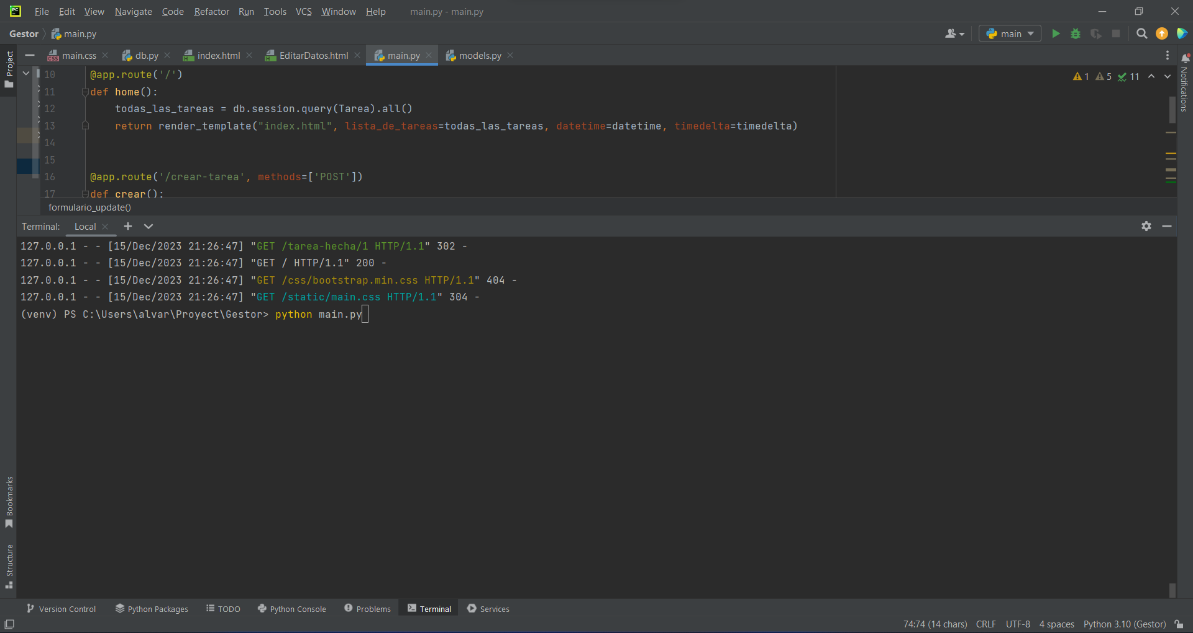
**Práctica de creación de una APP web con base de datos**

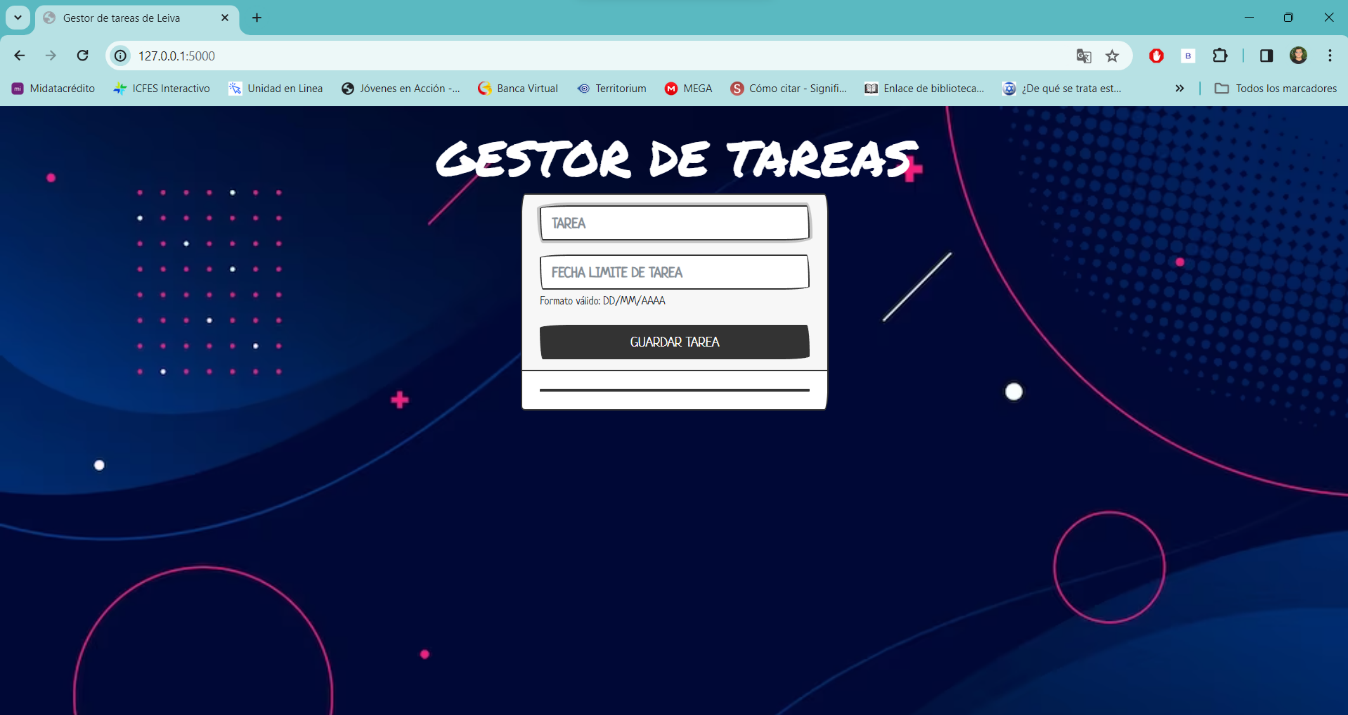
El siguiente informe tiene como objetivo, mostrar las funcionalidades de la aplicación web avanzada en el entorno de desarrollo integrado Pycharm, con la ayuda del framework Flask así mismo se aplicó el modelo de programación de base de datos ORM (Sqlalchemy), que nos permite tener mayor facilidad al momento de interactuar con los datos que nos proporciona el usuario.

Partiendo de lo anterior, se van a proporcionar imágenes, que nos van a permitir tener un primer acercamiento, a la aplicación web de *GESTOR DE TAREAS.*

