



Pontificia Universidad Javeriana
Departamento de Ingeniería de Sistemas
Estructuras de Datos, 2023-30
Parcial 1 - 16 de agosto de 2023

Reglas y recomendaciones

Durante el parcial, deben observarse las siguientes reglas:

1. El parcial se debe desarrollar en el computador, escribiendo las respuestas en los campos designados dentro del cuestionario de BrightSpace.
2. Archivos adicionales se aceptarán sólo en el caso del diseño, de acuerdo a las instrucciones que se encuentran en el enunciado del parcial. Guarde regularmente su trabajo (para evitar posibles pérdidas de información), con los nombres de archivo `par1_apellido1_apellido2_nombre1_nombre2.txt` (para el diseño de los TADs) y `par1_apellido1_apellido2_nombre1_nombre2_dis.pdf` (para el diagrama de relación entre TADs). Tenga en cuenta que los nombres y apellidos van todos en minúsculas, los espacios se reemplazan por guiones bajos y no se utilizan ni tildes ni 'ñ'. Estos archivos deben enviarse únicamente dentro de la pregunta correspondiente en el cuestionario de BrightSpace.
3. El parcial tiene una duración de dos horas, contadas a partir del inicio normal de la clase (9:00a.m.). El cuestionario en BrightSpace está programado para cerrarse al terminar las dos horas (11:00a.m.), por lo que no enviarlo dentro de ese lapso de tiempo se considerará como parcial no presentado y tendrá una calificación de 0.0.
4. El parcial es estrictamente individual y se debe desarrollar únicamente en la sala de computadores del día.
5. Se recomienda no utilizar compiladores, intérpretes de lenguajes de programación o entornos de desarrollo de cualquier tipo.
6. Puede utilizar sus apuntes, libros, e Internet para obtener la información que necesite para el parcial. Puede utilizar también herramientas de inteligencia artificial (tipo ChatGPT), bajo su propio riesgo y corroborando la información obtenida.
7. Sin embargo, está absolutamente prohibido comunicarse con cualquier otro ser humano para obtener información sobre el parcial, a través de cualquier medio (conversación directa, Teams, Skype, Zoom, Gtalk, Whatsapp, etcétera). También está totalmente prohibido copiar o transcribir texto o código de enunciados y respuestas de parciales anteriores, o de diseños e implementaciones que no sean propias.
8. Los celulares deben permanecer apagados, y no se debe enviar ni recibir ningún mensaje de texto.
9. La única excepción a lo anterior son los profesores y monitores de la asignatura, quienes sólo responderán consultas respecto a la claridad de las preguntas del parcial y no responderán consultas sobre la materia.
10. Si el estudiante incumple con cualquiera de las reglas, será evaluado con nota 0.0
11. **RECUERDE QUE DISEÑAR NO IMPLICA IMPLEMENTAR.**

1 (15%) Análisis de código

Considere la siguiente función (se garantiza que los argumentos contienen la cantidad de elementos necesarios para la correcta ejecución de la función):

```
typedef std::vector< string > TVector;
typedef std::list< string > TList;
typedef std::list< TList > TList2;
void f ( TList2& l, TVector v[] ) {

    for ( TList2::iterator it = l.begin( ); it != l.end( ); it++ ) {
        for ( TList::iterator it2 = it->begin( ); it2 != it->end( ); it2++ ) {
            if ( (*it2) == string("si") ) {
                v[0].push_back(*it2);
            } else if ( (*it2) == string("no") ){
                v[1].push_back(*it2)
            }
        }
    }
}
```

1.1 (7%) ¿Qué hace «func»? (describala en 15 palabras o menos)

1.2 (8%) ¿Cuál es el orden de complejidad de «func»? Justifique brevemente su respuesta.

