

# WUOLAH



Indalecia

[www.wuolah.com/student/Indalecia](http://www.wuolah.com/student/Indalecia)



## extraordinaria2018.pdf

EXAMEN JULIO 2018 (EXTRAORDINARIA)



1º Metodología de la Programación



Grado en Ingeniería Informática



Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación  
Universidad de Granada



Escuela de **LÍDERES**

## Master en Marketing Digital

Rafael García Parrado  
Director de  
Marketing Digital



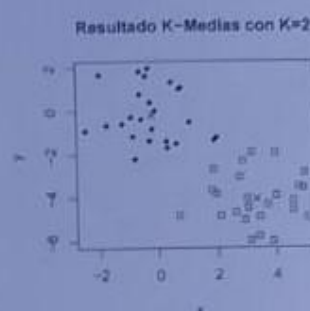
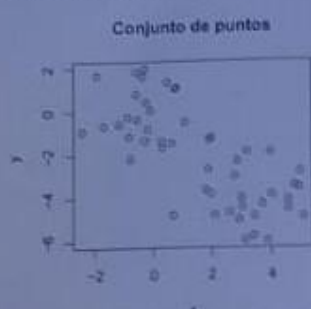


### Normas para la realización del examen:

Duración: 2.5 horas

- El único material permitido durante la realización del examen es un bolígrafo azul o negro.
- Debe disponer de un documento oficial que acredite su identidad a disposición del profesor.
- No olvide escribir su nombre completo y grupo en todos y cada uno de los folios que entregue.

En este examen se desea construir varias clases para trabajar con un algoritmo conocido como algoritmo de las K-medias. La entrada a este algoritmo es un conjunto de puntos y un número entero  $K$ . El algoritmo tiene como objetivo agrupar los puntos de entrada en  $K$  grupos (clústeres) disjuntos. Cada uno de los grupos estará formado por un subconjunto de los puntos originales y un punto que los represente (*centroide*), que es un punto que se calcula como el vector de medias de los puntos del grupo. Por ejemplo, en el gráfico de la derecha se muestra el resultado de aplicar el algoritmo al conjunto de puntos del gráfico de la izquierda, tomando  $K = 2$ . Se marcan con círculos rellenos, los puntos de uno de los grupos, y con cuadrados no rellenos los puntos del otro. También se marcan con una cruz, los centroides de cada uno de los dos grupos.



Para resolver el problema, se usarán los siguientes tipos de datos:

```
class Punto {
private:
    double x;
    double y;
public:
    // Interfaz de la clase
};

class Cluster {
private:
    Punto centroide; // centroide del grupo
    VectorPuntos puntos; // puntos del grupo (clúster)
public:
    // Interfaz de la clase
};

class VectorPuntos {
private:
    Punto *puntos; // array dinámico de puntos
    int nPuntos; // n° de puntos actualmente en el array dinámico
    int reservados; // capacidad del array dinámico
public:
    // Interfaz de la clase
};

class KMedias {
private:
    int K; // n° de grupos
    Cluster *clusters; // array dinámico de grupos
public:
    // Interfaz de la clase
};
```

Suponga que tiene ya implementados los siguientes métodos o funciones:

- Clase **Punto**
  - Constructor por defecto en la clase **Punto** que inicializa  $x$  e  $y$  al valor 0.0.
  - Métodos públicos de consulta `double getX(void)`, `double getY(void)` y `double getDistancia(const Punto& punto)`.
  - Métodos públicos de modificación `void setXY(double newX, double newY)`.
  - Operadores `>>` y `<<` (leen o escriben las coordenadas  $x$  e  $y$  separadas por un blanco).
- Clase **VectorPuntos**:
  - Métodos públicos de consulta `int size(void)` (devuelve el número de puntos).
  - Operadores `>>` y `<<` (leen o escriben el número de puntos y el conjunto de puntos).
- Clase **Cluster**:
  - Métodos públicos de consulta `int size(void)` (devuelve el número de puntos en el grupo), `Punto getCentroide(void)` (devuelve el centroide) y `Punto getPunto(int n)` (devuelve el punto  $n$ ).
  - Métodos públicos de modificación `void setCentroide(const Punto& punto)` (establece el valor del centroide) y `void add(const Punto& punto)` (añade un punto al grupo).
- Clase **KMedias**: destructor, constructor de copia y operador de asignación.
- Función `int uniforme(int minimo, int maximo)` que devuelve un número aleatorio entre mínimo y máximo.

