ESCOLA SUPERIOR DE MEDIA ARTES E DESIGN
POLITÉCNICO DO PORTO

ALP

FICHA DE EXERCÍCIOS
ATIVIDADE LETIVA

Programação Web I

UNIDADE CURRICULAR

Ficha 04 –Vue Router

FICHA

Abra o VSC e crie um projeto Vue separado para cada exercício desta ficha.

Lembre-se de selecionar Vue Router durante o setup.

1) Este exercício introduz os conceitos base de routing criando uma aplicação com múltiplas "páginas" navegáveis sem reload.
   a) Crie um projeto Vue chamado **my-portfolio** que vai simular um portfolio pessoal. O projeto deve conter 3 views (páginas):
      - HomeView – página inicial com uma mensagem de boas-vindas
      - ProjectsView – página que lista 3 projetos fictícios
      - ContactView – página com informação de contacto fictícia
   b) Configure o router para que estas 3 páginas sejam acessíveis através dos seguintes paths:
      - / para home
      - /projects para os projetos
      - /contact para o contacto
   c) No App.vue, crie uma barra de navegação com links que permitam navegar entre as 3 páginas. Certifique-se de que a RouterView está configurada para renderizar o componente correto.
   d) Adicione um footer fixo em todas as páginas com o seu nome e o ano letivo.

2) Vamos agora trabalhar com routing dinâmico, criando URLs que dependem de dados variáveis.
   a) Crie um projeto Vue chamado **tech-blog**. Este projeto deve ter:
      - HomeView – página inicial do blog
      - ArticleListView – página que mostra uma lista de artigos
      - ArticleView – página que mostra os detalhes de um artigo específico
   b) Crie um array com pelo menos 4 artigos na ArticleListView. Cada artigo deve ter:
      - id (UUID e único)
      - title (string)
      - author (string)

- excerpt (resumo curto do artigo)
- content (conteúdo completo do artigo)

c) Configure uma rota dinâmica para aceder a artigos individuais através do URL **/articles/:id**

d) Na ArticleListView, renderize os artigos numa lista. Cada item deve ter um RouterLink que redireciona para a página de detalhes desse artigo (/articles/[id])

e) Na ArticleView, utilize $route.params para obter o ID do artigo do URL e mostre as informações completas desse artigo.

Desafio: E se o utilizador tentar aceder a **/articles/999** (um ID que não existe)? Como poderia lidar com isso?

3) Este exercício foca-se em routing aninhado (nested routes), onde uma página "pai" contém sub-páginas "filhas".

a) Crie um projeto Vue chamado **sneaker-store**. A estrutura de routing deve ser:

- HomeView (/)
- ShopView (/shop) – página principal da loja que tem:
  - AllProductsView (/shop) – mostra todos os produtos
  - CategoryView (/shop/category/:name) – mostra produtos de uma categoria específica
- AboutView (/about)

b) Configure as rotas no router/index.js, criando as nested routes dentro de /shop.

c) Na ShopView, crie uma barra lateral com links para as categorias: "Running", "Casual" e "Basketball". Não se esqueça de renderizar as sub-rotas.

d) Crie um array de produtos na ShopView (ou um ficheiro separado se preferir) onde cada produto tem: id, name, category, price e image (URL ou placeholder).

e) Na AllProductsView, mostre todos os produtos.

f) Na CategoryView, filtre e mostre apenas os produtos dessa categoria.

Nota: Quando o utilizador muda de categoria (ex: de /shop/category/Running para /shop/category/Casual), os produtos devem atualizar automaticamente. Se isso não acontecer, investigue porquê e como resolver (dica: o componente está a ser reutilizado).

