



DOCUMENTACIÓN PROYECTO BASE DE DATOS ENTORNOS DE DESARROLLO

GRADO SUPERIOR EN DESARROLLO DE APLICACIONES WEB

Luis Rafael Arroyo Caballero

1º DAW 2020-2021

I.E.S. ALIXAR

ÍNDICE

- Objetivos del proyecto (página 3).
- Funcionalidades de la aplicación (página 4).
- Descripción de las tecnologías utilizadas (página 6).
- Diseño de la Base de Datos:
 - Modelo Entidad Relación (página 8).
 - Modelo Relacional (página 9).
- Conclusiones del proyecto (página 10).

OBJETIVOS

A día de hoy los videojuegos tienen una gran cantidad de información y es muy complicado saber todo al pie de la letra. En el videojuego en línea Genshin Impact, esta información es muy diversa y abundante, por lo que con los conocimientos adquiridos a lo largo del curso del ciclo de Desarrollo de Aplicaciones Web vamos a crear una base de datos que contenga esta información, para que sea fácil y de rápido acceder a esta información.

La idea de este proyecto surge de crear una “Wiki” con la información de estos elementos del videojuego y mostrando como interactúan entre ellos, para hacer la fase de juego mas amena y accesible para jugadores menos experimentados.

Este proyecto surge de uno de los módulos de este ciclo (base de datos), en el cual se nos propone crear una base de datos. Este proyecto tendrá gran importancia tanto en el desarrollo del primer curso como en el segundo, pues podremos utilizar esta base de datos para el proyecto final de ciclo.

La documentación de este proyecto surge de la asignatura de entornos de desarrollo, para desarrollar nuestra capacidad de trabajo en los distintos entornos de desarrollo.

La idea de este proyecto surge del gran impacto que ha generado este videojuego en el último medio año y aunque muchos lo conozcan no hay buenas guías y aplicaciones que nos permitan conocer la información de esta, haciendo del juego una experiencia complicada de entender y puede llegar a ser frustrante por la desinformación de muchos aspectos de este.

FUNCIONALIDADES DE LA APLICACIÓN

Para analizar las funcionalidades de nuestra aplicación debemos conocer los elementos del videojuego y los atributos de estos que queremos guardar.

En primer lugar tenemos habilidades, las cuales pueden ser de dos tipos: la habilidad Q y la habilidad E. Cada habilidad es poseída por tan solo un personaje y se diferencian en que la habilidad Q requiere de una cantidad de energía para ser utilizada.

Estos personajes (que tienen una habilidad de cada tipo) poseen un elemento innato, que se dividen en siete tipos: Pyro (fuego), Hydro (agua), Dendro (Planta), Anemo (Aire), Electro (Rayo), Geo (Roca), y Cryo (Hielo). Un personaje siempre tiene que tener un elemento, y estos elementos los tienen varios personajes.

Los personajes también tienen la posibilidad de llevar tanto artefactos como armas. Las armas, que se dividen en cinco tipos, son llevadas por estos personajes y los personajes pueden llevar un solo arma al mismo tiempo. Por otro lado están los artefactos, que se dividen en cinco tipos y los personajes pueden llevar hasta cinco (un artefacto de cada tipo).

Por otro lado existen materiales de mejora, que se utilizan para mejorar a los personajes y a las armas. Tanto las armas como los personajes necesitan varios materiales para mejorar y estos se consumen al ser utilizados. Los artefactos, sin embargo, se mejoran utilizando otros artefactos, es decir, necesitamos consumir varios artefactos para llevar a un único artefacto a su siguiente nivel.

Queremos guardar la siguiente información de estos:

- De la **habilidad del tipo Q** queremos guardar la energía necesaria para ser utilizada.
- De todas las **habilidades** queremos guardar su identificador, el tipo de habilidad y el tiempo de espera para volver a ser utilizada.
- De los **personajes** queremos guardar un identificador único que los diferencie y su calidad.
- De los **elementos** guardar tanto el identificador de estos como el tipo de elemento de los siete existentes.

- De los **artefactos** queremos guardar su identificador y el tipo de artefacto de los cinco existentes.
- De las **armas** queremos guardar su identificador, la calidad del arma y el tipo de arma.
- De los **materiales** queremos guardar su identificador y el tipo de material.

