Universidade Nova de Lisboa

Mestrado Integrado em Engenharia Informática

Conceção e Implementação de Aplicações para a Internet

CIAI Projecto fase 1

Autores:
41772 - Pedro Simão
41626 - André Correia
41915 - Cristiano Martins

28 de Outubro de 2016

Conteúdo

1 Análise do Domínio

1.1 Descrição contextual

O objetivo deste projeto é o de desenhar e implementar um aplicação web que consiga gerir cursos, as respetivas cadeiras, os alunos inscritos e toda a informação necessária para que ambos alunos e professores consigam gerir o necessário relacionado com o contexto em que se encontram. Para desenvolver a aplicação é necessário estabelecer alguns parametros primeiro. Cada curso é descrito pelo seu número total de ECTS, número de anos que demora a ser realizado e um conjunto de cadeiras. Cada cadeira é indentificada por 2 letras e 4 números, um nome, uma descrição e um conjunto de professores. Para além disso, uma cadeira é lecionada em apenas um dos semestre e tem um conjunto de alunos associados. Um estudante é identificado por um número e tem ainda um cojunto de atributos, como por exemplo, e-mail pessoal, e-mail da faculdade, morada, fotografia, ano do curso e as cadeiras em que se encontra inscrito. Cada estudante deve, numa determinada cadeira em que se encontre inscrito, ter notas associadas a avaliações (testes, exames e projetos). O aluno tem ainda aulas práticas. Professores são identificados por um nome de utilizador, nome, departamento e podem estar associados a várias cadeiras desempenhando diferentes funções: professor e assistente.

Em relação a funcionalidades necessárias para ambos, professores e alunos, temos, para professores:

- Apresentar toda a informação pessoal do mesmo, assim como possibilitar a alteração de alguns parametros.
- Apresentar as cadeiras em que se encontra ativo.
- Para uma dada cadeira deve ser possível ver todas as atividades agendas, como testes, exames, aulas práticas, assim como possibilitar a alteração, adicão e remoção das mesmas.
- Para uma dada cadeira deve ainda ser possível adicionar e remover alunos, assim como atribuir notas aos mesmos.

No caso de se tratar de um assistente é apenas possível ver as cadeiras em que o mesmo se encontra ativo. Para alunos são necessárias as seguintes funcionalidades:

- Apresentar toda a informação pessoal do mesmo, assim como possibilitar a alteração de alguns parametros.
- Apresentar notificações e futuros eventos.
- Apresentar as cadeiras em que se encontra inscrito e as que já concluíu, assim como os ECTS dessa cadeira e a nota final.
- Para cadeiras que já tenha concluído deve ser possível visualizar as notas que o aluno teve em testes, exames e trabalhos.

1.2 Modelo conceptual

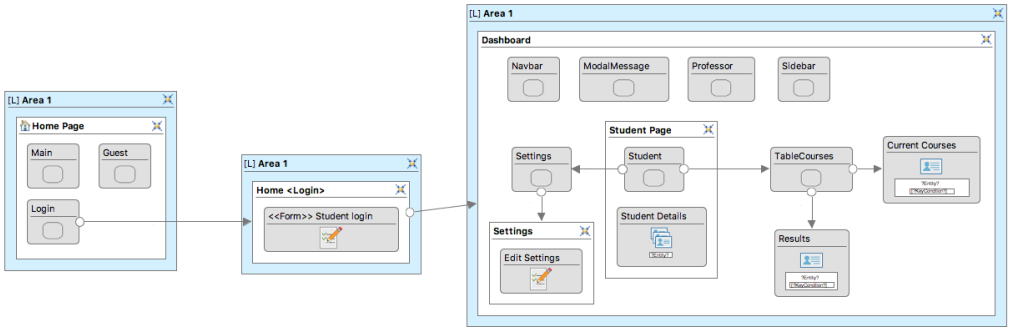

Figura 1: Modelo IFML

2 Tecnologias usadas

2.1 NPM

NPM é uma ferramenta utilizada para gerir pacotes de Javascript. Estes pacotes podem ser consultados no repositório do NPM. Uma vez instalada a dependência de um pacote no projeto, esta é referenciada num ficheiro *package.json*, previamente criado através do comando *npm init*, executado para definir algumas propriedades do projeto (nome, descrição, autor, dependências). Uma das vantagens desta ferramenta para além da reutilização de código por parte de outros *developers*, é os membros de um projeto poderem instalar as novas dependências através do comando *npm install*.

