

**Autores: Profesorado de Informática del Instituto
Nebrija de Formación Profesional**
DAM / DAW
Proyecto
Curso 2024-25

Proyecto: Elige tu propia aventura II

OBJETIVO: El objetivo del proyecto será mejorar el juego de Elige tu propia aventura previamente implementado o crear uno totalmente nuevo, estableciendo algunas mejoras con los conocimientos adquiridos, concretamente: programación orientada a objetos y conexión a base de datos.

-

1. Especificaciones y requisitos

- Fundamentalmente, como se ha mencionado anteriormente el proyecto tendrá dos líneas de actuación:

- Programación Orientada a Objetos. Con los conocimientos de programación orientada a objetos adquiridos, buscando dotar al código de abstracción, encapsulamiento y modularidad con el fin de potenciar su reusabilidad y mantenibilidad.

Es muy importante realizar un diseño orientado a objetos adecuado con:

- Diagrama de clases.
- Justificación de decisiones tomadas (herencia, composición, clases abstractas, interfaces etc.)
- El alumno, si lo estima oportuno, podrá realizar cambios en la historia, ampliarla o incluso optar por realizar una nueva aventura a su elección, dado que, todos los nuevos conceptos adquiridos desde la primera implementación del código, podrán utilizarse para potenciar el juego e incluir mejoras y ampliaciones con facilidad. De hecho, se valorarán positivamente la inclusión de nuevas funcionalidades e interacciones en el juego.
- En la memoria, deberán quedar documentadas todas las decisiones adoptadas para refactorizar y mejorar el código. Además, si se han ampliado funcionalidades en la lógica del juego respecto a la versión anterior, también han de quedar correctamente documentadas.
- El alumno, deberá, con la experiencia y conocimientos adquiridos, realizar cualquier cambio en el código inicial, aunque no sean específicos de BDD o POO como incluir más funciones, estructurar mejor el código, bucles, condiciones etc. De hecho, es una de las cuestiones fundamentales a cubrir con este proyecto: mejorar el código realizado inicialmente.

- Puesto que un aspecto básico para mejorar y refactorizar el código es la utilización de clases, es importante volver a destacar, como ya se ha comentado anteriormente, que se tendrá que realizar el diagrama de clases correspondiente y documentarlo en la memoria.
 - Además, se valorará incluir nuevas funcionalidades, es decir, nuevos puzzles o acertijos, batallas o combates más elaborados, nuevos enemigos o personajes etc.
 - En caso de que la inclusión de una nueva funcionalidad suponga la incorporación de un nuevo caso de uso, deberá reflejarse también en la memoria con el diagrama correspondiente.
- Conexión con Base de Datos.
- Se deberá incluir en el juego, la funcionalidad necesaria para gestionar el registro y el login del usuario.
 - Jugador. Todo jugador tendrá un usuario y una contraseña que deberá introducir para poder comenzar a jugar al juego. Este modo de acceso al juego, sustituye a la implementación de acceso mediante una palabra clave que se tenía en la implementación del juego realizada anteriormente.
 - La primera vez que se ingrese al juego, por tanto, se deberá preguntar si desea hacer login (usuario ya registrado con anterioridad) o registrarse como nuevo usuario.
 - En el segundo caso, se le pedirá al usuario un nombre de usuario, una contraseña y que repita dicha contraseña, no dejándole registrarse si las dos contraseñas no coinciden. Cuando coincidan, se ingresarán esos datos del usuario en la BBDD y la siguiente vez que se conecte al juego, ya podrá loguearse con total normalidad.
 - Además, se debe almacenar la puntuación de cada jugador. Todos los jugadores parten de cero puntos y cada vez que lleguen a un final satisfactorio se incrementará en 50 su puntuación. Asimismo, al completar un acertijo, juego, enigma, puzzle etc. de la aventura también se aumentará la puntuación del jugador (los puntos a incrementar son a elección del alumno y cada puzzle tendrá el suyo como se comentará posteriormente).
 - En el menú del juego se deberá dar la posibilidad de poder consultar:

