PDF – TP 3 – Gabriel Lopez Gasal.

Concepto del TP:

El concepto de mi tp se basa en un programa que permite dar de alta, modificar y eliminar personajes, de los cuales poseen ciertas características para posibilitar el enfrentamiento entre ellos. Es decir, se puede ingresar a una sección donde la posibilidad de enfrentar personajes existe. Además, el programa contiene una sección de impresión la cual guarda las filtraciones de personajes que se precisen hacer en cuestión de lo que se elija. Por ejemplo: Filtrar a todos los personajes con poder “Fuego”. El programa cuenta con una sección de “Testing” con más de 20 testeos, de todos los métodos y propiedades de la biblioteca principal de clases. Además de la documentación absoluta y total de todo método y propiedad de excepciones, biblioteca de clases entidades, testers e inclusive forms. El programa incluye un directorio con archivos y carpetas que contienen todos los errores que surgieron en el transcurso del programa.

Temas realizados y aplicados:

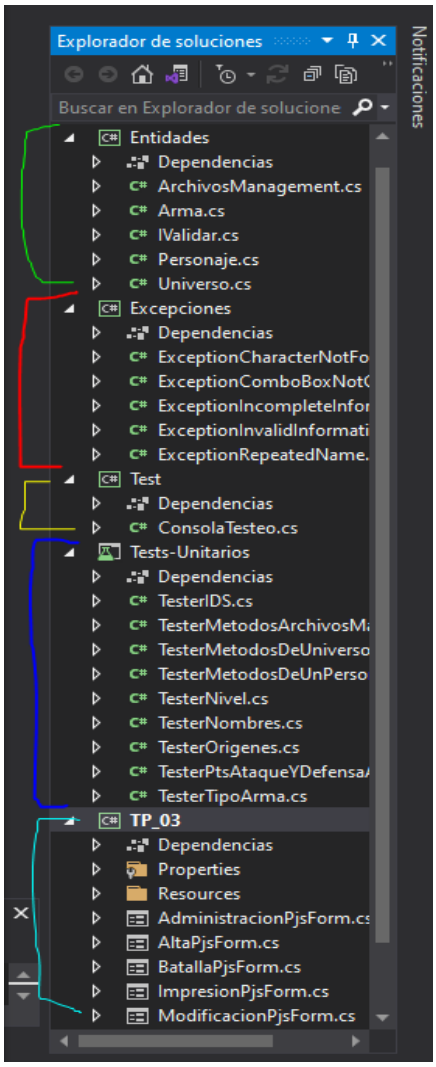
Se realizaron **excepciones personalizadas y utilizadas** para situaciones donde surgían errores específicos.

También se incluyó una **consola básica** que al ser ejecutada, se realizan todas las funcionalidades más básicas e importantes del programa, tales como guardar en archivos, enfrentar héroes, etc.

Se utilizó 3 tipos de formatos de archivos que a día de hoy siguen utilizándose, tales como **escritura y lectura de archivos xml, json y txt.** Remarcando el detalle de que se permite utilizar el escritor y lector con cualquier objeto más allá de un personaje o una lista de estos, ya que tantos los escritores como lectores de todos los formatos, fueron creados **de forma genérica**. Permitiendo lo mencionado anteriormente.

Se utilizó también una **interfaz validadora** que fue la encargada de validar, personajes y las armas de los mismos.

Finalmente, como se mencionó en la conceptualización del tp al principio de página, todo lo realizado en el programa, fue testeado y verificado con **Pruebas Unitarias.**



La ubicación exacta de cada tema aplicado está bastante a la vista.

- La **interfaz** se vio aplicada en la biblioteca de clases, entidades.

- Los **test-unitarios** se incluyeron en la solución como un nuevo proyecto, este mismo incluyendo los testeos de casi todo método y o propiedad existente.

- La **consola** que prueba las funcionalidades mas importantes se vio, también, agregada como un proyecto a la solución del programa, llamándose “Test”.

- Las **excepciones personalizadas** se agregaron de la misma forma, con un nuevo proyecto agregado a la solución del programa, sin mucho desarrollo por encima de estas, ya que el principal objetivo era personalizar el error, no desarrollarlo a fondo.

- Los métodos de **escritura y lectura de archivos** se encuentran en la biblioteca de clases, en la clase “ArchivosManagement”.

- La posibilidad de **utilizar un método o clase de forma genérica** se vio aplicado a los métodos de escritura y lectura de archivos, ubicados, como se mencionó anteriormente, en la biblioteca de clases, clase “ArchivosManagement”.