

Curso Técnico Superior Profissional em: Tecnologias e Programação de Sistemas de Informação

2.º Ano/1.º Semestre

Unidade Curricular: Aplicações Centradas em Redes

Data Entrega: 5/01/2021

Docente: Michael Silva / Hugo Perdigão

Época: Normal

ENUNCIADO DO TRABALHO

Projeto final - versão 1.2

Tecnologias as usar

Criar uma aplicação web usando as tecnologias:

- A Framework **Laravel** (PHP)
- **HTML + CSS** (ou SASS)
- **JavaScript** (podem usar jQuery)
- Base de dados **MySQL**

Tema

O projeto é individual e o tema é livre desde que cumpra os requisitos.

Notas na escolha do projeto

- 1) Devem escolher temas diferentes.
- 2) Idealmente deve ser semelhante ao exemplo dado nas aulas com 2 classes além da Users e com todas as funcionalidades pedidas.
 - a. poderão eventualmente utilizar a tabela Users como principal mas têm que ter obrigatoriamente 3 níveis (ex: cliente, funcionário e Admin) e tem que ser bem justificado no relatório).
 - b. Podem criar outras classes além das principais, mas não há necessidade de criar um grande número.
- 3) Poderão instalar pacotes (ex. carrinho de compras) mas deve ser indicado e bem justificado no relatório quais os pacotes instalados e porquê, bem como descrito o seu funcionamento

Exemplos de temas / portais:

- Portal de vendas
- Portal Imobiliário
- Portal de Streaming (Filmes /Séries)
- Portal de Jogos
- Portal Escolar
- Portal de Notícias

Requisitos:

- Área públicas.
 - Listagens publicas (da(s) classe(s) existentes – devidamente justificado)
 - Página(s) de Detalhes – toda a informação de um objeto
 - Ações limitadas (pesquisa/visualização)
- Área Reservada.
 - Gestão de objetos (criado pelo próprio)
 - (criação/ Edição/ Eliminação)
 - Administrador (pode gerir tudo)

Estrutura de dados

- Pelo menos 2 Classes (ex: Tema: Notícias → Classes: Autores e Artigos)
 - Deve existir pelo menos um campo Imagem/foto

Relação

Tem de existir pelo menos uma relação de 1 para muitos.

Exemplo:

- 1 Autor tem Muitos Artigos
- A relação deve estar definida na base de dados (migration) e nos objetos(Model).

Requisitos Funcionais

Parte interna (acesso com login)

1. Gestão de dados (CRUD) das classes escolhidas:
 - a. Criar objetos (Create)
 - b. Listar objetos (restrito ao utilizador – só vê os objetos criados por ele)

Para cada objeto:

- i. Visualizar (Read)
- ii. Editar (Update)
- iii. Eliminar (Delete)

2. Usar a validação de campos:

Cofinanciado por:



- a. pelo menos 1 campo obrigatório e com mínimo de caracteres (podem explorar e usar outros tipos de validação, se forem úteis)
3. **Apresentar a relação** de 1 para muitos, onde se justificar
Exemplo:
 - a. Ao mostrar os detalhes de um Autor, mostrar uma lista com os seus Artigos.
4. Criar um sistema de **Autenticação** (através do módulo do Laravel)
 - a. Com registo e login de utilizadores
5. Definir as seguintes **Permissões**
 - a. Utilizador tem de fazer login para gerir as entradas.
 - b. Um utilizador **comum**, ao fazer login **só pode ver e gerir as suas entradas**.
 - c. Ter pelo menos 1 utilizador **"admin"** que pode ver e editar **todas** as entradas

Parte pública (sem login)

6. **Página(s) de lista(s)** (da(s) classes principais - justificado), que cumpram o seguinte:
 - a. Apresentar os principais dados de cada objeto listado
 - b. Permitir aceder aos detalhes de um objeto (apenas dados públicos)
7. **Pesquisa (pelo menos numa classe)** que cumpra o seguinte:
 - a. por **2 campos** e com **ordenação** (por 2 campos, ascendente e descendente), em **simultâneo** (campos e ordenação).
 - b. Exemplo: pesquisar pelo título e autor da notícia, ordenando por mais recente ou por título (ordenação alfabética)
8. **Páginas de Detalhes públicas das 2 classes principal:** com imagem em dimensões maiores. Caso existam, mostrar lista das entradas da classe que tem relação, ex: nos detalhes do autor, mostrar ligações para os seus artigos.
9. Página de **contacto** com informações de contacto.

Regras de negócio

Criar 2 regras de negócio.

- Limitar o número de entradas que um utilizador pode ter (ex 1 autor só pode ter 10 artigos).

Cofinanciado por:



- Existir uma entrada que é um destaque **aleatório** que aparece sempre nas páginas públicas (lista e ficha). A cada refresh das páginas aparece uma entrada de destaque aleatória

Requisitos Não Funcionais

- Usabilidade:
 - a aplicação deve ser fácil de utilizar;
 - ter boa apresentação e organização na interface do utilizador (principalmente na parte pública)

Mais-valias

- Paginação nas listas
- Formulário de Envio de Email na página de contactos

Cofinanciado por:



Mini Relatório

Conteúdo

- 1) identificação do tema, disciplina e aluno
- 2) Apresentação do trabalho (justificação do tema e classes/relações escolhidas) e quais as suas regras de negócio
- 3) Indicações gerais das opções tomadas, decisões próprias e log de atividade

Decisões próprias

Indicar e justificar decisões, de estrutura/organização, listagens, dados a mostrar, entre outras, tomadas por iniciativa própria.

Log de atividade:

Resumo do trabalho efetuado por data indicando o que foi feito nesse dia.

Contactos

Em caso de dúvidas contactar através de:

- Moodle
- Email do professor:
 - Turma A: michael.silva@staff.uma.pt
 - Turma B: hugo.perdigao@staff.uma.pt

Entrega

- **Datas Previstas:**
 - **Entrega:** 04/01/2021 (Prof. Michael)
 - **Entrega:** 05/01/2021 (Prof. Hugo)
 - **Apresentação/defesa:** 06/01/2021 (Prof. Michael)
 - **Apresentação/defesa:** 26/01/2021 (Prof. Hugo)
- **Defesa: 2 a 4 perguntas, 9.5 mínimo**
- Para o relatório, usar o modelo de capa: Modelo da capa dos trabalhos académicos CTeSP
- Enviar o **relatório** em **formato .pdf** e incluir no ficheiro .zip enviado.
- No phpmyadmin: exportar a base de dados para um ficheiro .sql
- Comprimir os ficheiros num ficheiro .zip
- Ou enviar por email, we transfer, google drive, dropbox ou serviço similar

Cofinanciado por:

