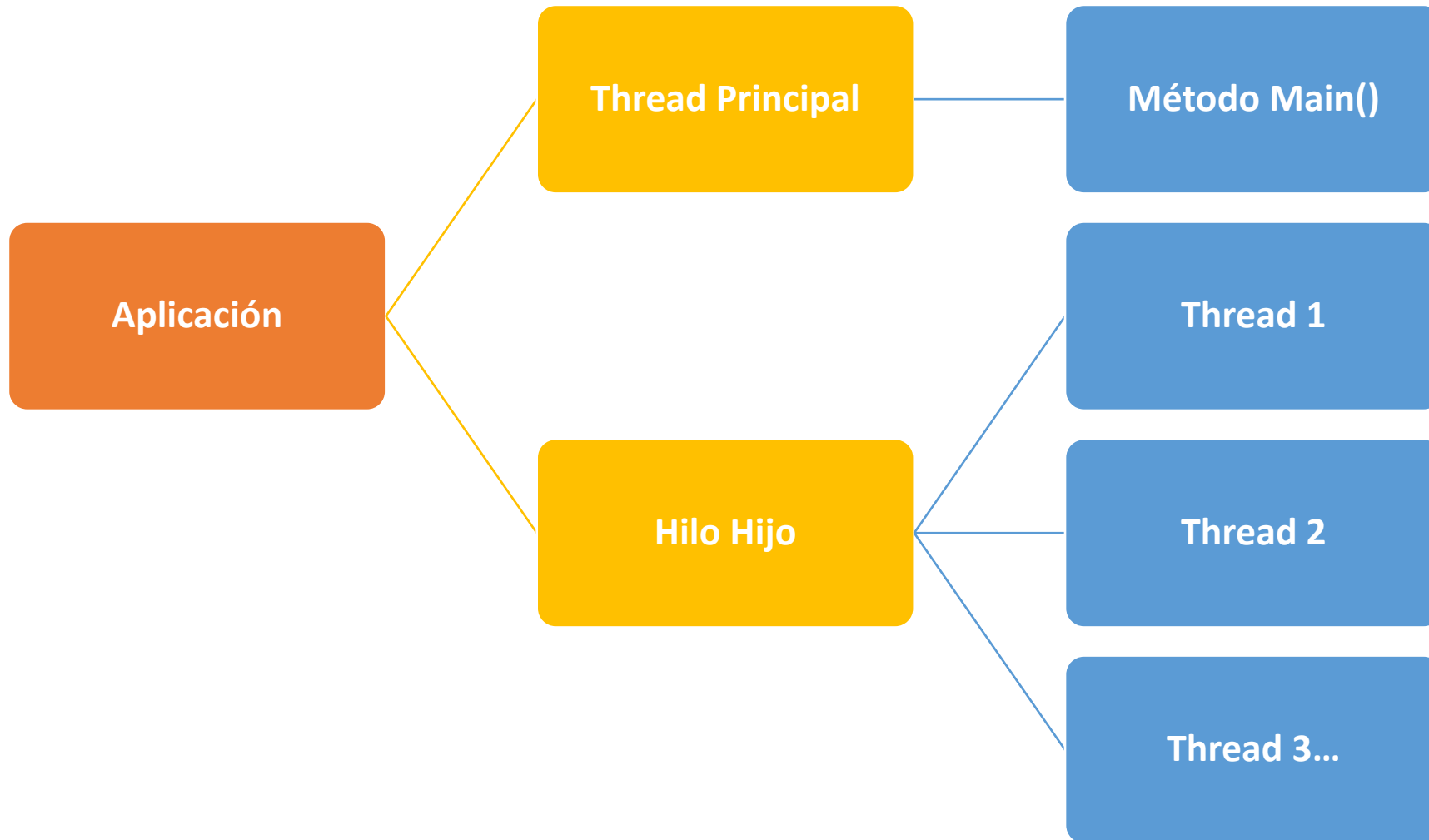
Introducción a Threads



Introducción a Threads



Introducción a Threads



Introducción a Threads

Propiedades y Descripción

CurrentContext

Obtiene el contexto actual en el que se está ejecutando el hilo.

CurrentCulture

Obtiene o establece la cultura para el hilo actual.

CurrentPrinciple

Obtiene o establece el hilo principal actual (seguridad basada en roles).

CurrentThread

Obtiene el hilo actual que se está ejecutando.

CurrentUICulture

Obtiene o establece la cultura actual utilizada por el Administrador de recursos para buscar recursos específicos de la cultura en tiempo de ejecución.

ExecutionContext

Obtiene un objeto `ExecutionContext` que contiene información sobre los diversos contextos del hilo actual.

IsAlive

Obtiene un valor que indica el estado de ejecución de la secuencia actual.

IsBackground

Obtiene o establece un valor que indica si un hilo es o no un hilo de fondo o background.

IsThreadPoolThread

Obtiene un valor que indica si un hilo pertenece o no al grupo de hilos administrados.

ManagedThreadId

Obtiene un identificador único para el hilo administrado actual.

Name

Obtiene o establece el nombre del hilo.

Priority

Obtiene o establece un valor que indica la prioridad de programación de un hilo.

ThreadState

Obtiene un valor que contiene los estados del hilo actual.

Introducción a Threads

Métodos y Descripción

`public void Abort()`

Levanta una `ThreadAbortException` en el hilo en el que se invoca, para comenzar el proceso de terminar el hilo. Llamar a este método generalmente termina el hilo.

`public static LocalDataStoreSlot AllocateDataSlot()`

Asigna una ranura de datos sin nombre en todos los hilos. Para un mejor rendimiento, use campos marcados con el atributo `ThreadStaticAttribute` en su lugar.

`public static LocalDataStoreSlot AllocateNamedDataSlot(string name)`

Asigna una ranura de datos con nombre en todos los hilos. Para un mejor rendimiento, use campos marcados con el atributo `ThreadStaticAttribute` en su lugar.

`public static void BeginCriticalRegion()`

Notifica a un host que la ejecución está a punto de ingresar a una región de código en la cual los efectos de un aborto de subprocesso o excepción no controlada pueden poner en peligro otras tareas en el dominio de aplicación.

`public static void BeginThreadAffinity()`

Notifica a un host que el código administrado está a punto de ejecutar instrucciones que dependen de la identidad del hilo del sistema operativo físico actual.

