Alberto Ballesteros Rivas

Documentación técnica

Contenido

[¿Qué solución se ha aportado y cómo se ha resuelto? 2](#_Toc61471510)

[Modelos conceptual, lógico y físico 2](#_Toc61471511)

[Ventajas/Inconvenientes de cambio de paradigma de base de datos 4](#_Toc61471512)

[Requisitos fase 1 4](#_Toc61471513)

[Requisitos fase 2 4](#_Toc61471514)

[Requisitos fase 3 4](#_Toc61471515)

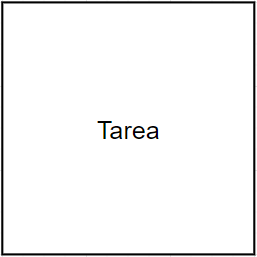
[Requisitos fase 4 4](#_Toc61471516)

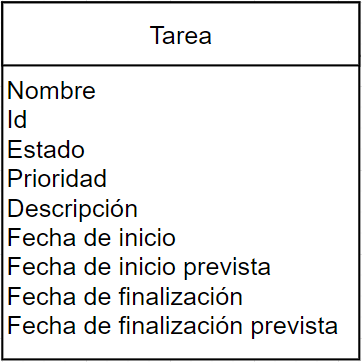
# ¿Qué solución se ha aportado y cómo se ha resuelto?

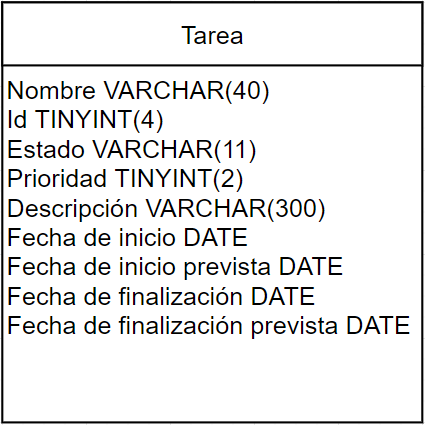
He solucionado la incorporación de una SGBD a mi proyecto utilizando MySQL como base de datos y las sentencias preparadas en php para mostrar la información almacenada en la base de datos.

He solucionado los requisitos de mostrar las tareas por orden de prioridad o alfabético usando las sentencias preparadas adecuadas y php.

# Modelos conceptual, lógico y físico







Tarea

Nombre: VARCHAR(40). El nombre de la tarea, con un máximo de 40 caracteres.

Id: TINYINT(4). El id de la tarea, con primary key y auntoincrement.

Estado: VARCHAR(11). Hay 3 posibles estados: pendiente, en progreso y finalizada. El más largo es en progreso con 11, así que VARCHAR(11).

Prioridad: TINYINT(2). La prioridad que el usuario considere oportuna, del 0 al 10.

Descripción: VARCHAR(300). Una descripción de la tarea que el usuario considere oportuna.

Fecha de inicio: DATE. Fecha en que se inició la tarea.

Fecha de inicio prevista: DATE. En caso de que no se haya iniciado aún, la fecha en la que se planea iniciar.

Fecha de finalización: DATE. Fecha en que se finalizó la tarea.

Fecha de finalización prevista: DATE. En caso de que no se haya finalizado aún, fecha en que se planea finalizar.

He optado por esta solución porque me parecía la más lógica. Crear una sola tabla porque todas son tareas, la entidad es tareas, con distintos atributos como el nombre o el estado. Usar una sola tabla también facilita las cosas a la hora de usar sentencias: para pasar una tarea de una tabla a otra (me refiero a las tablas html aquí), sólo hay que cambiar un atributo.

# Ventajas/Inconvenientes de cambio de paradigma de base de datos

## Requisitos fase 1

Los requisitos son tan simples que casi diría que tiene más desventajas que ventajas el SGBD con respecto a los .txt.

Ventajas: en un futuro cuando sepamos acceder al SGBD desde cualquier dispositivo sin tener que instalar xamp, crear la base de datos, la tabla, etc. sí será una ventaja.

Desventajas: tardas más tiempo creando la base de datos y la tabla que haciendo los ficheros .txt y un while para leerlos.

## Requisitos fase 2

Aquí ya es otro mundo, se te quitan las ganas de volver a usar .txt, el SGBD es mucho más fácil, rápido, eficaz, potente, etc. A partir de aquí ya no le veo ninguna ventaja a los .txt.

La única desventaja del SGBD es que hay que aprender a usarlo.

## Requisitos fase 3

Lo mismo, con el SGBD preparas alguna sentencia y ya lo tienes casi todo hecho, con el .txt tenías que escribir y pensar mucho más.

Todo son ventajas con el SGBD. Y eso que todavía ni sé usarlo prácticamente, no me quiero imaginar cuando sepa.

## Requisitos fase 4

Lo mismo que los anteriores, muy fácil con el SGBD. Todo ventajas, ninguna desventaja. Bueno con el SGBD me siento como al principio con los .txt, y los .txt ya sentía que los dominaba bastante, pero no los echaré de menos.