

Relatório de Laboratório de Projeto Integrado

Sistema de registo e organização de publicações científicas “Scientific Papers”



Realizado por: João Rosa, nº 35088

Orientador: Professor Luís Borges Gouveia

Docente da disciplina: Professor Paulo Rurato

Índice

- Proposta de trabalho.....4
- Planeamento.....5
- LogBook.....6
- Soluções tecnológicas.....7-9
- Arquitetura da aplicação.....10-13
- Funcionalidades (Cronologicamente).....14-17
- Possíveis melhorias/Conclusão.....18-19

Proposta de trabalho

No âmbito da disciplina de Laboratório de Projeto Integrado, do 3º ano, 2º semestre, da licenciatura em Engenharia Informática na *Faculdade de Ciências da Universidade Fernando Pessoa*, foram propostos vários projetos aos alunos, pelos docentes do curso.

Foi-nos dada a escolha do projeto, podendo este ser entre um de três tipos: autoproposto, continuação de um outro projeto da nossa autoria ou desenvolvimento de um dos projetos propostos pelos docentes do curso.

Eu escolhi a última opção, tendo optado por desenvolver o projeto “*Sistema de registo e organização de publicações científicas*” proposto pelo Professor Luís Borges Gouveia.

Para este projeto, pretendia-se desenvolver uma aplicação que permitisse o registo e a organização/filtragem de artigos científicos de acordo com diferentes parâmetros fundamentais tais como as palavras-chave, o resumo, o tipo de publicação, o(s) autor(es), etc...

Planeamento

Foi elaborado um plano de desenvolvimento de modo a atingir as metas propostas dentro dos prazos estipulados. O plano contém não só as tarefas a serem realizadas como também uma estimativa do tempo de duração de cada uma delas.

Tarefa	Estimativa de duração (horas)
Análise e compreensão do tema proposto	0,5
Escolha de tecnologias	0,5
Deteção das funcionalidades básicas necessárias a desenvolver	2
Desenvolvimento do código da aplicação	20
Introspeção sobre futuras possíveis melhorias da aplicação	1
Preparação para a apresentação do projeto	3
Apresentação/defesa do projeto	0,25
Realização do relatório do projeto	3

Tabela de planeamento

LogBook

De modo a manter o desenvolvimento do projeto organizado, manteve-se atualizado um LogBook com as tarefas a desenvolver e as datas em que estas foram realizadas.

Tarefa	Período de realização (2019)
Análise e compreensão do tema proposto	14/02
Escolha de tecnologias	21/02
Deteção das funcionalidades básicas necessárias a desenvolver	28/02
Desenvolvimento do código da aplicação	1/03 - 31/05
Introspeção sobre futuras possíveis melhorias da aplicação	1/04
Preparação para a apresentação do projeto	2/06 - 10/06
Apresentação/defesa do projeto	11/06
Realização do relatório do projeto	13/06 - 19/06

Tabela do LogBook

Soluções tecnológicas

Para desenvolver o projeto foi necessário escolher os diferentes tipos de tecnologia a serem utilizados, mas antes disso, teve de se selecionar o tipo de aplicação: **nativa** ou **web**.

Uma **aplicação nativa** é desenvolvida para um sistema específico (Windows, Android, iOS...) e é descarregada e instalada para a máquina do utilizador de modo a poder ser usada. O seu desenvolvimento faz-se recorrendo a linguagens de programação específicas ao sistema onde o programa vai executar como o Objective C para iOS ou Java para Android e Windows. As aplicações nativas são ainda caracterizadas pela sua rapidez e por oferecerem uma experiência de utilização otimizada ao dispositivo. Estas aplicações permitem a utilização de todas as funcionalidades do dispositivo como por exemplo a câmara, sistema de geolocalização ou serviço de notificações.

