AGenda Tareas

Documentación Técnica

Moisés Antonio Muñoz Aranda

Contenido

[Modelos conceptual, lógico y físico 2](#_Toc61469871)

[Modelo Conceptual 2](#_Toc61469872)

[Modelo Lógico 2](#_Toc61469873)

[Modelo Físico 2](#_Toc61469874)

[Creación base de datos, tabla e inserción de tareas 3](#_Toc61469875)

[Preparación, ejecución, asignación, lectura y cierre de sentencias 4](#_Toc61469876)

[Comprobación de lectura de tareas DB 5](#_Toc61469877)

[Inserción tareas 7](#_Toc61469878)

[Comprobación de la inserción 8](#_Toc61469879)

[Cambio de estado y Borrado de tareas 9](#_Toc61469880)

[tarea en pendiente: paso a en progreso 9](#_Toc61469881)

[tarea en progreso: paso a finalizadas 9](#_Toc61469882)

[Comprobación cambio estado de tareas 10](#_Toc61469883)

[Borrado de tareas 11](#_Toc61469884)

[Comprobación borrado 11](#_Toc61469885)

[Ventajas/Inconvenientes de cambio de paradigma -> Ficheros vs SBGD 12](#_Toc61469886)

# Modelos conceptual, lógico y físico

### Modelo Conceptual

Las entidades más importantes en el proyecto son:

Tarea, Tareas pendientes, Tareas en progreso, Tareas finalizadas, Id

**Justifico:**

En el proceso de la agenda de tareas y sus requerimientos podemos destacar estas entidades existiendo por sí mismas y siendo contabilizables.

### Modelo Lógico

-Entidad principal:

Tarea

-Atributos:

identificador, tarea que realizar, fecha alta,

estado pendiente, estado en progreso, estado finalizada,

prioridad, borrada, activa

**Justifico:**

Distingo tarea como entidad maestra, asignado los atributos especificados.

### Modelo Físico

Considerando todos los atributos de la tarea anteriormente mencionados

DB: agenda

TABLA: tareas

COLUMNAS Y TIPOS DE DATO:

id: identificador único // BIGINT AUTO\_INCREMENT PRIMARY KEY

tarea: tarea que realizar // VARCHAR(255) NOT NULL

fhalta: fecha de alta // TIMESTAMP DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP

estado: pendiente o enprogreso o finalizada // CHAR(10) NOT NULL DEFAULT 'pendiente'

prioridad: comun o urgente o absoluta // CHAR(10) NOT NULL DEFAULT 'comun'

borrado: no=1 o si=0 // CHAR(1) NOT NULL DEFAULT 1

**Justifico:**

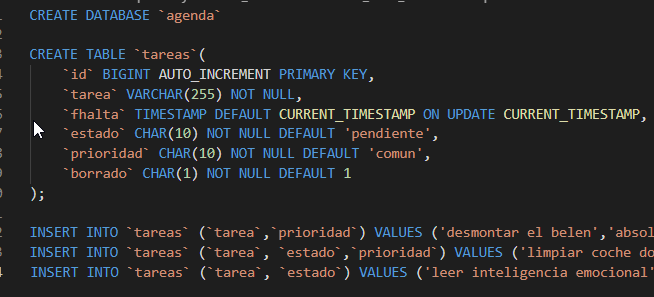
Por simplicidad, compacidad y eficiencia la DB contendrá una sola tabla. Con un id tipo BIG INT, generado automático por AUTO\_INCREMENT.

Fecha de alta: generada automáticamente, en el momento de la creación del registro.

Estado, prioridad y borrado: selección por defecto. Posibilidad cambio.

