

PDF - TP 4 - 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 un aproximado total de 40 tests, de todos los métodos y propiedades de la biblioteca principal de clases. Además, se encuentra desarrollada la documentación absoluta y total de todo método y propiedad de excepciones, biblioteca de clases entidades, testers e inclusive forms. También las "líneas de código" con una lógica ciertamente compleja o abstracta, se encuentran documentadas.

Correcciones ya implementadas del TP3

Se fijaron e implementaron las ideas dadas en la corrección del TP3, mejorando así aún más el programa, tanto desde la lógica como desde lo visible. Ahora existe un sistema automatizado de batallas, el cual deja atrás la necesidad de escribir obligatoriamente el nombre de cada personaje. Pero eso no quitó la posibilidad de escribir el nombre que queramos... sosteniendo así tanto el concepto de realizar una "pelea de personajes random" como una "pelea personalizada". Un pequeño, pero no insignificante detalle fue la implementación de la lista de personajes en esta sección de Batalla. Ahora no tenemos que recordar el nombre de nuestro personaje ni tampoco ir a la sección de Impresiones para ver la lista de personajes. La lista ya se encuentra al alcance y comodidad del mismo form "Batalla". También se agregó un total de 11 filtros extra (sin contar los ya existentes de la versión pasada), todos con resultados estadísticos, tales como porcentajes, máximos, mínimos, entre otros, abriendo así, el panorama en cuanto a las posibilidades para filtrar.

Temas realizados y aplicados:

Contenidos TP3:

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. 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 tanto 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 testado y verificado con **Pruebas Unitarias**.

Contenidos TP4:

A partir de la implementación de los contenidos del TP3 y las correcciones, **se incluyó SQL**, es decir, una base de datos que reemplazará el uso de archivos pero que no lo descartará. Los archivos pasaron a ser back-up de lo que sucede en la base, obviamente manteniendo las funcionalidades como la de poseer un log de errores, archivos que contengan la información filtrada que se desee, etc. El **uso de delegados** se encuentra presente tanto en la creación del **evento Batallar** como en la carga de personajes de la base de datos. **La utilización de hilos** ocurre al principio del programa, encargándose de realizar la carga de personajes mientras el usuario recorre las funcionalidades del programa que no precisen que la carga esté finalizada. Para iniciar la tarea que se ejecutará en otro hilo, **se utiliza una expresión Lambda** que contiene el delegado que carga a los personajes y un flag booleano representativo a si la tarea finalizó. En la sección de Impresiones, al obtener los porcentajes tanto de poderes como de armas, **se utiliza un método de extensión** de la colección "List" de tipo Personaje. (List<Personaje>).

La ubicación exacta de cada tema aplicado.

TEMAS TP3

- 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".

TEMAS TP4

- El desarrollo de **SQL** tanto lo que es el uso y la conexión se encuentran dentro de Entidades, en "DBStuff.cs". Las principales referencias provienen del alta, baja, modificación y eliminación de personajes que se deben realizar también en la base de datos.
- Los **delegados** son utilizados tanto en el loader de "PanelControlForm.cs" como en la clase "Universo.cs" dentro de Entidades, para la creación del evento batallar.

- El uso de una **expresión lambda** se ve presente en el loader de "PanelControlForm.cs" al momento de iniciar la Task.
- La utilización de **hilos** está presente en el loader de "PanelControlForm.cs" al momento de crear e iniciar la task "taskTraerPersonajesDB".
- La creación de un **evento** se puede observar en la clase "Universo.cs" ubicada dentro de Entidades y su invocación dentro del método enfrentamiento de esta misma clase. El manejador de este mismo evento, también se encuentra en la clase ya mencionada.
- Se realizó un **método de extensión** para la colección de List del tipo Personaje. Se puede visualizar la declaración en la clase "MetodoExtension.cs" ubicada en la biblioteca de clase Entidades y la utilización dentro de "ImpresionPjsForm.cs" dentro del manejador del botón filtrar, dentro del bloque else, en el switch case 7. Se llama al método añadido a la colección tanto para obtener los porcentajes de armas como poderes.