Uma **aplicação web** é obrigatoriamente acedida através de um web browser (Chrome, FireFox, Opera...), não podendo ser utilizada sem ligação à internet e é desenvolvida em linguagens de programação web tais como HTML, CSS e JavaScript. Uma vez que são acedidas através de um *browser* não requerem instalação e, tipicamente, têm um tempo de desenvolvimento curto, sendo mais fáceis de manter que as aplicações nativas.

No contexto do projeto a realizar decidiu-se fazer uma aplicação web uma vez que permite aos utilizadores usufruírem das funcionalidades da plataforma sem terem de descarregar/instalar a aplicação e aceder à mesma a partir de qualquer dispositivo que suporte browsers sem obrigar a quem está a realizar o projeto a ter de desenvolver código diferente para diferentes sistemas.

Tecnologias Web:

As tecnologias web utilizadas podem ser separadas em dois grupos distintos relativos ao desenvolvimento de websites: **Front-End** e **Back-End**.

A parte de **Front-End** de um website é relativa a tudo aquilo que se vê. Todas as características visuais de uma página como o tamanho, organização e visualização de elementos fazem parte deste grupo, que usa linguagens de programação específicas que permitem a quem está a desenvolver uma aplicação web controlar estas características.

A parte **Back-End** de um website é relativa à lógica, armazenamento e organização de dados. É a parte que funciona como suporte àquilo que a Front-End apresenta ao utilizador, possibilitando a troca de informações com o mesmo.

Tecnologias Front-End utilizadas:

- **HTML** – Permite estruturar uma página através de elementos.
- **CSS** – Permite editar visualmente os elementos estruturados em HTML.
- **JavaScript** – Permite tornar uma página mais interativa.

Para além das linguagens de programação acima mencionadas utilizou-se a framework de HTML e CSS o **Bootstrap** que facilita o desenvolvimento front-end.

Tecnologias Back-End utilizadas:

- **PHP** – Permite realizar todas as operações de lógica e funcionamento da aplicação como gestão e armazenamento de dados.

Para além da linguagem de programação acima mencionada utilizou-se a framework de PHP **Laravel** que facilita o desenvolvimento back-end.

Arquitetura da aplicação

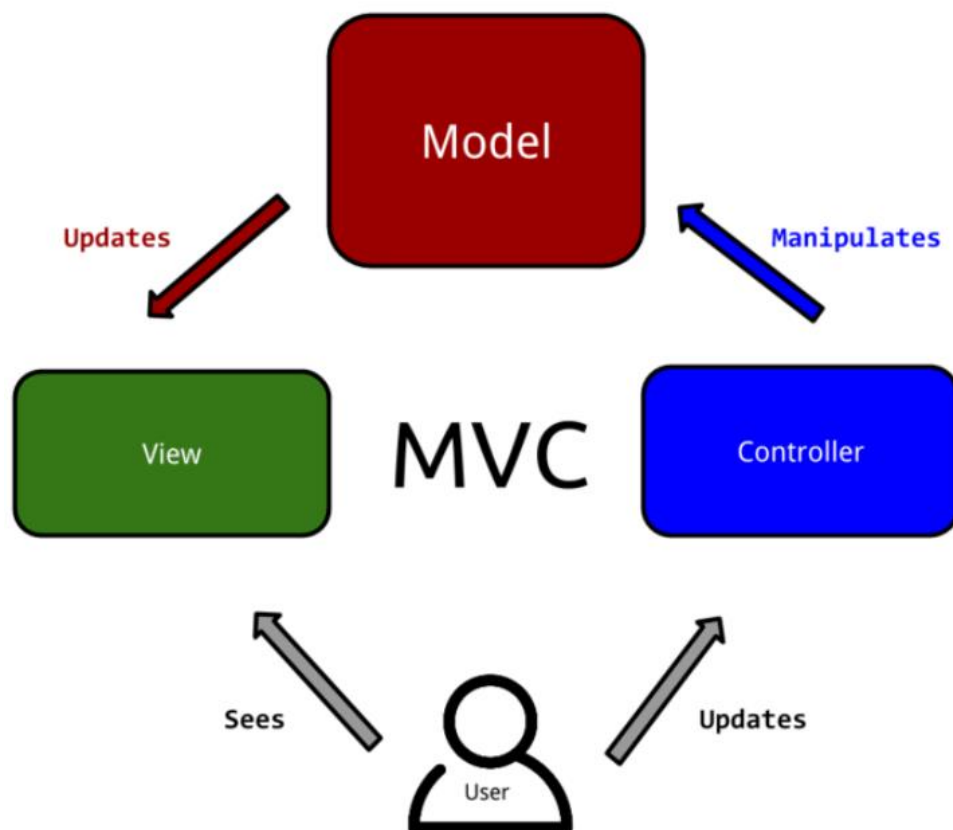
De modo a facilitar e agilizar o desenvolvimento e organização do código da aplicação utilizou-se um padrão de arquitetura de software (solução geral e reutilizável para um problema que ocorre com alguma frequência em arquitetura de software dentro de um determinado contexto). Neste caso, o padrão de arquitetura utilizado foi o que é usado como base na framework Laravel, o padrão **MVC (Model View Controller)**.