`public static void EndCriticalRegion()`

Notifica a un host que la ejecución está a punto de ingresar a una región de código en la que los efectos de un aborto de subprocesso o una excepción no controlada se limitan a la tarea actual.

`public static void EndThreadAffinity()`

Notifica a un host que el código administrado ha terminado de ejecutar instrucciones que dependen de la identidad de la secuencia actual del sistema operativo físico.

`public static void FreeNamedDataSlot(string name)`

Elimina la asociación entre un nombre y un espacio, para todos los hilos en el proceso. Para un mejor rendimiento, use campos marcados con el atributo `ThreadStaticAttribute` en su lugar.

Introducción a Threads

```
public static Object GetData(LocalDataStoreSlot slot)
```

Recupera el valor de la ranura especificada en la secuencia actual, dentro del dominio actual de la secuencia actual. Para un mejor rendimiento, use campos marcados con el atributo `ThreadStaticAttribute` en su lugar.

```
public static AppDomain GetDomain()
```

Devuelve el dominio actual en el que se está ejecutando el hilo actual.

```
public static AppDomain GetDomainID()
```

Devuelve un identificador de dominio de aplicación único

```
public static LocalDataStoreSlot GetNamedDataSlot(string name)
```

Busca una ranura de datos con nombre. Para un mejor rendimiento, use campos marcados con el atributo `ThreadStaticAttribute` en su lugar.

```
public void Interrupt()
```

Interrumpe un hilo que está en el estado del hilo `WaitSleepJoin`.

```
public void Join()
```

Bloquea el hilo que llama hasta que termina un hilo, mientras continúa realizando el bombeo estándar de COM y `SendMessage`. Este método tiene diferentes formularios sobrecargados.

```
public static void MemoryBarrier()
```

Sincroniza el acceso a la memoria de la siguiente manera: El procesador que ejecuta el hilo actual no puede reordenar las instrucciones de tal manera que la memoria tenga acceso antes de que la llamada a `MemoryBarrier` se ejecute después de los accesos de memoria que siguen a la llamada a `MemoryBarrier`.

```
public static void ResetAbort()
```

Cancela un aborto solicitado para el hilo actual.

```
public static void SetData(LocalDataStoreSlot slot, Object data)
```

Establece los datos en el intervalo especificado en el hilo actualmente en ejecución, para el dominio actual de ese hilo. Para un mejor rendimiento, use los campos marcados con el atributo `ThreadStaticAttribute` en su lugar.

```
public void Start()
```

Inicia un hilo.

Introducción a Threads

```
public static void Sleep(int millisecondsTimeout)
```

Hace que el hilo se detenga por un período de tiempo.

```
public static void SpinWait(int iterations)
```

Hace que un hilo espere el número de veces definido por el parámetro de iteraciones.

```
public static byte VolatileRead(ref byte address)
```

```
public static double VolatileRead(ref double address)
```

```
public static int VolatileRead(ref int address)
```

```
public static Object VolatileRead(ref Object address)
```

Lee el valor de un campo. El valor es el último escrito por cualquier procesador en una computadora, independientemente de la cantidad de procesadores o el estado del caché del procesador. Este método tiene diferentes formularios sobrecargados. Solo algunos se dan arriba.

```
public static void VolatileWrite(ref byte address, byte value)
```

```
public static void VolatileWrite(ref double address, double value)
```

```
public static void VolatileWrite(ref int address, int value)
```

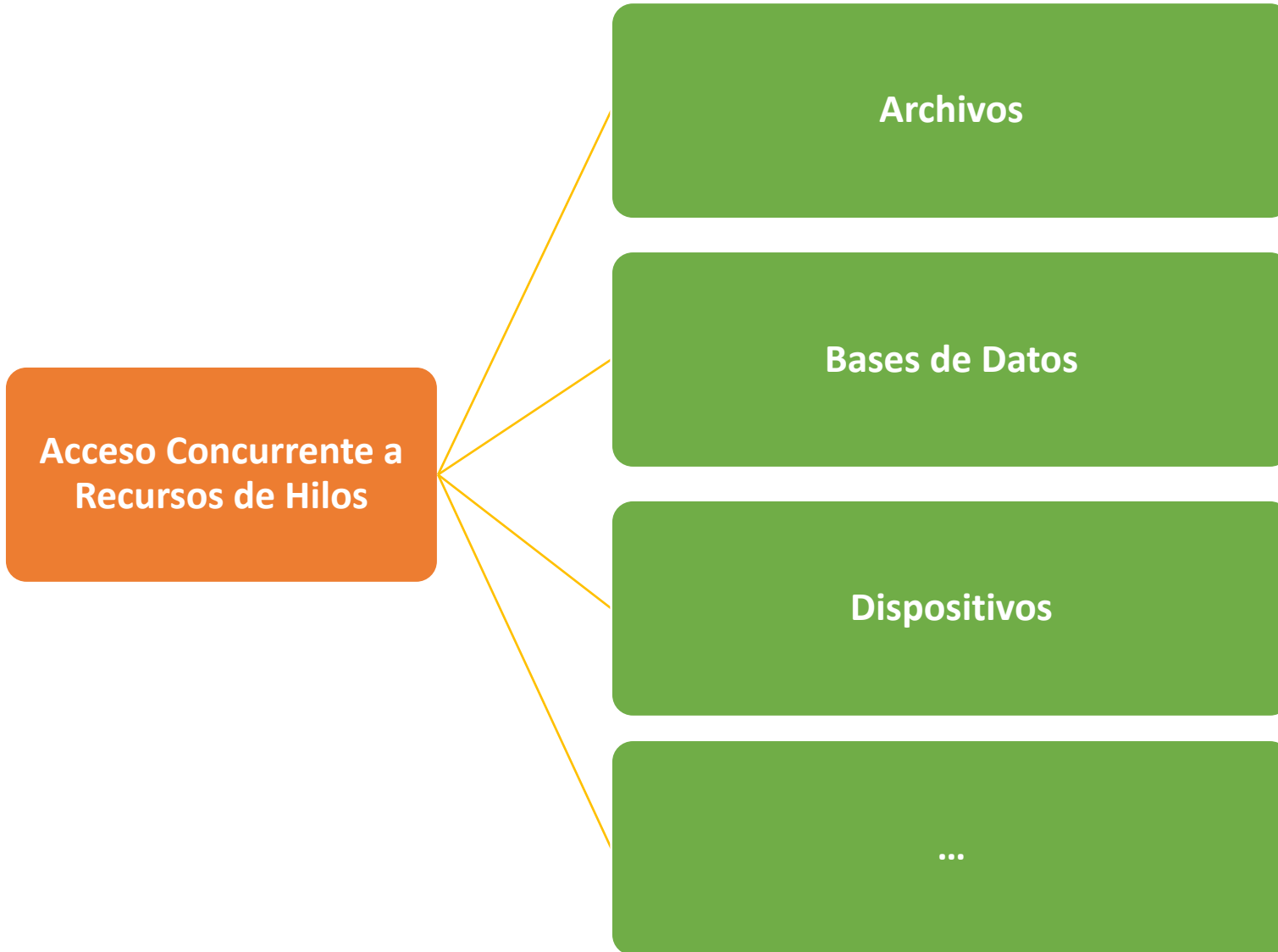
```
public static void VolatileWrite(ref Object address, Object value)
```

