

Alumno: _____

Este examen vale un 40% de la nota de exámenes.

1.- Dada la clase Punto que permite gestionar puntos del plano .

```
public class Punto {
    private double x;
    private double y;
    public Punto (double x, double y) {
        this.x = x;
        this.y = y;
    }
    public double getX () {
        return x;
    }
    public double getY () {
        return y;
    }
    public double distancia () {
        return Math.sqrt(x*x+y*y);
    }
}
```

Y sea la clase Datos, que contiene una lista de puntos del plano:

```
import java.util.*;
public class Datos {
    private ArrayList <Punto> datos;
    // , , , ,
    // , , , ,

    public void add (Punto p) {
        datos.add(p);
    }
}
```

Se pide escribir:

- Un constructor para Datos que instancie la variable datos con un arraylist vacío. **(2 puntos)**
- Un constructor para Datos que reciba un arraylist de puntos como parámetro y lo asigne a la variable datos. **(2 puntos)**
- Un método *distanciaMedia()*, de la clase *Datos*, que devuelva la distancia media de todos los puntos contenidos en el arraylist datos al origen del plano. El resultado es de tipo *double* (-1 si la lista de puntos está vacía) **(4 puntos)**

2- En Java, ¿a qué nos estamos refiriendo si hablamos de 'Swing'? **(1 Punto)**

- Una función utilizada para intercambiar valores
- Es el sobrenombre de la versión 1.3 del JDK
- Un framework específico para Android
- Una librería para construir interfaces gráficas

3.- Dada la clase PaisCiudades:

```
public class PaisCiudades {
    private HashMap< String, ArrayList<String>> paisCiudades;

    public PaisCiudades() {
        paisCiudades = new HashMap< String, ArrayList<String>>();

        ArrayList<String> ciudades = new ArrayList<String>();
        ciudades.add("Granada");
        ciudades.add("Madrid");
        ciudades.add("Valencia");
        ciudades.add("Jaén");
        this.paisCiudades.put("España", ciudades);

    }
    //inserta país-ciudad si no está ya en el hashMap
    public boolean insertaPaisCiudad(String pais, String ciudad) {
        .....
    }
}
```

Escriba el método `public boolean insertaPaisCiudad(String pais, String ciudad)` dentro de la clase `PaisCiudades`, que inserte en el `hashMap`: si el país ya está en el Map, añade la nueva ciudad al `arraylist` de ciudades del país (si la ciudad no está en la lista, si la ciudad ya está no hace nada); si el país no está, añade el país como nueva clave y la ciudad será la única dentro del `arraylist` de ciudades. Devuelve `true` si se produce la inserción o `false` si no se produce (el país y la ciudad ya estaban en el Map) **(8 puntos)**

`paisCiudades.get("Francia")` devuelve un `arrayList` con las ciudades de "Francia", `null` si la clave "Francia" no está en el Map. El método `get()` es propio de la clase `HashMap`;

`String[] paises = paisCiudades.keySet().toArray(new String[paisCiudades.keySet().size()])` devuelve un array con las claves (países) del Map.

`paisCiudades.put(pais, ciudades)` mete la clave país y el `arrayList` ciudades en el Map, hay que tener en cuenta que el método `put()` (propio de la clase `HashMap`) machaca el `arraylist` de ciudades si el país ya existía como clave dentro del Map.

4- ¿Qué código asociarías a una Interface en Java? **(1 Punto)**

- `public class Componente interface Product`
- `Componente cp = new Componente (interfaz)`
- `public class Componente implements Printable`
- `Componente cp = new Componente.interfaz`