

# Tarea N°3 Física Computacional I - 1S 2023

Prof. Guillermo Fonseca Kuvacic

Junio 2023

Instrucciones generales:

- Todas las entregas deben realizarse a través de algún repositorio de **Git** (puede utilizar Github o de otro servicio de su preferencia). La entrega puede realizarse mediante un repositorio general para el ramo, o un repositorio específico para esta tarea.
- La entrega de la tarea debe contener un archivo **README** con instrucciones de como ejecutar sus archivos, incluyendo instrucciones de *Make*, scripts (bash o awk) o archivos a compilar en C.
- En el caso de programar en C, se sugiere realizar programas modulares que consiste en repartir de forma ordenada las rutinas en múltiples archivos, junto con un “*header*” en el que se declaren estructuras de datos, funciones y librerías. Todo esto para tener un orden en el código. Ejemplo: Se tiene un programa que realiza una simulación:
  - Archivo1: Contiene la función `main`, que orquesta la ejecución del programa.
  - Archivo2: Contiene funciones para iniciar, asignar y eliminar la estructura de datos con la que se realiza la simulación.
  - Archivo3: Contiene funciones para ejecutar la simulación a partir de la estructura.
  - Archivo4: Contiene funciones para escribir los resultados de la simulación.
  - Archivo5: Contiene utilidades que sirven para todas las funciones de los archivos anteriores.
  - *Header*: Contiene las declaraciones de las estructuras y funciones que se utilizarán en otro archivo diferente al cual se definen. Ejemplo, la función `RK4()` se define en el archivo `rk4.c` pero se requiere utilizar en el archivo `main.c`, en este caso es necesario declarar la función en el archivo `header.h`.

## Problema

Considere una caja cúbica de volumen  $V = 1[m^3]$ , dentro de ella se ubicarán 64 partículas esféricas de diámetro  $0.25[m^3]$ .

Construya un algoritmo que calcule las posiciones de las partículas. Consideraciones:

- Las partículas no deben solaparse unas con otras, aunque pueden tener contacto (se sugiere añadir una tolerancia).
- Las posiciones de las partículas deben siempre estar dentro de la caja, aunque se permite que parcialmente las partículas sobrepasen los límites de la caja (el centro debe estar dentro de la caja).
- Las partículas no necesariamente deben ocupar todo el espacio de la caja.

Como salida de su algoritmo, debe entregar un archivo en formato XYZ.

**Hint:** Puede considerar la estructura cristalina con factor de empaquetamiento más alto.