Escribe un valor en un campo inmediatamente, de modo que el valor sea visible para todos los procesadores en la computadora. Este método tiene diferentes formularios sobrecargados. Solo algunos se dan arriba.

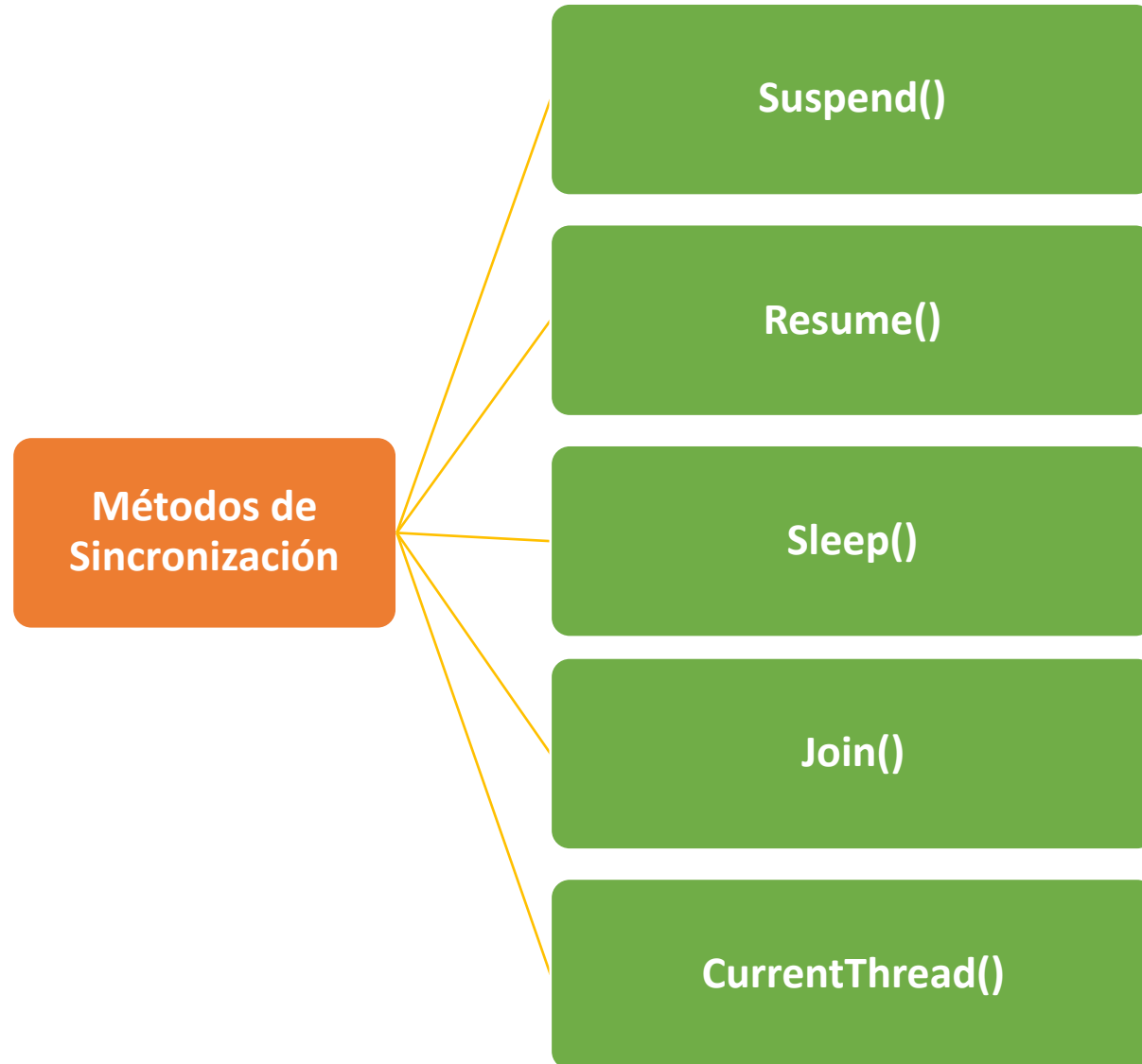
```
public static bool Yield()
```

Hace que el subproceso de llamada ceda la ejecución a otro subproceso que está listo para ejecutarse en el procesador actual. El sistema operativo selecciona el hilo para ceder.

