

Programación 4

PARCIAL FINAL EDICIÓN 2012

SOLUCIÓN

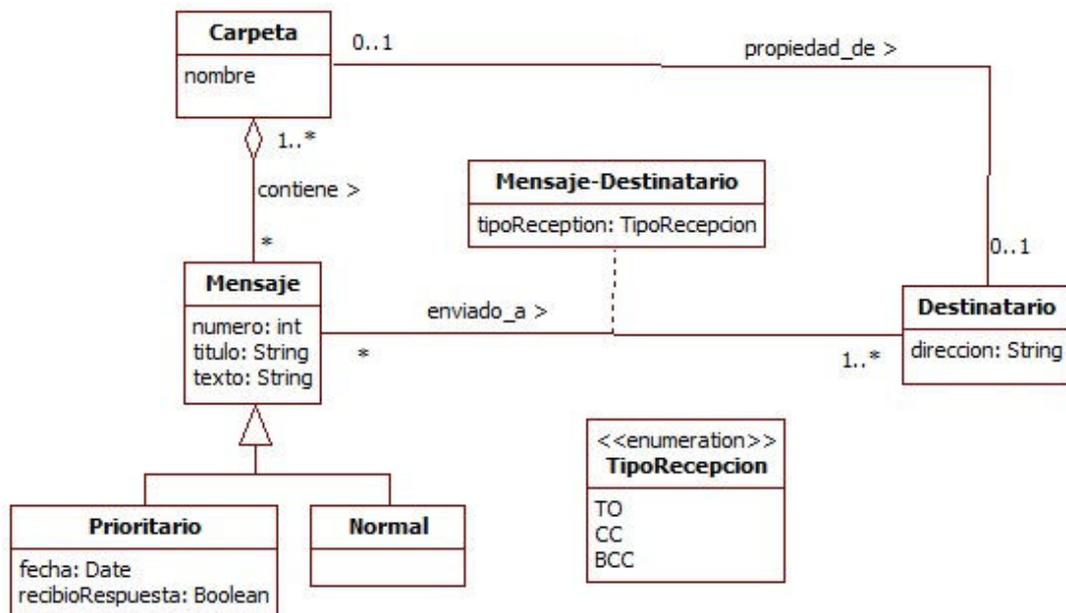
Por favor siga las siguientes indicaciones:

- 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 (35 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 [...]

- 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.

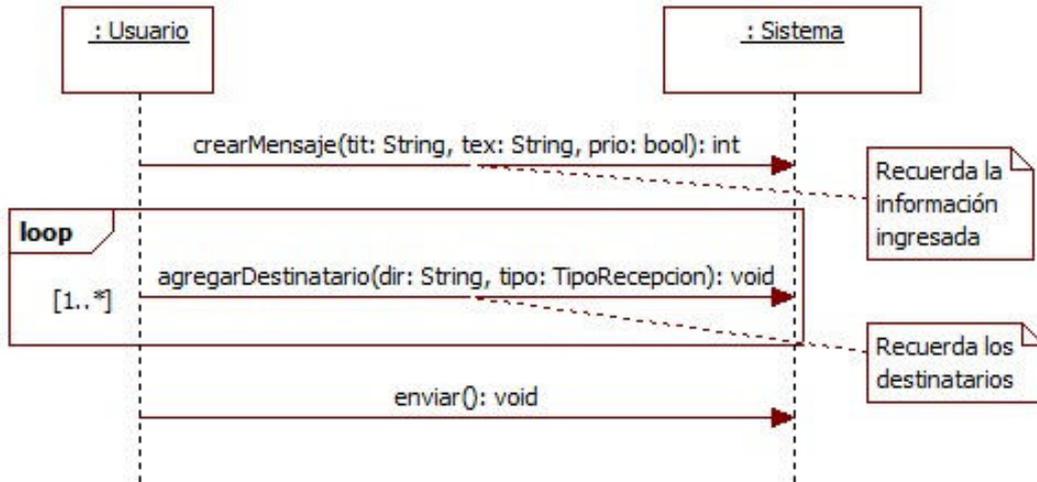


Restricciones:

- El atributo nombre identifica a la Carpeta
- El atributo direccion identifica al Destinatario
- El atributo numero identifica al Mensaje
- La Carpeta asociada a un Destinatario está compuesta por mensajes enviados a dicho Destinatario
- Existe una Carpeta de nombre ENVIADOS que contiene todas las instancias de Mensaje

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).

