

Certamen Final 1er. Sem. 2004

Todas las preguntas tienen igual puntaje.

1.

a) Con sus palabras explique ¿qué es un diagrama de Secuencia?

Un diagrama de secuencia es un dibujo que muestra, para un escenario particular de un caso de uso, los eventos que actores externos generan, su orden, y los eventos generados al interior del sistema.

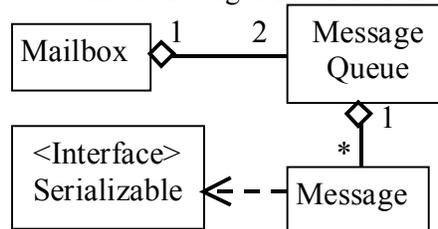
7pts.

b) ¿Qué es un Diagrama de Estados?

Un diagrama de estado ilustra los eventos de interés y estados de un objeto, y la transición de un estado a otro en reacción a un evento.

7pts.

c) ¿Qué lectura puede dar usted a cada una de las siguientes relaciones?



Mailbox/Message queue: Cada mailbox tiene dos Message Queue.

Message queue/Message: Cada Message Queue tiene cero o más Messages.

Message/Serializable: La clase Message implementa la Interfaz Serializable.

6pts.

2. Un fabricante de tarjetas de sonido se propone fabricar un nuevo modelo que incluye amplificación en la tarjeta. Los parlantes se podrán conectar directamente a ella sin requerir alimentación extra. La tarjeta de sonido tendrá tres registros que permitirán controlar encendido, volumen y balance. A usted le encargan desarrollar un programa para que el usuario pueda controlar estas variables. Se desea que su interfaz gráfica dibuje un parlante con los controles que le permitirán operarlo.

a) Invente y escriba dos casos de uso; uno será "Encender parlante" y el otro será "Aumentar volumen".

Escenario: Nombre "Encender parlante"

Actor: Usuario

Descripción: El usuario enciende el parlante para dejarlo operativo.

1. El usuario mueve el mouse a posición en la figura que tiene dibujado un interruptor, el cual muestra OFF.
2. El usuario presiona el interruptor.
3. El interruptor cambia a posición ON.
4. Se inicia reproducción según los valores prefijados para volumen y balance.

5pts.

Escenario: nombre "Aumentar volumen"

Actor: Usuario

Descripción: El usuario aumenta el volumen del parlante.

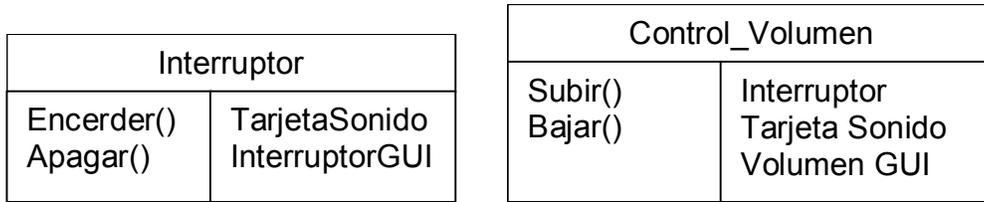
1. El usuario mueve el cursor a la posición donde se encuentra la perilla de volumen.
2. El usuario presiona el botón derecho del mouse.
3. Si el parlante está ON, la perilla gira aumentando el volumen. Si el parlante está OFF, no hay cambio.
4. A medida que la perilla gira, el aumento de volumen se refleja con un cambio en un indicador numérico del porcentaje respecto al máximo y el mayor sonido.

5pts.

b) Proponga dos tarjetas CRC para este sistema.

Hay varios objetos que intervienen en este sistema. Externos al sistema son el usuario, y la tarjeta de sonido. El sistema de software puede definir varias clases por ejemplo: Interruptor, control de volumen, y control de balance, cada uno de los cuales puede

tener su propia interfaz usuario; además se puede considerar un contenedor para todos los elementos de la interfaz (siguiendo el patrón modelo, vista, controlador). Dos tarjetas preliminares son:



5+5 pts.