4) Vamos agora combinar tudo: rotas dinâmicas, nested routes e uma arquitetura mais complexa.

a) Crie um projeto Vue chamado **admin-dashboard**. A arquitetura deve ser:

- LoginView (/)
- DashboardView (/dashboard) – página principal com:
  - OverviewView (/dashboard) – visão geral
  - UsersView (/dashboard/users) – lista de utilizadores
  - UserDetailView (/dashboard/users/:id) – detalhe de um utilizador específico
  - SettingsView (/dashboard/settings) – configurações

b) Crie um array de utilizadores fictícios na UsersView (pelo menos 5), cada um com id, name, email e role.

c) Configure todas as rotas necessárias utilizando nested routes (DashboardView é a rota pai de todas as sub-rotas /dashboard/*).

d) No DashboardView, crie um menu lateral com links para Overview, Users e Settings.

e) Na UsersView, renderize uma lista de utilizadores. Cada utilizador deve ter um link "Ver detalhes" que redireciona para /dashboard/users/[id].

f) Na UserDetailView, mostre todos os dados do utilizador selecionado. Inclua um botão "Voltar" que redireciona para /dashboard/users/.

Nota: Repare como o componente DashboardView está sempre visível enquanto navega entre as suas sub-rotas. Isto é útil para manter elementos comuns (como o menu lateral) sempre presentes.

5) Este exercício explora navegação programática, named routes e tratamento de rotas não encontradas (404).

a) Crie um projeto Vue chamado **photo-gallery**. Estrutura de rotas:

- HomeView (/) – página inicial com lista de fotos
- PhotoView (/photo/:id) – visualização de uma foto específica
- NotFoundView (rota wildcard) – página 404 para rotas inexistentes

b) Crie um array de fotos (pelo menos 6) onde cada foto tem id, title e url.

c) Configure as rotas utilizando a propriedade **name** em cada rota. Adicione também uma rota wildcard no final para capturar todas as rotas inexistentes.

d) Na HomeView, mostre as miniaturas das fotos. Utilize RouterLink com `:to="{ name: 'photo', params: { id: photo.id } }"` para navegação.

e) Na PhotoView:

- Mostre a foto em tamanho maior com o título
- Valide se o ID existe no array de fotos. Se não existir, navegue programaticamente para a página 404.
- Adicione 3 botões:
  - "Anterior" – navega para a foto anterior (se existir)
  - "Próxima" – navega para a próxima foto (se existir)
  - "Foto Aleatória" – navega para uma foto aleatória do array

f) Na NotFoundView, mostre uma mensagem "Página não encontrada" e um botão "Voltar ao Início".

Desafio: Implemente navegação por teclado na PhotoView. As setas esquerda/direita devem navegar entre fotos, e a tecla ESC deve voltar à página inicial.

6) Vamos agora combinar named views, redirect e alias para criar diferentes layouts numa aplicação.

a) Crie um projeto Vue chamado **news-portal**. As rotas devem ser:

- / – redireciona para /home
- /home (alias: /inicio) – layout com 3 named views

- /category/:name – mostra notícias filtradas por categoria (apenas view default)

b) No App.vue, configure 3 RouterView com nomes diferentes:

```
<RouterView />                <!-- view default -->
   <RouterView name="sidebar" />
   <RouterView name="banner" />
```

c) Crie os seguintes componentes:

- NewsListView – lista de notícias (view default)
- SidebarView – lista de categorias e notícias em destaque
- BannerView – banner rotativo com notícias principais
- CategoryView – mostra notícias filtradas por categoria

d) No ficheiro de rotas:

- Configure a rota / com redirect: '/home'
- Configure a rota /home com alias: '/inicio' e utilize componentes para mapear as 3 views: `{ default: NewsListView, sidebar: SidebarView, banner: BannerView }`
- Configure a rota /category/:name a apontar para CategoryView

e) Crie um array de notícias (pelo menos 10) onde cada notícia tem id, title, category, summary e featured (boolean).

f) Na NewsListView, mostre todas as notícias. Na SidebarView, crie links para as diferentes categorias. No BannerView, mostre cada notícia featured alternadamente.

g) Na CategoryView, filtre e mostre as notícias pela categoria. Adicione também um botão "Voltar ao início".

Desafio: Adicione uma rota `/category/:name/article/:id` que mostra o artigo completo numa única view, sem sidebar nem banner. Utilize nested routes para manter a estrutura organizada.