

Explicación de mejoras de UML y de diseño y arquitectura del esquema PokemonTCG

En ésta segunda versión del diseño del esquema, se implementaron mejoras de la normalización, para que cumpla correctamente tanto la segunda como la tercera forma normal, separando campos que dependían únicamente del identificador de cada tabla lo mayor posible, sin hacer muy tedioso el número total de tablas y que éste sea una cifra manejable y cómoda, además de incluir más detalles acerca de cada movimiento dentro de una partida.

El esquema PokemonTCG, está en la primera forma normal porque la base de datos solo tiene un elemento por fila y columna, cada campo cumple con solo contener un tipo de dato, cada campo tiene un nombre único respecto a su tabla, cada fila tiene una llave primaria única, y el orden del almacenamiento de los datos no tiene importancia.

Por lo tanto el esquema está en la segunda forma normal porque cumple con la primera forma normal y adicionalmente todos los atributos de cada tabla solo dependen de la llave primaria, y no están anclados a ningún otro atributo.

Y finalmente el esquema cumple con la tercera forma normal porque cumple con la segunda y porque hay una separación entre varios elementos en tablas que no dependen de los datos de los otros campos, claro que éste criterio siempre se va a poder optimizar más, pero se deja un poco a criterio del arquitecto de base de datos, qué tanto se debe de normalizar el esquema, de manera que no sea muy tedioso que hayan demasiadas tablas