

**CICLO: Desenvolvimento de Aplicações
(Web/Multiplataforma)**

MÓDULO: Programación

Examen evaluación 3

Normas

- Hay que enseñar y presentar un documento de identidad válido al inicio del examen.
- Hay que apagar los móviles.
- Bajo ningún concepto se puede utilizar sistemas de mensajería, correo electrónico o cualquiera tipo de comunicación.
- Se puede utilizar [Alexia Classroom](#) para subir el examen y consultar la documentación del módulo, así como se pueden consultar las tareas realizadas durante el módulo. Se puede consultar [StackOverflow](#), [Jenkov](#), [W3Schools](#), la [documentación oficial](#) de Java. En caso de necesitar acceder a otra página web, se deberá consultar antes con el profesor.
- No se puede utilizar ChatGPT ni ninguna otra IA durante la realización del examen.
- Al acabar el examen se debe avisar al profesor para enseñarle el resultado de la ejecución del programa.
- Para entregar la actividad se debe exportar el proyecto de IntelliJ en formato zip y subirlo a la plataforma Alexia Classroom.
- Hay que grabar la pantalla de todo el examen mediante el programa OBS Studio y asegurarse de copiarlo al profesor.

Desarrollo de un proyecto en Java en el que se implementará un programa sobre alumnos, este debe leer un fichero de texto en donde hay guardados los datos de cada alumno. Los datos almacenados son:

- Nombre completo.
- Edad.
- Ciudad de nacimiento.
- Nota de Matemáticas (numérico con decimales).
- Nota de Inglés (numérico con decimales).
- Nota de Programación (numérico con decimales).

Por lo que será necesario crear una clase Alumno en donde se gestionará la información necesaria de cada alumno. Además, en esta clase habrá que tener un método que nos proporcione la nota media de las tres materias. Nuestro programa debe hacer lo siguiente:

1) (2 ptos) Al arrancar el programa, este debe acceder al fichero de registros y leerlo todo.

- a) En caso de no existir el fichero o no haber registros o producirse un error de lectura, ... nuestro programa no puede mostrar errores, solo debe decir que no hay registros.
- b) En caso de haber registros, éstos se deben guardar en una colección de Alumnos (nuestra clase).

2) (1 ptos) Una vez leído el fichero y guardados los registros, debemos mostrar los datos de los alumnos por pantalla. La salida debe de como se puede ver en la Figura 1: Salida de la lista de los registros.



```
-----  
Luis Gómez  
18 años (Lugo)  
M: 7.6 I: 6.9 P: 8.3  
NOTABLE (7.6)  
-----  
Carlos Sánchez  
22 años (Valencia)  
M: 7.5 I: 6.9 P: 8.8  
NOTABLE (7.73)  
-----
```

Figura 1: Salida de la lista de los registros.

3) Después, hay que mostrar un menú con las opciones:

- a) (2 ptos) Mostrar estadísticas: El programa debe mostrar unas estadísticas con los siguientes datos de cada materia:
 - 1) Nota media de todo el alumnado.
 - 2) Nota mínima.
 - 3) Nota máxima.
 - 4) Número de alumnos suspensos.
 - 5) Número de alumnos aprobados.
- b) (2 ptos) Mostrar todos los datos de nuevo: El programa vuelve a mostrar los datos como en el apartado 2).
- c) (2 ptos) Agregar registros.
 - 1) Se deben pedir al usuario los datos necesarios para agregar un nuevo registro a nuestra colección.
 - 2) Estos datos también se deben persistir en el fichero de registros.
- d) Salir: El programa debe salir.

El punto restante sobre 10 es para la estructura general del programa, el menú, la opción de salir, etc.

Normas sobre el código fuente del examen y algunas cuestiones que os pueden ayudar:

- Se debe utilizar la Orientación a Objetos tal y como se explica.
- Es mejor empezar el programa por el orden en el que el enunciado se presenta.
- Se debe utilizar algún tipo de colección (lista, mapa, cola, conjunto) para almacenar los datos de los alumnos leídos del fichero.
- En caso de no existir el fichero o no haber registros o producirse un error de lectura, escritura, ... nuestro programa no puede mostrar errores, solo debe avisar de que no hay registros, o que la operación no se ha podido realizar, pero no puede parar la ejecución.
- Intentar encapsular el código susceptible de reutilización en funciones, tal y como se ha visto en la tutorías y ejercicios, como por ejemplo ciertas acciones que se realizan varias veces, etc.
- Recordar seguir las convenciones sobre como nombrar las clases, variables, objetos, ...
- En el examen podéis consultar los ejemplos de la página web de Jenkov sobre las colecciones, ficheros, ...
- Acordaros de que todos los String tienen un método split por el que se puede dividir un String por un "token" y esto nos genera una lista de Strings con los distintos pedazos del String original.