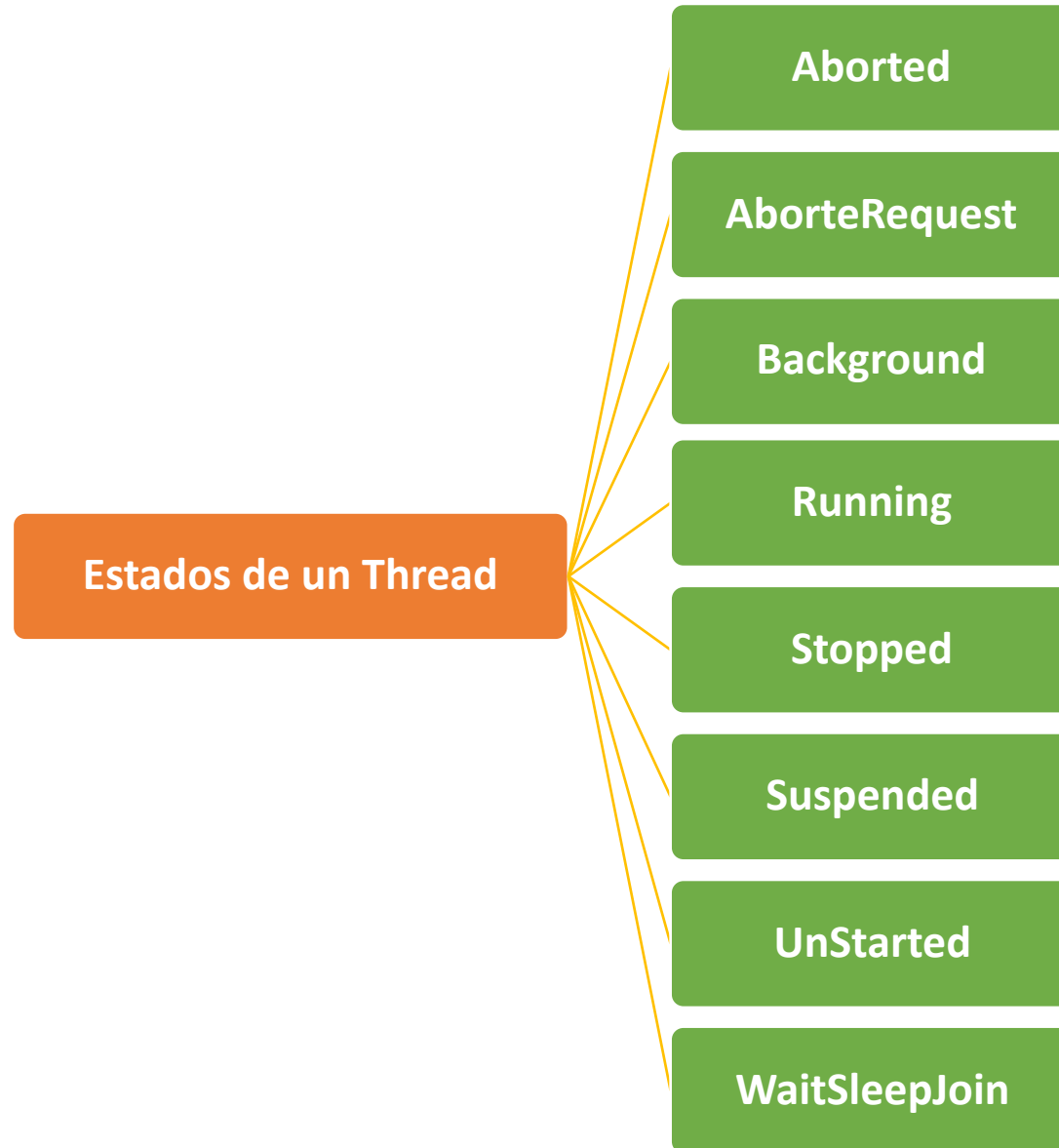
Threads – Programación Asíncrona



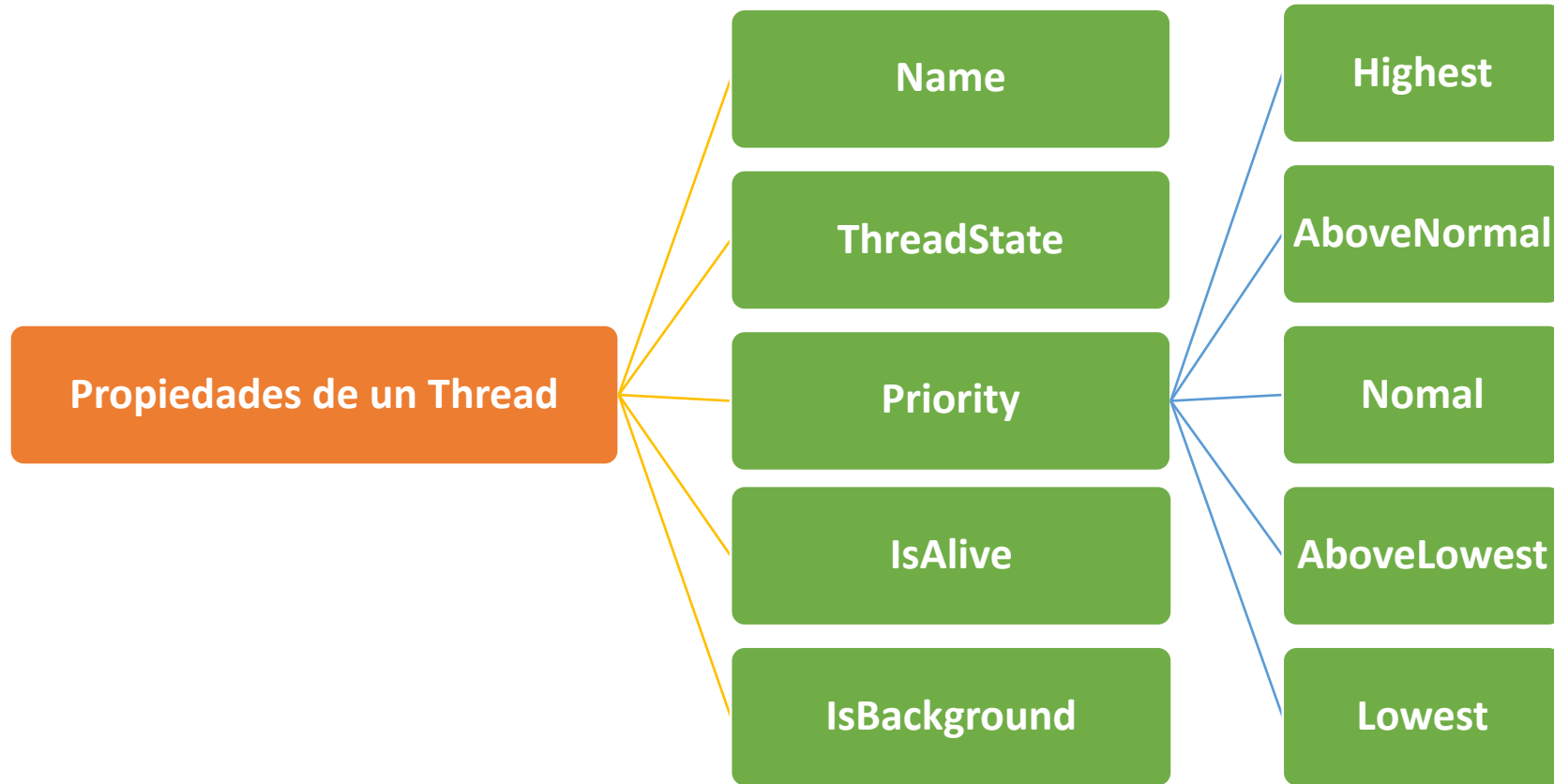
Threads – Programación Asíncrona