**

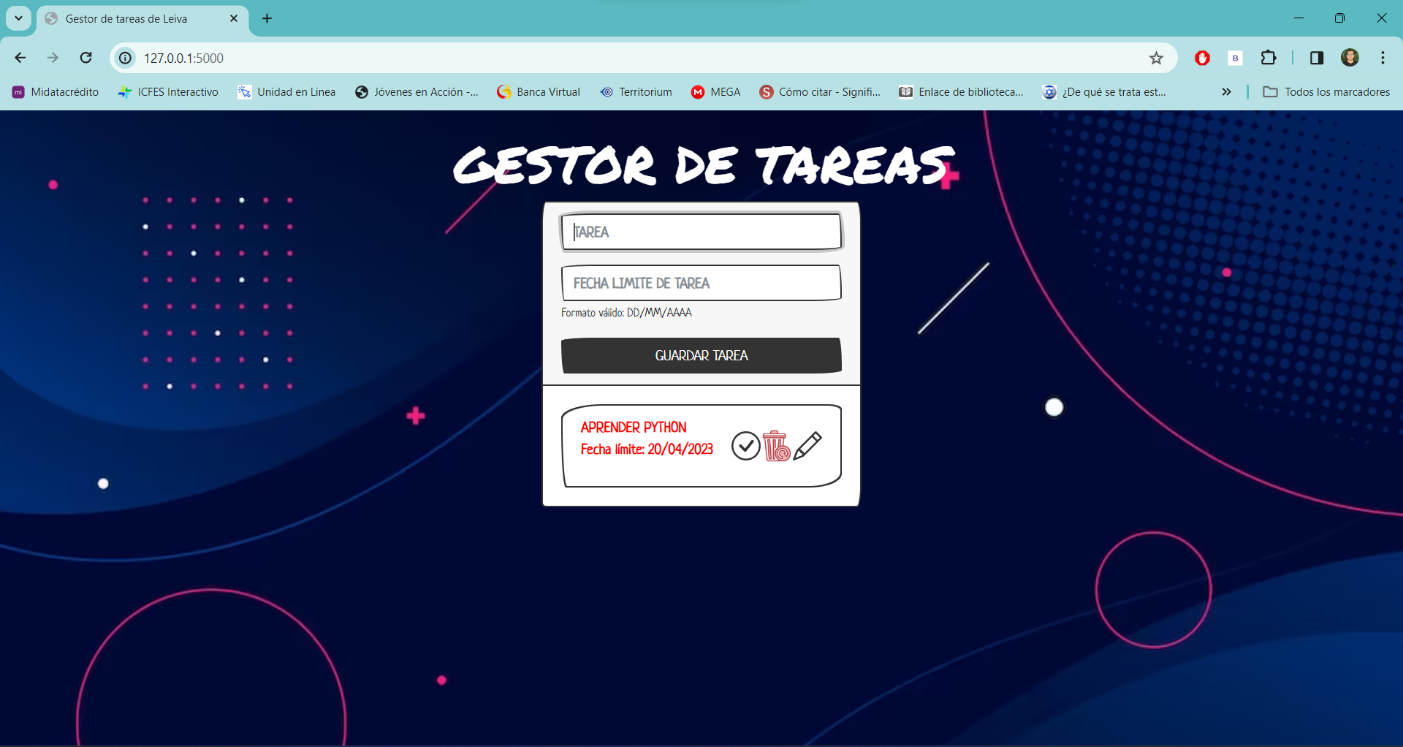
En esta imagen se evidencia el modo de arranque de la aplicación web, donde se nos proporciona el medio de acceso a su funcionamiento.