**NOTA IMPORTANTE:** Para la realización de los ejercicios del examen, puede usar los métodos y funciones especificados anteriormente, así como los que usted vaya implementando. Si necesita usar otros métodos adicionales, deberá implementarlos en el ejercicio correspondiente.





**INEAF**  
BUSINESS SCHOOL

Formación  
**Práctica**

Contenidos  
**Actualizados**


Profesores  
**Especializados**

Es el momento  
**DE CRECER**

Referentes en formación  
Fiscal, Contable  
y Jurídica




# Master en Asesoría Fiscal de Empresas



**UNIVERSIDAD DE GRANADA**  
Departamento de Ciencias de la Computación e Inteligencia Artificial

Metodología de la Programación (2017/18)  
1º GII / GII-M / GII-ADE  
Conv. Extraordinaria - 6 de Julio de 2018  
Examen de teoría



---

◀ **Ejercicio 1** ▶ Métodos básicos de la clase

[1.5 puntos]

- (0.5 punto) Constructor por defecto de las clases VectorPuntos y KMedias. El constructor por defecto de VectorPuntos crea un vector de puntos vacío, pero con capacidad para guardar 10 puntos. El constructor por defecto de KMedias crea un array para 2 grupos.
- (0.75 puntos) Constructor de copia, operador de asignación y destructor de la clase VectorPuntos.
- (0.25 puntos) Razone si es necesario implementar el constructor por defecto, constructor de copia, operador de asignación y destructor en la clase Cluster, o si no es necesario implementar alguno (o ninguno) de estos métodos.

---

◀ **Ejercicio 2** ▶ Métodos de la clase

[1 punto]

- (0.75 puntos) Método add(punto) en la clase VectorPuntos para añadir un punto. Este método debe redimensionar automáticamente el array dinámico al doble de la capacidad que tenga actualmente cuando la capacidad del VectorPuntos esté agotada.
- (0.25 puntos) Método clearAndSetK(entero) en la clase KMedias que debe reconfigurar el objeto para contener el número de grupos deseado, destruyendo la información que contenga actualmente el objeto KMedias y reservándole la memoria dinámica necesaria.

---

◀ **Ejercicio 3** ▶ Sobrecarga de operadores

[1.75 puntos]

Implemente los siguientes operadores en la clase indicada:

- (0.5 puntos) Operador [] de la clase VectorPuntos para poder usarlo tanto en el lado derecho como en el lado izquierdo de una asignación.
- (0.5 puntos) Operador de salida << para poder insertar en un flujo de salida el contenido de un objeto KMedias en formato texto. Implemente también el operador << para la clase Cluster (recuerde que dispone de este operador para las clases VectorPuntos y Punto). En concreto, para un objeto KMedias deberá aparecer:
  - El número de grupos K seguido del carácter fin de línea.
  - La lista de grupos. Para cada uno se mostrará:
    - Las coordenadas x e y del centroide (separadas por un blanco) seguido del carácter fin de línea.
    - El número de puntos del grupo, seguido del carácter fin de línea.
    - El conjunto de puntos del grupo, uno en cada línea.
- (0.75 puntos) Operador de entrada >> para poder obtener desde un flujo de entrada el contenido de un objeto KMedias. Implemente también el operador >> para la clase Cluster (recuerde que dispone de este operador para las clases VectorPuntos y Punto). Se asume el mismo formato indicado en el apartado anterior.

---

◀ **Ejercicio 4** ▶ Métodos de cálculo

[1.75 puntos]

- (0.5 puntos) Implemente el método calcularCentroide() en la clase Cluster que recalcule las coordenadas x e y del centroide de un grupo. La coordenada x (de forma similar con la y) se calcula como la media de las coordenadas x (de forma similar con la y) de los puntos del grupo.
- (0.75 puntos) Implemente el método clusterMasCercano(punto) en la clase KMedias que devuelva el índice (un entero) del grupo más cercano a un punto dado. Para ello se comparará la distancia del punto al centroide de cada grupo.
- (0.5 puntos) Implemente el método cohesion() en la clase KMedias que devuelva la distancia media entre los centroides de los grupos. Para ello debe sumar la distancia entre todas las combinaciones de centroides tomados dos a dos y dividir por el número de combinaciones posibles. Por ejemplo, si tenemos un conjunto de cuatro grupos {A, B, C, D}, la cohesión se calcularía como:
 
$$[d(A, B) + d(A, C) + d(A, D) + d(B, C) + d(B, D) + d(C, D)]/6$$

Pág. 2 / 2

