



FECHA: 4 de diciembre

Hora inicio: 16:55

Hora de finalización: 19:40

Examen 1º Evaluación

CRITERIOS DE CALIFICACIÓN

*Cada ejercicio tiene una serie de apartados con una ponderación concreta, será **requisito imprescindible** que para obtenerlos, se cumpla con todo lo que se pide en cada uno. No se calificarán apartados a medias.*

No debe haber errores en consola de los programas para que tengan validez.

*Se deberá entregar **OBLIGATORIAMENTE**, un archivo comprimido con vuestros apellidos y nombre, ejemplo: RodriguezFernandezPepe.rar.*

*Adicionalmente, deberéis subir una nueva versión a vuestro **repositorio** en un commit llamado, "Examen".*

OBSERVACIONES

- El examen no podrá entregarse una vez que acabe el tiempo establecido.
- Disponéis de 15 minutos desde el comienzo del examen para realizar preguntas de los enunciados, luego no se responderá a ninguna duda del examen.
- Sólo se permitirán usar las prácticas entregadas que han realizado este curso los alumnos.
- En caso de ver el uso de dispositivos móviles, aplicaciones que permitan establecer alguna comunicación con otras personas (Discord, correo, WhatsApp web...) o herramientas de IA, sin consentimiento, como extensiones del Visual Studio (la cual se podrá pedir al alumnado que muestre si la tiene activa en el examen), el examen estará suspenso.



Proyecto inicial (4 puntos)

- Que esté correctamente generado el JSDoc del proyecto. **(0.5 puntos)**
- Proyecto organizado correctamente, utilizando módulos, archivo de constantes, archivo de utilidades... **(0.5 puntos)**
- Que el proyecto funcione correctamente como se ha pedido en la entrega inicial. **(3 puntos)**

Defensa (6 puntos)

Deberéis obtener como mínimo 3 puntos en esta parte para que se considere apta.

- **Formulario para crear al jugador: (2 puntos)**

La primera escena del juego deberá ser un formulario que permita añadir al jugador, el nombre, ataque, defensa y vida. Se deberán comprobar con expresiones regulares si cumplen estos requisitos antes de crearlo y avanzar en la historia.

- **Nombre:** Sólo permitirá letras y espacios en blanco, siendo la primera letra mayúscula y que el total sea como máximo de 20 caracteres, no debe permitir que sólo haya espacios en blanco.
- El **ataque** y la **defensa** empezarán en 0 y podrán aumentarse en valores enteros.
- La **vida** empezará en 100 y podrá aumentarse en valores enteros, nunca podrá ser inferior a 100 puntos.
- La **cantidad total** entre el ataque, defensa y vida no podrá ser mayor a 110 puntos. Permitiendo al jugador poder repartir 10 puntos entre estas 3 propiedades. Ninguna propiedad podrá ser negativa.

Permite añadir al jugador con los datos que se piden (1 punto)

Se controlan todas las regex (1 punto)

- **Dinero: (2 puntos)**

- Los jugadores empezarán con una **nueva propiedad** llamada **dinero**, que tendrá un valor inicial de 500 monedas (En el mercado no habrá precios superiores a 500 por objeto).



- En la escena del mercado, tras añadir o retirar objetos de la cesta, esta propiedad se modificará, siendo esta siempre positiva, de tal forma que no se permitirá añadir un producto a la cesta si no hay suficiente dinero. **(1 punto)**
 - En la escena del mercado, se deberá mostrar el **dinero disponible** en función de lo que se haya añadido en la cesta. **(1 punto)**
-
- **Combates: (1 punto)**
 - En cada batalla, si el jugador gana, además obtendrá una **suma extra de dinero**, si es un enemigo normal, serán 5 monedas y si es un jefe, serán 10 monedas. **(0.5 puntos)**
 - A la puntuación total se le sumará el total de monedas restantes. **(0.5 puntos)**
-
- **Ranking: (1 punto)**
 - Al finalizar todos los combates, se debe guardar en **LocalStorage**, un registro del nombre del jugador, junto con su puntuación y monedas totales. **(0.5 puntos)**
 - Añadir un botón en la última escena para **mostrar el ranking** por consola al pulsarlo. **(0.5 puntos)**