<http://127.0.0.1:5000/>

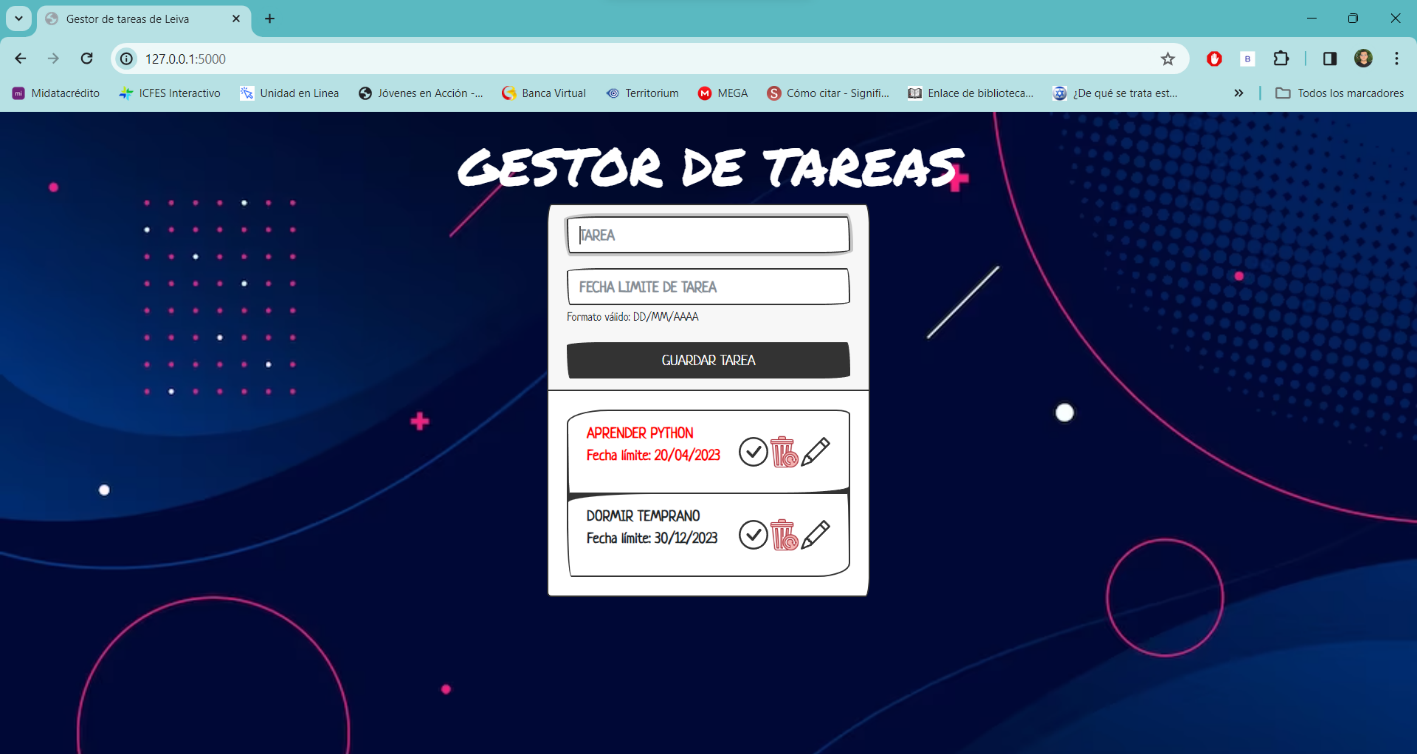
**Página principal**

**CREAR TAREA**

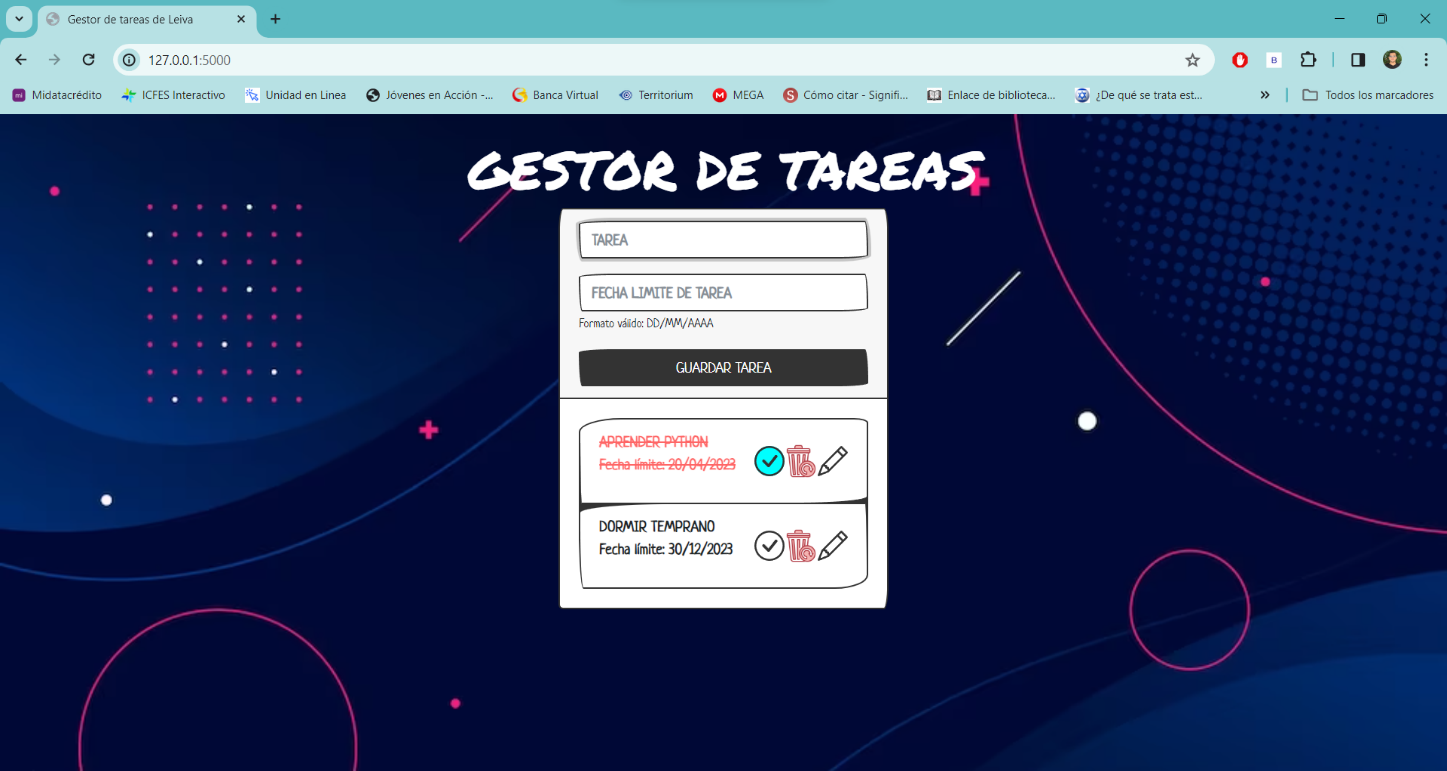
Se genera la tarea, nuestra tarea generada sale en rojo, dado que gracias a Jinja podemos validar, si el tiempo restante para realizar, es decir, la fecha limite esta cerca por dos días, o la fecha limite en tiempo real, ha quedado atrás en el tiempo.

