

# Programación Avanzada

## PRIMER PARCIAL 2014

Por favor siga las siguientes indicaciones:

- Escriba con lápiz.
- Escriba las hojas de un solo lado
- Escriba su nombre y número de documento en todas las hojas que entregue.
- Numere las hojas e indique el total de hojas en la primera de ellas.
- Recuerde entregar su numero de parcial junto al parcial

### Problema 1 (25 puntos)

Se está desarrollando un software para gestionar el envío de mensajes (mail) de un usuario y en este proceso se ha relevado lo siguiente:

El sistema apunta a soportar las funcionalidades clásicas de envío de mensajes. En este sentido, cada mensaje tendrá un título y un texto, así como un número único asignado por el sistema. Además, cada mensaje será enviado a un conjunto de destinatarios (al menos uno) de los que se conocerá su dirección electrónica. A esta dirección serán enviados los mensajes y al mismo tiempo servirá para identificar a los destinatarios. No todos los destinatarios recibirán el mensaje de la misma forma. En particular, existen tres tipos de recepción de un mensaje: normal (TO), con copia (CC), con copia oculta (BCC). Por ende, para cada mensaje se desea conocer de qué forma recibe el mensaje cada destinatario (un destinatario podrá recibir cada mensaje sólo de una forma).

Por otro lado, el sistema apunta a incluir algunas funcionalidades que faciliten el almacenamiento y búsqueda de los mensajes. Para ello se desea poder crear carpetas en las cuales se puedan almacenar los diferentes mensajes que se envían. Cada carpeta será identificada por un nombre y particularmente existirá una carpeta (de nombre ENVIADOS) en la que se guardarán por defecto todos los mensajes enviados. El sistema tendrá la particularidad de que un mensaje podrá ser almacenado en más de una carpeta. Adicionalmente, se podrán definir carpetas cuyo contenido sea propiedad de un destinatario, es decir, que contengan mensajes enviados (de cualquier manera) a un destinatario particular. Estas carpetas las definirá el usuario en la medida de lo necesario. Una última funcionalidad permitirá determinar la prioridad de un mensaje y controlar si se recibió alguna respuesta al mismo. Para ello es necesario determinar cuándo un mensaje es prioritario, y en este caso, asociar al mismo la fecha de envío.

**a) Se pide:** Modelar la realidad planteada mediante un **Diagrama de Modelo de Dominio UML** y expresar todas las restricciones del modelo en lenguaje natural.

Como parte del relevamiento se especificó el siguiente caso de uso:

**Nombre:** Envío de Mensaje

**Actores:** Usuario

**Sinopsis:** El caso de uso comienza cuando el usuario desea enviar un mensaje. Para ello el usuario crea un nuevo mensaje indicando el título del mismo y el texto que contiene, así como indica si el mensaje es prioritario o no. Ante esto, el sistema registra el mensaje asignándole un número que lo identifica. Luego, el usuario ingresa el conjunto de destinatarios que recibirán el mensaje, indicando para cada uno el tipo de recepción del mensaje. El sistema registra la información y controla que haya al menos un destinatario para el envío del mensaje.

Finalmente, el usuario envía el mensaje y el sistema almacena el mismo en la carpeta por defecto, así como en las propias de los destinatarios, en caso de que existan.

**b) Se pide:**

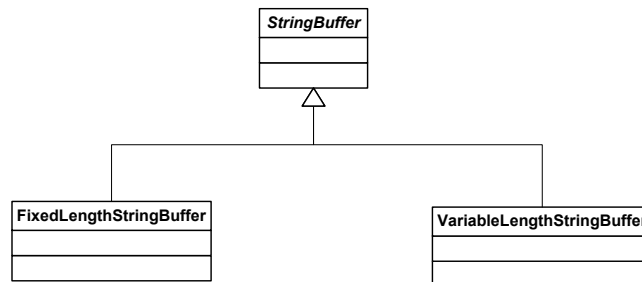
- i. Realizar el **Diagrama de Secuencia del Sistema (DSS)** correspondiente al caso de uso.
- ii. Expresar las pre- y post-condiciones, en lenguaje natural, de los contratos correspondientes a las operaciones del DSS anterior, de acuerdo al modelo de dominio realizado en la parte a).

**c) Responda brevemente:**

- i. ¿Qué técnicas existen para la identificación de conceptos del Modelo de Dominio?
- ii. Defina “tipo asociativo” haciendo especial énfasis en las multiplicidades de éste.
- iii. ¿Qué especifica y cómo está estructurado un Contrato?

## Problema 2 (15 puntos)

- Implementar en C++ el datatype String, que represente cadenas de caracteres de largo variable.
- Agregar la sobrecarga de los siguientes operadores:
  - Concatenación (operador +).
  - Acceso a un caracter del String (operador []).
- Agregar la siguiente operacion:
  - *substring*, que retorna el substring que se encuentra entre dos posiciones dadas de la cadena (incluyendo los caracteres en dichas posiciones).
- Agregar manejo de excepciones de forma que:
  - Al recibir un parámetro inválido en una operación se lance la excepción "std::invalid\_argument".
- El datatype String implementado representa una cadena de caracteres de valor variable. Si se quiere modificar dicha cadena, se puede utilizar un StringBuffer. La clase StringBuffer posee varios métodos como ser inserción, reemplazo y borrado de caracteres del buffer. Existen varias formas de representar un StringBuffer. Dos posibles tipos de buffers son:
  - **FixedLengthStringBuffer**, especialización de StringBuffer de largo fijo. En los casos que las operaciones sobrepasan el largo del buffer, el mismo se trunca.
  - **VariableLengthStringBuffer**, especialización de StringBuffer de largo variable. Si las operaciones sobrepasan el largo del buffer, el mismo se redimensiona al largo necesario.



Implementar en C++ la clase StringBuffer (abstracta) y las dos especializaciones definidas, con las siguientes operaciones:

- *insert*, que permite insertar un String en el buffer a partir de una posición dada.
- *capacity*, que retorna el largo del buffer.