Operación	<code>crearMensaje(tit:string, tex:string, prio:bool):int</code>
Pre	
Post	<ul style="list-style-type: none"> • Si <code>prio=false</code>, entonces se crea una instancia de <code>Normal</code> con <code>titulo=tit</code>, <code>texto=tex</code> y <code>numero generado</code> • Si <code>prio=true</code>, entonces se crea una instancia de <code>Prioritario</code> con <code>titulo=tit</code>, <code>texto=tex</code>, <code>numero generado</code> y <code>fecha de hoy</code> • Se retorna el <code>numero generado</code> • Se recuerda la instancia de <code>Mensaje</code>

Operación	<code>agregarDestinatario(dir:string, tipo:TipoRecepcion)</code>
Pre	<ul style="list-style-type: none"> • Existe una instancia de <code>Mensaje</code> recordada • Existe una instancia de <code>Destinatario</code> con <code>direccion=dir</code> • No existe un link entre la instancia de <code>Mensaje</code> recordada y la de <code>Destinatario</code> con <code>direccion=dir</code>
Post	<ul style="list-style-type: none"> • Se crea una instancia de <code>Mensaje-Destinataro</code> con <code>tipoRecepcion=tipo</code> y se asocia con la instancia de <code>Mensaje</code> recordada y la de <code>Destinatario</code> con <code>direccion=dir</code>

Operación	<code>enviar()</code>
Pre	<ul style="list-style-type: none"> • Existe una instancia de <code>Mensaje</code> recordada • Existe al menos un link entre la instancia de <code>Mensaje</code> y alguna instancia de <code>Destinatario</code>
Post	<ul style="list-style-type: none"> • Se crea un link entre la instancia de <code>Mensaje</code> recordada y la de <code>Carpeta</code> con <code>nombre="ENVIADOS"</code> • Para cada instancia de <code>Destinatario</code> linkeada con la instancia de <code>Mensaje</code>, si existe un link entre la instancia de <code>Destinatario</code> y una instancia de <code>Carpeta</code>, entonces se crea un link entre esa instancia de <code>Carpeta</code> y la del <code>Mensaje</code> • Se deja de recordar información

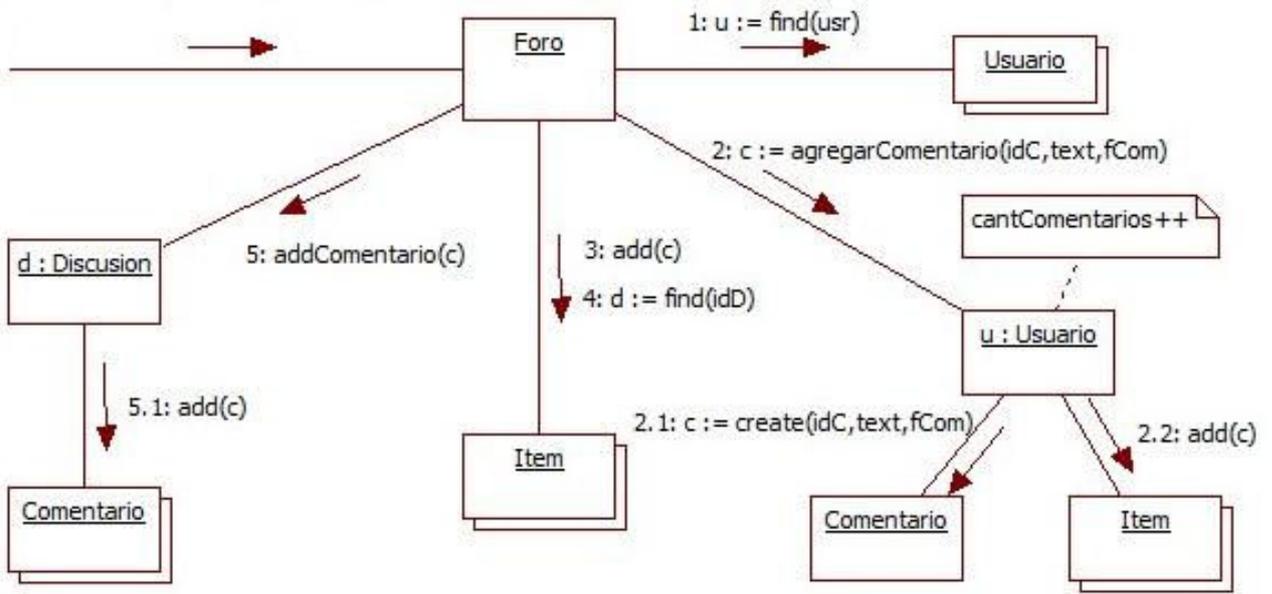
Problema 2 (30 puntos)

Se cuenta con el siguiente Diagrama de Clases de Diseño (DCD) parcial para la implementación de un foro de discusión donde los usuarios pueden crear discusiones y/o comentar sobre ellas.

