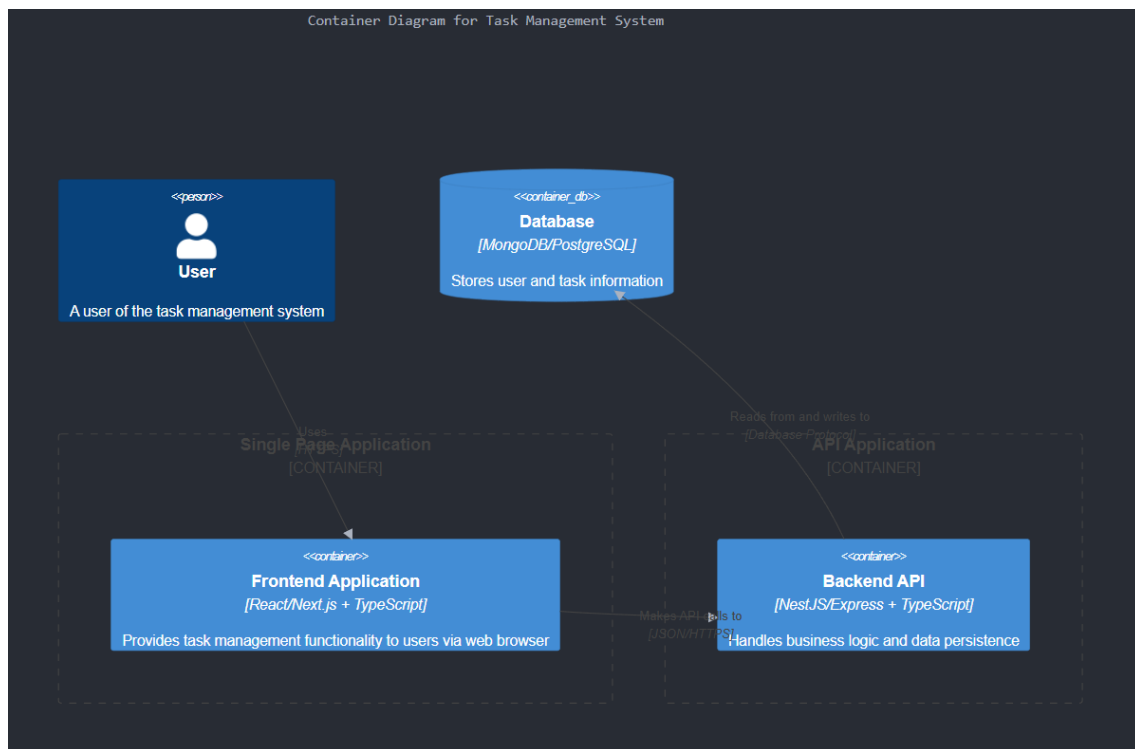


# Explicación de cada diagrama

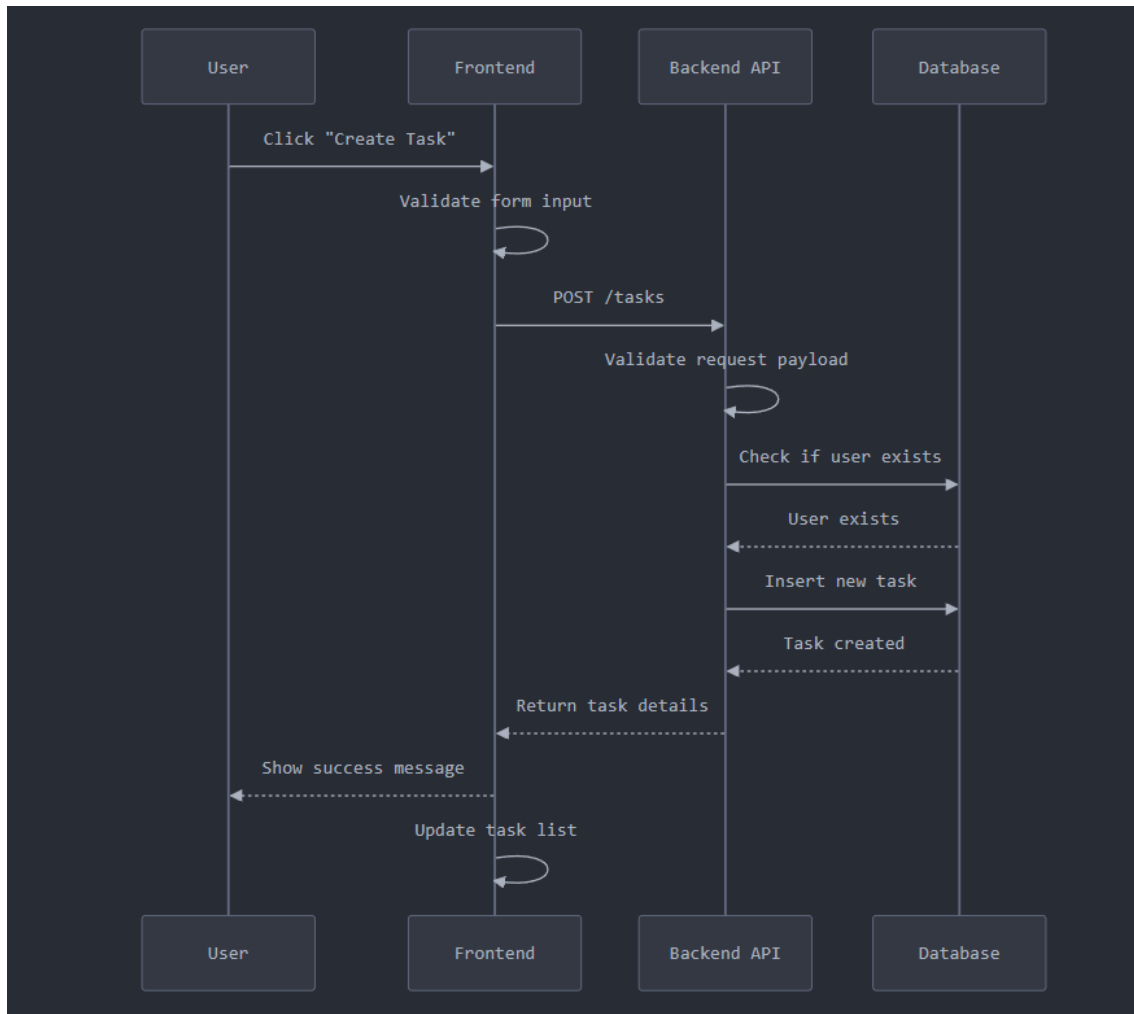
## Diagrama de contenedor C4 (Nivel 2):

**Componentes principales:** Frontend SPA, Backend API y Base de datos. Ilustra las relaciones y protocolos entre componentes. Incluye al usuario como actor externo.

Muestra claramente la pila de tecnología utilizada en cada contenedor.