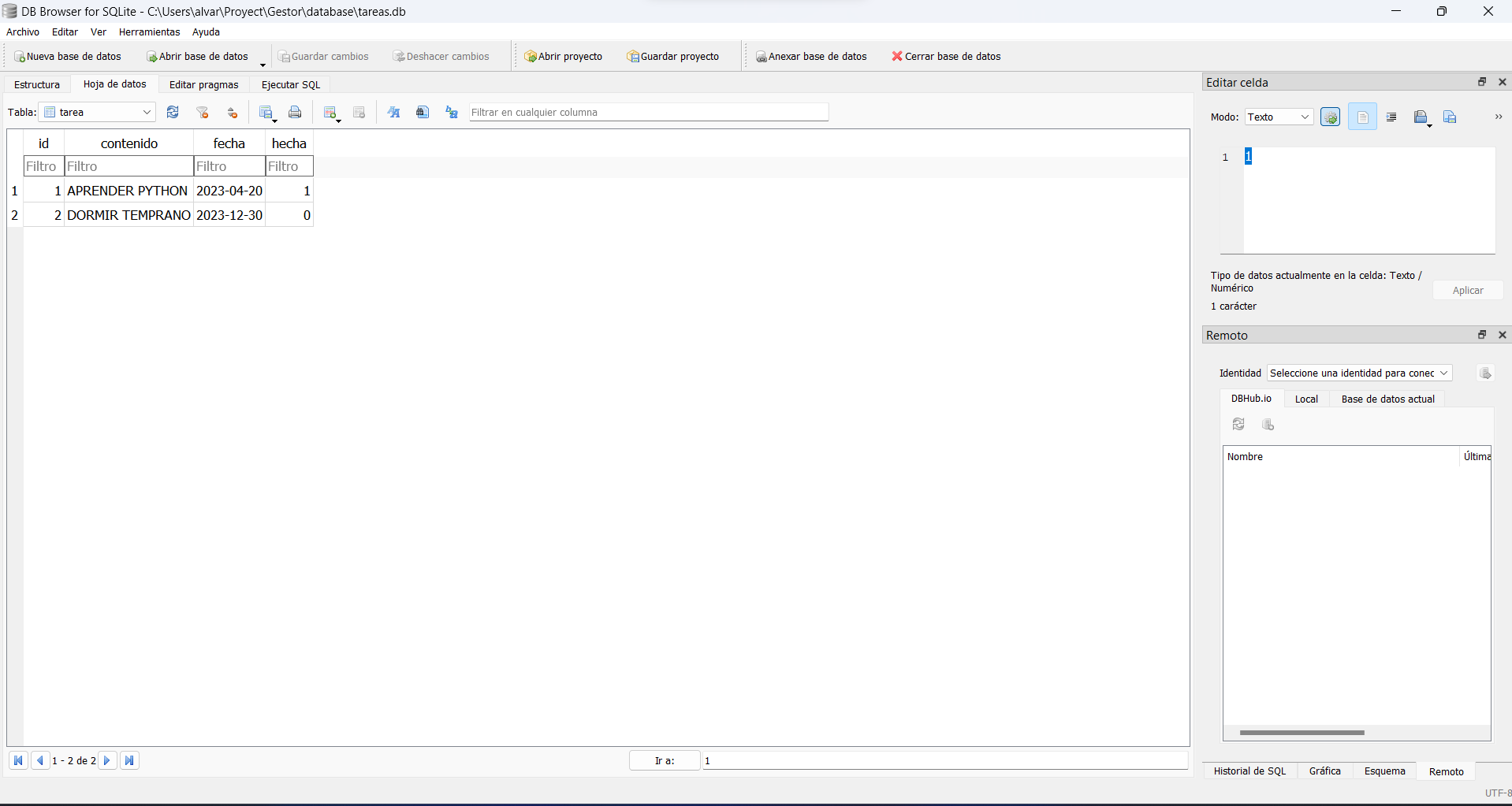


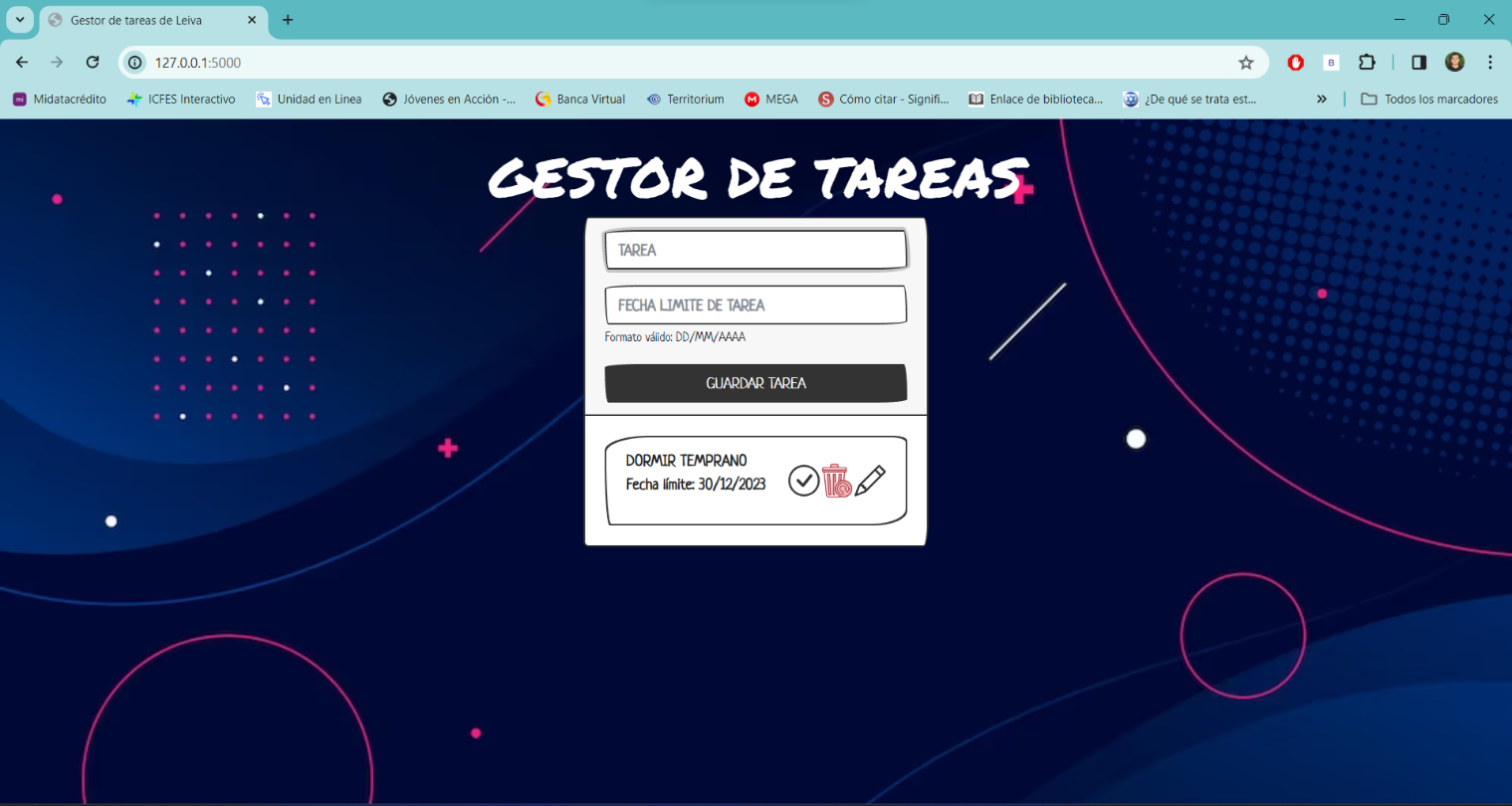
En la imagen que viene se puede evidenciar que al generar otra tarea fuera de los tiempos del condicional, esta sale en color negro, lo que podíamos considerar una tarea normal, y la que se ve en rojo, como una tarea próxima.