Este padrão separa a representação da informação da interação do utilizador com ela. Normalmente usado para o desenvolvimento de interfaces de utilizador que divide uma aplicação em três partes inter-relacionadas. Isto é feito para separar as representações de informação internas da aplicação, dos modos como a informação é apresentada aos utilizadores. O padrão MVC separa estes componentes possibilitando a reutilização de código e desenvolvimento paralelo eficiente.

Como explicado anteriormente, o padrão MVC separa o desenvolvimento do software em três camadas independentes:

1. Model - é responsável por tudo que a aplicação vai fazer a partir dos comandos da camada de controle em um ou mais elementos de dados, respondendo a perguntas sobre a sua condição e a instruções para mudá-las. O modelo sabe o que a aplicação quer fazer e é a principal estrutura computacional da arquitetura, pois é ele quem modela o problema que se está a tentar resolver. Modela os dados e o comportamento por detrás do processo de negócios. Preocupa-se apenas com o armazenamento, manipulação e geração de dados. É um encapsulamento de dados e de comportamento independente da apresentação.
2. View - Uma View deve garantir que sua expressão reflita o estado do Model. Sempre que os dados do Model são alterados, o Model altera as Views que dependem dele. Em resposta, cada View tem a oportunidade de modificar-se. É a camada de interface com o utilizador. É utilizada para receber a entrada de dados e apresentar visualmente o resultado.
3. Controller - É responsável por interpretar as ações de entrada através do rato e do teclado realizadas pelo utilizador. O Controller envia essas ações para o Model e

para a janela de visualização (View) onde serão realizadas as operações necessárias.



Esquema simplificado que mostra a relação entre as três camadas do MVC e o utilizador

Funcionalidades (Cronologia de Implementação)

Após feita a análise dos requisitos e escolhidas as tecnologias e arquitetura da aplicação, começou-se a parte de implementação de código.

A primeira página a ser desenvolvida foi a *Home Page*, estando numa primeira fase apenas definido o seu aspeto visual, com uma barra de navegação, uma imagem de fundo e uma caixa central com uma mensagem de boas vindas à plataforma e dois botões, um para o utilizador fazer login, outro para se registar (neste momento apenas o visual está definido e nenhuma das funcionalidades estão implementadas ainda).

Após ter a página inicial concebida, implementou-se o login e registo de utilizadores, tendo cada uma destas funcionalidades a sua própria página. Ao registar-se, o utilizador tem de fornecer um e-mail que ainda não esteja a ser utilizado por outro utilizador na plataforma e definir uma password de acesso pessoal com o mínimo de oito caracteres (se tentar definir uma com menos é apresentada uma mensagem num retângulo vermelho a avisá-lo da situação). Uma vez registado, o utilizador pode fazer o login com as suas credencias de acesso que definiu durante o registo.