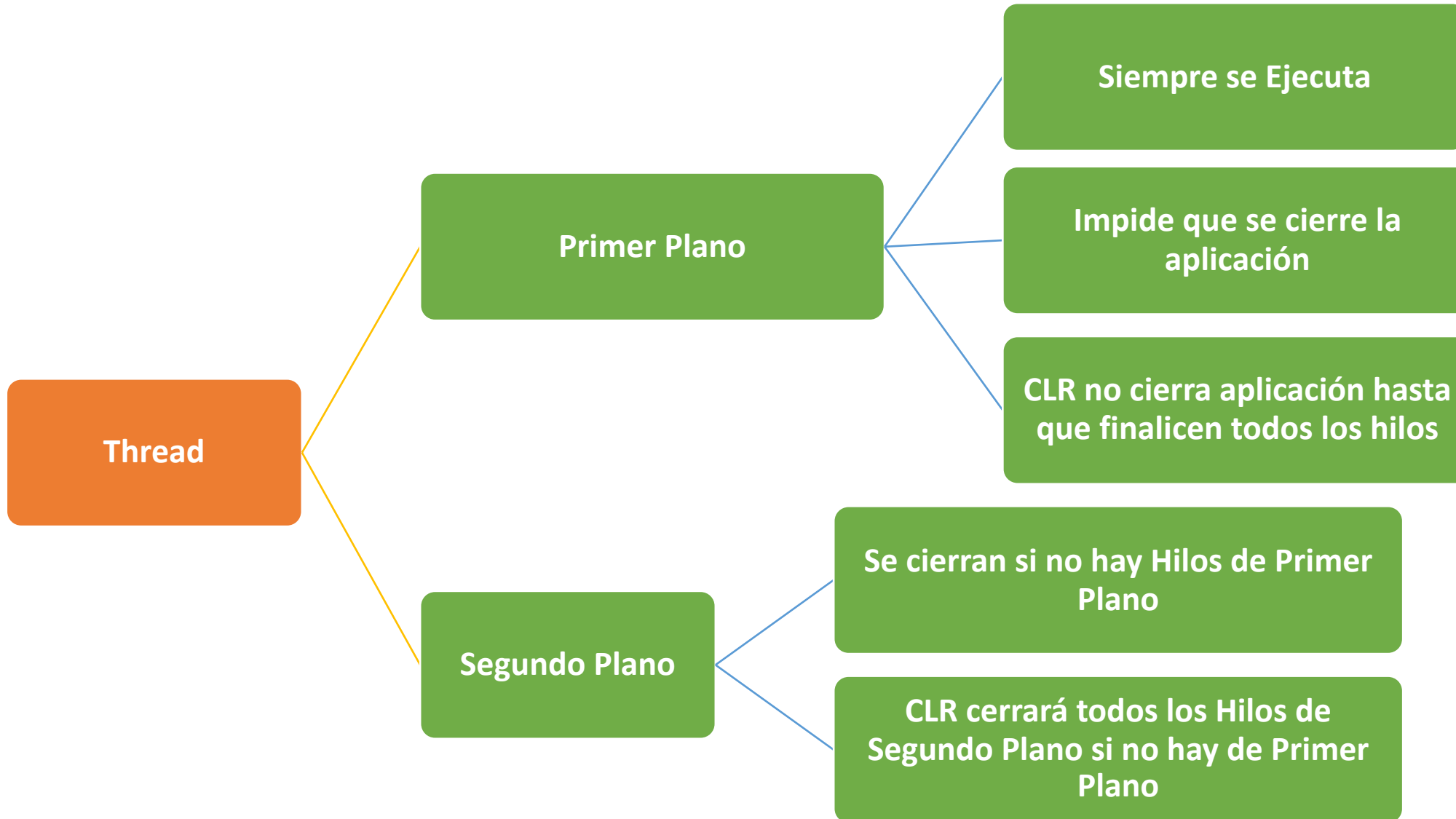
Threads – Programación Asíncrona



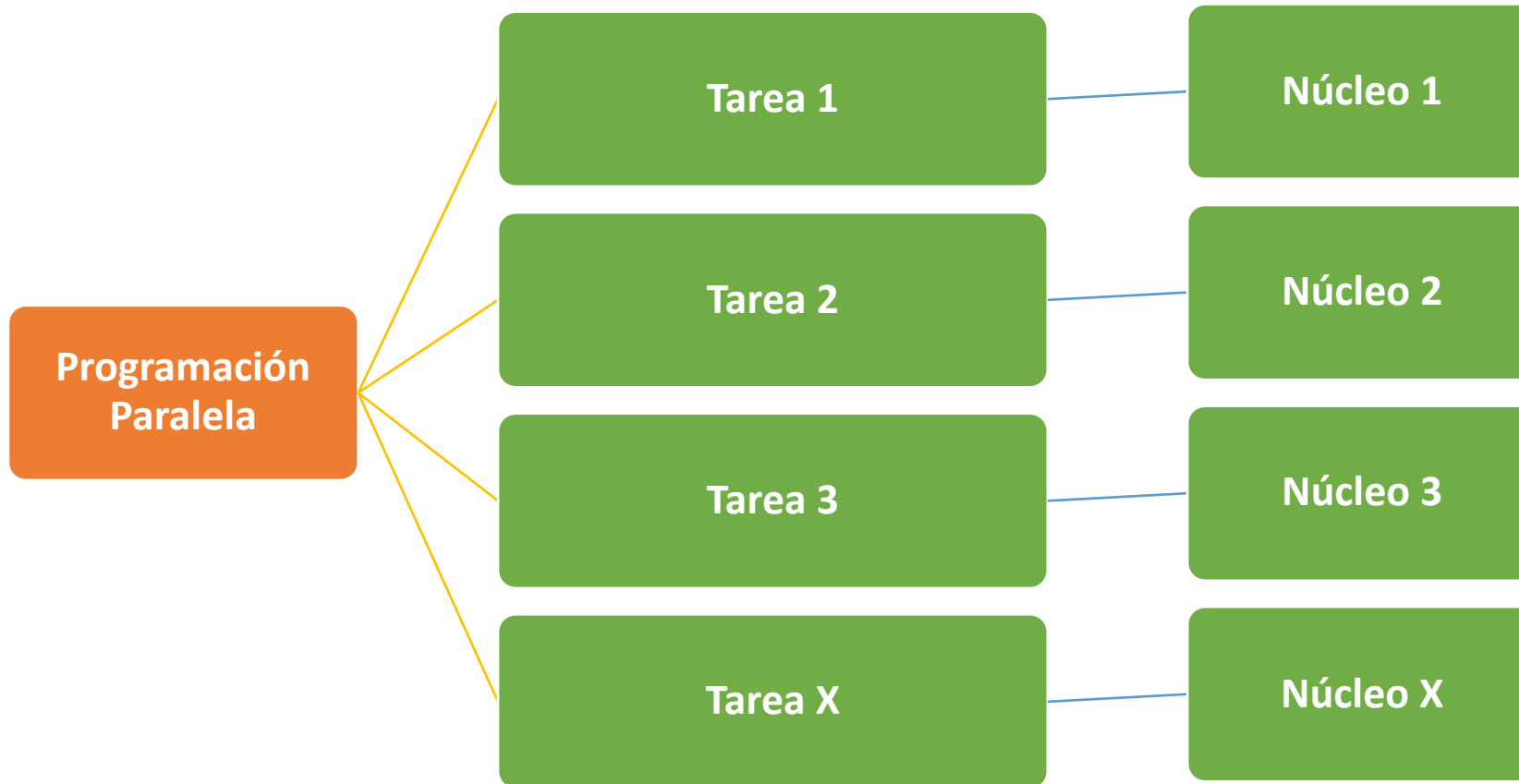
Threads – Programación Asíncrona



