Informe de pruebas de Guardar\_cargar

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Función** | **Descripción** | **Resultado** | **Posible solución** | **Autor informe** | **Autor pruebas** |
| Guardar | Imprime el estado actual de la partida en un fichero de texto | \*Satisfactorio  (aunque imprimía algunos espacios que no encajaban con el formato buscado y que quizás luego darían problemas al cargar) | Solución de Elvira: revisar el código y corregir aquellos huecos no deseados | David López | Elvira Vegas |
| Cargar | Realiza una llamada al lector para leer los datos sobre el estado de una partida | Satisfactorio (dependía del funcionamiento del lector) | Nada que destacar | David López | Elvira Vegas |

**Análisis**

**• Homogeneidad:** ambas funciones (guardar y cargar) tienen la misma forma en los argumentos, ya que son los personajes, objetos y espacios (y sus respectivos índices de tablas) y el fichero donde se va a realizar la acción. Además, las dos devuelven STATUS.

**• Redundancia:** es claro ver que no hay funciones que realicen la misma acción, ya que son muy diferenciadas: una para guardar partida y otra para cargarla.

**• Documentación:** con la cabecera debería ser suficiente para entender la función que realiza cada rutina. Es conceptualmente sencillo entender los parámetros que pasamos, los necesarios para gestionar correctamente una partida.

**• Funcionalidad:** con las funciones implementadas se puede realizar un guardado y cargado de partida para continuar desde el punto exacto donde lo dejamos al salir.

**• Tratamiento de errores:** el tratamiento de errores se basa en que los parámetros sean los adecuados para poder imprimir y leer las tablas de personajes, objetos y espacios con el fichero correspondiente. Si alguno de ellos no encaja, devuelve error, que será procesado por el main principal.