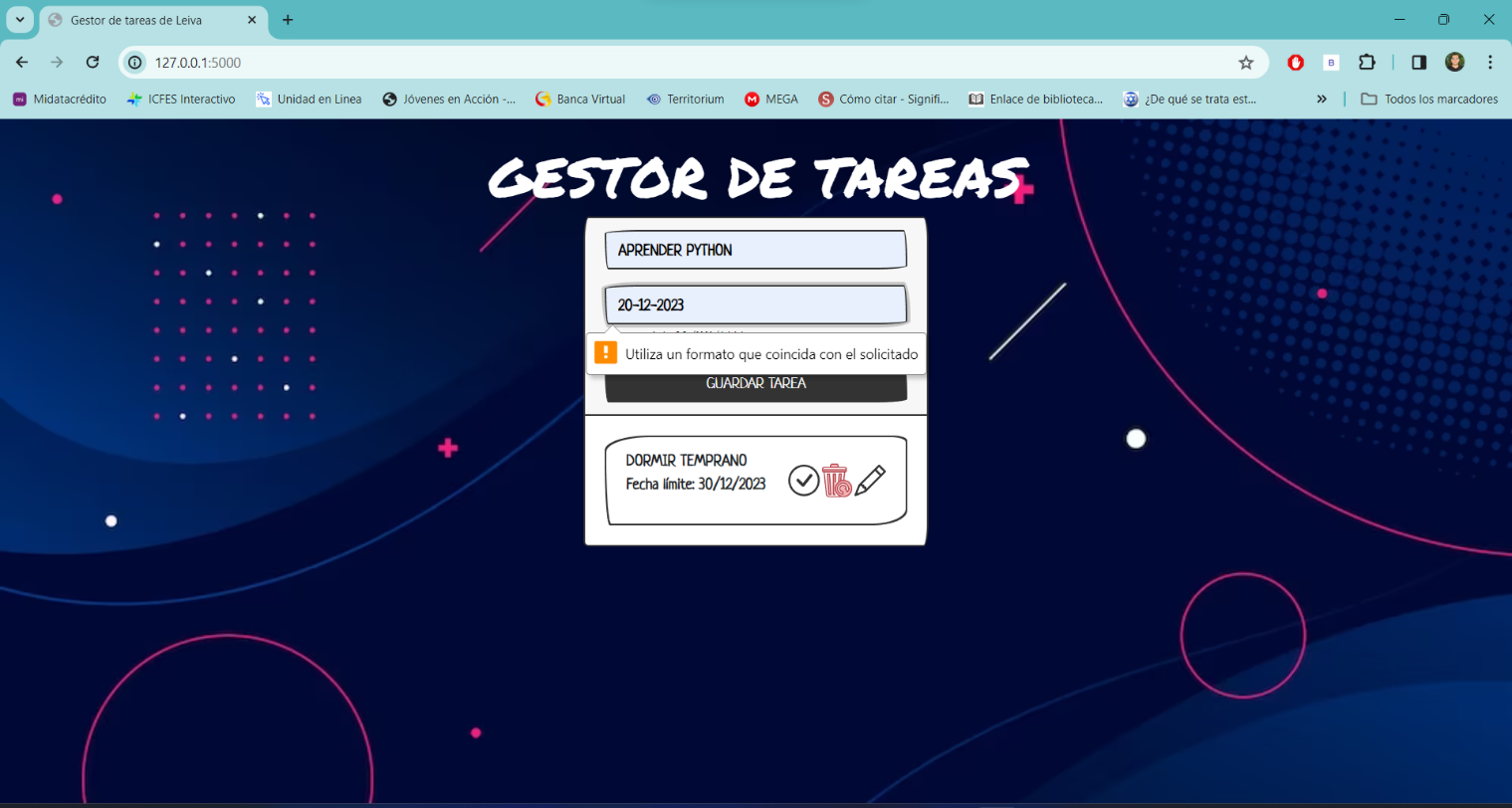


**TAREA HECHA**

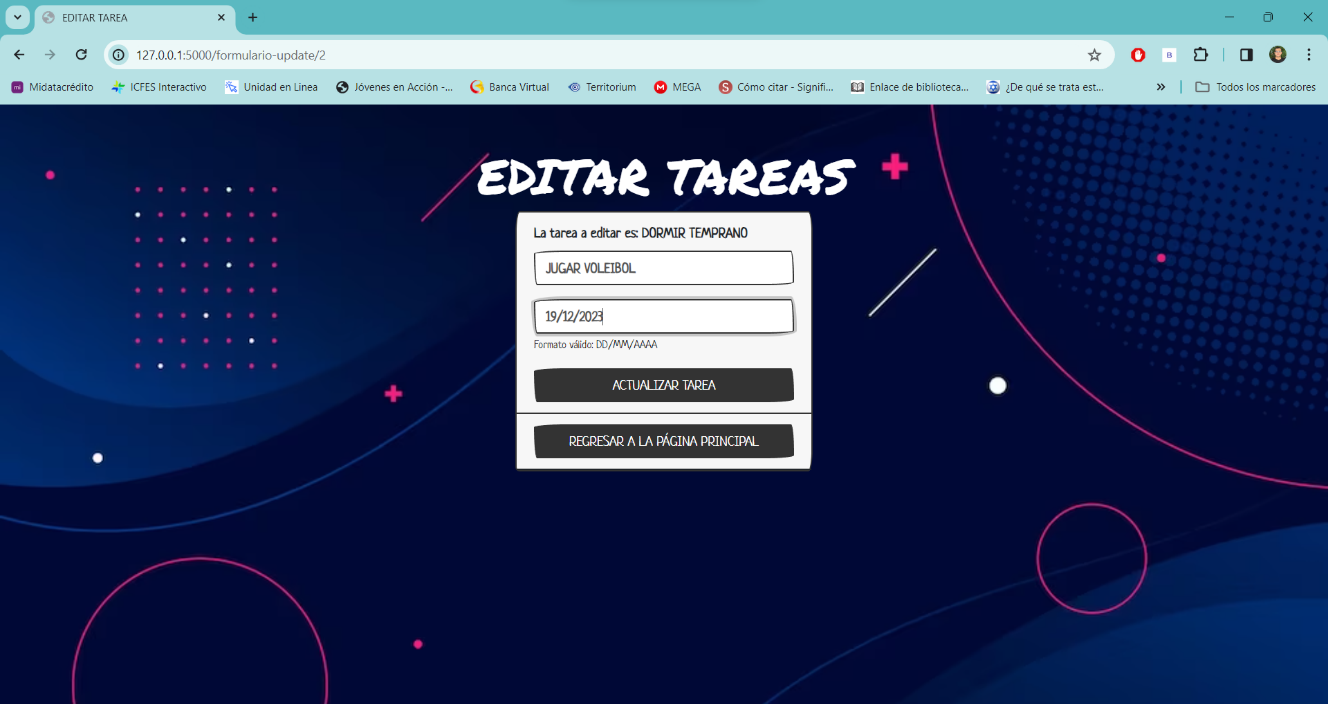
Cuando se selecciona el icono del círculo, este pasa de estar vacío a ser coloreado por color Cian, que se configuro con números hexadecimales, además de cambiar el icono, la tarea es subrayada con una línea del color equivalente al texto de la tarea, adicionalmente, la tarea es opacada.

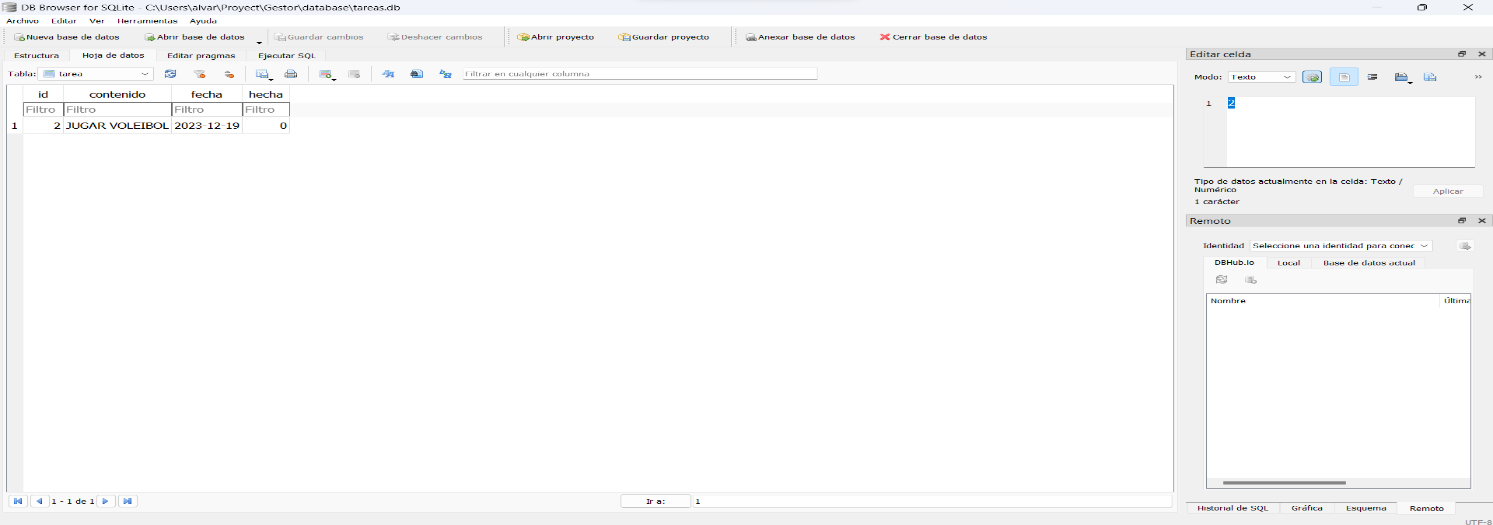
Al mirar la base de datos, podemos ver las columnas que conforman nuestra DB, que en este casi es id, que se configura como PRIMARY KEY, con tipo de datos integer, también el contenido de la tarea que esta configurada como un Varchar(200), la columna fecha, configurada como dato NUMERIC y por ultimo. Al tener en cuento lo realizado en la aplicación web, podemos ver que coinciden tales acciones, por ejemplo, la columna hecha, que ha sido declarada como Booleana, podemos ver que en la primera tarea esta en 1, es decir, en verdadero, tal como quedo en la ultima actividad realizada en la APP.