Algumas das dependências que utilizamos neste projeto:

1. **bootstrap**, *framework* de HTML, CSS e Javascript que ajuda no desenvolvimento de sites *responsive*. Disponibiliza um variado leque de classes e componentes já predefinidos.

2. **jquery**, biblioteca de Javascript que facilita a manipulação de elementos HTML, animações, gestão de eventos e simplifica o uso de Ajax.

3. **react**, biblioteca de Javascript que ajuda a desenvolver interfaces interativas. Permite a implementação de componentes com um estado próprio e cada um com uma vista. Uma vez que o estado de um componente altere este encarrega-se de renderizar a vista dos componentes associados.

4. **react-router**, biblioteca de Javascript para gerir rotas. Mantém a interface em sincronização com um URL específico.

2.2 Webpack

Webpack é uma ferramenta que avalia todas as dependências entre módulos instalados num projeto (a partir do NPM) e gera um ficheiro estático que representa esses módulos. A vantagem desta ferramenta é que disponibiliza um desenvolvimento por módulos. Podemos criar vários módulos que representam um determinado componente do site. Simplificando a estrutura geral do projeto. Os módulos que foram instalados por NPM também vão ser encupidos no ficheiro gerado. Este ficheiro Javascript será o ponto de entrada no

site. Uma vez que instalamos os módulos correspondentes ao bootstrap, react e jquery então deixa de ser necessário importar diretemente estas bibliotecas. Para o webpack funcionar é preciso criar um ficheiro de configuração *webpack.config.js* que contém algumas informações como também o ficheiro de ponto de entrada para o Webpack. A partir do ficheiro de entrada (app.js) o webpack começa por ver os módulos dependentes. Avaliando recursivamente as expressões *require(¡module name¿')* presentes nos ficheiros. Para cada dependência o webpack verifica se a mesma existe no caminho especificado.

2.3 Gulp

Gulp é uma ferramenta para automação de tarefas. Usamos como complemento ao webpack, mas nao sendo por si uma ferramenta importante no contexto deste projeto. Existe apenas uma tarefa designada para o Gulp. Executar o webpack, uma vez compilado o ficheiro pelo webpack este é copiado para uma pasta destino (/public/resources/assets/js).

2.4 Babel

Babel é um compilador de Javascirpt. Uma vez que o código em React não segue as notações puras do Javascript, o Babel compila o código em React para código nativo de Javascript. Utilizamos como módulo, instalado através de NPM. O babel podia ser importado diretamente no browser e compilar o codigo react diretamente no browser, desde que o ficheiro que contém o codigo react seja importado com o atributo *type="text/babel"*. Uma vez que fazemos esta transformação localmente então deixa de ser necessário acrescentar este atributo. Tendo ainda como vantagem o facto de a compilação do código ser feito localmente e não diretamente no browser. Reduzindo também o tempo de espera de loading da página.

2.5 Spring

Spring é uma framework para implementação de sistemas e aplicações em Java. Utilizamos esta framework para o desenvolvimento do Servidor. A framework é baseada em módulos. Para este projeto usamos *Spring - MVC Framework*, que fornece uma arquitetura MVC (*model-view-controller*) e um conjunto de componentes preparados para desenvolver aplicações web.

2.6 Sass

Syntactically Awesome StyleSheets (SASS) é uma extensão ao CSS que fornece algumas vantagens à linguagem base do CSS. Permite-nos usar variáveis, regras em cadeia e importar outras folhas de estilo. Ajuda e manter o código das folhas de estilo mais organizado. Como estas folhas de estilo não são interpretadas diretamente pelo browser é necessário um compilador. Para compilar usamos Compass que instalamos através de NPM. As folhas de estilo são importdas através do javascript como módulos (requeire('stylefile.scss')) e o webpack uma vez mais fica encarregue de compilar este código com auxílio do módulo Compass, sendo gerada uma folha de estilo em linguagem CSS básico.

3 Implementação

3.1 Resultado final

Todas as funcionalidades requeridas foram implementadas, faltando apenas a funcionalidade que permite ao professor adicionar e remover notas, no entanto, é possível ver notas que já tenham sido lançadas.