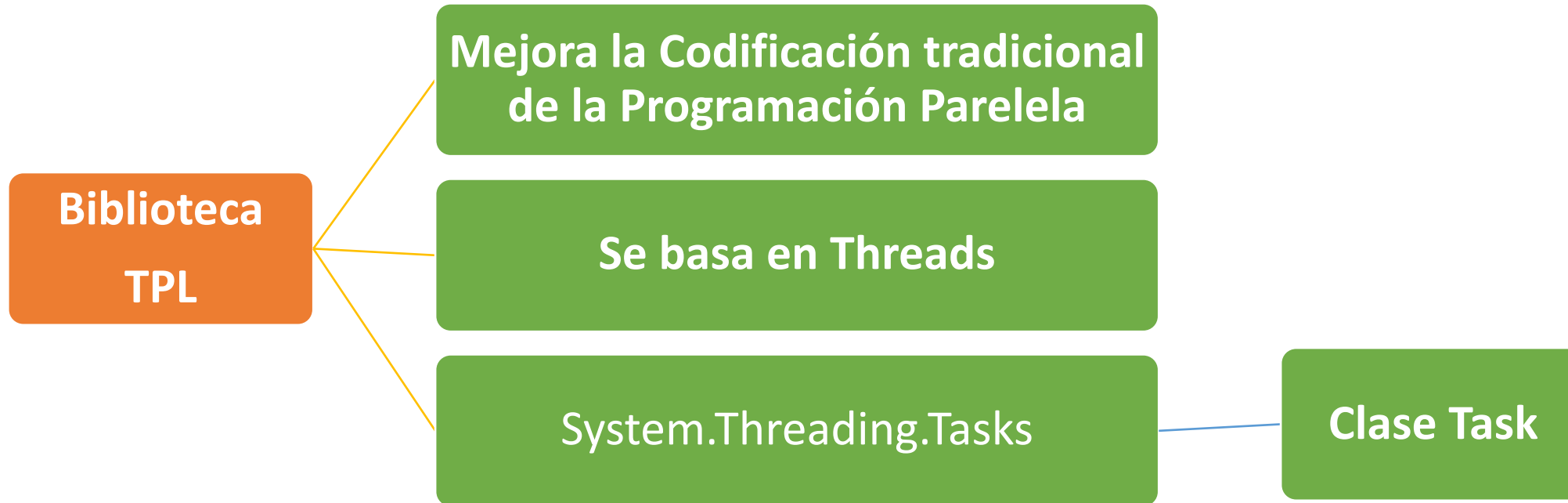
Threads – Primer Plano y Background



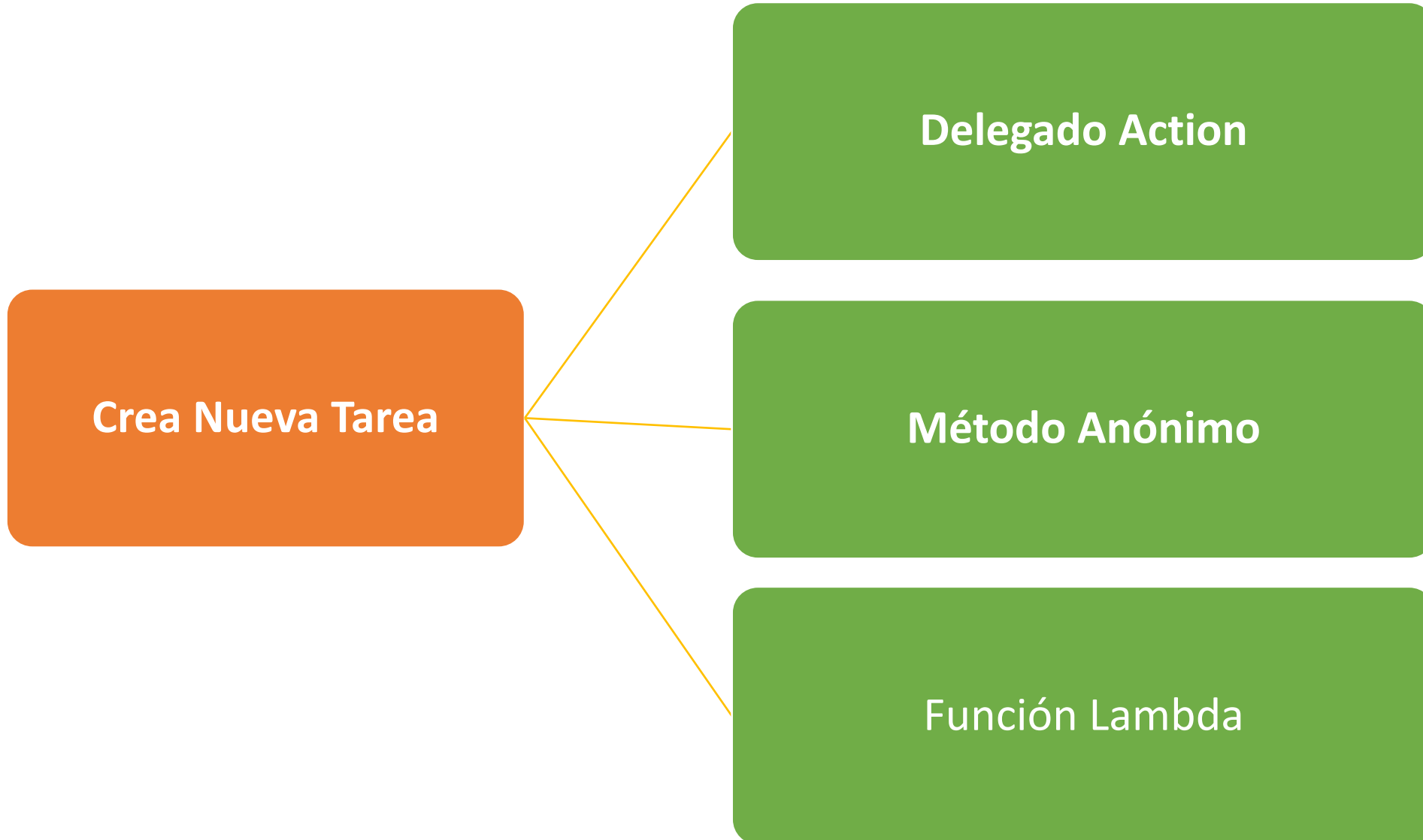
Threads – Programación Paralela