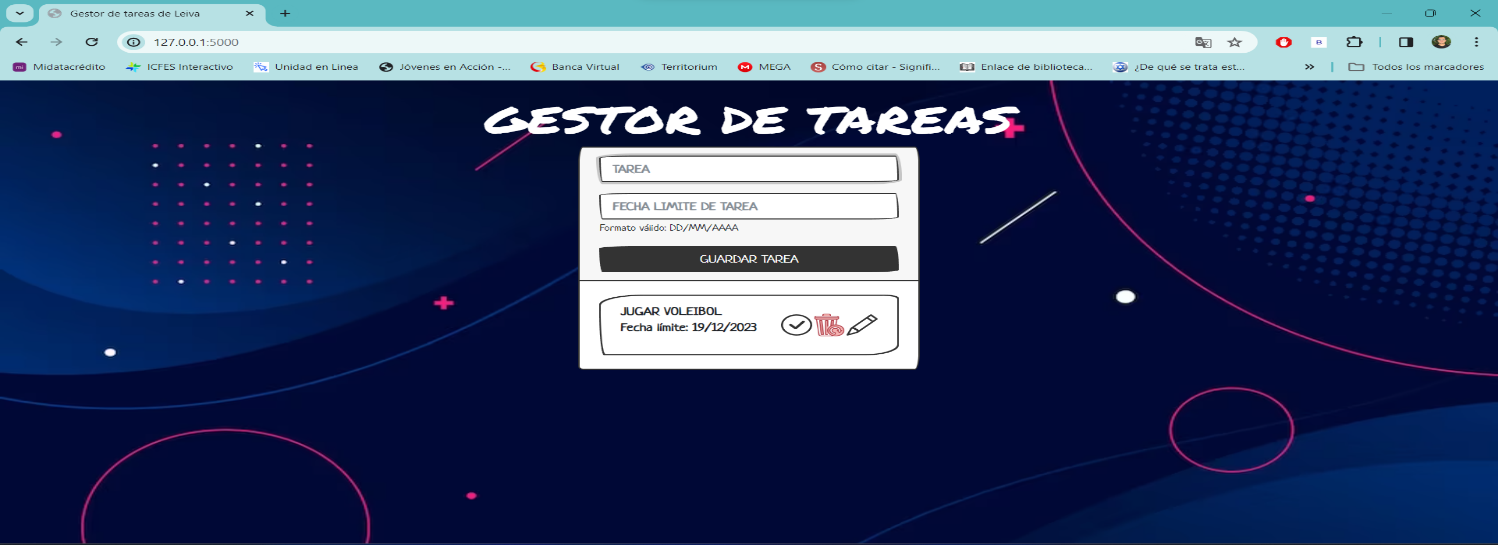
**ELIMINAR TAREA**

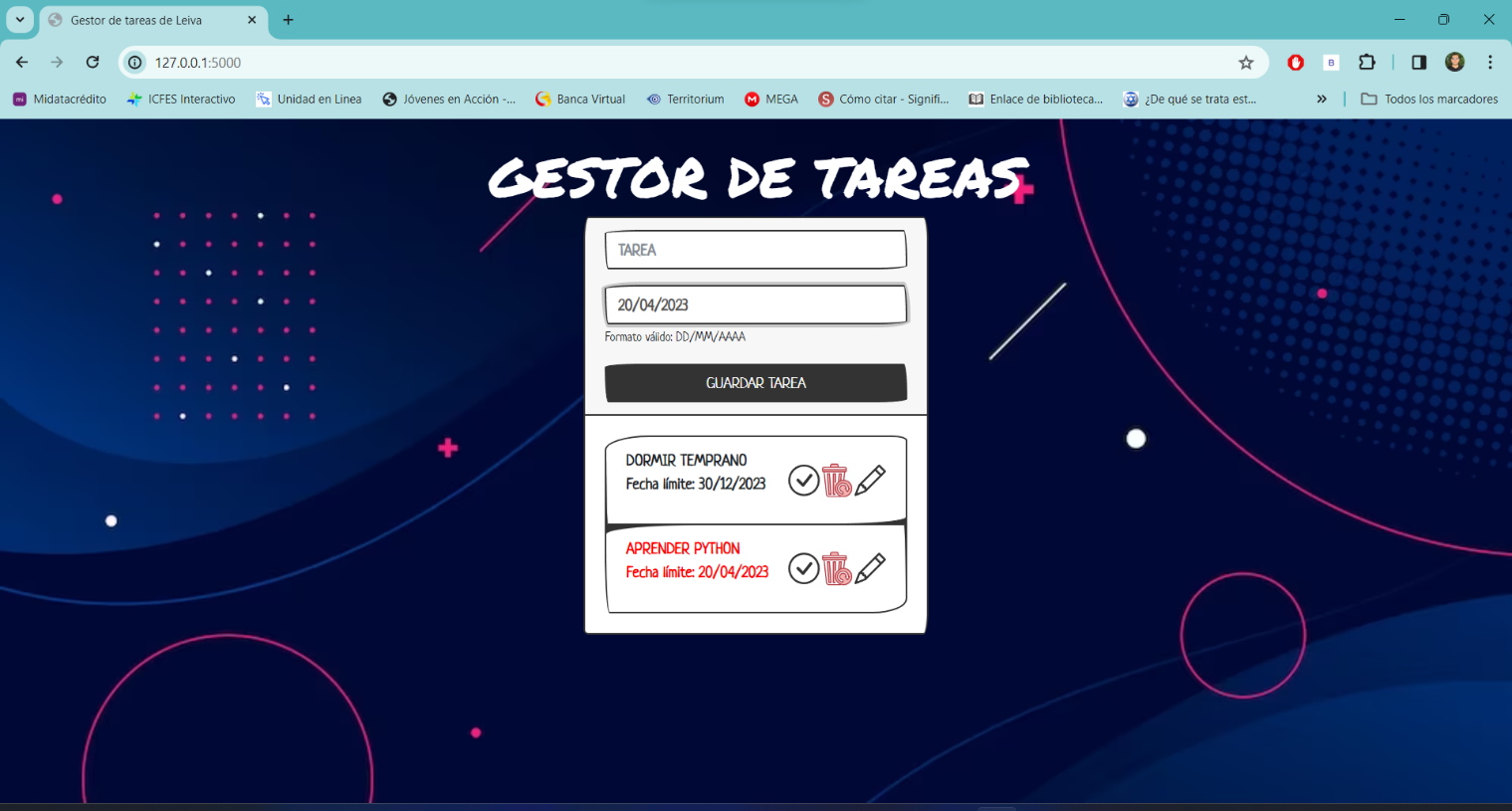
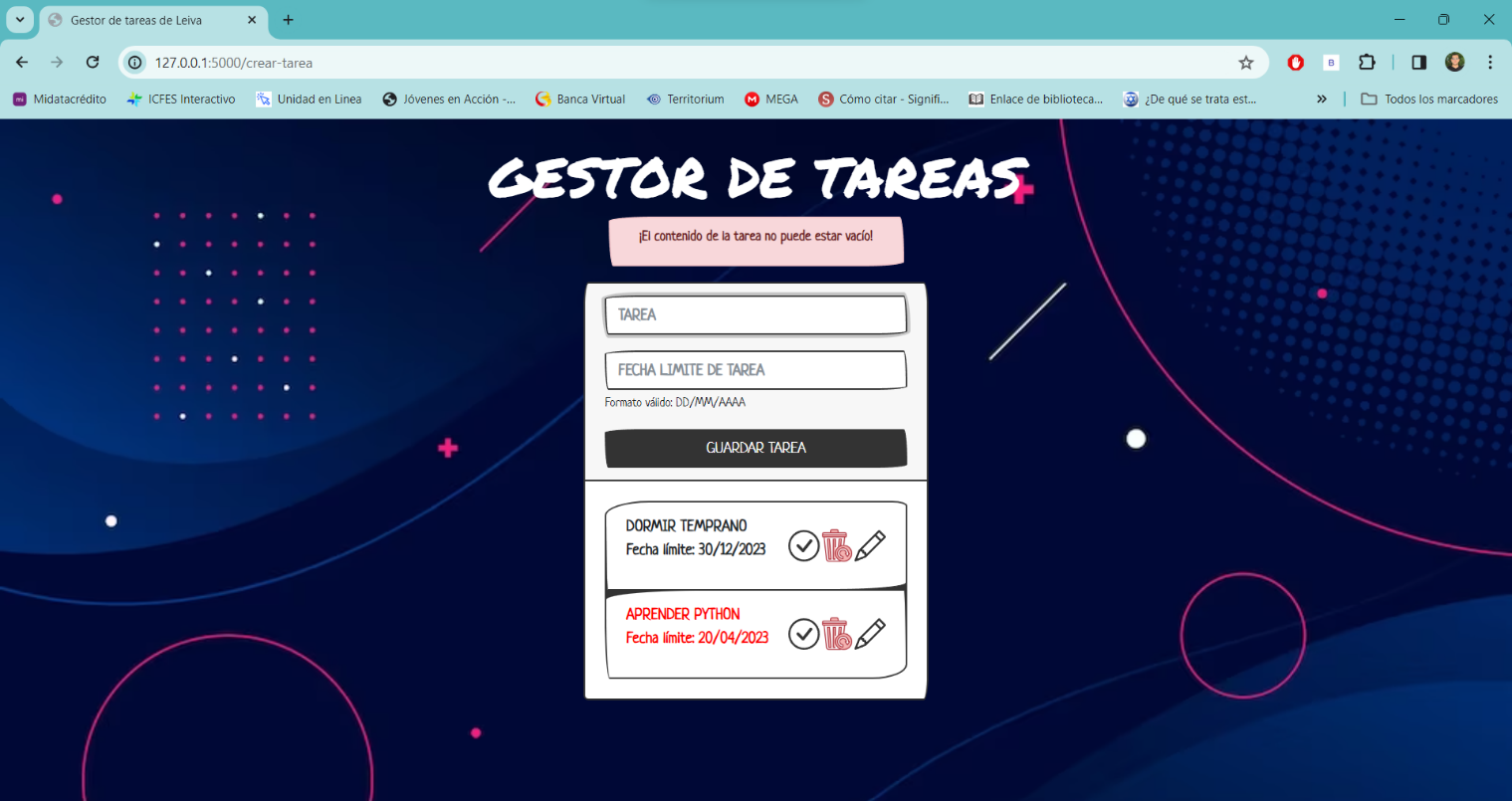
Dentro de la parte gráfica, se evidencia una forma de una canasta de basura, la cual tiene como función principal, eliminar la tarea deseada, tomando como filtro del id de la tarea.

