



Reglas y recomendaciones

Durante el parcial, deben observarse las siguientes reglas:

1. El parcial se debe desarrollar en el computador, escribiendo las respuestas en un archivo de texto plano (TXT). Guarde regularmente su trabajo (para evitar posibles pérdidas de información), con el nombre de archivo
`par1_apellido1_apellido2_nombre1_nombre2.txt`
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'. El archivo de respuestas debe enviarse únicamente a través de la actividad correspondiente en UVirtual.
2. Archivos adicionales se aceptarán sólo en el caso del diseño, de acuerdo a las instrucciones que se encuentran en el enunciado del parcial. El archivo debe guardarse como
`par1_apellido1_apellido2_nombre1_nombre2_dis.pdf`
y enviarse junto con el archivo de respuestas a través de la actividad correspondiente en UVirtual.
3. El parcial tiene una duración de dos horas, contadas a partir del inicio normal de la clase. La actividad en UVirtual está programada para cerrarse al terminar las dos horas, por lo que cualquier archivo no entregado dentro de ese lapso 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. Sin embargo, está absolutamente prohibido comunicarse con cualquier otro ser humano para obtener información sobre el parcial, a través de cualquier medio (conversación directa, MSN, Skype, gtalk, whatsapp, blackberry messenger, etcétera).
7. Los celulares deben permanecer apagados, y no se debe enviar ni recibir ningún mensaje de texto.
8. 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.
9. Si el estudiante incumple con cualquiera de las reglas, será evaluado con nota 0.0
10. **Recuerde que diseñar no implica implementar.**

“La suerte favorece a las mentes más preparadas.”

– Louis Pasteur

1 (15%) Análisis de código

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

```
typedef std::vector<int> TVector;
typedef std::list<string> TList;
typedef std::list<TList> TList2;
void func(TList2& l, TVector& v) {
    v.resize(l.size());    // Esta línea es  $O(1)$ 
    for(TVector::iterator it = v.begin(); it != v.end(); it++) {
        (*it) = 0;
    }
    TVector::iterator itv = v.begin();
    for (TList2::iterator it = l.begin(); it != l.end(); it++) {
        for (TList::iterator it2 = it->begin(); it2 != it->end(); it2++) {
            if ((*it2) == string("hola")) {
                (*itv)++;
            }
        }
        itv++;
    }
}
```

1.1 ¿Qué hace «func»? (describala en 12 palabras o menos)

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

2 (20%) Falso o verdadero

2.1 Para realizar la búsqueda de un elemento en cualquiera de las estructuras lineales (lista, pila, cola) es suficiente con utilizar un elemento cursor o ventana que se vaya moviendo sobre la estructura para revisar cada elemento.

Verdadero Falso

2.2 Teniendo en cuenta el orden de complejidad de la búsqueda binaria, un algoritmo que aplique esta búsqueda para encontrar un número entero en un conjunto de n arreglos de enteros diferentes tiene complejidad $O(n \log_2 n)$.

Verdadero Falso

3 (65%) Diseño e Implementación de TADs

El auge de las comunicaciones móviles ha generado una gran demanda de servicios por parte de las empresas de telefonía celular. De esta forma, la empresa *LlamaYa* ha inaugurado recientemente un gran Centro de Atención al Usuario (CAU) para poder atender adecuadamente a todos sus suscriptores, y también a los nuevos usuarios que se quieran vincular. Este CAU tiene un conjunto de asesores, los cuales se encargan de realizar la atención personalizada a cada usuario. El usuario a su llegada solicita un turno, y espera que sea llamado por el asesor para realizar su operación.

El CAU ofrece atención a los usuarios en 3 temas fundamentales: ventas, servicio técnico y reclamos. De esta forma, al llegar al CAU cada usuario tiene asociado un número de identificación personal (único), un indicador de prioridad (si es una persona de la tercera edad, con alguna discapacidad o está en embarazo, por ejemplo), un tema sobre el cual va a consultar, representado por dos letras (VE, ST, RE), y un número de turno o número consecutivo de espera para el tema escogido. Por otro lado, cada asesor tiene asociado un número de identificación personal (único), un estado actual (que puede ser ocupado, disponible o fuera de servicio), un tema que puede atender (cada asesor se especializa en atender sólo un tipo de solicitudes), y un registro de los turnos que ha atendido hasta el momento.

Para implementar toda esta dinámica en el nuevo CAU, la empresa *LlamaYa* requiere el apoyo de un sistema de información que se encargue de la asignación de los turnos a los usuarios y de la asignación de usuarios a los asesores para realizar la atención. Además, el sistema de información debería permitir extraer algunas estadísticas u otra información relevante sobre el proceso de atención de usuarios. De esta forma, se han identificado inicialmente tres grandes operaciones que el sistema debería ofrecer para el nuevo CAU:

1. Asignar un turno. Dado un nuevo usuario que llega (con su identificación, prioridad y tema de consulta), el sistema debe generar un número de turno para el usuario, y debe almacenar la información del usuario en una estructura adecuada que permita la posterior atención bajo el esquema: primero que llega, primero en ser atendido. Si el usuario tiene alguna prioridad, debe ubicarse en la estructura de forma que sea el próximo (primero) en ser atendido.
2. Atender un usuario. El sistema debe identificar el tema con mayor cantidad de usuarios en espera para realizar ahí la atención. De esta forma, dentro del conjunto de asesores del CAU, el sistema debe encontrar el primer asesor que esté disponible y que además pueda atender el tema identificado anteriormente. La información del usuario debe retirarse de la estructura de atención y su número de turno debe almacenarse en el registro de atendidos del asesor seleccionado.
3. Conteo de usuarios atendidos. Para identificar la cantidad de usuarios atendidos hasta el momento en el CAU, el sistema debe revisar los registros de usuarios atendidos de cada asesor del CAU.

Se le pide entonces diseñar e implementar (en C++) los componentes ya descritos del sistema de apoyo a la empresa de telefonía celular *LlamaYa*.

3.1 (20%) Diseño

Diseñe el sistema y el (los) TAD(s) solicitado(s). Recuerde que diseñar es un proceso previo a la implementación, por lo que no debería contener ninguna referencia a lenguajes de programación (código fuente). Para simplicidad del diseño, no es necesario incluir los métodos obtener y fijar (*get / set*) del estado de cada TAD. Para el diagrama de relación entre TADs, anéxelo en formato PDF como parte de su entrega.

Algoritmos

De acuerdo al diseño realizado en el punto anterior, escriba la implementación de los siguientes algoritmos en C++. Estas implementaciones deberán 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),
- el correcto uso del (de los) TAD(s) diseñado(s) en el punto anterior, y
- la escritura de todo el código que pueda llegar a necesitar que no esté incluido en la STL (excepto los métodos *get* / *set* del estado de cada TAD).

3.2 (15%) Asignar un turno.

3.3 (20%) Atender un usuario.

3.4 (10%) Conteo de usuarios atendidos.