# Creación base de datos, tabla e inserción de tareas

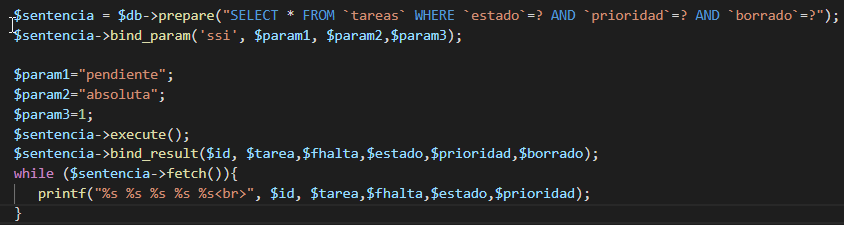
El programa contiene un archivo llamado db\_tabla\_inserciones.sql para la creación de la Base de datos, tabla y algunos registros de prueba.



# Preparación, ejecución, asignación, lectura y cierre de sentencias

El mostrado de las tareas por orden de prioridad, se tiene en cuenta en las consultas:

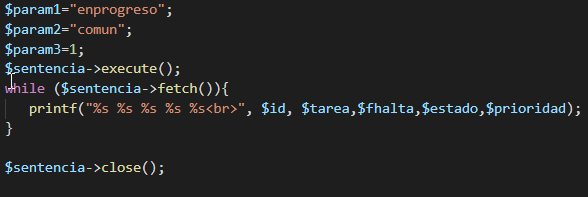
* Preparo sentencia para lectura estado: pendiente, prioridad: absoluta, borrado:1 ejecuto, asigno resultados a variables e imprimo por pantalla.



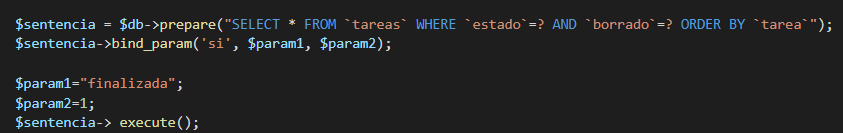
También realizo para estado: pendiente, borrado:1, prioridad: urgente y prioridad: común.

* Ejecuto asigno columnas a variables e imprimo por pantalla resultados de misma sentencia para:

tareas estado: enprogreso, prioridad: absoluta, prioridad: urgente y prioridad: común, borrado:1

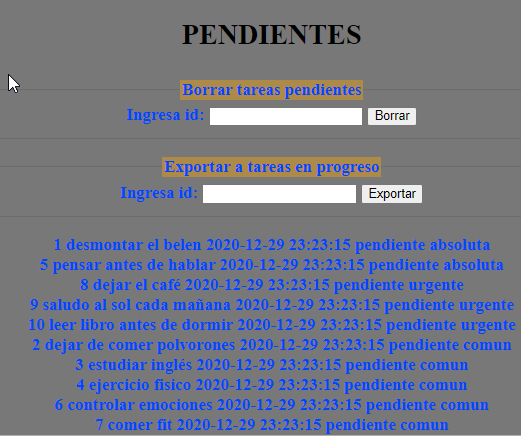


Para la lectura de las tareas finalizadas, mostramos por orden alfabético ascendente:

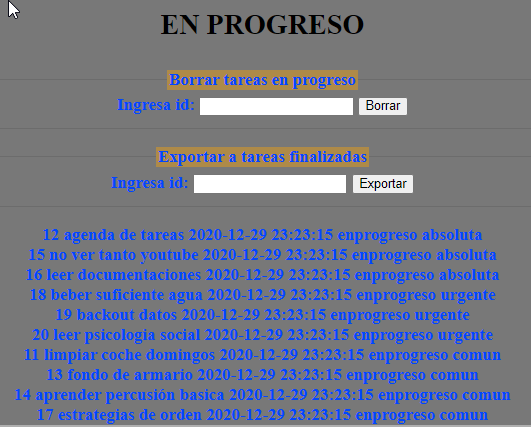


### Comprobación de lectura de tareas DB

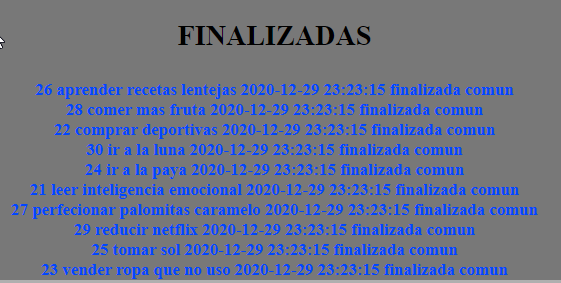
* Lectura tareas pendientes por orden de prioridad