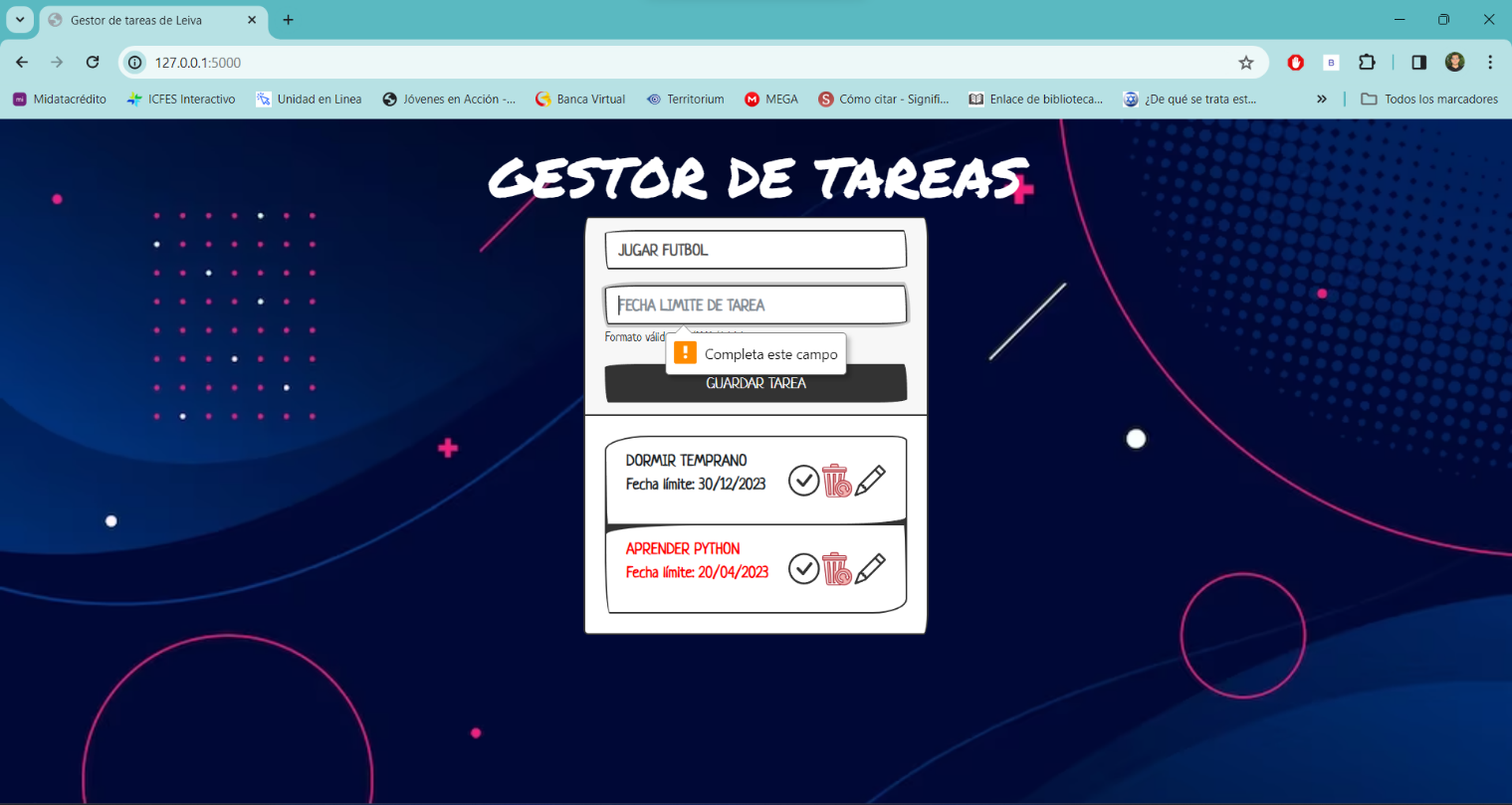
En la siguiente imagen, se puede ver que se estable un formato para el ingreso de la fecha, esto para no presentar problemas con el tipo de datos que se debe ingresar a la base de datos.

En la aplicación de este formulario, se hizo necesario otro archivo HTML, para gestionar los datos nuevos, de tal manera que al momento del usuario pulsar en el icono del lápiz, se traslada a otra página para realizar la acción deseada, cabe recalcar que, dentro del formato, el usuario podrá ver el nombre de la tarea a editar.

