

APELLIDOS

NOMBRE

Nº Mat.

ASIGNATURA: SISTEMAS INFORMÁTICOS INDUSTRIALES

Calificación

CURSO 4º

GRUPO

Julio 2015

2. Problema de Algoritmia (5 puntos - 20 minutos)

En una planificación de un proyecto se dispone de la secuencia de 10 tareas T_i ordenadas parcialmente que aparece a continuación, y se necesita encontrar un orden total (el símbolo \mapsto indica precedencia).

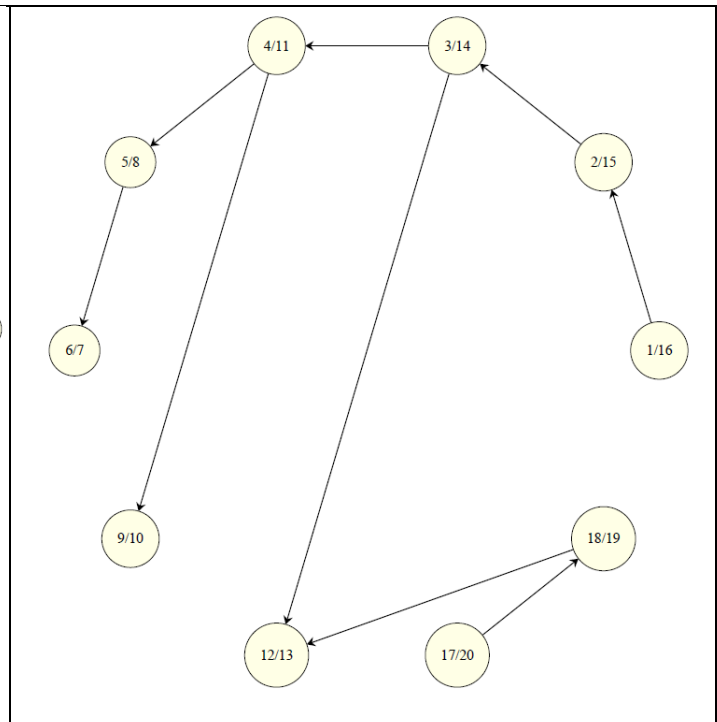
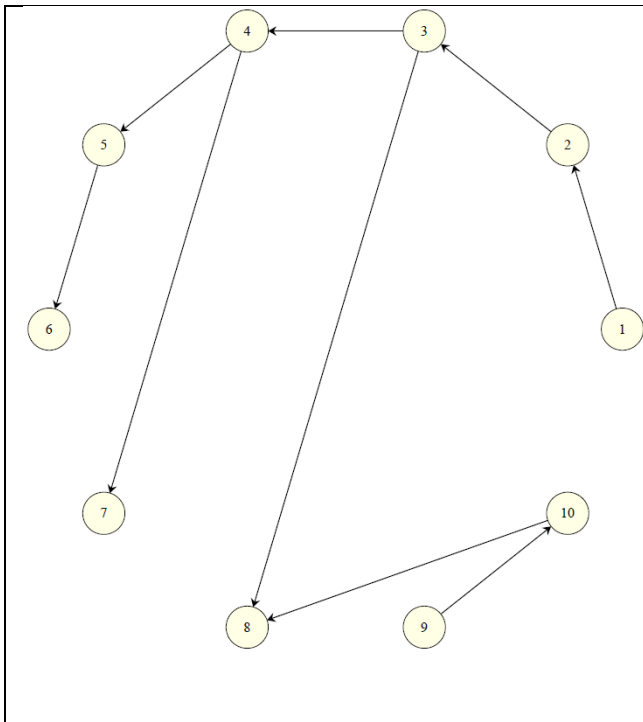
$$T = \{T_1, T_2, \dots, T_{10}\}$$

$$T_1 \mapsto T_2, T_2 \mapsto T_3, T_3 \mapsto T_4, T_4 \mapsto T_5, T_5 \mapsto T_6, T_4 \mapsto T_7, T_3 \mapsto T_8, T_9 \mapsto T_{10}, T_{10} \mapsto T_8$$

Se pide:

1. Grafo que representa las relaciones de orden parcial (emplee el índice de las tareas para la numeración de los vértices). ¿Qué tipo de grafo es? (1 punto)
2. Escriba un algoritmo que permita establecer el ordenamiento deseado recorriendo sistemáticamente el grafo del apartado anterior. (1.5 puntos)
3. Escriba los pasos de inicio y final de cada vértice tras la ejecución del algoritmo, suponiendo que se sigue el orden natural de vértices en cada ramificación, y determine el orden total en consecuencia. (2.5 puntos)

SOLUCION



CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

Cartagena99

The logo for Cartagena99 features the text 'Cartagena99' in a stylized, teal-colored font. The text is set against a light blue, irregular shape that resembles a splash or a stylized 'C'. Below the text, there is a horizontal orange bar with a slight gradient and a drop shadow effect.

**CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70**

**ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70**

APELLIDOS

NOMBRE

Nº Mat.

