



SISTEMA DE GESTION DE CALIFICACIONES

Nos han contratado para realizar parte de una aplicación de gestión de calificaciones del alumnado de 1DAM en el módulo de Programación.

Nuestra aplicación debe permitirnos gestionar el alumnado y con ello poder obtener datos de un alumnado en concreto o del todo el grupo, poder modificar sus datos o indicar que este alumno no está disponible. Además, cuando consultemos la lista del alumnado debemos obtener los resultados ordenados por número de lista, o por orden alfabético. Del alumnado nos interesa saber su DNI (NNNNNNNNL) nombre, apellidos, email, teléfono de contacto en formato (NNN-NNNNNN) y si ha perdido la evaluación continua. Es conveniente tener la fecha de matriculación del alumnado y que esta siempre nos aparezca como DD/MM/AAAA. Tenemos que tener en cuenta que un alumno no se puede eliminar si ya está en una evaluación, alteraría el resultado de la misma, por lo que es interesante que os planteéis que mejor que eliminar quizás sea interesante indicar si está disponible o no.

Por otro lado, sabemos que tenemos distintas categorías de evaluación: práctica, examen, ejercicio, exposición... Nuestro problema es que no podemos saberlas todas, por lo que necesitaremos gestionarlas para en todo momento poder añadir, o modificar las existentes.

Por supuesto, tenemos la prueba de evaluación que se realizará en una fecha, y que siempre se mostrará como DD/MM/AAAA HH:MM:SS. Además dicha prueba tendrá una descripción y siendo de una categoría de las existentes. Por cada prueba tendremos unas calificaciones donde para cada alumno asociaremos la calificación obtenida y su momento de entrega. Debemos tener en cuenta, como se ha dicho con anterioridad, que solo podemos evaluar a alumnado disponible o que no haya perdido la evaluación continua.

- Respecto a la evaluación, podemos listarlas, consultarlas, pero no modificarlas una vez ya han sido creadas, pero sí eliminarlas. Además en la propia evaluación debe almacenar la nota máxima, mínima, media, porcentaje de aprobados, porcentaje de suspensos. Además mostrará el alumnado y sus calificaciones en orden de nota. No hace falta decir que la nota se presenta con dos decimales en formato español. Finalmente podremos exportar esta calificación en un fichero markdown (formato .md), dándole formato adecuado (encabezados, negritas, cursivas, listas, etc.) y preguntando al usuario dónde quiere que lo guarde. El informe debe terminar con la fecha en la que ha sido generado, y el tiempo que ha tardado en realizarse. Por ejemplo: Informe generado el DD/MM/AAAA HH:MM:SS en SS. Ejemplo: Informe generado el día 05/10/2021 a las 12:14:34 en 7,45 segundos.

Finalmente, debemos ser capaces de exportar e importar toda la información en formato JSON, para no perderla o simplemente poder recuperarla con otros sistemas en un futuro en el fichero Backup.

Debemos ser cuidadosos y asegurar que nuestra aplicación no corrompe la información y falla en la entrada de datos. Por lo que hay que prestar especial interés en ello.

Ningún software debe estar libre de errores, por lo que se puede testear unitariamente el sistema de almacenamiento de datos.

Antes de programar a lo loco, os recomiendo que hagas un diagrama de clases del modelo de negocio. Más que nada porque se entregará.

No hemos visto SOLID, Arquitecturas y Patrones para nada, por lo que se valorará muy positivamente el uso de ellos.

Se deberá usar GIT/GitHub, aplicando GitFlow en todo momento entre los componentes del equipo. Si no se aplica GitFlow la práctica quedará invalidada.

Nuestro programa debe llamarse con un JAR de la siguiente manera

```
$ java -jar calificaciones.jar
```

Debes tener en cuenta que los datos deben expresar y mostrar las unidades numéricas y de fecha en formato oficial de España.

Cuidado que para buscar algún dato, se puede meter en mayúsculas o minúscula y no debe importar.

Se recomienda tener una batería de datos de prueba con los que inicializar y ver el funcionamiento del programa.

Se recomienda que el proyecto esté gestionado por maven o gradle.

Puede que te falten detalles, o puede que no tengas toda la información en este anuncio y necesites entrevistarte con el cliente. Recuerda que puedes hacerlo, pero no siempre. Esto es tan real como la vida misma.

Recuerda no verlo como un todo... Divide, diseña bien y construye las cosas poco a poco y vencerás.

Se debe entregar:

- La práctica es en parejas y en la documentación como en la salida del informe debe aparecer el nombre y apellidos de los dos integrantes.
- Repositorio GitHub con la solución en el que incluyas:
 - o Readme explicando el proyecto y el nombre de los integrantes.
 - o Código fuente comentado y perfectamente estructurado con JDoc. Además de los gitignore adecuados y que siga el flujo de trabajo GitFlow. No se deben incluir los ejecutables si no se deben poder crear los jar desde maven. Usa .gitignore bien.
 - o Documentación en PDF donde expliques describas el proceso de solución:
 - Requisitos funcionales, no funcionales y de información.
 - Diagrama de Casos de Uso.
 - Diagrama de clases.
 - Arquitectura de software aplicada.
 - Patrones de diseño y ejemplo de como se han aplicado.
 - Principios SOLID aplicados y ejemplos.
 - Diseño de Pruebas unitarias.

- Ejemplo de uso y capturas del mismo.

Se valorará enlace a un vídeo de Youtube donde se muestre su funcionamiento.

La práctica no es obligatoria, pero no realizarla implica que los Resultados de aprendizaje no se podrán evaluar a través de ella, lo que implica que deberán ser evaluados por otros instrumentos

Aprobar la práctica no implica no se haga el examen, si no es un complemento al mismo.

La copia de la práctica o fragmentos de ella implica la evaluación de todos Resultados de Aprendizaje con un 0, no pudiéndose recuperar hasta la evaluación ordinaria.

Fecha de entrega 8 de abril de 2022 a las 08:00.

Repositorio de entrega: https://classroom.github.com/a/Xgo58m5_

“¿Saber quieres la diferencia entre maestro y aprendiz? El maestro más veces ha fallado que intentar el aprendiz ha hecho” Maestro Yoda.