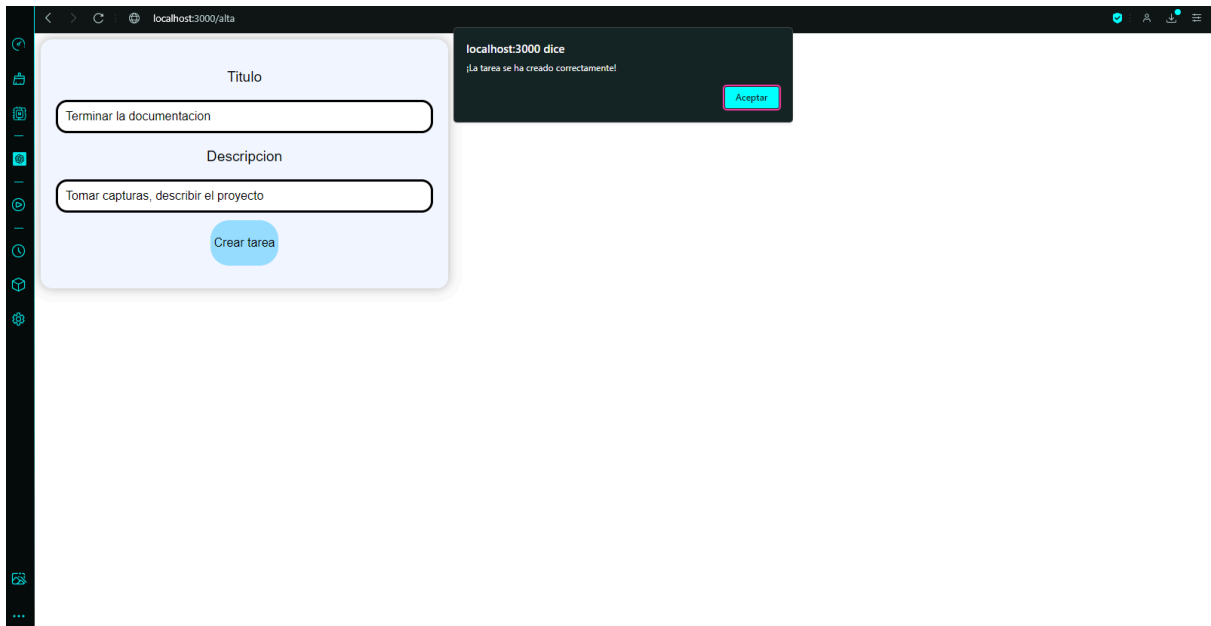


# TODO APP Ejemplos de uso y capturas

Esta es la pantalla principal, donde por defecto se nos cargan todas las tareas consumiendo el servicio GET de la api, cada tarea cuenta con su título, descripción, y el estado, si esta completada o no, aquí tenemos tres botones para manipular cada tarea, el botón azul permite marcarla como completada o no completada este estado se actualiza gracias al servicio PUT, el rojo permite eliminar la tarea y el amarillo permite editar la tarea, además tenemos un div fixed para agregar una nueva tarea:



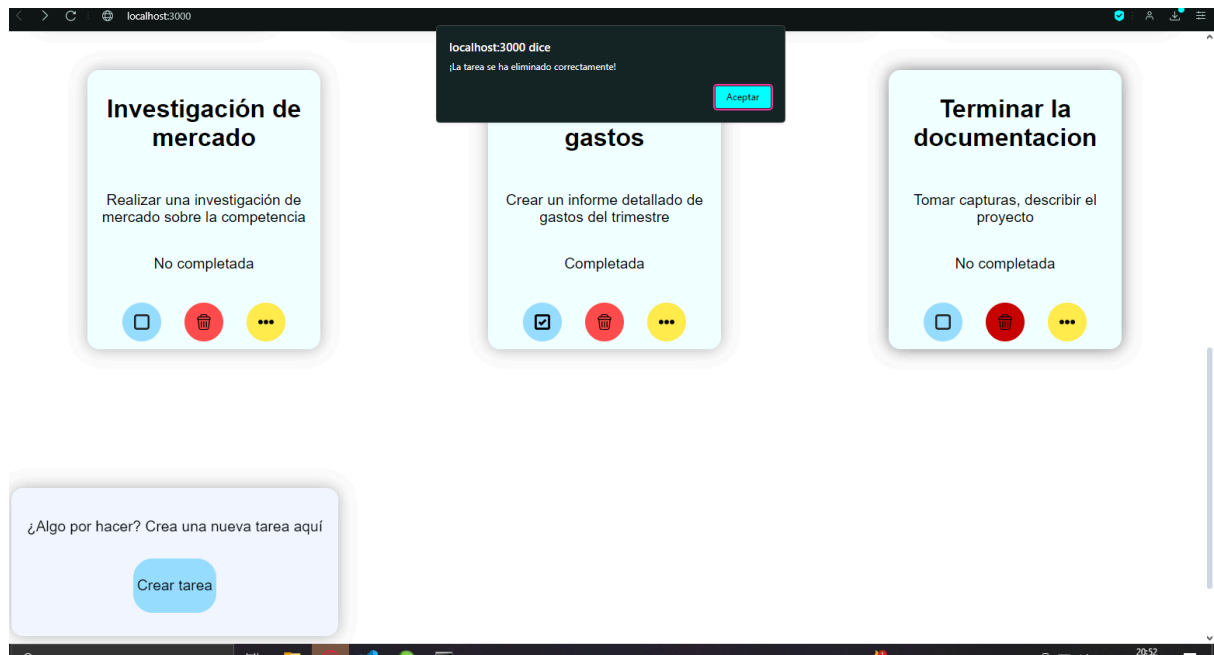
Cuando usamos el botón de crear tarea nos lleva a esta pantalla con el siguiente formulario, el cual solo acepta el título y la descripción de la tarea, ya que la fecha es generada automáticamente, este dato puede modificarse en la opción de editar, aquí consumimos el servicio POST:



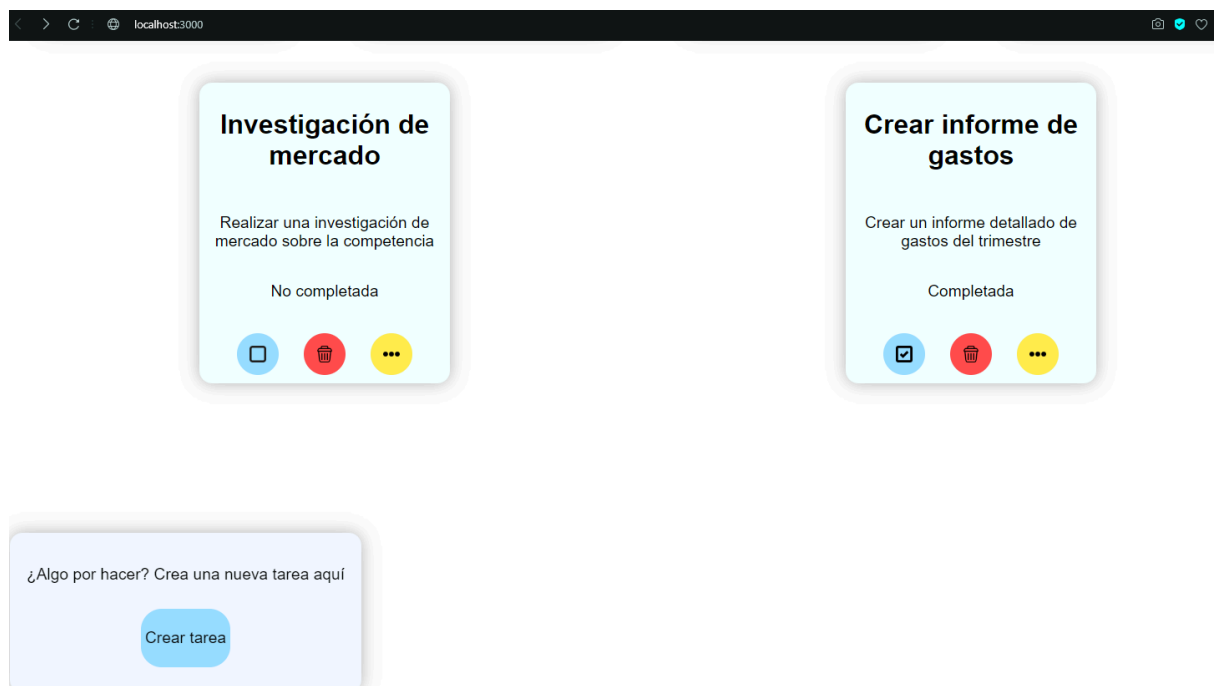
Volviendo al dashboard podemos encontrar la nueva tarea:



Intentemos eliminarla:



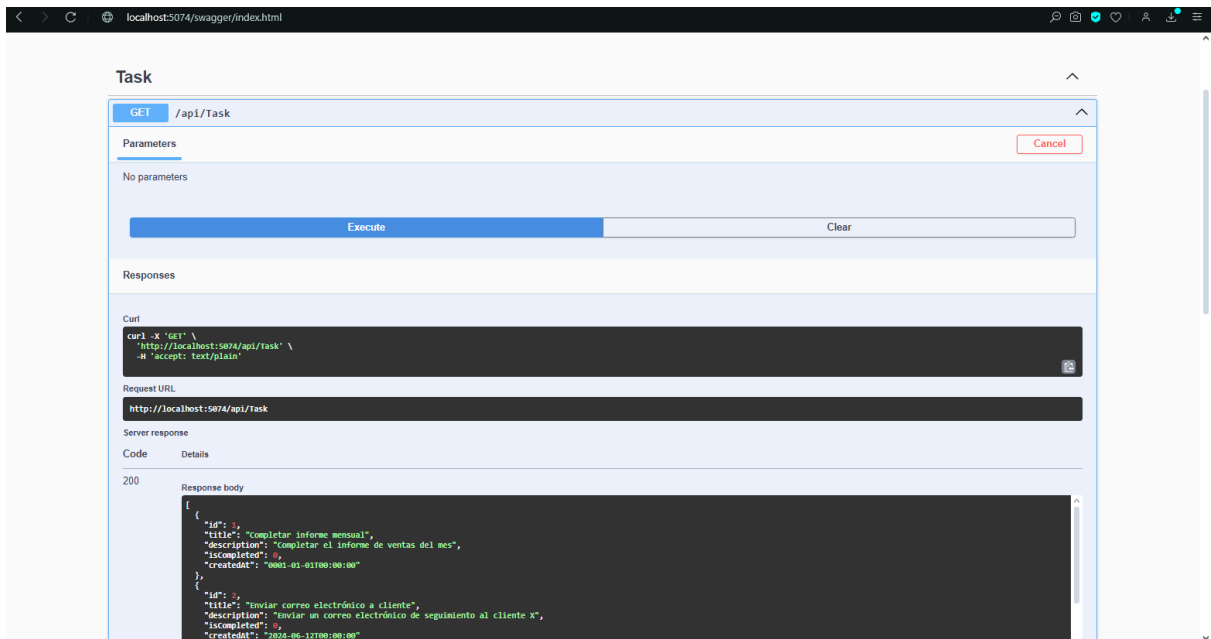
la tarea se ha eliminado:



Ahora vamos a editar la primer tarea, por defecto, en el formulario se cargan los datos de la tarea, usando GET, y para la modificación de los datos usamos PUT:



Además se usó swagger para verificar el correcto funcionamiento de la API:



## Servicios y modelo de datos:

The image shows a Swagger UI interface in a web browser at localhost:5074/swagger/index.html. The top section displays a 200 OK response with a media type of text/plain. Below this, there are four API endpoints listed with their respective HTTP methods: POST /api/Task, GET /api/Task/{id}, PUT /api/Task/{id}, and DELETE /api/Task/{id}. The bottom section, titled 'Schemas', shows a TaskDataModel with the following fields: id, title, description, iscompleted, and createdat, each with a dropdown arrow indicating further details.

200 OK No links

Media type:

Controls: Accept header:

Example Value | Schema

```
{
  "id": 0,
  "title": "string",
  "description": "string",
  "iscompleted": 0,
  "createdat": "2024-06-09T02:53:44.362Z"
}
```

**POST** /api/Task

**GET** /api/Task/{id}

**PUT** /api/Task/{id}

**DELETE** /api/Task/{id}

**Schemas**

**TaskDataModel** {

- id > [...]
- title > [...]
- description > [...]
- iscompleted > [...]
- createdat > [...]

}