

---

---

Desenvolvimento de Aplicações Empresariais – 2025-26 – 1S

Engenharia Informática – 3º Ano – Ramo SI

**Enunciado do Projeto**

---

---

## Plataforma de Gestão de Publicações Científicas

### Objetivo

O objetivo deste projeto é implementar e testar uma aplicação empresarial de gestão de publicações científicas para um Centro de Investigação.

### Cenário

O Centro de Investigação e Desenvolvimento XYZ (O Centro) atua em múltiplas áreas científicas tais como Ciência de Dados e Ciência dos Materiais. No decurso da sua atividade, os membros do Centro adquirem e produzem publicações científicas tais como: artigos científicos com revisão por pares, comunicações em conferências, capítulos de livros ou livros científicos, relatórios técnicos, patentes, dados científicos (*datasets*), software (em código aberto), modelos de IA, bases de dados, teses de mestrado ou de doutoramento, e artigos de divulgação científica. Algumas destas publicações são confidenciais, não estando disponíveis para pessoas externas ao Centro.

Para aumentar as sinergias dentro do Centro, pretende-se desenvolver uma Plataforma de Gestão de Publicações Científica (A Plataforma) que permita: armazenar estas publicações; que os membros as enriqueçam com classificações, comentários, resumos e análises críticas; e um acesso facilitado tanto às publicações previamente adquiridas como às informações relacionadas anexadas pelos membros.

### **Exemplo 1**

O João A, é um investigador júnior recentemente contratado e afeto ao Projeto X, na área científica de Ciência de Dados, e está a fazer a sua pesquisa bibliográfica inicial. Começa por pesquisar na Plataforma publicações com o critério área científica “Ciência de Dados” e obtém uma lista com milhares de publicações! Antes de entrar em desespero com a perspetiva de ter de analisar milhares de resumos, alguns dos quais poderão simplesmente não ser relevantes para o Projeto X, decide alterar o critério de pesquisa para publicações com a *tag* “Projeto X”. Não só obtém uma lista de resultados muito mais pequena (menos de uma centena), como todos os artigos têm um breve comentário atribuído por colegas do Centro que lhe permite rapidamente avaliar o grau relevância do artigo para o contexto do Projeto X. Mais, a plataforma permite-lhe “subscrever” a *tag* Projeto X, de modo a ser notificado por e-mail sempre que alguém no Centro acrescentar essa *tag* a alguma publicação ou acrescentar ou modificar algum conteúdo relacionado com uma publicação com essa *tag*.

### **Exemplo 2**

A Joana B, é uma investigadora sénior que tem trabalhado no Projeto X desde que este começou. No decorrer de uma conferência, conheceu um investigador que tinha desenvolvido uma nova técnica que a Joana B identificou extremamente promissora para o Projeto X. Imediatamente depois de obter do dito investigador um PDF com a descrição desta nova técnica, submeteu esse documento na Plataforma com a *tag* “Projecto X” e com o comentário “Parem com tudo! Esta nova técnica pode revolucionar a nossa abordagem ao Projeto X!” Imediatamente, chega ao correio do João A uma mensagem a dizer “Há novidades na tag Projeto X” com uma ligação para a entrada recém-criada por Joana B na Plataforma.

### **Exemplo 3**

O Manuel C, é um programador afeto ao Projeto Y e está a implementar um algoritmo incluído num artigo de referência tanto para o Projeto Y como para o Projeto X. No decurso da sua atividade, deteta que há um erro grave neste artigo e que provavelmente irá resultar em que esteja retirado pela revista que o publicou. Mas este é um processo que pode demorar meses. Enquanto isso, na Plataforma acrescenta um comentário a

este artigo avisando do erro. Todos os membros do Centro que subscrevem as tags Projeto X e Projeto Y, são imediatamente notificados.

## Requisitos Funcionais

A Plataforma deverá suportar os seguintes roles e respetivas funcionalidades:

- **Colaborador:** fazer upload de uma publicação (PDF ou ZIP); corrigir informação gerada automaticamente (ex.: resumo gerado por IA); atribuir rating a publicação (ex.: 1 a 5 estrelas); comentar a publicação; associar *tags* a uma publicação; consultar todas as publicações visíveis para todos; consultar as publicações submetidas pelo próprio (incluindo as não visíveis para todos) e o histórico de edições sobre cada uma delas; pesquisar publicações (por título, autor, área científica, *tag*, data, etc.); ordenar listas de publicações (por número de comentários, por rating médio, por número de ratings); subscrever *tag*; consultar histórico de atividade do próprio (ex.: uploads, edições); editar dados pessoais (nomeadamente o endereço de e-mail); alterar palavra passe; recuperar palavra passe através do e-mail.
- **Responsável:** definir e remover *tags*; desassociar *tags* de publicação; ocultar ou mostrar comentários; ocultar ou mostrar publicações; consultar informação oculta (*tags*, comentários, publicações). Adicionalmente, pode realizar todas as funcionalidades de um Colaborador.
- **Administrador:** criar, editar, remover, ativar e desativar (suspender) utilizadores; alterar roles dos utilizadores; consultar histórico de atividade de qualquer utilizador. Adicionalmente, pode realizar todas as funcionalidades de um Responsável.

**Nota:** Esta secção não pretende ser exaustiva e nem todos estes requisitos funcionais têm o mesmo peso. Deverá clarificar com o docente das práticas quaisquer dúvidas quanto à interpretação e priorização destes requisitos.