- El jugador con mayor puntuación.
- El top 3 de jugadores con mayor puntuación.
- Aunque, para almacenar esta información, se trata de una base de datos simple a nivel de diseño, debe quedar especificado en la memoria todo el proceso de modelado datos:
 - Toma de requisitos de datos de la aplicación
 - Entidad-Relación
 - Paso a tablas
 - Razonamiento de consultas que se necesiten.
 - Consultas necesarias con sintaxis SQL.
- Todas las fases del modelado de datos anteriormente comentadas, justificadas y razonadas, deberán quedar reflejadas en la memoria.
- Almacenamiento de datos relevantes del juego. Se deberán almacenar todos los datos relevantes del juego en base de datos. De esta forma, en caso de que no estén implementados ya en la versión inicial, se deben tener en cuenta los siguientes aspectos:
 - Personajes protagonistas (3): de cada personaje, será necesario almacenar:
 - Nombre.
 - Finales: Cada personaje tenía, como mínimo, cinco finales diferentes.
 - Puntos de vida
 - Objetos de inventario (si lleva)
 - Personajes secundario (NPC): Personajes que se dedican a charlar y darte información en la aventura.
 - Nombre
 - Especie: humano, animal, extraterrestre, elfo, pokemon ..
 - Frases disponibles para interacción con el usuario.
 - Personajes secundarios enemigos. De cada personaje secundario enemigo, será información relevante:

- Nombre
 - Especie: humano, animal, extraterrestre, elfo, pokemon ..
 - Puntos de vida
 - Ataques (como se comentó en el enunciado de la primera parte, como mínimo tendrá tres posibles ataques).
- Ataques: En cuanto a los ataques, de cada ataque, será necesaria la siguiente información:
 - Nombre del ataque
 - Puntos que quita (recordad que, como se indicó en el enunciado de la primera parte de este proyecto, habrá tres posibilidades, dependiendo de si impacta totalmente, parcialmente o no impacta en el adversario).
 - Juegos, puzzles, acertijos.. o como los queráis nombrar. Como se mencionó en el enunciado inicial del proyecto cada personaje tiene asociado al menos 5 juegos.

En cuanto a los juegos, de cada juego, será necesario tener disponible la siguiente información:

- Nombre del juego.
 - Descripción del juego
 - Puntos que proporciona a la puntuación del jugador en caso de que sea superado.
- Batallas. Como sabemos los personajes principales tendrán que librar batallas con los enemigos. De cada batalla, será necesario disponer de la siguiente información:
 - Nombre de la batalla.
 - Escenario en el que se desarrolla.
 - Enemigo contra el que se lucha.
 - Escenarios. Para este proyecto, no estaba contemplado en el primero, habrá que considerar el escenario. Así, el escenario, influirá en la batalla, de tal forma que bajo determinadas condiciones puede haber

cierta penalización (como veremos a continuación). De cada escenario, será necesario disponer de la siguiente información:

- Nombre del escenario.
- Condiciones atmosféricas. Podrán ser:
 - Despejado
 - Penalización: 0
 - Nuboso
 - Penalización: 0
 - Lluvia
 - Penalización en ataque: 1
 - Nieve
 - Penalización en ataque: 2
- Momento del día. Podrá ser:
 - Mañana
 - Penalización: 0
 - Tarde
 - Penalización 0
 - Noche
 - Penalización en defensa: 1
 - Bonificación en ataque: 2
- Condiciones del terreno.
 - Llano
 - Penalización: 0
 - Rocoso
 - Penalización en ataque y defensa: 1
 - Montañoso
 - Penalización en ataque y defensa: 2

En cuanto a las penalizaciones, quedará a elección de los alumnos cómo integrarlas en el juego (por ejemplo, que dependiendo de los turnos de combate

cambie la hora del terreno, que el personaje tenga una habilidad para moverse a otro terreno distinto, que si un enemigo o personaje lanzan un conjuro, cambie el tiempo atmosférico...)