2 (21%) Selección múltiple con única respuesta

2.1 (7%) Juanito toma una secuencia de números enteros desde 1 hasta n e inserta la primera mitad en una cola y la segunda mitad en una pila. Luego, llena una lista extrayendo primero los elementos de la pila, y luego los elementos de la cola. ¿Cómo quedan los elementos en la lista final?

1. Los primeros $n/2$ en orden inverso, luego los segundos $n/2$ en orden inverso.
2. Los segundos $n/2$ en orden inverso, luego los primeros $n/2$ en orden.
3. Los primeros $n/2$ en orden inverso, luego los segundos $n/2$ en orden.
4. Los segundos $n/2$ en orden inverso, luego los primeros $n/2$ en orden inverso.

2.2 (7%) Para cualquier contenedor secuencial estándar de la STL, el iterador `reverse_iterator` permite:

1. Recorrido principal hacia atrás, sólo lectura.
2. Recorrido principal hacia adelante, lectura/escritura.
3. Recorrido principal hacia atrás, lectura/escritura.
4. Recorrido principal hacia adelante, sólo lectura.

2.3 (7%) Para realizar un cálculo se cuenta con un algoritmo dividido en dos bloques secuenciales de instrucciones. El primer bloque de instrucciones tiene complejidad $O(n^2)$, mientras que el segundo bloque de instrucciones tiene complejidad $O(\log_{10} n)$. ¿Cuál es la complejidad del algoritmo completo?

1. $O(n^2 \log_{10} n)$
2. $O(n^2)$
3. $O(\log_{10} n)$
4. $O(n^2 + \log_{10} n)$

3 (64%) Diseño e Implementación de TADs

Las conferencias académicas o científicas (también conocidas por otros nombres como congresos, simposios, talleres o reuniones) son espacios dispuestos para que investigadores, integrantes de la industria, académicos y hasta el público en general puedan presentar y discutir sus trabajos de investigación, profundización, o temas de actualidad. Son espacios importantes en el quehacer de docentes, investigadores y científicos, pues facilitan el intercambio de información entre investigadores, incentivando la práctica de habilidades de presentación efectiva, la retroalimentación y comentarios entre pares académicos, la posibilidad de concretar oportunidades conjuntas de trabajo y colaboración, y la posibilidad de estar actualizado en la investigación actual en el área de interés.

Estas conferencias se programan para uno o varios días de actividades secuenciales, donde cada día puede dedicarse a un subtema específico dentro de la temática general de la conferencia. Durante cada día, pueden realizarse algunas de las siguientes actividades, organizadas en bloques y siempre en forma secuencial: presentaciones cortas (**short**) en bloques de 1 hora para presentar trabajos específicos, presentaciones principales (**main**) de 1 hora y media realizadas por expertos en el área, paneles de discusión (**panel**) alrededor de un tema entre varios expertos invitados, exposiciones de posters (**poster**) en las que trabajos específicos se presentan a través de un formato afiche, y talleres guiados (**workshop**) en los que los asistentes pueden profundizar en tecnologías o herramientas específicas. Actividades adicionales, como recesos (**pause**) y almuerzos o comidas (**lunch**) se intercalan entre éstas para poder dar a los asistentes tiempos de descanso. Cada una de estas actividades suele tener asociado un título o nombre, un tipo (uno entre **short**, **main**, **panel**, **poster**, **workshop**, **pause** o **lunch**) una hora de inicio y una duración estimada (en minutos). En el caso de las presentaciones cortas y las exposiciones de posters, en éstos se encolan por orden de llegada los trabajos aceptados para presentación en la conferencia, los cuales se identifican por un número asignado al momento del envío. A continuación se presenta un ejemplo de organización de actividades en una conferencia:

