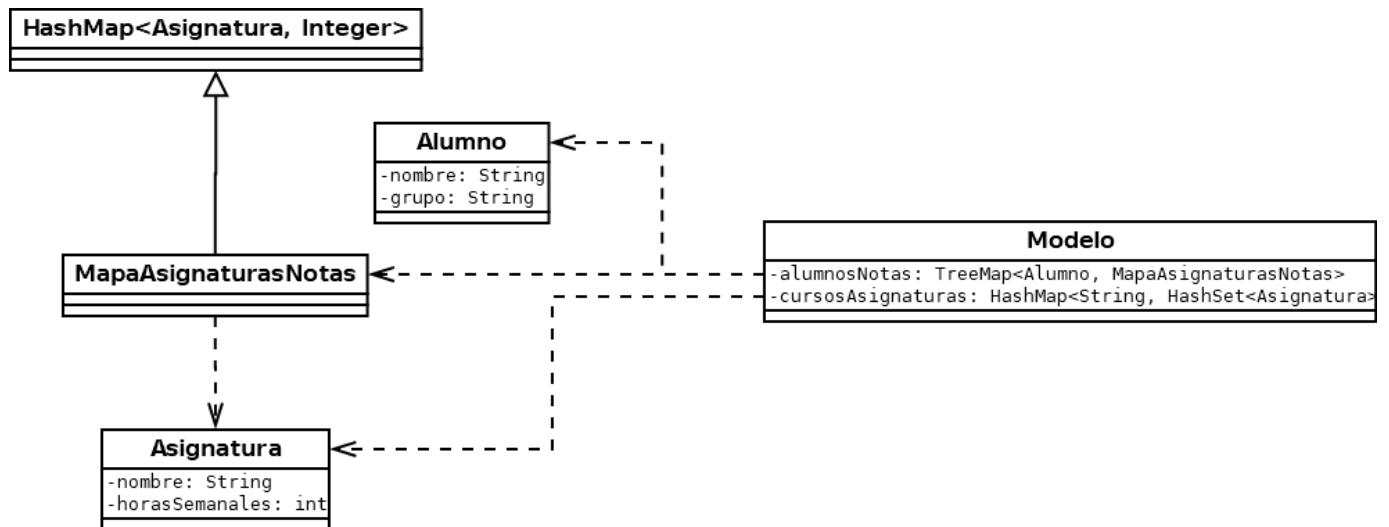


C.F.G.S. DAM - PROGRAMACIÓN
CONTROL 3ªEVALUACIÓN (04-04-2022)

Para gestionar los alumnos y las notas de las asignaturas en las que están matriculados, se debe hacer un programa en java. Se dan al alumno todas las clases, de las cuales **Modelo.java** tiene métodos incompletos. El diagrama UML que representa al modelo y sus clases asociadas es el siguiente. Además, se proporcionan las clases **Principal.java** y **VentanaPpa1.java** que representan al controlador y la vista respectivamente.



Se deben realizar esos métodos, para que el programa funcione como el ejecutable proporcionado. Se deben entregar los ficheros fuente de todas las clases con los métodos completados y lo que sea necesario añadir en el resto de clases. **No se deben añadir atributos ni modificar los métodos ya implementados en las clases aportadas.**

Los métodos a completar son:

a) `public void addAsignatura(Asignatura asig, String curso)`

Añade dicha asignatura al mapa `cursosAsignaturas`. Ten en cuenta que, si el curso no estuviera añadido previamente, este método deberá añadirlo.

b) `public String getListadoCompleto()`

Devuelve los alumnos, asignaturas y notas tal y como se ve en el cuadro de texto al pulsar el botón "Listar Alumnos Notas".

c) `public boolean alumnoEnMasDeUnCurso(Alumno a)`

Devuelve un booleano indicando si un alumno tiene asignaturas de más de un curso.

d) `public float notaMediaAlumno(Alumno alumno)`

Devuelve la nota media del alumno recibido por parámetro.

e) `public List<Asignatura> asignaturasSuspensas(Alumno alumno)`

Devuelve una lista con las asignaturas que tiene suspensas.

f) [solo Alumnos DUAL] `public Set<Asignatura> getConjuntoAsignaturas()`

Devuelve un conjunto con todas las asignaturas de todos los cursos (comprobar en `comboBoxAsignaturas`).

g) [solo Alumnos DUAL] `public void borrarAsignatura(String nomAsignatura)`

Borra a los alumnos los pares asignatura-nota de la asignatura recibida por parámetro.

h) [solo Alumnos NO DUAL]

Añade el menú Alumnos → Añadir Alumno tal que al pulsarlo, se muestre el diálogo de alta (`DialogoAltaAlumno.java`), que debe ser un diálogo modal y tener las características que se muestran en el ejecutable.

Se deben completar los métodos incompletos de `DialogoAltaAlumno`, además de llamar a los métodos correspondientes de otras clases (y crearlos si no existen) siguiendo el estilo modelo-vista-controlador.

Estos métodos son:

- `private void botonAltaAlumnoPulsado()`: Método que debe ser llamado cuando se pulse el botón “Añadir Alumno”.

Este método añade un alumno con los datos de los campos anteriores. Una vez añadido, debe salir un mensaje de confirmación y cerrarse el diálogo. Si no hay curso seleccionado, debe salir un mensaje de error.

- `public void actualizaComboCursos()`

La llamada a este método ya está realizada. Solo debes completar de tal manera que el combo de cursos se cargue con los datos de los cursos existentes.

PUNTUACIÓN:

- Un programa que NO COMPILE o del que se haya modificado otra clase o método que no sean los que se piden tendrá una nota desde 0 (si no hay código) a 4 como máximo.
- Por cada situación distinta que provoque UNA EXCEPCIÓN, se restará UN PUNTO de la nota final del examen que será obtenida según lo siguiente:

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

(a)	Método addAsignatura	8 Puntos
(b)	Método getListadoCompleto	10 Puntos
(c)	Método alumnoEnMasDeUnCurso	14 Puntos
(d)	Método notaMediaAlumno	10 Puntos
(e)	Método asignaturasSuspensas	10 Puntos
(f)	[DUAL] Método getConjuntoAsignaturas	10 Puntos
(g)	[DUAL] Método borrarAsignatura	14 Puntos
(h)	[NO DUAL] Menú y botones	8 Puntos
(i)	[NO DUAL] Método botonAltaAlumnoPulsado	10 Puntos
(j)	[NO DUAL] Método actualizaComboCursos	6 Puntos