


**De:** Nuno Santos nsantos@icognitus.com   
**Assunto:** [iCognitus] - Recrutamento fase 2  
**Data:** 13 de janeiro de 2025 às 16:58  
**Para:**

---



Boa tarde,

Antes de mais votos de um bom ano!

Conforme combinado, envio o problema para resolução até final do mês.

**Problema:** Sistema de Gestão de Tarefas

**Descrição do Problema:**

Desenvolver uma aplicação de gestão de tarefas simples. A aplicação terá duas partes principais: uma API desenvolvida em .NET Core e uma interface de utilizador criada em Angular. O sistema deve permitir aos utilizadores gerir tarefas, onde eles podem adicionar, listar, editar e excluir tarefas.

**Requisitos:**

1. API (Backend - .NET Core):

A API deve ser construída com ASP.NET Core e deve expor os seguintes endpoints RESTful:

GET /tasks: Retorna a lista de todas as tarefas.

GET /tasks/{id}: Retorna detalhes de uma tarefa específica.

POST /tasks: Cria uma nova tarefa.

Parâmetros (no corpo da requisição): title (string), description (string), status (string: "Pendente", "Em Progresso", "Concluída").

PUT /tasks/{id}: Atualiza uma tarefa existente.

Parâmetros (no corpo da requisição): title, description, status.

DELETE /tasks/{id}: Deleta uma tarefa.

A API deve utilizar um banco de dados em memória (como o In-memory Database do Entity Framework Core ou um repositório simples, se preferir não usar EF Core) para armazenar as tarefas.

2. Frontend (Interface - Angular):

A interface do utilizador deve ser construída com Angular e deve permitir as seguintes ações:

Visualizar todas as tarefas: Ao carregar a página, deve ser exibida uma lista de todas as tarefas.

Adicionar uma nova tarefa: Deve existir um formulário para criar novas tarefas, que permita o preenchimento do título, descrição e seleção do status.

Editar uma tarefa: A interface deve permitir editar qualquer tarefa já existente.

Excluir uma tarefa: A interface deve ter uma opção para deletar tarefas.

3. Considerações Técnicas:

Use HTTP Client no Angular para consumir a API RESTful.

O backend em .NET Core deve validar os dados de entrada e retornar os códigos de status HTTP apropriados.

O frontend Angular deve exibir mensagens de erro ou sucesso conforme as respostas da API.

**Funcionalidades opcionais:**

Adicionar funcionalidades de filtro de tarefas por status (por exemplo: "Pendente" ou "Concluída").

Implementar autenticação e autorização simples, por exemplo, usando JWT.

Melhorar a experiência do utilizador com feedbacks visuais para ações como "a carregar", "sucesso" ou "erro".

Com os melhores cumprimentos,

**NUNO SANTOS**



Rua do Campo Alegre 1517, LJ24 C62

4150-182 Porto, Portugal

[www.icognitus.com](http://www.icognitus.com) | [nsantos@icognitus.com](mailto:nsantos@icognitus.com) | +351 928 143 477 (Escritório)

Horário do escritório: Seg-Sex: 09:00-18:00 (WET)

Esta mensagem destina-se apenas à(s) pessoa(s) mencionada(s). Se recebeu esta mensagem por engano, por favor elimine-a imediatamente, bem como eventuais cópias existentes no seu sistema, destrua eventuais impressões e notifique o remetente. Não é permitida, directa ou indirectamente, utilizar, distribuir, imprimir ou copiar a totalidade ou parte desta mensagem se não for um dos destinatários. A ICOGNITUS4ALL reserva-se ao direito de monitorizar todas as comunicações de correio electrónico efectuadas através das suas redes. Quaisquer opiniões expressas na mensagem são do próprio remetente, não representando a posição da instituição, excepto quando explicitamente indicado o contrário e por remetentes autorizados pela ICOGNITUS4ALL.