ASIGNATURA: SISTEMAS INFORMÁTICOS INDUSTRIALES

Calificación

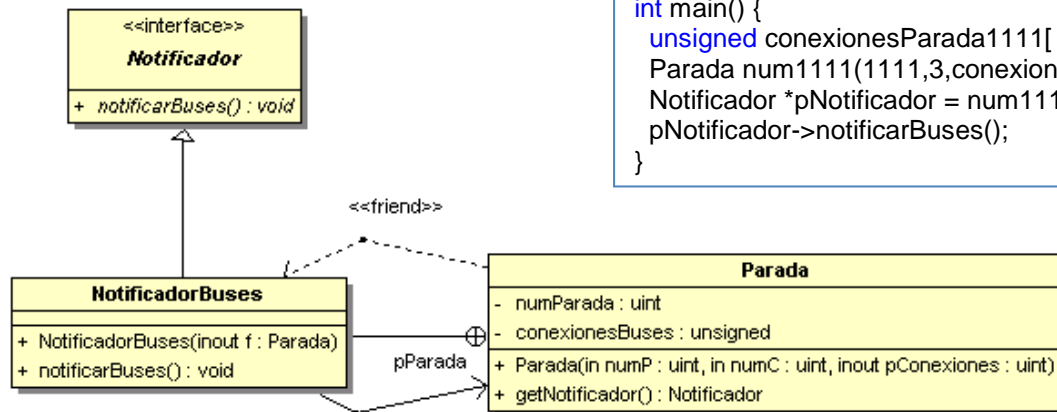
CURSO 4º

GRUPO

Julio 2015

3. Problema de Análisis y Diseño Orientado a Objetos (10 puntos - 40 minutos)

Dado el diagrama de clases de diseño y el código de test, así como el resultado de su ejecución, obtener la implementación en C++ del diseño propuesto (5 puntos).



```

int main() {
    unsigned conexionesParada1111[ ] ={60,148,78};
    Parada num1111(1111,3,conexionesParada1111);
    Notificador *pNotificador = num1111.getNotificador();
    pNotificador->notificarBuses();
}
    
```

```

d:\compartido\cplatero\docencia\SII\teoría\cap6\A
Parada 1111. Conexiones: 60 148 78
    
```

```

class Notificador{
public: virtual void notificarBuses() = 0;
};
class Parada {
    unsigned numParada;
    vector<unsigned> conexionesBuses;
public:
    class NotificadorBuses; // Clase interna:
    friend class Parada::NotificadorBuses;
    class NotificadorBuses : public Notificador {
        Parada* pParada;
    public:
        NotificadorBuses(Parada* f) : pParada(f){}
        void notificarBuses() {
            cout << "Parada " << pParada->numParada << ". Conexiones: ";
            for(unsigned i=0;i<pParada->conexionesBuses.size();i++)
                cout << pParada->conexionesBuses[i] << " ";
            cout << endl;
        }
    };
}
    
```

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70



Para el siguiente código se pide: a) Ingeniería inversa: Diagrama de clases. B) Diagrama de secuencia de la función `main()`. (5 puntos).

```
#include <iostream>
#include <string>
#include <cmath>
using namespace std;

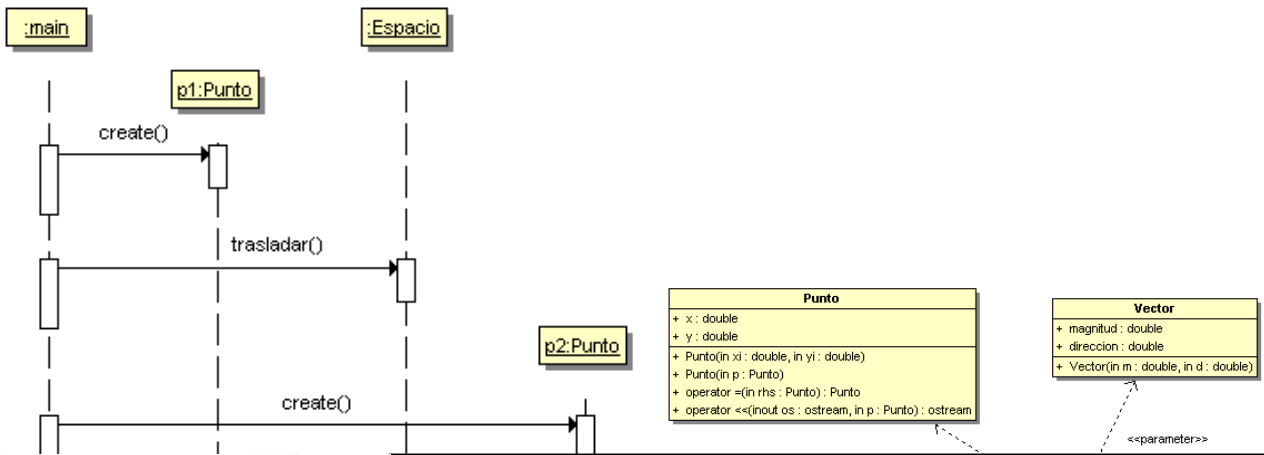
class Punto {
public:
    double x, y;
    Punto(double xi, double yi) : x(xi), y(yi) {}
    Punto(const Punto& p) : x(p.x), y(p.y) {}
    Punto& operator=(const Punto& rhs) {
        x = rhs.x;
        y = rhs.y;
        return *this;
    }
    friend ostream&
    operator<<(ostream& os, const Punto& p) {
        return os << "x=" << p.x << " y=" << p.y;
    }
};

class Vector {
public:
    double magnitud, direccion;
    Vector(int m, int d) : magnitud(m), direccion(d) {}
};

class Espacio {
public:
    static Punto trasladar(Punto p, Vector v) {
        p.x += (v.magnitud * cos(v.direccion));
        p.y += (v.magnitud * sin(v.direccion));
        return p;
    }
};

int main() {
    Punto p1(1, 2);
    Punto p2 = Espacio::trasladar(p1, Vector(3,
3.1416/3));
    cout << "p1: " << p1 << " p2: " << p2 << endl;

    return 0;
}
```



**CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70**

**ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP:689 45 44 70**

