

Práctica de laboratorio # 5

Puede desarrollar esta práctica en parejas o tríos. Procure trabajar con alguien diferente cada vez.

En este laboratorio el o la estudiante implementará el uso de clases parametrizadas por medio de la implementación de plantillas para generar una solución flexible, utilizando para ello el lenguaje C++.

Problema único. Matrices de relación.

Una matriz de relación es una representación en 2 dimensiones y de forma cuadrada, de la **relación 1:1** que se establece entre 2 objetos del mismo tipo.

Por ejemplo, Facebook podría utilizar una matriz de relación, en donde cada persona A se relaciona con una persona B por medio de una amistad. Si entre A y B no hay relación, podría de igual forma representarse esta "no-relación" por otros valores en la matriz.

Como otro ejemplo, una ciudad en un mapa digital X puede relacionarse con otra ciudad Y, y su relación podría ser la distancia en kilómetros que hay entre cada una.



En una matriz de relación:

- La **matriz es cuadrada**, porque la representación de filas y columnas corresponde con una misma lista de objetos o elementos base.
- Las **relaciones siempre tienen un valor entero**. Cero significa que no hay relación, mientras que diferente de 0 permite no solo establecer que sí hay relación, sino también que podría significar algo más.
- Las relaciones **no necesariamente son conmutables**, lo que significa que la relación de A hacia B no necesariamente es la misma que de B hacia A.
- La cantidad de elementos que componen las dimensiones de la matriz **puede cambiar con el tiempo** (agregándose o eliminándose elementos), por lo que la matriz también podría crecer y decrecer en tamaño.

Usted debe diseñar una **plantilla de matriz de relación** que permita establecer y leer relaciones entre objetos del mismo tipo. Tome en cuenta que debe implementar al menos las siguientes funcionalidades:

- Agregar un elemento
- Eliminar un elemento
- Establecer relación entre 2 elementos y colocarle un valor si corresponde
- Eliminar relación entre 2 objetos, y modificar el valor de su relación
- Ver el valor de la relación entre los 2 elementos.

Una vez implementada la plantilla, diseñe e implemente el código para que funcione para solucionar 3 problemas totalmente diferentes. En el diseño debe existir un menú que permita manipular y ver las relaciones.

- 1) Una red social de citas románticas, en donde una persona usuaria con su cuenta, contraseña, número de contacto, entre otros, da "likes" a otra persona si le gusta su apariencia. Si el contacto es mutuo, se anuncia de forma específica.

- 2) Una aplicación de montos de pasajes de metro entre ciudades, en donde cada ciudad con su nombre, extensión, y demás características se relaciona con otras por medio del costo que tiene el pasaje correspondiente. Este costo podría no ser el mismo de ida que de vuelta.
- 3) Un sistema de relaciones en una empresa, que indica si las personas trabajadoras son compañeras de departamento o si más bien no trabajan juntas. En este caso, la relación sí sería conmutable.

Implemente sobrecarga de operadores para 3 diferentes operaciones:

- la eliminación de elementos en la plantilla
- el agregado de elementos en la plantilla
- la impresión de los objetos que use en los 3 problemas diferentes, directamente usando cout

Haga un main que permita probar su programa. Si tiene supuestos, haga un README y adjúntelo.

Rubro	Peso
Definición de clase plantilla	40%
Definición de clases para cada problema	30%
Sobrecarga de operadores	20%
Interacción y pruebas	10%