3. Alguien propone la siguiente clase en C++ para almacenar la hora.

```
class Hora {
public:
    Hora(int h, int m, int s);
    Hora(Hora hms); // mejor habría sido Hora (const Hora &hms);
    ~Hora();
    friend ostream & operator <<( ostream & os,
        const Hora &t );
private:
    int * h, *m, *s; // horas, minutos, segundos
};
```

Implemente cada una de las funciones mostradas en esta declaración de la clase Hora.

```
Hora::Hora(int _h, int _m, int _s) {
    h = new int;
    m = new int;
    s = new int;
    *h= _h;
    *m= _m;
    *s= _s;
}

Hora::Hora(Hora hms) { // Hay otras formas de implementar esto.
    Hora(*(hms.h), *(hms.m), *(hms.s)); // hay otras formas OK.
}

Hora::~~Hora() {
    delete h;
    delete m;
    delete s;
}

ostream & operator <<( ostream & os, const Hora &t ){
    os << *(t.h) << ":" << *(t.m) << ":" << *(t.s);
    return os;
}
```

Nota: Cabe destacar que esta clase debería tener definido el constructor por defecto. También debería sobre cargar el operador asignación. Esto no fue pedido en la pregunta.

4.

- a) Haga una plantilla que permita calcular el promedio de los objetos contenidos en un **arreglo vector**. Como parámetro de entrada se pasa un vector (STL) con los objetos. Como salida se retorna un objeto del mismo tipo de los contenidos en el arreglo.

```
template <class T>
T promedio( const vector <T> &v) {
    T sum=0;
    for(int i=0; i<v.size(); i++)
        sum= sum+v[i];
    return sum/v.size();
}
```

- b) Se tiene un vector con horas de llegada de los alumnos a almorzar. ¿Cómo se debe modificar la clase presentada en la pregunta 3 para que este vector pueda usarse en una invocación a la función promedio de la parte a) ? Sólo muestre la declaración, no la implementación.

Los métodos que deben incluirse para que la template pueda ser usada están destacados. Nota: para un mejor uso de la clase otros métodos podrían incluirse, pero esto es lo mínimo para usar la plantilla.

```
class Hora {
public:
    Hora ();    // Necesario por T sum ..
    Hora(int h, int m, int s);
    Hora(Hora h);
    ~Hora();
    Hora operator=(int i); // Necesario por sum=0;
    Hora operator+(const Hora & hms); // Necesario por sum+v[i]
    Hora operator=(const Hora & hms); // Necesario por sum=sum+..
    Hora operator/(int n); // Necesario por sum/v.size()
    friend ostream & operator <<( ostream & os,
        const Hora &t );
private:
    int * h, *m, *s;
};
```

5. En la declaración de clase de la pregunta 3, se detecta que un usuario podría invocar al constructor con valores inapropiados, por ejemplo con segundos o minutos mayores a 59. Despreocúpese de las horas.
- a) Proponga una clase de excepción para indicar la presencia de valores segundos y/o minutos superiores a 59. En su estado el objeto lanzado debe almacenar los valores que generaron la excepción.

```
class ExcepcionRango {
public:
    ExcepcionRango() {
        msg="Se superó el rango 0..59";
    };
};
```

```
ExceptionRango(int m, int s, string m) {
    this->m=m;
    this->s=s;
    msg = m;
};
int getMinutos() {
    return m;
};
int getSegundos() {
    return s;
};
string getMensaje() {
    return msg;
};
private:
    int m,s;
    string msg;
};
```

b) Muestre la nueva versión del constructor que incorpora el lanzamiento de esta excepción.

```
Hora::Hora(int _h, int _m, int _s) {
    if ((_m > 59) || (_s > 59))
        throw ExceptionRango( _m, _s, "Minutos y/o segundos
superiores a 59");
    h = new int;
    m = new int;
    s = new int;
    *h= _h;
    *m= _m;
    *s= _s;
}
```