

# 1 Seminarios C y C++

Los requerimientos de cada ejercicio del seminario serán expuestos desde el punto de vista práctico y teórico; es decir, para su exposición, cada equipo se basará en el caso práctico en cuestión para introducir y explicar el elemento teórico requerido. La exposición no es una mera enunciación de código. Preguntas como: *¿Por qué?*, *¿Basándose en qué?*, *¿Cómo se logra esto en el lenguaje X?* entre otras, deben hacerse.

Todos los miembros del equipo deben participar en la solución del ejercicio y estar preparados para exponer todo el trabajo. **La persona a exponer** se decide el día de la exposición. Quién no esté presente en la exposición de su equipo tiene 0 en la evaluación. (Note que estas notas se promedian y hay distinción entre 0 y 2).

## 1.1 Seminario 4 (C++11 y C++14):

1. Implemente utilizando **TMP** (*Template Metaprogramming*) de C++:
  - a. Un algoritmo que calcule en tiempo de **COMPILACIÓN** el n-ésimo valor de la secuencia de *Fibonacci*.
  - b. Implemente un **template** que permite desenrollar (*unroll*) una expresión **for** de manera que se genere código secuencial que simule la ejecución de esta expresión.
2. ¿Cómo surge la meta-programación utilizando **templates**?
3. Explique el principio **SFINAE** de C++.
4. Implemente utilizando **constexpr** de C++:
  - a. Un algoritmo que calcule en tiempo de compilación el n-ésimo valor de la secuencia de *Fibonacci*.
  - b. Una clase **Point** que se pueda instanciar en tiempo de compilación.
  - c. Un algoritmo que calcule el punto intermedio entre dos puntos en tiempo de compilación.
5. Explique hasta dónde se pueden utilizar las **constexpr** de C++14.
6. Sobre las características propuestas para C++17.
  - a. Explique en que consiste *template argument deduction for class templates* y en qué puede ser útil.
  - b. Explique la propuesta **if (init; condition)** y **switch (init; condition)**. (Puede ser útil en C#?)
  - c. Exponga brevemente otras 2 propuestas al lenguaje que le parezcan atractivas.