	Martes 13 de junio	Miércoles 14 de junio	Jueves 15 de junio
8:00am			
8:30am			
9:00am	Conferencia principal 1 9am – 60 minutos	Panel de discusión 8:30am - 90 minutos	Taller 2 8:30am – 90 minutos
9:30am			
10:00am	Receso 10am – 30 minutos	Receso 10am – 30 minutos	Receso 10am – 30 minutos
10:30am	Presentaciones cortas 1 10:30am – 90 minutos Trabajos: 4, 7, 23, 45	Presentaciones cortas 2 10:30am – 90 minutos Trabajos: 6, 15, 25, 36	Presentaciones cortas 4 10:30am – 90 minutos Trabajos: 8, 12, 27, 40
11:00am			
11:30am			
12:00m			
12:30pm	Almuerzo libre 12m – 120 minutos	Almuerzo libre 12m – 120 minutos	Almuerzo libre 12m – 120 minutos
1:00pm			
1:30pm			
2:00pm	Conferencia principal 2 2pm – 60 minutos	Presentaciones cortas 3 2pm – 90 minutos Trabajos: 9, 18, 21, 38	Conferencia principal 3 2pm – 60 minutos
2:30pm			
3:00pm	Receso 3pm – 30 minutos		Receso 3pm – 30 minutos
3:30pm		Receso 3:30pm – 30 minutos	Conferencia principal 4 3:30pm – 60 minutos
4:00pm	Taller 1 3:30pm – 90 minutos	Exposición de pósteres 4pm – 60 minutos Trabajos: 50, 53, 56, 59, 64, 67, 73, 78, 81, 85	
4:30pm			
5:00pm			

Para facilitar la programación de actividades en cualquier conferencia, se espera poder desarrollar una herramienta informática que organice por días los bloques de actividades y permita almacenar de ellos la información necesaria. Dentro de esta herramienta, se ha identificado inicialmente una operación muy importante: la que permite agregar una nueva actividad a la programación de la conferencia. Esta

operación debe recibir los datos de la actividad (título, tipo, hora de inicio, duración, y trabajos aceptados, cuando aplica) y el día en el que se espera programarla, y con esta información valida inicialmente si no se cruza con algo más ya programado, antes de realizar el ingreso de la actividad en la programación de la conferencia con toda su información adecuadamente organizada.

3.1 (22%) Diseño

Diseñe el sistema y el (los) TAD(s) solicitado(s). Utilice la plantilla de especificación de TADs vista en clase para el diseño. Recuerde que diseñar es un proceso previo a la implementación, por lo que no debería contener ninguna referencia a lenguajes de programación (es decir, si escribe encabezados o código fuente, el punto no será evaluado y tendrá una calificación de cero). Para simplicidad del diseño, no es necesario incluir los métodos obtener y fijar (get/set) del estado de cada TAD.

3.2 (10%) Diagrama de relación

El diseño incluye el diagrama de relación entre TADs, cuando se definen dos o más TADs en el punto anterior. En ese caso, adjúntelo en formato PDF, JPG o PNG como parte de su entrega.

3.3 (16%) Diseño: Agregar actividad

Dado el (los) TAD(s) ya diseñado(s), diseñe la operación que permite agregar una nueva actividad a la programación de la conferencia. El diseño deberá especificar, en lenguaje natural, los pasos necesarios para agregar una actividad, teniendo en cuenta los datos y operaciones ya definidos en el (los) TAD(s).

3.4 (16%) Implementación: Agregar actividad

Dado el (los) TAD(s) ya diseñado(s), y el diseño de la operación escrito en el punto anterior, escriba la implementación en C++ del algoritmo que permite agregar una nueva actividad a la programación de la conferencia. La implementación deberá tener en cuenta:

- la definición apropiada de los prototipos de los métodos/funciones (i.e. recibir/retornar los datos suficientes y necesarios para su correcta ejecución),
- el NO uso de salidas/entradas por pantalla/teclado (i.e. paso/retorno correcto de valores y/o objetos),
- la coherencia y el correcto uso del diseño definido en los puntos anteriores, y
- la escritura de todo el código que pueda llegar a necesitar que no esté incluido en la STL.