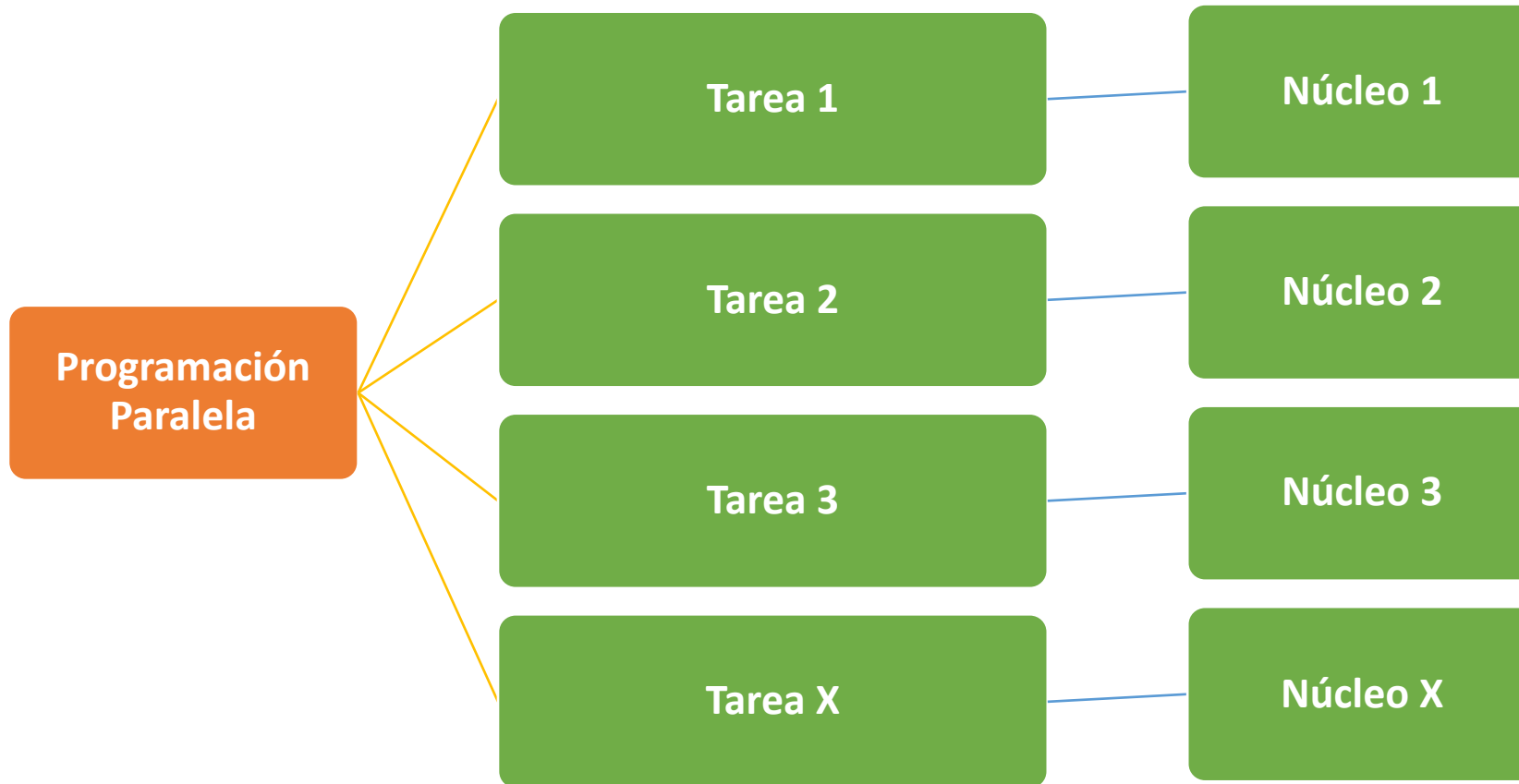
Threads – Programación Paralela



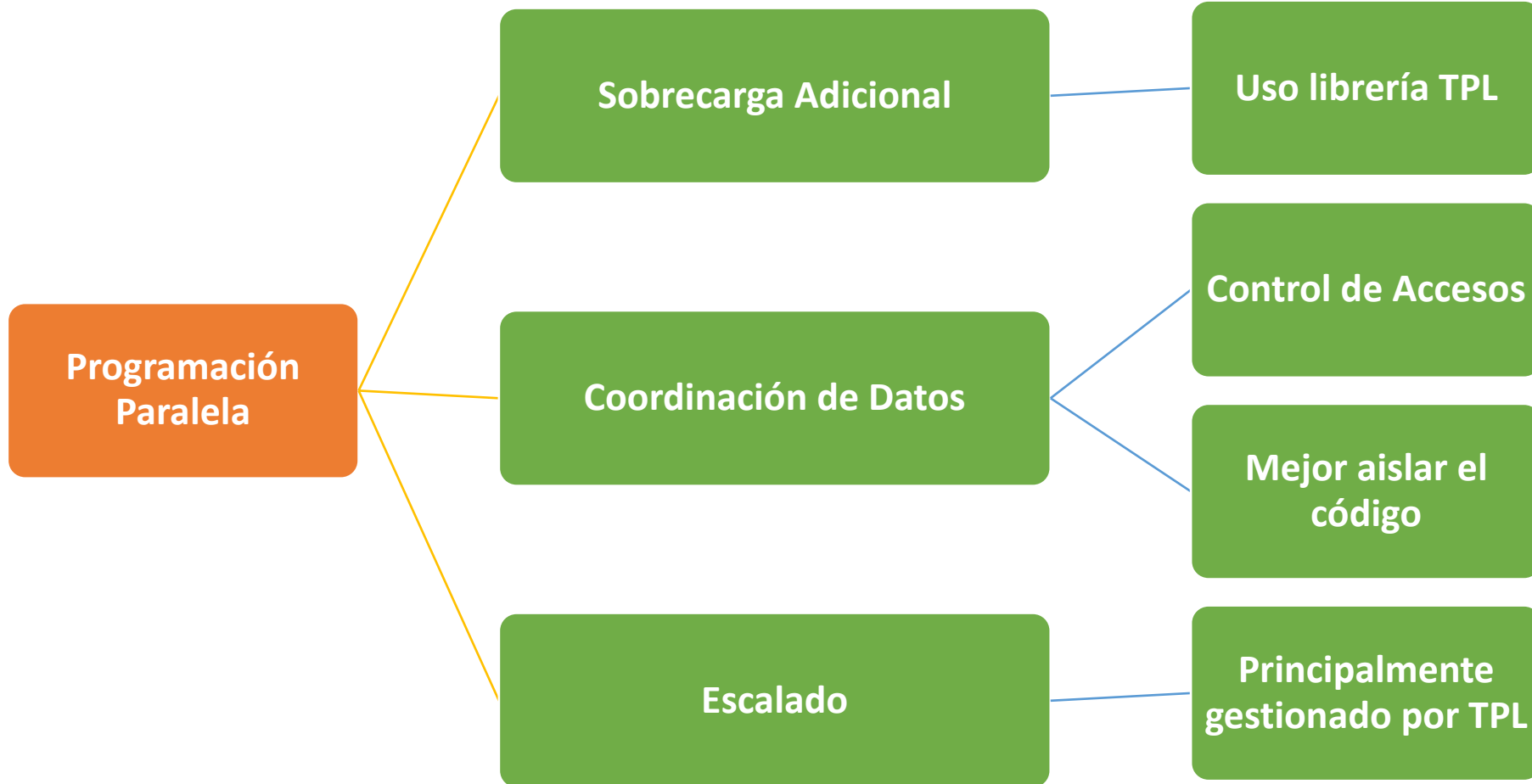
Threads – Programación Paralela