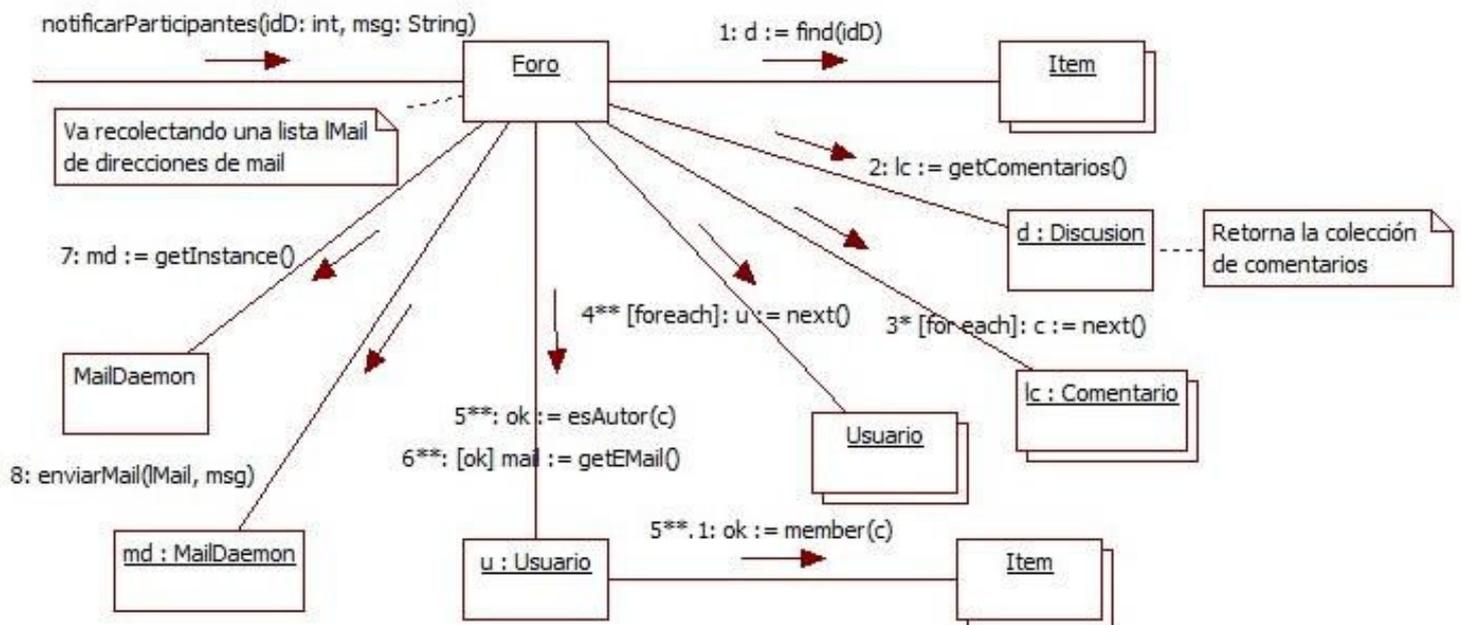
Se pide:

- i. Realizar el diagrama de comunicación de la siguiente operación.

agregarComentario(usr: String, idD: int, idC: int, text: String, fCom: Date)



- ii. Realizar el diagrama de comunicación de la operación `notificarParticipantes(idDiscusion:int, msg:String)`, ejecutada cada vez que se agrega un comentario en una discusión. Esta operación envía un email con el mensaje `msg` a todos los usuarios que hayan comentado en dicha discusión, haciendo uso de la operación `enviarMail (...)` de la clase `MailDaemon`.



Problema 3 (35 puntos)

Se desea implementar un tipo de objetos que represente el concepto de *sombra*. [...]

Se pide: implementar en C++

i. La clase Sombra.

```
class Sombra : public ICollectible {
private:
    ICollection *sombras;
public:
    Sombra();
    Sombra *nuevaSombra();
    void propagar();
    virtual Sombra *getCopia() = 0;
    virtual void copiarContenido() = 0;
}

Sombra()::Sombra() {
    this->sombras = NULL;
}

Sombra *Sombra::nuevaSombra() {
    if (this->sombras == NULL) {
        this->sombras = new List;
        this->sombras->add(this);
    }
    Sombra *nueva = this->getCopia();
    nueva->sombras = this->sombras;
    this->sombras->add(nueva);
    return nueva;
}

void Sombra::propagar() {
    if (sombras != NULL) {
        Sombra *s;
        IIterator *it = sombras->getIterator();
        while (it->hasNext()) {
            s = (Sombra *)it->next();
            this->copiarContenido(s);
        }
        delete it;
    }
}
```

ii. La clase Persona.

```
class Persona : public Sombra {
private:
    String nombre;
public:
    Sombra *getCopia();
    void copiarContenido(Sombra *);
    void setNombre(String);
}
```

```
Sombra *Persona::getCopia() {
    Persona *nueva = new Persona;
    nueva->nombre = this->nombre;
    return nueva;
}

void Persona::copiarContenido(Sombra *p) {
    (Persona *)p->nombre = this->nombre;
}

void Persona::setNombre(String nom)
{
    this->nombre = nom;
    this->propagar();
}
```