Diagrama de Flujo para Dinámica Molecular

Módulos necesarios:

general.c: en este módulo van funciones que pueden ser utilizadas en los distintos módulos, como my_rand(), distribucion_gaussiana(), distancia_al_cuadrado(), y delta_x_ij(): (delta_x devuelve un vector que va de x_i a x_j, útil para calcular las fuerzas).

inicializar.c: en éste módulo van las funciones que setean las condiciones iniciales, set_box() y set_v(), una arma la red cuadrada y la otra asigna velocidades a las partículas según la distribución de MB.

Para programar las funciones de inicializar.c van a necesitar usar funciones del módulo general. Otra ventaja de la programación modular es que nos permite ir de lo más general a lo más particular, aprovechando todos los módulos que construimos anteriormente.

Módulos necesarios:

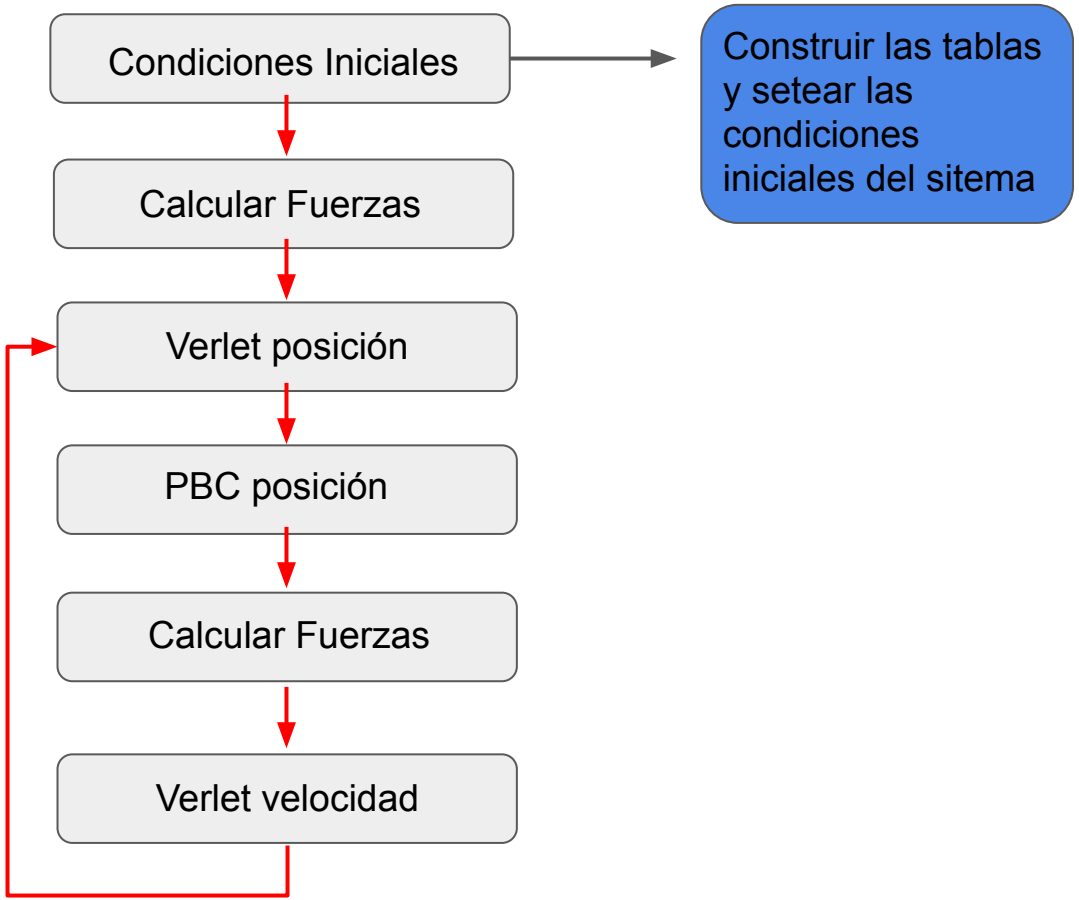
interaccion.c:

- Una función que haga las tablas
- Una función que mirando las tablas devuelva la fuerza y el potencial
- Aplicar las PBC a las fuerzas

avanzar.c:

- Verlet en posición
- Verlet en velocidad
- Calcular las fuerzas. Esta última función va a utilizar las tres funciones del módulo interacción
- Aplicar PBC a las posiciones luego de actualizarlas (por si alguna partícula se va de la caja)

Diagrama de Flujo:



Distribución gaussiana para las velocidades

```
double Gaussiana(double mu, double sigma) // devuelve un float con proba gaussiana.
{
int n = 15, i;
double z = 0;

for(i = 0; i < n; i++)
    {
    z  += Random();
    }

z = sqrt(12 * (double) n)*(z/n-0.5); //normal estándar.

return z*sigma + mu;
}
```

Devuelve números ‘aleatorios’ con distribución de probabilidad de MB. En nuestro caso mu = 0 porque no queremos corrientes y sigma se relaciona con la temperatura