

C.F.G.S. DESARROLLO DE APLICACIONES WEB - PROGRAMACIÓN CONTROL JAVA TERCERA EVALUACIÓN. EDD – 25-3-2025

Una plataforma de aprendizaje en línea quiere analizar la actividad de sus estudiantes en diferentes cursos y módulos. Se dispone de los nombres de los cursos, los IDs de los estudiantes y los nombres de los módulos que cada estudiante ha completado en cada curso. Un estudiante puede completar varios módulos en diferentes cursos.

Estructura de Datos:

Los datos se almacenan en un **HashMap<String, TreeMap<String, TreeSet<String>>>**, donde:

- La clave del HashMap es el nombre del curso (String).
- El valor asociado a cada curso es un TreeMap donde:
- La clave es el ID del estudiante (String).
- El valor asociado al ID del estudiante es un TreeSet con los nombres de los módulos completados por ese estudiante en ese curso (String).

Ejemplo:

```
{
    "Programación Java": {
        "EST12345": {"Introducción", "Arrays", "Objetos"},
        "EST67890": {"Arrays", "Interfaces"},
        "EST13579": {"Introducción", "Objetos", "Herencia", "Excepciones"}
    },
    "Bases de Datos SQL": {
        "EST67890": {"Consultas"},
        "EST24680": {"Introducción", "Joins", "Triggers"}
    }
}
```

Significa que en el curso “**PROGRAMACIÓN JAVA**” cursan los estudiantes **EST12345**, **EST67890** y **EST13579**. En este curso, el estudiante **EST12345** ha completado el módulo **Introducción, Arrays y Objetos**.

Se dan al alumno las dos clases. La clase **Modelo.java** está incompleta y es la única que debe completar y entregar (**en un fichero comprimido con el nombre del alumno**), con los métodos completos.

NOTAS ACLARATORIAS:

- Se proporciona en la clase **Principal.java** el método **creaDatosDePrueba** para poder probar los métodos. A la clase **Modelo.java** hereda **HashMap** y se le debe completar los métodos siguientes:

```
public String cursoConMasEstudiantesUnicos()
```

```
    //Devuelve el nombre del curso que ha tenido la mayor cantidad de estudiantes
    únicos
    //(es decir, estudiantes con IDs diferentes).
```

```
public Boolean matricularAlumnoCuros(String curso, String Alumno)
```

```
    //incluir en al curso pasado por parámetro, el alumno que se pasa por parámetro. Si el
    //curso no existe, se añade nuevo. Si el alumno ya está en el curso se devuelve un
    //booleano de valor false. Por el contrario, se incluye al aalumno en dicho curso y
    //devolvemos el valor booleano true.
```

```
public Map<String, Integer> numeroCursosPorAlumno()
```

```
// Devuelve un mapa cuyas claves son los identificadores de los alumnos ordenados
//alfabéticamente,
// y en valores el número de cursos en los que se encuentra matriculado dicho alumno
```

```
public HashMap<String, TreeMap<String, TreeSet<String>>> datosReorganizadosPorEstudiante()
```

```
// Devuelve un mapa con los mismos datos del modelo, pero en lugar de estar
// organizados por los nombres de los cursos, deben estar
// los identificadores de los estudiantes como claves y asociados a esos
// identificadores, los nombres de
// los cursos y los módulos superados de los mismos
```

.

No se debe modificar la clase Principal, pues será usada en la corrección del examen.

Puntuación ponderada (el máximo corresponde a una nota de 10):

| | |
|---|-----------|
| Método cursoConMasEstudiantesUnicos | 14 Puntos |
| Método Boolean matricularAlumnoCursos | 16 Puntos |
| Método Map<String, Integer> numeroCursosPorAlumno() | 15 Puntos |
| Método datosReorganizadosPorEstudiante() | 15 Puntos |

Soluciones:

Apartado a) Curso con más estudiantes: Base de Datos

Apartado b) Matricular al alumno 34567890C en FOL: true

Apartado b) Matricular al alumno 11111111A en Base de Datos: false

Apartado c) Alumnos y número de cursos en los que están matriculados: {11111111A=1, 12345678A=2, 34567890C=3, 45678901D=1, 56789012E=3, 67890123F=1}

Apartado c) Alumnos y número de cursos en los que están matriculados: {56789012E={Base de Datos=[Consultas, Introducción, Joins, Modelo Entidad Relación, Tablas, Triggers], Programación Java=[Arrays, EDD, Introducción, objetos], Python=[Bibliotecas, Funciones]}, 34567890C={FOL=[], Lenguaje de Marcas=[CSS, HTML, XML], Programación Java=[Arrays, EDD, Ficheros, Herencias, Interfaces, Introducción, objetos]}, 11111111A={Base de Datos=[Consultas, Introducción, Modelo Entidad Relación]}, 67890123F={Base de Datos=[Consultas, Introducción, Joins, Modelo Entidad Relación]}, 45678901D={Base de Datos=[Consultas, Introducción, Joins, Modelo Entidad Relación]}, 12345678A={Programación Java=[EDD, Ficheros, Herencias, Introducción, Objetos], Python=[Arrays, Biblioteca, Control, Funciones, Introducción, Módulos]}}