UNIVERSIDAD POLITÉCNICA DE MADRID
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA Y DISEÑO INDSUTRIAL

Departamento de Ingeniería Eléctrica, Electrónica, Automática y Física Aplicada

APELLIDOS

NOMBRE Nº Mat.

ASIGNATURA: SISTEMAS INFORMÁTICOS INDUSTRIALES

CURSO 4º **GRUPO** **Julio 2015**

Calificación

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).

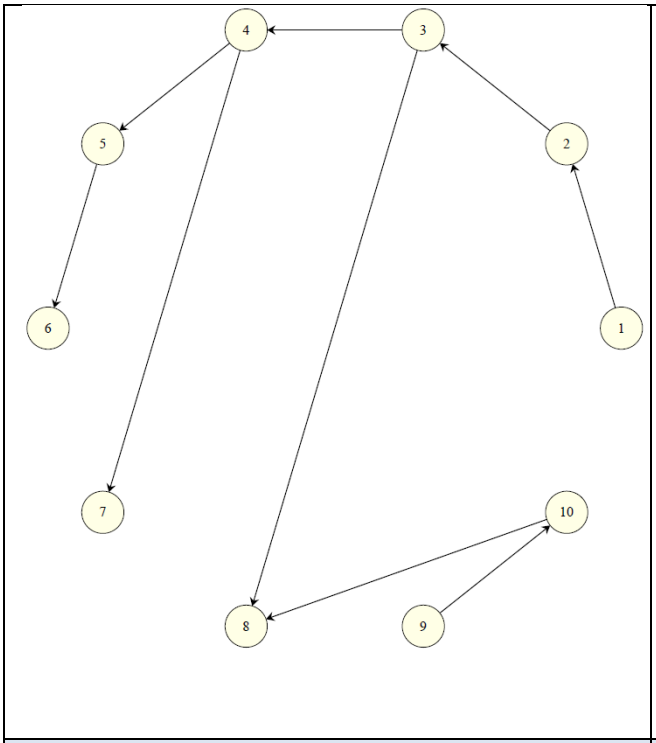
$$T = \{T_1, T_2, \cdots, T_{10}\}$$

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

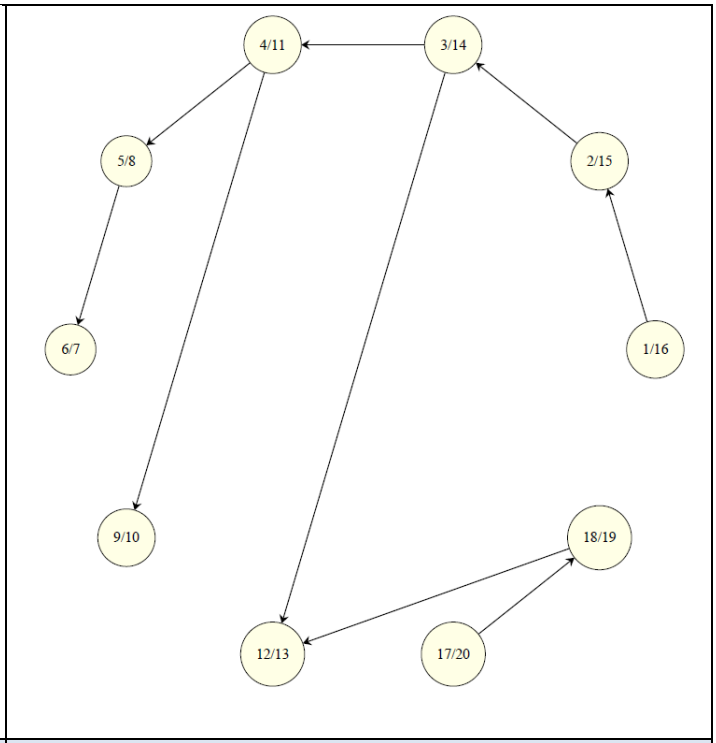
Se pide:

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



Grafo de ordenes parciales



Pasos de tiempo tras lanzar DFS

Grafo simple dirigido

Algoritmo: 1.Lanzar una búsqueda-primero-en-profundidad (DFS(y almacenar los pasos temporales 2.Ordenar los vértices por tiempo de finalización decreciente

ORDEN TOTAL: T9, T10, T1, T2, T3, T8, T4, T7, T5, T6

UNIVERSIDAD POLITÉCNICA DE MADRID
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA Y DISEÑO INDSUTRIAL

Departamento de Ingeniería Eléctrica, Electrónica, Automática y Física Aplicada

APELLIDOS

NOMBRE Nº Mat.

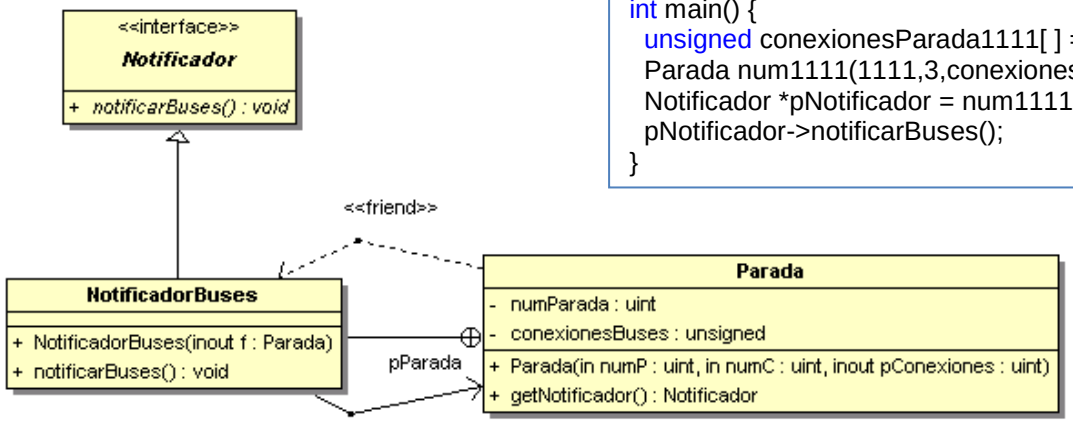
ASIGNATURA: SISTEMAS INFORMÁTICOS INDUSTRIALES

CURSO 4º **GRUPO** **Julio 2015**

Calificación

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).

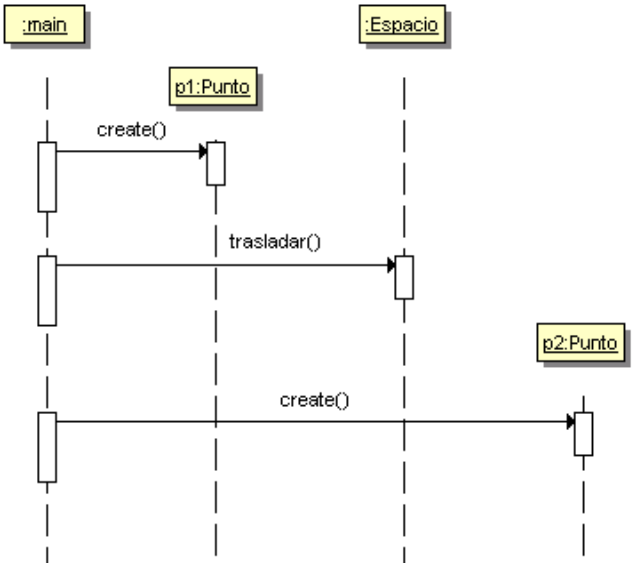


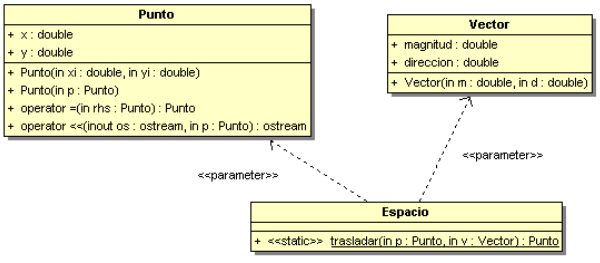

```
int main() {
  unsigned conexionesParada1111[ ] ={60,148,78};
  Parada num1111(1111,3,conexionesParada1111);
  Notificador *pNotificador = num1111.getNotificador();
  pNotificador->notificarBuses();
}
```


```
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;
           }
    } elNotificardorLineasBuses;
  Parada(unsigned nP, unsigned nC, unsigned *pConexiones ) : numParada(nP), elNotificardorLineasBuses(this) {
            for(unsigned i=0;i<nC;i++)
                    conexionesBuses.push_back(pConexiones[i]);
  }
  Notificador* getNotificador() {return &elNotificardorLineasBuses;}
};
```

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;
}
```





escuela técnica superior de ingeniería y diseño industrial
UNIVERSIDAD POLITÉCNICA DE MADRID
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA Y DISEÑO INDSUTRIAL
Departamento de Ingeniería Eléctrica, Electrónica, Automática y Física Aplicada

APELLIDOS

NOMBRE Nº Mat.

ASIGNATURA: SISTEMAS INFORMÁTICOS INDUSTRIALES

CURSO 4º **GRUPO** **Julio 2015**

Calificación

4. Problema de Sistemas Operativos (10 puntos - 30 minutos)

Referente al **capítulo de Procesos** el sistema operativo nos ofrece los siguientes servicios según la interfaz POSIX en lenguaje de programación C;

```
fork(), exec(), signal()
```

- **1º- (1 punto)** Elija un servicio de los anteriores, escriba el prototipo de la función (parámetros y valor devuelto) y explique para qué sirve.
- **2º- (1 punto)** Del servicio elegido codifique en C un **breve** ejemplo, diciendo qué hace.

Referente al **capítulo de Comunicación y Sincronismo** el sistema operativo nos ofrece los siguientes servicios según la interfaz POSIX en lenguaje de programación C;

```
sem_wait(), sem_post(), sem_open()
```

- **3º- (1 punto)** Elija un servicio de los anteriores, escriba el prototipo de la función (parámetros y valor devuelto) y explique para qué sirve.
- **4º- (1 punto)** Del servicio elegido codifique un **breve** ejemplo, diciendo qué hace.

Referente al **capítulo de Sistema de Ficheros** el sistema operativo nos ofrece los siguientes servicios según la interfaz POSIX en lenguaje de programación C;

```
read(), creat(), open()
```

- **5º- (1 punto)** Elija un servicio de los anteriores, escriba el prototipo de la función (parámetros y valor devuelto) y explique para qué sirve.
- **6º- (1 punto)** Del servicio elegido codifique un **breve** ejemplo, diciendo qué hace.

Referente al **capítulo de Distribuidos** el sistema operativo nos ofrece los siguientes servicios según la interfaz POSIX en lenguaje de programación C;

```
socket(), sendto(), accept()
```

- **7º- (1 punto)** Elija un servicio de los anteriores, escriba el prototipo de la función (parámetros y valor devuelto) y explique para qué sirve.
- **8º- (1 punto)** Del servicio elegido codifique un **breve** ejemplo, diciendo qué hace.

Referente al capítulo de Gestión de Memoria en la siguiente tabla se da a modo de ejemplo las características de la región de código de un proceso. Complete dicha la tabla con dos nuevas filas con las características de las siguientes regiones de un proceso:

- **9º- (1 punto)** Región de heap.
- **10º- (1 punto)** Región de pila.

Nombre de la región	**Permisos de la región**	**Procedencia de los datos**	**¿Se rellena de ceros?**	**¿Es una región dompartida o privada?**	**Posibles ubicaciones de las páginas de esta región**	**¿Puede crecer?**
Código	R, X	del fichero ejecutable	No	Compartida	Memoria o ejecutable	No
...						