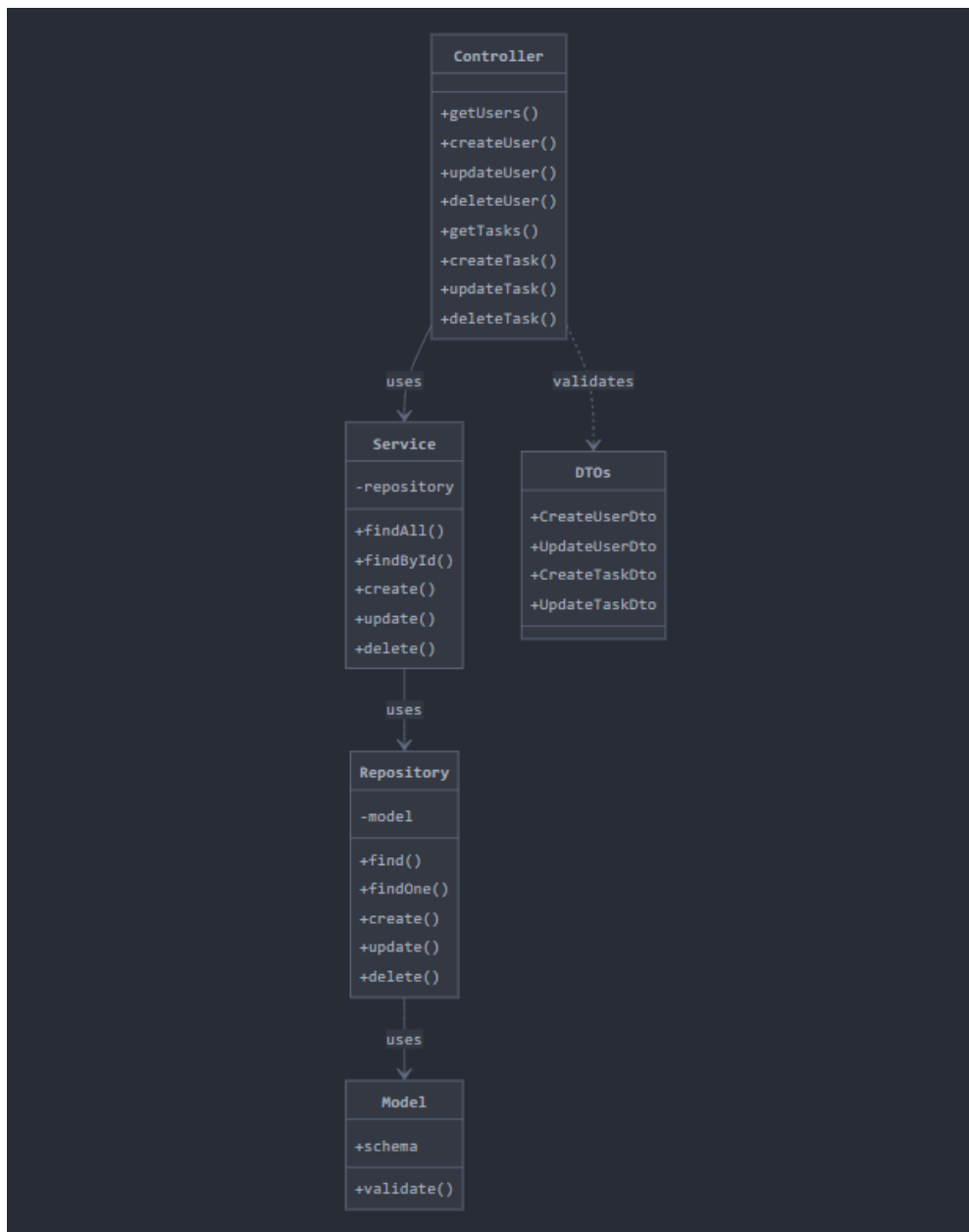


**Diagrama de secuencia:** Demuestra el flujo de creación y asignación de una tarea. Muestra los pasos de validación tanto en el frontend como en el backend. Incluye interacciones con bases de datos. Demuestra manejo de errores y retroalimentación de éxito.



## Patrones de diseño y diagrama de arquitectura

Patrón MVC/Arquitectura Limpia (capas Controlador-Servicio-Repositorio) Patrón de repositorio para la abstracción de acceso a datos Patrón DTO para validación y transferencia de datos. Demuestra una clara separación de preocupaciones. Muestra dependencias entre componentes. La arquitectura sigue estos principios clave: Arquitectura en capas para la separación de preocupaciones Patrón de repositorio para abstracción de bases de datos Patrón DTO para validación y transformación de datos. Capa de servicio para lógica empresarial Capa de controlador para manejo de solicitudes.