* Lectura tareas en progreso por orden de prioridad

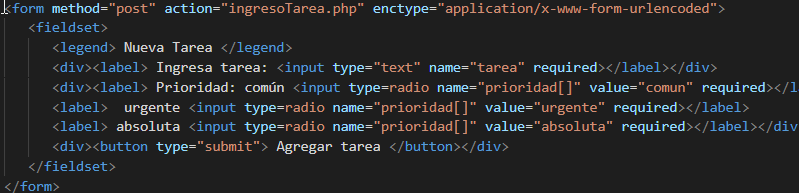


* Lectura por orden alfabético ascendente

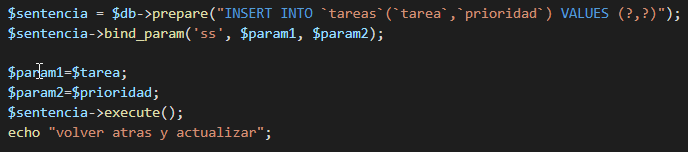


# Inserción tareas

Creación de formulario y recopilación de datos necesarios: tarea y prioridad



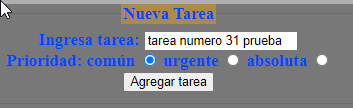
Campos id y fhalta son generados automáticamente, para los campos: estado y borrado el dato es generado por defecto (se especificó en el tipo de dato de la tabla tareas)

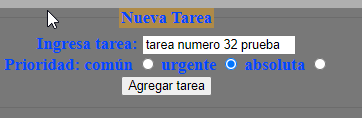


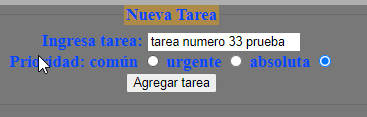
En referencia al validador de fases anteriores del proyecto:

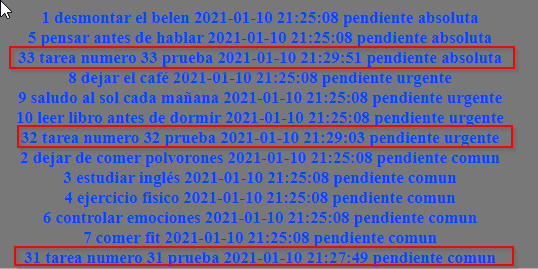
Las tareas tendrán identificador único, id comenzará 1, no existirán huecos y no se repetirán debido al tipo de dato especificado para este campo.

### Comprobación de la inserción







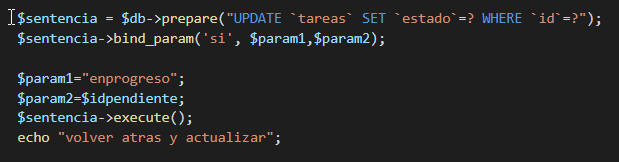


# Cambio de estado y Borrado de tareas

### tarea en pendiente: paso a en progreso

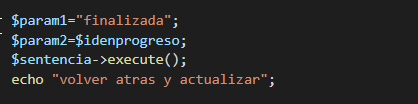
Conectamos con la DB, preparamos consulta y actualizamos el campo correspondiente al id de tarea pendiente recibido en formulario, nuevo estado:

enprogreso.



