

### Indicaciones específicas:

- Esta evaluación contiene 6 páginas (incluyendo esta página) con 2 preguntas. El total de puntos son 20.
- El tiempo límite para la evaluación es 100 minutos.
- Cada pregunta deberá ser respondida en un solo archivo con el número de la pregunta.
  - p1.cpp
  - p2.cpp
- Deberás subir estos archivos directamente a [www.gradescope.com](http://www.gradescope.com), uno en cada ejercicio. También puedes crear un .zip

### Competencias:

- Para los alumnos de la carrera de Ciencia de la Computación
  - Aplicar conocimientos de computación y de matemáticas apropiadas para la disciplina. (Evaluar)
  - Analizar problemas e identificar y definir los requerimientos computacionales apropiados para su solución.(Usar)
  - Utilizar técnicas y herramientas actuales necesarias para la práctica de la computación. (Usar)
- Para los alumnos de las carreras de Ingeniería
  - Capacidad de aplicar conocimientos de matemáticas (nivel 3)
  - Capacidad de aplicar conocimientos de ingeniería(nivel 2)
  - Capacidad para diseñar un sistema, un componente o un proceso para satisfacer las necesidades deseadas dentro de restricciones realistas (nivel 2)

## Calificación:

Tabla de puntos (sólo para uso del professor)

Question	Points	Score
1	10	
2	10	
Total:	20	

1. (10 points) **Pregunta 1**

Crear la clase plantilla **MiLista**, en la cual se pueda almacenar datos. Para ello, el programa debe cumplir con los siguientes requisitos:

- En el constructor se debe ingresar los elementos con los cuales se inicializará el objeto **MiLista**.
- Realizar la sobrecarga del operador '<<' para visualizar los objetos **MiLista**.
- Otra sobrecarga del operador '<<' para agregar elementos al objeto **MiLista**.
- Realizar la sobrecarga del operador '+' para retornar la suma de dos objetos **MiLista**.

El ejercicio debe ser validado con el siguiente bloque de código:

Listing 1: Ejemplo 1

```
int main(){
    MiLista<> lista1;
    MiLista<int> lista2(1);
    MiLista<float> lista3(2.1,3.2,0.4);

    cout << lista1 << endl;
    cout << lista2 << endl;
    cout << lista3 << endl;

    lista1<<2<<5;
    lista2<<9;
    cout << lista1 << endl;
    cout << lista2 << endl;

    MiLista<int> lista4 = lista1 + lista2;
    cout << lista4 << endl;
}
```

Para lo cual, el resultado del código debe ser el siguiente:

Listing 2: Resultado 1

```
[ ]
[ 1 ]
[ 2.1 3.2 0.4 ]
[ 2 5 ]
[ 1 9 ]
[ 2 5 1 9 ]
```

La rúbrica para esta pregunta es:

Criterio	Excelente	Adecuado	Mínimo	Insuficiente
Clases y Objetos	Desarrolla las clases con todos los atributos y métodos requeridos, se realiza un adecuado encapsulamiento, se utiliza adecuadamente los constructores que permiten copiar y asignar objetos, se realiza las sobrecargas adecuadas, se realiza un uso adecuado de la memoria dinámica (5pts)	Desarrolla las clases con todos los atributos y métodos requeridos, se realiza un adecuado encapsulamiento, se realiza las sobrecargas adecuadas, se realiza un uso adecuado de la memoria (4pts)	Desarrolla las clases con todos los atributos y métodos requeridos, se realiza las sobrecargas adecuadas, errores en el funcionamiento pasa algunas pruebas (2pts).	Contiene errores que no hace que funcione el programa (1pts)
Template de Funciones	Buen nivel de abstracción, uso adecuado de los diferentes parámetros de templates, selección correcta del tipo de template, selección correcta de la especialización o recursividad de templates. (5pts)	Buen nivel de abstracción, uso adecuado de los diferentes parámetros de templates. Selección correcta del tipo de template, uso de especializaciones o recursividad. (4pts)	Buen nivel de abstracción, uso adecuado de los diferentes parámetros de templates. selección correcta del tipo de template, errores en el funcionamiento pasa algunas pruebas. (2pts).	No se hizo un uso adecuado de los templates. (1pts)

2. (10 points) **Pregunta 2**

Implemente la clase plantilla **MatVector** para representar un vector. Esta clase debe tener las siguientes especificaciones:

- En el constructor se debe ingresar el tamaño del vector y se debe rellenar con valores aleatorios entre 0 y 9 (enteros o decimales). Utilice **arrays dinámicos** para la creación del vector.
- Realizar la sobrecarga del operador '<<' para visualizar objetos **MatVector**.
- Implementar la sobrecarga al operador '[' para asignar y obtener un elemento del vector.
- Implementar la sobrecarga al operador '\*' para realizar el productor escalar de dos vectores.

El ejercicio debe ser validado con el siguiente bloque de código:

Listing 3: Ejemplo 2

```
int main(){
    MatVector<> v1(4);
    MatVector<float> v2(4);

    cout << v1[2] << endl;
    cout << v1 << endl;

    cout << v2 << endl;
    v2[3] = 7.1;
    cout << v2 << endl;

    cout << (v1 * v2) << endl;
}
```

Para lo cual, el resultado del código debería ser similar–no necesariamente igual–al siguiente:

Listing 4: Resultado 2

```
5
[ 4 8 5 1 ]
[ 5.9 7.2 1.2 5.6 ]
[ 5.9 7.2 1.2 7.1 ]
94.3
```

La rúbrica para esta pregunta es:

Criterio	Excelente	Adecuado	Mínimo	Insuficiente
Clases y Objetos	Desarrolla las clases con todos los atributos y métodos requeridos, se realiza un adecuado encapsulamiento, se utiliza adecuadamente los constructores que permiten copiar y asignar objetos, se realiza las sobrecargas adecuadas, se realiza un uso adecuado de la memoria dinámica (5pts)	Desarrolla las clases con todos los atributos y métodos requeridos, se realiza un adecuado encapsulamiento, se realiza las sobrecargas adecuadas, se realiza un uso adecuado de la memoria (4pts)	Desarrolla las clases con todos los atributos y métodos requeridos, se realiza las sobrecargas adecuadas, errores en el funcionamiento pasa algunas pruebas (2pts).	Contiene errores que no hace que funcione el programa (1pts)
Template de Funciones	Buen nivel de abstracción, uso adecuado de los diferentes parámetros de templates, selección correcta del tipo de template, selección correcta de la especialización o recursividad de templates. (5pts)	Buen nivel de abstracción, uso adecuado de los diferentes parámetros de templates. Selección correcta del tipo de template, uso de especializaciones o recursividad. (4pts)	Buen nivel de abstracción, uso adecuado de los diferentes parámetros de templates. selección correcta del tipo de template, errores en el funcionamiento pasa algunas pruebas. (2pts).	No se hizo un uso adecuado de los templates. (1pts)