## Instalación de NestJS:

`npm i -g @nestjs/cli`

`nest new backend`

`cd backend`

### Instalación de dependencias:

*npm install @nestjs/mongoose mongoose*

*npm install @nestjs/typeorm typeorm pg*

*npm install class-validator class-transformer*

### módulos, controladores y servicios necesarios:

*nest generate module users*

*nest generate controller users*

*nest generate service users*

*nest generate module tasks*

*nest generate controller tasks*

*nest generate service tasks*

**users/users.controller.ts**

**users/users.service.ts**

**DTOs:** Crea un archivo **user.dto.ts** en la carpeta **users/dto** para definir los DTOs.

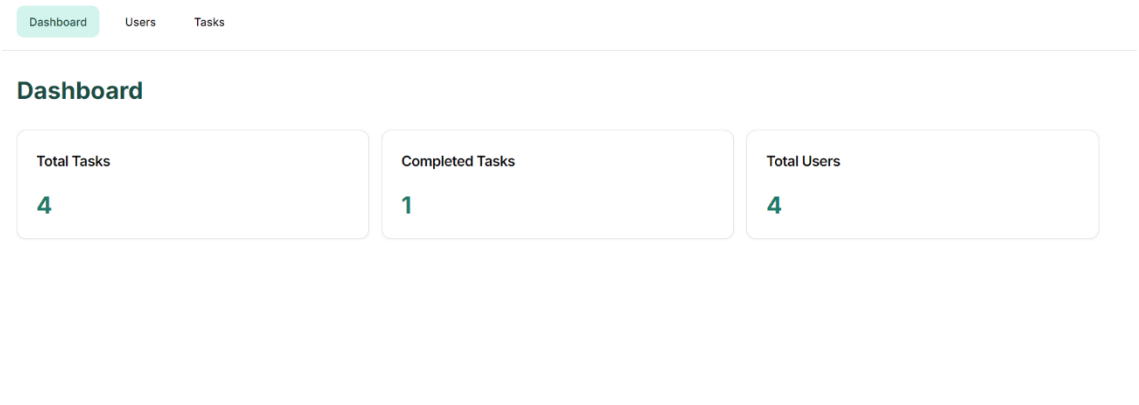
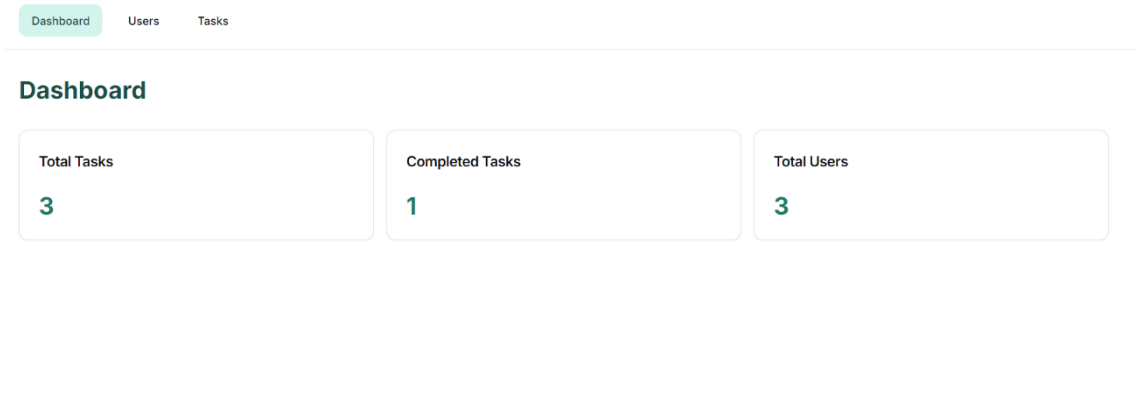
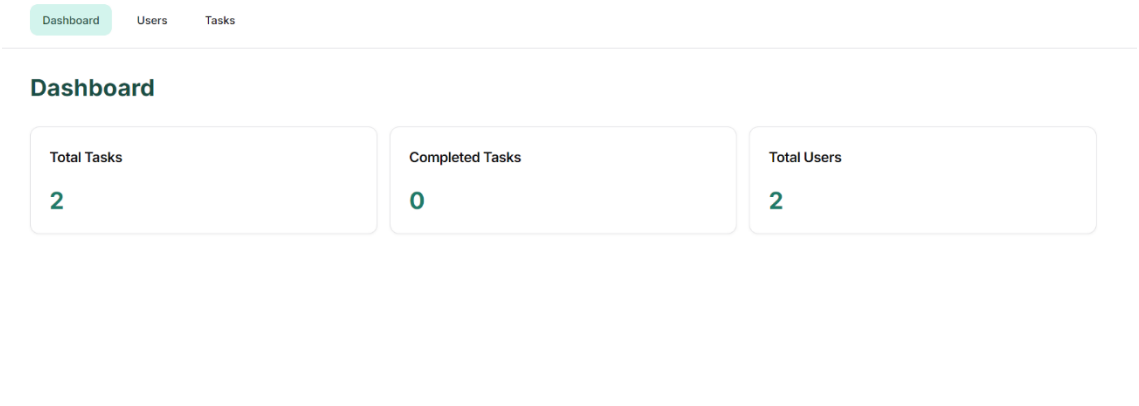
### Implementación de Tareas

Un enfoque similar para las tareas, creando un controlador y un servicio para **tasks**.

**tasks/tasks.controller.ts:**

**tasks/tasks.service.ts:**

# Imágenes



# Usuarios

Dashboard

Users

Tasks

Users

+ Add User

Juan Pérez

juan@example.com

Joined: 28/12/2024

Maria García

maria@example.com

Joined: 28/12/2024

Maria García

Add New User

X

Name

Email

Create