- Casos de Uso. Realizar los diagramas de casos de uso. Tendrán que incluirse en la memoria.

2. Desarrollo del proyecto

- A continuación, se indica como se desarrollará el proyecto.
- El proyecto se realizará en grupo.
- Con los conceptos de cada área especificada (programación orientada a objetos, entornos de desarrollo, y BDD) el alumno podrá ir acometiendo las tareas especificadas en el proyecto.
- Es importante destacar que, tanto por la naturaleza de los trabajos a realizar como por ser la mejora de un proyecto anterior es fundamental realizar una labor de análisis y toma de decisiones antes de realizar cada tarea que deberán quedar documentadas en la memoria.

Es importante tener en cuenta:

- Antes de realizar la refactorización de código se tendrá que estudiar bien la implementación inicial de la que se dispone y las medidas que serían interesantes para mejorarlo, así como los mecanismos a utilizar para ello.
- Para realizar los casos de uso será necesario pensar bien con anterioridad los actores que interactuarán con el sistema (el juego) y las posibles acciones a realizar por ellos.
- Por último, lógicamente antes de introducir cualquier BDD en el motor, es necesario realizar todo el proceso de modelado de datos, como se ha indicado con anterioridad.

3. Entregables.

- Se deberá entregar en la plataforma en el curso de proyectos en la actividad indicada para ello. Los entregables son dos:
- Memoria en pdf. El documento ha de llamarse: Nombre_Apellido1_ETPA_II.pdf (donde nombre será el nombre del alumno y apellido1 su primer apellido).
- Código implementado. Para ello, se deberá exportar el proyecto desde eclipse a un .zip. El Archivo debe llamarse Nombre_Apellido1_ETPA_II. (donde nombre será el nombre del alumno y apellido1 su primer apellido)..

- Un fichero de texto, llamado GrupoX_ETPA_II.txt cuyo contenido será el nombre y apellidos todos los componentes del grupo.
- En cuanto a la entrega es muy importante destacar los siguientes aspectos:
 - Deberán realizar la entrega en la plataforma todos los componentes del grupo.
 - Se deberá realizar antes de la fecha límite indicada en la plataforma y comunicada por los profesores. Si algún grupo, por causa justificada, no pudiera llegar a la fecha límite ha de ponerse en contacto con los profesores.

4. Evaluación.

- Para la evaluación se tendrán en cuenta los siguientes aspectos:
 - Exposición (20 %)
 - Código (70%)
 - Memoria (20%)
 - Los profesores podrán realizar una prueba de autoría oral (al margen de la exposición) y/o escrita en caso de que exista alguna duda sobre la misma en todo el grupo o en componentes en concreto. También se podrá realizar dicha prueba de manera generalizada si se considera el proyecto como clave dentro del desarrollo del curso. En ambos casos, la calificación será de Apto o No Apto.
 - Se valorará positivamente la originalidad y la imaginación a la hora de implementar funcionalidades, buenas prácticas, código bien estructura y limpio, así como el uso de elementos que los alumnos hayan aprendido por su cuenta, pero SIEMPRE sabiéndolos explicar correctamente.

5. Extras

Como trabajo extra, que siempre será valorado positivamente, el alumno podrá:

- Incluir otros aspectos del juego que considere que requieran persistencia de datos y que, por tanto, se deberán incorporar a la Base de Datos.
- Ampliar de forma significativa la aventura desarrollada anteriormente para que sea más completa.
- Guardar encriptadas las contraseñas en la Base de Datos.
- A la hora de registrarse el usuario en la BBDD, se podrá comprobar que no exista ya un usuario con ese nombre de registro o apodo, en cuyo caso el programa

informará de que ya existe un usuario con ese nombre y que debe elegir otro si desea registrarse.

6. Fechas y plazos

- Trimestre 3, Se concretará la fecha exacta en las próximas semanas.