## Requisitos Tecnológicos

**RT1.** O Sistema a desenvolver neste projeto deverá ser auto-contido (não dependerá de nenhum sistema externo, mesmo que público na *cloud*), e deverá ser constituído por:

- uma aplicação *backend* e o respetivo motor de base de dados;
- uma aplicação de *frontend* (que testar todas as funcionalidades expostas pelo *backend*)

**RT2.** A aplicação de *backend*, que abrange as camadas de lógica de negócio e de acesso a dados do sistema, deverá ser desenvolvida com base na plataforma Jakarta Enterprise Edition seguindo o modelo REST Service.

**RT3.** A aplicação de *frontend*, que abrange a camada de apresentação do sistema, pode utilizar qualquer tecnologia de *frontend* web capaz de utilizar serviços REST, nomeadamente a plataforma Vue.JS/NUXT utilizada nas aulas práticas.

**RT4.** O motor de base dados utilizado pelo *backend* deverá ser uma base dados relacional com licença GPL ou LGPL, nomeadamente o PostgreSQL utilizado nas aulas práticas.

**RT5.** O Sistema deverá pontuar-se pela adoção de padrões arquiteturais que promovam a modularidade em cada uma das camadas de aplicação (e.g., MVC, modelo de domínio, técnicas de ORM para a persistência de dados, *lazy loading*, controlo de concorrência, etc.)

**RT6.** O Sistema deve incluir funcionalidades implementadas com o recurso a IA (ex.: fazer resumos automáticos das publicações através de um LLM). A componente de IA deve ficar num contentor Docker padrão (ex.: <https://docs.ollama.com/docker>), cabendo à aplicação Jakarta EE a implementação do cliente da API padrão exposta por este contentor.

## Grupos: Regras

1. O projeto deverá ser efetuado em grupos compostos por 4 estudantes.
2. Não haverá distinção de classificações entre membros do grupo.
3. A apresentação e defesa da entrega final pode contemplar perguntas individuais a membros do grupo. **A má prestação de um dos membros do grupo afetará a prestação do grupo inteiro.**

## Entrega Intermédia (Primeira Entrega)

1. A entrega intermédia consiste na especificação, em ficheiro PDF, da API REST a implementar pela aplicação *backend*.
2. Este ficheiro PDF deverá ter o nome:  
DAE-2025-26-API-Projeto-#####-#####-#####-#####.pdf  
Onde ##### corresponderá a cada um dos números dos estudantes que compõem o grupo de trabalho.
3. Este ficheiro deve ser entregue no repositório disponibilizado para o efeito na página Moodle desta UC, até à data agendada no calendário de avaliações do curso.
4. Neste ficheiro, cada *endpoint* da API deverá ser especificado como se exemplifica a continuação:

## Exemplo 1

**EP01.** Um utilizador **consulta o catálogo dos medicamentos** utilizando o protocolo HTTP, verbo **GET**, para o sítio: /farmacia/api/medicamentos

A resposta **devolvida** por este recurso segue o formato JSON:

```
[
  {
    "id": 101,
    "nome": "Paracetamol",
    "dosagem": "400 mg"
  },
  {
    "id": 102,
    "nome": "Prozac",
    "dosagem": "10 mg"
  },
  . . .
]
```

## Exemplo 2

**EP02.** Um utilizador, autenticado com **médico**, **adiciona um medicamento a uma receita** utilizando o protocolo HTTP, verbo **PUT**, para o sítio:  
/farmacia/api/receitas/1000/medicamentos

O corpo do pedido **recebido** por este recurso segue o seguinte formato JSON:

```
{
  "id": 101,
}
```

A resposta **devolvida** por este recurso segue o seguinte formato JSON:

```
{
  "id": 1000,
  "estado": "aberto",
  "utente": 10001,
  "medicamentos": [
    {
      "id": 101,
      "dosagem": "100mg",
      "nome": "Aspirina"
    },
    . . .
  ]
}
```

## Entrega Final

1. A entrega final consiste num arquivo ZIP, RAR ou 7Z que contenha:
  - Projeto Java com aplicação *backend*
  - Projeto Vue.JS/NUXT (ou outro, de acordo com o RT3) com a aplicação *frontend*
  - Especificação final da API REST implementada (caso seja diferente da entregue previamente na entrega intermédia)
2. Este arquivo deverá ter o nome  
DAE-2025-26-Projeto-#####-#####-#####-#####  
Onde ##### corresponderá a cada um dos números dos estudantes que compõem o grupo de trabalho.
3. Este arquivo deve ser entregue no repositório disponibilizado para o efeito na página Moodle desta UC, até à data agendada no calendário de avaliações do curso.

## Avaliação

15% - Entrega Intermédia (Especificação da API)

85% - Entrega Final (Implementação do Projeto), avaliada da seguinte forma:

85% - Funcionalidades implementadas (*backend* e *frontend*) do cenário descrito

15% - Extras implementados

### Notas importantes:

1. Na entrega final, é obrigatório incluir a criação de um cenário inicial com dados que cubram todas as possibilidades suportadas pelo sistema.
2. Como extras podem ser considerados: importação de dados de configuração através de ficheiros CSV (*comma separated values*), ficheiros Excel ou API; *frontend* com tema responsivo para tablet, smartphone e monitor de PC; notificações por e-mail (ex.: alertas relacionados com *tag* subscrita); ou outras funcionalidades não essenciais para o cenário, mas que tirem partido da informação recolhida pelos sensores.

3. A entrega final do projeto está sujeita a apresentação e defesa, onde deverão ser explanadas as opções tomadas na análise, o desenho e a implementação; demonstradas as funcionalidades do projeto com o recurso a um cenário previamente preparado (que pode ser uma versão melhorada do cenário referido na nota 1).

**BOM TRABALHO!**