Dashboard

Users

Tasks

Users

+ Add User

Juan Pérez

juan@example.com

Joined: 28/12/2024

Maria García

maria@example.com

Joined: 28/12/2024

andres O

andresv@gmail.com

Joined: 28/12/2024

jose

jose@gmail.com

Joined: 28/12/2024

# Tareas

Dashboard

Users

Tasks

Tasks

+ Add Task

Completar proyecto

Finalizar el proyecto de desarrollo web

Assigned to: Juan Pérez

Created: 28/12/2024

pending

Revisar documentación

Revisar la documentación del proyecto

Assigned to: María García

Created: 28/12/2024

in-progress

Add New Task

Title

diseño arquitectura

Description

desarrollar una arquitectura DLT para apuestas P2P

Status

Completed

Assign to

Juan Pérez

Create

Dashboard

Users

Tasks

Tasks

+ Add Task

Completar proyecto

Finalizar el proyecto de desarrollo web

Assigned to: Juan Pérez

Created: 28/12/2024

pending

Revisar documentación

Revisar la documentación del proyecto

Assigned to: María García

Created: 28/12/2024

in-progress

diseño arquitectura

desarrollar una arquitectura DLT para apuestas P2P

Assigned to: Juan Pérez

Created: 28/12/2024

completed

maquetacion

stack html css

Assigned to: jose

Created: 28/12/2024

in-progress