# Estructura de Archivos para un CRUD de Tareas en Next.js con Material UI

Este documento describe una estructura de archivos recomendada para desarrollar una aplicación CRUD de tareas utilizando Next.js y Material UI. La estructura facilita la organización de componentes, páginas, estilos y otros elementos necesarios para un desarrollo eficiente y escalable.

## Estructura Principal del Proyecto

### 📁 /components

Contiene los componentes reutilizables que se utilizan en varias partes de la aplicación.

### 📁 /pages

Carpeta de rutas en Next.js. Aquí se alojan las páginas de la aplicación como /index, /tasks, etc.

### 📁 /styles

Almacena los archivos de estilo global y temas personalizados de Material UI.

### 📁 /utils

Funciones de utilidad para facilitar operaciones como validaciones, formateos y manejo de datos.

### 📁 /services

Define las interacciones con APIs para las operaciones CRUD.

### 📁 /hooks

Custom hooks para manejar lógica de estado y efectos específicos.

### 📁 /context

Contexto global de la aplicación para compartir datos entre componentes.

## 📁 components

Esta carpeta contiene los componentes reutilizables de la aplicación, organizados en subcarpetas según su función.

* 📄 TaskForm.js

Formulario para crear y editar tareas.

* 📄 TaskList.js

Lista de tareas que muestra todas las tareas creadas.

* 📄 TaskItem.js

Componente individual que representa una tarea en la lista.

* 📄 Navbar.js

Barra de navegación para la aplicación.

## 📁 pages

Contiene las páginas principales del proyecto en Next.js. Cada archivo representa una ruta de la aplicación.

* 📄 index.js

Página de inicio que presenta una visión general del CRUD de tareas.

* 📄 tasks/index.js

Página principal donde se muestra la lista de tareas.

* 📄 tasks/[id].js

Página para ver o editar una tarea específica, utilizando routing dinámico.

* 📄 tasks/new.js

Página para crear una nueva tarea.

## 📁 styles

Aquí se almacenan los estilos globales y temas personalizados de Material UI.

* 📄 globals.css

Estilos globales que se aplican en toda la aplicación.

* 📄 theme.js

Tema personalizado para Material UI, definiendo colores, tipografías y otros elementos.

## 📁 utils

Carpeta para utilidades o funciones que ayudan a realizar operaciones comunes.

* 📄 api.js

Funciones para manejar las llamadas a la API de CRUD.

* 📄 validators.js

Funciones para validar entradas de datos.

## 📁 services

Define las funciones para interactuar con la API, organizadas en funciones específicas para las operaciones CRUD.

* 📄 taskService.js

Funciones para realizar operaciones CRUD con la API de tareas.

## 📁 hooks

Custom hooks para manejar lógica de estado o efectos relacionados con las tareas.

* 📄 useTasks.js

Hook para manejar la lógica de obtención y gestión de tareas.

* 📄 useForm.js

Hook para manejar lógica de formularios de manera reutilizable.

## 📁 context

Proporciona contexto global para compartir datos entre componentes sin necesidad de prop drilling.

* 📄 TaskContext.js

Contexto para compartir la información de tareas a lo largo de la aplicación.