

## CS2013 - Programación III

### Simulacro Práctica Calificada #2 (PC2)

2022 - 1

Profesor: Rubén Rivas

---

#### Complejidad Algorítmica - 6 puntos

1. Diseñar y desarrollar el template de función **valores\_iguales** que en un tiempo lineal  $O(n)$  permita identificar si 2 contenedores del mismo tamaño contienen los mismos elementos no importando que sus valores se encuentren en diferente orden.  
Ejemplo:

```
container_1 = { 1, 4, 5, 19, 20, 11, 22, 1 };  
container_2 = { 1, 1, 4, 5, 11, 19, 20, 22 };
```

#### Casos de uso:

```
// Caso de uso #1  
int n = 0;  
cin >> n;  
vector<int> vec1(n);  
vector<int> vec2(n);  
for(auto& item: vec1)  
    cin >> item;  
for(auto& item: vec2)  
    cin >> item;  
cout << valores_iguales(vec1, vec2) << endl;  
// Caso de uso #2  
deque<int> deq1(n);  
list<int> lis1(n);  
for(auto& item: deq1)  
    cin >> item;  
for(auto& item: lis1)  
    cin >> item;  
cout << valores_iguales(deq1, lis1) << endl;
```

## Programación Concurrente - 10 puntos

2. Elaborar la función *find\_fibonacci* que utilizando concurrencia obtenga de una colección todos los números que pertenezcan a la serie fibonacci y que los almacene en un contenedor. El programa deberá asegurarse que los números sean almacenados en orden.

Considere el siguiente link:

- [Determinar si un número es un número Fibonacci](#)

Considere utilizar la siguiente formula para determinar la cantidad de threads.

Cantidad de thread:

```
int get_number_of_threads(int sz, int rng) {  
    int max_threads = (sz + rng - 1) / rng;  
    int k_threads = static_cast<int>(thread::hardware_concurrency());  
    return min(k_threads != 0? k_threads: 2, max_threads);  
}
```

Casos de uso:

```
int n = 0;  
cin >> n;  
List<unsigned int> fs(n);  
for (auto& item: fs)  
    cin >> item;  
  
auto rs = find_fibonacci(fs);  
auto current = begin(rs);  
if (current != end(rs))  
    cout << *current++;  
while(current != end(rs))  
    cout << " " << *current++;  
cout << endl;
```

## Patrones de Diseños - 4 puntos

3. Se desea desarrollar un programa que permita simular estructuras urbanas desde cuartos (sala, comedor, baños, etc.) departamentos y casas que incluyan cuartos, departamentos y casas que se encuentren en cuadras, cuadras que se encuentren en distritos y distritos en ciudades.  
¿Qué tipo de patron(es) recomendaría? Dibujar el diagrama de clases usan el patron(es) seleccionado que represente la solución a este programa.

Barranco, 3 de junio 2022.