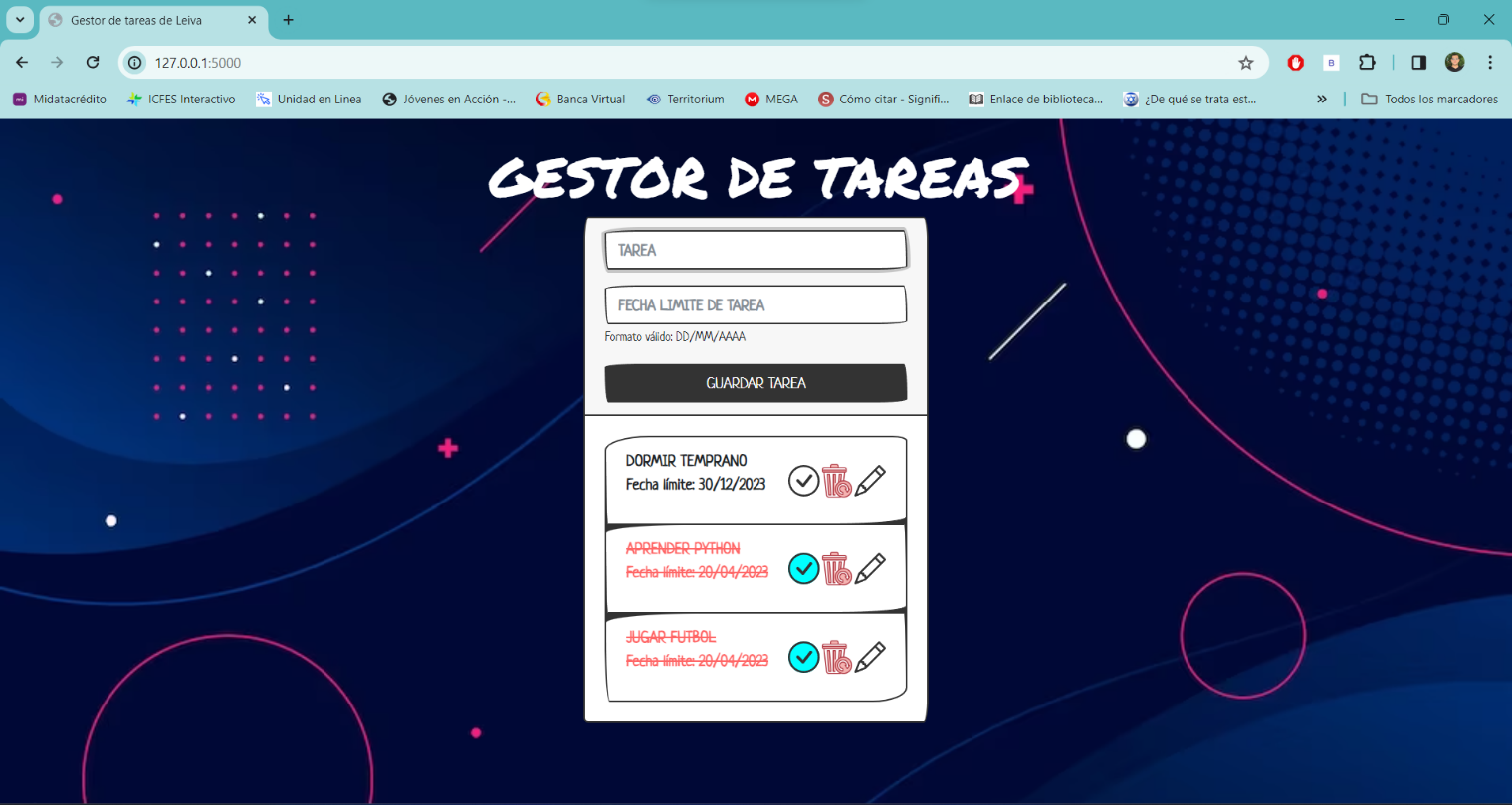
****Al usuario llenar los datos nuevos y pulsar el botón ACTUALIZAR TAREA, esta se va a reflejar en la base de datos y así mismo en la página principal, también se le da la opción al usuario de regresar a la página principal, dado el caso de cambiar de parecer respecto a la edición.

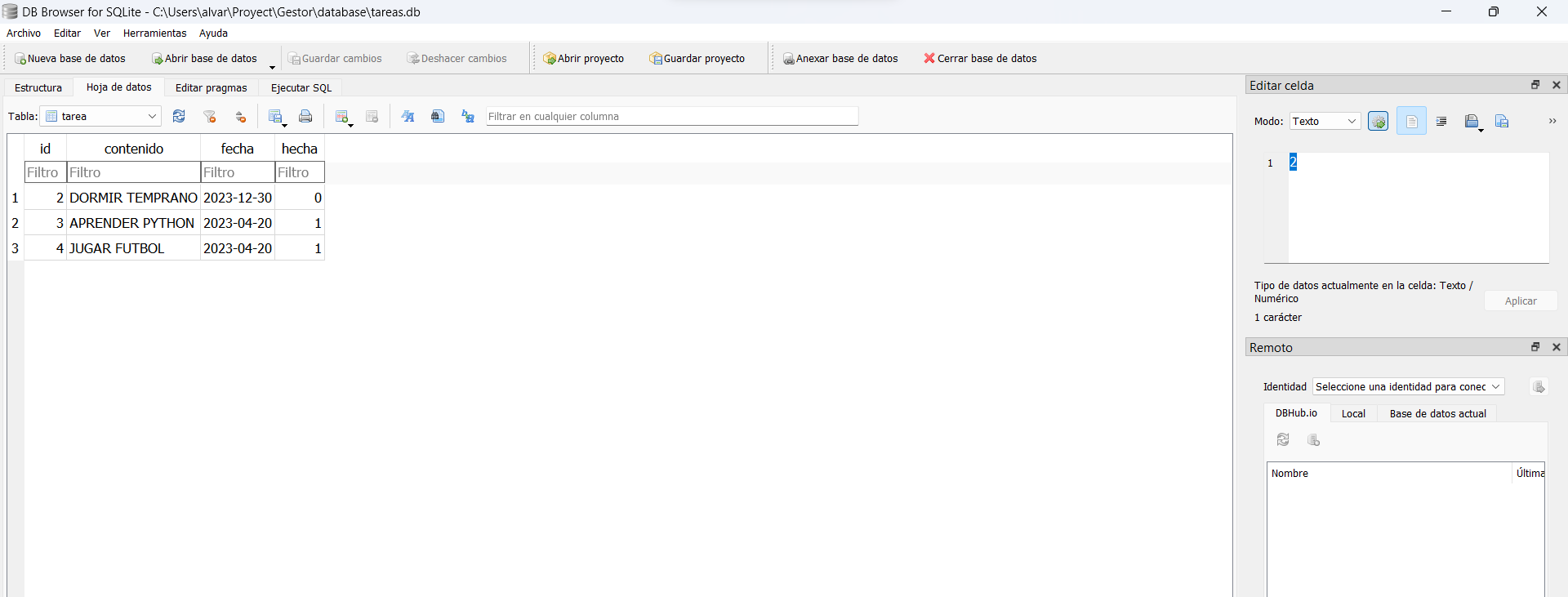


En las siguientes capturas de pantallas, se muestra el condicional que da un mensaje de error, esta verificación advierte al usuario que no puede crear tareas si no llena el formulario de la tarea, es decir, sino asigna el nombre, esto va a permitir el óptimo funcionamiento entre los datos de la tarea y la base de datos, dado que, en la configuración de esta, se dejó establecido que la columna de los nombres de las tareas no aceptará datos nulos.

En esta imagen, se muestra también la importancia de llenar el formulario de la fecha límite, para que el usuario pueda crear bien la tarea deseada.

Por último, creamos varias tareas y las asignamos como hechas y posteriormente vamos a la base de datos para ver el reflejo de lo realizado en nuestra aplicación.





Nota: No realice la opción de categoría, dado que no me pareció necesaria, considero que tarea pendiente siempre será una tarea por hacer, sin importar la categoría que esta tenga.