

Práctica 1 (manejo del tiempo)

1. Medición de tiempos y espera

Experimente con las funciones vistas en clase para medir intervalos de tiempo y suspender la ejecución de un thread.

2. Temporización con intervalos discretos

Implemente una función que calcule el tiempo que se debe esperar hasta que se cumpla un tiempo $t = t_0 + n \cdot dt$, desde un instante inicial t_0 , donde n es el número de intervalo (0, 1, 2, etc) y dt es la duración de un intervalo. Todos los tiempos deben estar medidos en segundos.

```
#include <chrono>
// funcion que calcula cuanto tiempo falta para que deba
// iniciar el siguiente intervalo (n) y realiza una
// espera con ese tiempo
using duration = std::chrono::duration<double>;
using time_point = std::chrono::steady_clock::time_point;

std::chrono::duration<double> espera_siguiete_intervalo(const
std::chrono::steady_clock::time_point & t0, int n, const
std::chrono::duration<double> &dt );
```

3. Programa con control de tiempo

Utilizando el esqueleto de programa con control de tiempo dado y la función del ejercicio 2, completar el programa de control de tiempo.

4. Ensayar programa

Corra el programa del punto 3 para diversos valores de Dt {0.1, 0.05, 0.02, 0.01, 0.005, 0.002, 0.001} segundos. Analice los valores de espera y el tamaño de cada intervalo temporal y discuta.