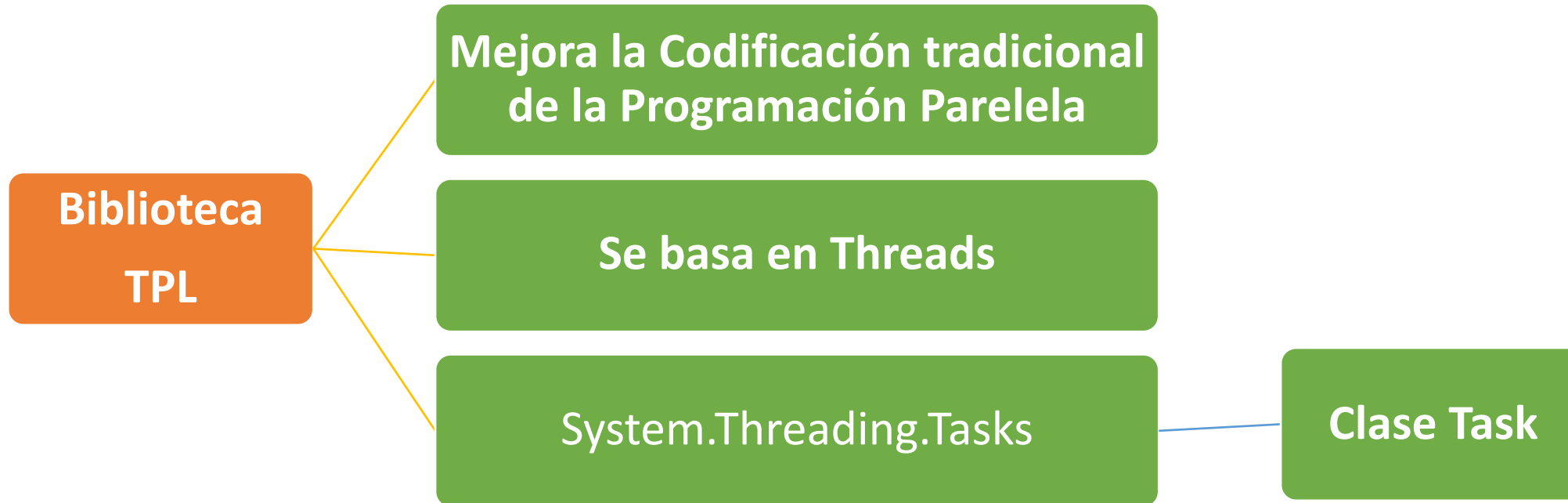
Threads – Programación Paralela



Threads – Programación Paralela



Threads – Programación Paralela



Threads – Programación Paralela