TECNOLOGÍAS UTILIZADAS

Para generar este proyecto hemos hecho uso de ciertas herramientas, tanto para el modelo relacional y el planteamiento de la base de datos como para propia creación de esta. Este es el listado de las herramientas utilizadas para la creación de esta base de datos y su función:

- **MariaDB:** MariaDB es un sistema gestor de bases de datos (**SGBD**), es decir, un conjunto de programas que permiten modificar, almacenar, y extraer información de una base de datos. Disponiendo de otro tipo de funcionalidades como la administración de usuarios, y recuperación de la información si el sistema se corrompe, entre otras.

MariaDB surge a raíz de la compra, de la compañía desarrolladora de otro (SGBD) llamado MySQL, por la empresa Sun Microsystems. El desarrollador original, decide tomar el código fuente original de MySQL y genera un derivado con mejoras y cambios a los que llama MariaDB. Permitiendo así la existencia de una versión de este producto con licencia GPL (General Public License).

HeidiSQL: HeidiSQL, inicialmente conocido como MySQL-Front, es un software libre y de código abierto que permite conectarse a servidores MySQL (y sus derivaciones como MariaDB y Percona Server), así como Microsoft SQL Server y PostgreSQL.

MySQL-Front comenzó a ser desarrollado en Delphi por el programador alemán Ansgar Becker, quién por motivos personales dejó el proyecto sin terminar. Más tarde el desarrollador alemán Nile Hoyer contactó a Ansgar y adquirió los derechos para utilizar el nombre "MySQL-Front" en su propio proyecto, sin embargo tuvo que cancelarlo porque surgió una infracción de derechos de autor con MySQL Labs sobre el uso del nombre "MySQL". Finalmente, Ansgar y otros colaboradores retomaron el proyecto MySQL-Front renombrándolo HeidiSQL.

Para administrar las bases de datos con HeidiSQL, los usuarios deben iniciar una sesión en un servidor MySQL local o remoto. Sus características permiten realizar las operaciones de base de datos más comunes y avanzadas, sin embargo aún sigue en desarrollo a fin de integrar la máxima funcionalidad que se espera en una interfaz de base de datos de SQL.

- **MySQL:** MySQL es un sistema de gestión de bases de datos relacionales de código abierto (RDBMS, por sus siglas en inglés) con un modelo cliente-servidor. RDBMS es un software o servicio utilizado para crear y administrar bases de datos basadas en un modelo relacional

XAMPP: XAMPP es un servidor independiente de plataforma de código libre. Te permite instalar de forma sencilla Apache en tu propio ordenador, sin importar tu sistema operativo (Linux, Windows, MAC o Solaris). Y lo mejor de todo es que su uso es gratuito.

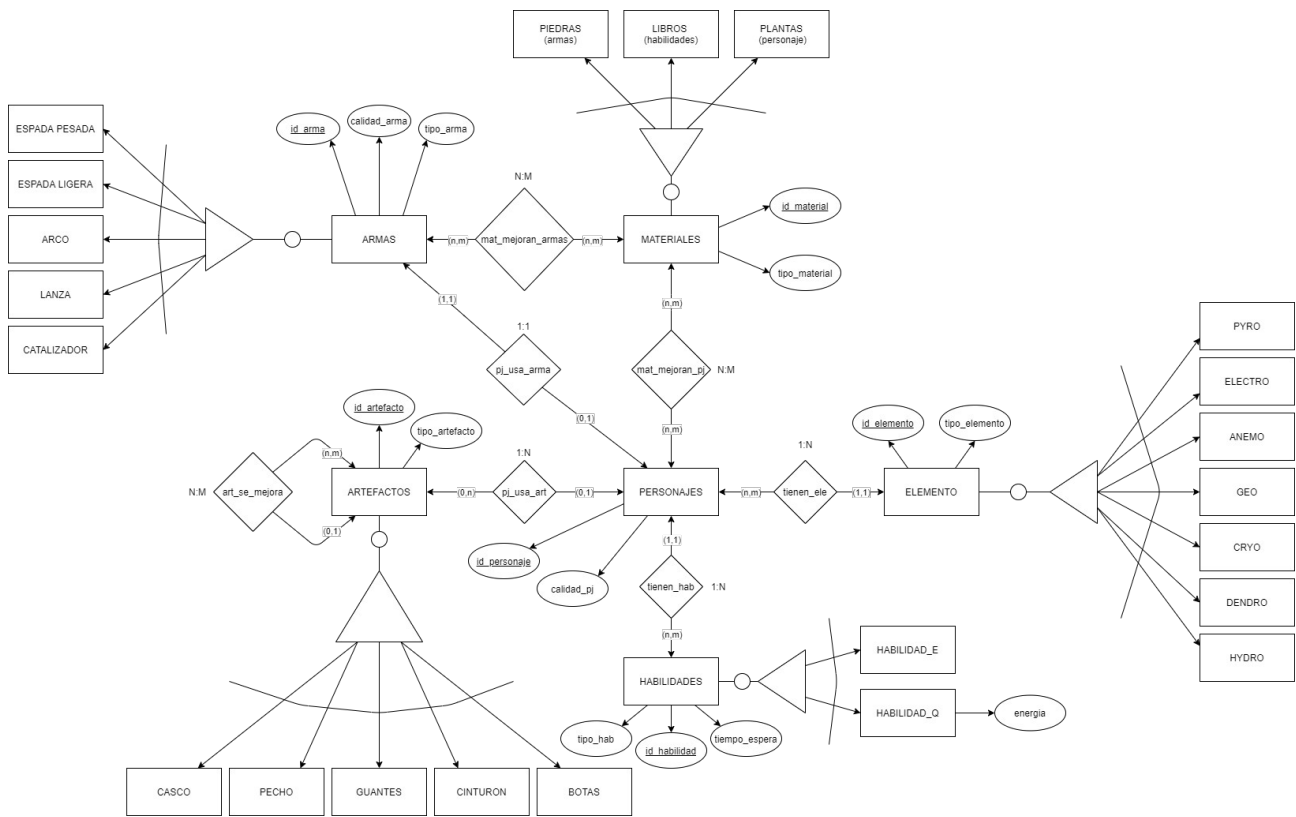
XAMPP incluye además servidores de bases de datos como MySQL y SQLite con sus respectivos gestores phpMyAdmin y phpSQLiteAdmin. Incorpora también el intérprete de PHP, el intérprete de Perl, servidores de FTP como ProFTPD ó FileZilla FTP Serve, etc. entre muchas cosas más.

