

**Nombre:** Juan Diego Figueroa Hernández  
Juan Andrés Guarín Rojas  
Gabriela Sánchez Ariza  
Nicolas Toledo Parra

**Código:** 2200815  
2201870  
2200816  
2200017

## Paralelización Problema N cuerpos conceptual

### 1. Partición

En el caso de la partición del problema se tomó en cuenta que la estructura de los datos está bien definida y estructurada. Con lo cual el tipo de partición más útil sería el de descomposición de dominio. Primero, se debe mencionar que los datos del problema corresponden a las posiciones y velocidades de los  $N$  cuerpos a lo largo del tiempo. Cada uno de estos datos puede agruparse de acuerdo al cuerpo del que hace parte. De manera el conjunto de datos se puede particionar en  $N$  partes, cada una correspondiente a cada cuerpo. Lo anterior se puede hacer basado en que en la función *siguiente\_valor* (ver imagen adjunta) cada valor de posición y velocidad de un cuerpo es hallado de forma independiente. Puesto que cada dato nuevo de posición y velocidad solo depende del conjunto de datos del sistema asociado al frame anterior, y no al frame actual. En conclusión, esto permitiría asignar un procesador a cada cuerpo del problema gracias a la partición de los datos.

### 2. Comunicación

En este caso existe una comunicación sincrónica y estructurada en donde productores y consumidores se ejecutan de forma coordinada y cooperan en las operaciones de transferencia de datos. La estructura de canal que gestiona la transferencia de datos y vincula las tareas que los requieren son las funciones, las funciones son creadas y tienen un papel tanto de consumidor (como tarea que requiere datos) como de productor (tarea que posee estos datos), en otras palabras, las funciones pueden ser vistas como recolectores de datos que transforman estos en nuevos datos que serán usados posteriormente por otras funciones que recolectarán los datos transformados (estos datos serían mensajes de enviada y llegada), surgiendo así funciones anidadas como canal usado para primero comunicar los datos y luego transformarlos. También hay una comunicación global porque para generar una instrucción como, por ejemplo, mostrar en pantalla un cierto frame de la órbita, es necesario llamar a los datos generados por una función, datos que a su vez son datos transformados recibidos de otra función, siguiendo así de forma anidada, comunicando datos iniciales con el resto del código a través de las funciones.

### 3. Aglomeración

Para el caso del proceso de aglomeración de procesamiento en el algoritmo desarrollado se toma en cuenta que la capacidad de crear un número variable de tareas es fundamental para que un programa sea portátil y escalable, puesto que los buenos algoritmos paralelos están diseñados para ser resistentes a los cambios en el recuento de procesadores. Esta flexibilidad también puede ser útil al ajustar un código para una computadora en particular. Si las tareas a menudo bloquean la espera de datos remotos, puede ser ventajoso asignar varias tareas a un procesador. Entonces, una tarea bloqueada no tiene por qué hacer que un procesador se vuelva inactivo, ya que otra tarea puede ejecutarse en su lugar. De esta manera, la comunicación de una tarea se superpone con el cálculo de otra tarea, en nuestro caso las distintas tareas aglomeradas se enfocaban en partir las distintas fuerzas que actuaban en un cuerpo celeste determinado debido a la presencia de otros cuerpos masivos a su alrededor, de esta manera se dividían las tareas de procesamiento de cálculos de estas fuerzas, dependiendo del objeto estudiado, permitiendo generar el algoritmo en paralelo lo cual disminuiría el tiempo de cómputo al dividir tareas entre distintos procesadores.

### 4. Mapeo

En el caso del mapeo, el objetivo principal es disminuir el tiempo total de ejecución del problema de  $N$  cuerpos, de forma que se puedan ejecutar simultáneamente las diferentes tareas que le corresponde a cada procesador, para lo que fue necesario aglomerar las tareas, las cuales se enfocaban en las distintas fuerzas que actúan sobre un cuerpo celeste. Como se realizó la descomposición de dominio se crea una especie de malla lógica conformada por las diferentes tareas en cada uno de los procesadores, de forma que periódicamente cada procesador compara su carga computacional con la de sus vecinos para no detener el proceso.

Diagrama de flujo del pseudocódigo realizado para el problema de los N-cuerpos (Los diagramas de flujo de las funciones auxiliares se encuentran aquí: [https://drive.google.com/drive/folders/1\\_uNw-jl6k4LKu6wxAv9cgghN8-huy3mo?usp=sharing](https://drive.google.com/drive/folders/1_uNw-jl6k4LKu6wxAv9cgghN8-huy3mo?usp=sharing)).