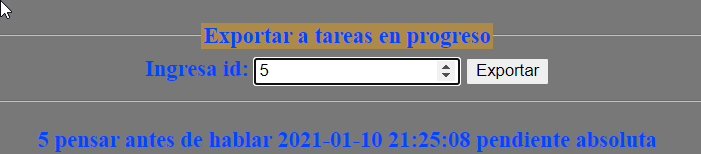
### tarea en progreso: paso a finalizadas

Mediante la misma consulta actualizo registro perteneciente al id de tarea en progreso recibido, nuevo estado: finalizada



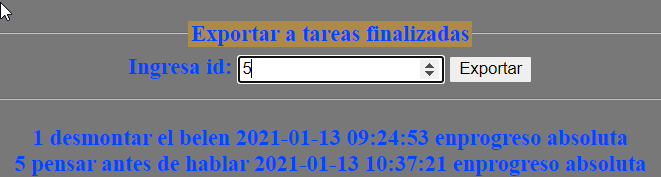
### Comprobación cambio estado de tareas

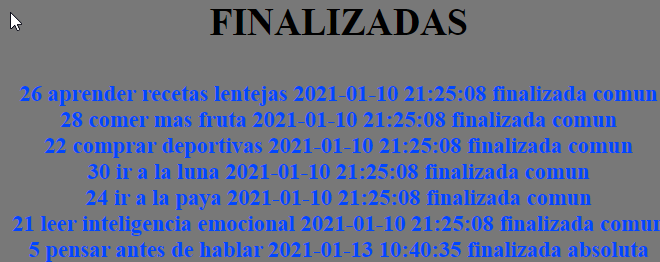
* De pendientes a en progreso





* De en progreso a finalizadas

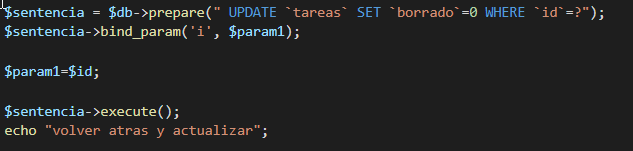




### Borrado de tareas

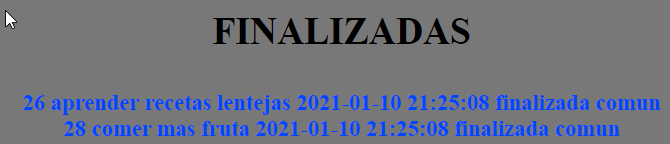
Preparamos BD, preparamos consulta, actualizamos campo borrado = 0,

De esta manera no mostraremos en la lectura de nuestra DB, realizamos un borrado lógico.



### Comprobación borrado

* Borraremos tarea id 26







## Ventajas/Inconvenientes de cambio de paradigma -> Ficheros vs SBGD

Requisitos

Fase1

Existencia de 3 ficheros planos

Mostrar 3 listas diferenciadas

Mostrar datos de ficheros

* Ante estos requerimientos cabe decir: la compacidad del SGBD es mayor a los ficheros.

Fase2

Ordenadas por id

Crear tareas pendientes desde aplicación

Comprobar ids tareas correctos

* Ante estos requerimientos cabe decir: El SGBD aporta facilidad para la creación de un identificador único, que no se repita, no nulo.
* La creación de tareas aún más sencilla, y con más detalles, lo haremos con una sencilla inserción.

Fase3

Borrar pendientes y en progreso

Pasar tareas pendientes a en progreso y en progreso a finalizadas

Dar estilo

* Ante estos requerimientos cabe decir: lo realizaremos fácilmente con una actualización de campo en nuestro registro seleccionado.

Solicito dato identificador del registro a modificar id, en el caso de borrado, actualizo borrado =0, en el otro actualizo estado = pendiente, estado =progreso, estado=finalizada.

Fase4

Finalizadas orden alfabético ascendente

Pendientes y en progreso orden de prioridad

* Ante estos requerimientos cabe decir: El filtrado de los registros se hace muchísimo más sencillo que de otra forma, con el uso del lenguaje SQL tratamos los datos de nuestra base de datos ágilmente.

A DESTACAR:

El equipo de desarrollo no ha detectado ninguna Ventaja del uso de ficheros como método de almacenar Información.

El SGBD supera en todos los aspectos, y contempla muchas funcionalidades inexistentes en los ficheros.