Si alguna vez has intentado instalar Apache, sabes que no es una tarea fácil, sin embargo con XAMPP todo es diferente. Una de las ventajas de usar XAMPP es que su instalación es de lo mas sencilla, basta descargarlo, extraerlo y comenzar a usarlo.

Diagrams.net: Es una aplicación web gratuita y de código abierto que te permite crear una gran variedad de diagramas desde cualquier navegador web. La herramienta también posee una versión de escritorio disponible para Windows, Linux y macOS.

Con esta herramienta puedes crear y editar una gran variedad de diagramas como: diagramas de flujo, diagramas entidad-relación, diagramas UML, organigramas, diagramas de procesos, mapas mentales, modelos de procesos de negocios, entre otros. Permite agregar imágenes, configurar los textos, elegir el tipo de fuente, cambiar el idioma y además puedes guardar tu archivo para configurarlo después. Asimismo incorpora notaciones matemáticas y capas que facilitan la edición.

MODELO RELACIONAL



MODELO RELACIONAL

HABILIDAD_Q (**id_habilidad**, energia)

HABILIDADES (**id_habilidad**, tiempo_espera, tipo_hab, **id_personaje**)

PERSONAJES (**id_personaje**, calidad_pj, **id_arma**, **id_elemento**)

ELEMENTOS (**id_elemento**, tipo_elemento)

ARTEFACTOS (**id_artefacto**, tipo_artefacto)

ARMAS (**id_arma**, calidad_arma, tipo_arma)

MATERIALES (**id_material**, tipo_material)

MAT_MEJORAN_ARMAS (**id_material**, **id_arma**)

MAT_MEJORAN_PJ (**id_material**, **id_personaje**)

PJ_USA_ART (**id_artefacto**, **id_personaje**)

ART_SE_MEJORA (**id_artefacto**, **id_artefacto_usado**)

CONCLUSIONES

Con este proyecto he aprendido a realizar y gestionar una base de datos simple sobre un videojuego, algo que me ha hecho darme cuenta de la complejidad real que existe a la hora de crear una base de datos de este calibre, pues hay mucha cantidad de información y resulta muy complejo guardarla toda de manera coherente y relacionada entre sus entidades, pues existen una inmensa cantidad de estas.

Por otro lado creo que los desarrolladores de aplicaciones web de este tipo hacen un gran favor a la comunidad de videojuegos, no solo a este en concreto, pues nos facilitan mucho el llegar a cierta información compleja de obtener y que nos resultan muy útiles y pueden hacer de una experiencia frustrante y complicada a algo ameno y divertido.

En conclusión, creo que este proyecto me ha permitido acercarme y darme cuenta del trabajo que lleva realizar un tipo de aplicación que uso con bastante frecuencia y he disfrutado basando el proyecto en una de mis aficiones mas grandes e importantes.