Justificación de patrones

Enrique Alonso

Singleton

Singleton es el patrón más fácil de implementar para una conexión a una base de datos SQL, porque así solo tengo que manejar una instancia de una conexión a SQL que es movida a lo largo de mis Forms, y solo es instanciada una vez, en vez de los 9 casos en los que seria instanciada.

Prototype

Prototype me ha facilitado el proceso de copiar una variable a otra sin que se quede guardado por apuntador. Esto es útil debido a las múltiples listas que utilizo a lo largo de mi programa que requieren ser copiadas y actualizadas múltiples veces.

Observer

Observer me permite conectar mis múltiples User Controls, sin necesidad de utilizar un acercamiento basado en eventos, se utiliza en mi control TestForm para controlar los demás controles encargados de editar perfil, mostrar perfil, hacer publicaciones, crear historias, etcétera.

Adapter

Adapter me facilita tanto en tiempo como en comprensión, simplemente tomar un objeto Image y convertirlo en una imagen guardada dentro del sistema, o viceversa.

Iterator

Después del examen del parcial pasado, encontré que era muy complejo repetir cálculos una y otra vez para poder navegar un conjunto de clases y mostrarlo gráficamente, pero al crear un iterador que al inicializar se declara el numero de saltos, me facilito mucho el trabajo.

Builder

Debido a que iba a generar 2 clases relativamente similares (Story y Publication), era mucho más fácil simplemente crear un Builder y según mis necesidades usar polimorfismo para poder hacer la creación de ambas.