Tendo o login e o registo de utilizadores a funcionar, o próximo passo foi desenhar e implementar a página pessoal a que cada utilizador tem acesso uma vez registado. Esta, pode ser acedida através da barra de navegação, assim que o utilizador se logar (o nome do utilizador aparece do lado direito e funciona como um menu dropdown onde existem as opções de ir para a sua página pessoal ou para fazer logout. Nessa página, o utilizador tem uma lista dos títulos dos seus trabalhos inseridos (com botões para apagar ou editá-los) e um botão para inserir um novo trabalho.

Sendo a plataforma "*Scientific Papers*", uma aplicação com o objetivo de armazenar e gerir informação (trabalhos científicos neste caso) desenvolveram-se seguidamente os métodos **CRUD** (**Create**, **Read**, **Update**, **Delete**) que representam as quatro operações básicas utilizadas em bases de dados relacionais. Foram implementados pela seguinte ordem (com as respetivas páginas):

- **Create** (página de criação de posts) - Esta página permite ao utilizador inserir trabalhos/pesquisas académicos. Aqui estão contidas caixas de texto para o utilizador caracterizar da melhor maneira possível o trabalho que está a guardar na plataforma. Os campos a preencher são o "Título", o "Tipo" (que pode ser um entre nove que são selecionados num menu dropdown), o "Resumo", as "Keywords", os "Comentários" e pode ainda opcionalmente ser enviado um ficheiro.

- **Read** (páginas de pesquisa e visualização de posts) - Na barra de navegação existe uma opção “Search” em que ao se fazer click aparecem por baixo três opções, que correspondem às pesquisas por título, autor e palavras-chave. Em cada uma dessas três páginas existe um caixa de texto onde o utilizador pode inserir o título, autor ou palavras-chave que deseja procurar. Para além destas três páginas fez-se uma página onde se podem visualizar todos os posts inseridos na plataforma, também acessível através da navbar. Esta página foi implementada de maneira a apresentar os posts inseridos mais recentemente no topo, com um máximo de dez (pode-se ver mais para além dos últimos dez posts inseridos uma vez que a listagem dos posts é paginada, isto é, é possível avançar para a “página 2” se houverem mais de dez posts). Nesta página de visualização, os títulos das publicações são hyperlinks que levam o utilizador a uma página de visualização da publicação que contém toda a informação relativa a esse post.
- **Update** (página de edição de posts) - A página de edição de posts é praticamente igual à de criação. A única diferença é que em vez dos campos da publicação estarem vazios, estão preenchidos com o seu conteúdo original, de modo a permitir ao utilizador editar apenas aqueles que quiser. É possível aceder a esta página de duas maneiras diferentes, ou através da página pessoal do utilizador carregando no botão *edit* ao lado do post

que deseja editar, ou através da página de visualização do post específico a apagar (um post que pertença a um utilizador que o está a visualizar aparece com um botão que permite editar).

- **Delete** – A opção de apagar publicações não necessita de uma página própria. Apenas foi necessário criar botões (e programá-los de modo a apagarem da base de dados a publicação em questão) e colocá-los em locais oportunos. Esses locais são na página pessoal do utilizador (ao lado do nome de cada post, para além do botão de editar, há também um botão para apagar) e na página de visualização de uma determinada publicação (um post que pertença a um utilizador que o está a visualizar aparece com um botão que permite apagar).

Após desenvolvidas as funcionalidades chave CRUD, implementaram-se medidas de controlo de acesso lógicas ao bom funcionamento da plataforma:

- Um guest (utilizador não logado) não pode criar, editar ou apagar nenhuma publicação.
- Um utilizador logado não pode apagar ou editar as publicações de outros utilizadores

Possíveis melhorias/Conclusão

Existem inúmeras melhorias e novas funcionalidades que poderão e irão ser implementadas ao longo do tempo como forma de praticar e aperfeiçoar o meu conhecimento na vasta e importantíssima área que é o desenvolvimento web. Algumas das funcionalidades a implementar são:

- Permitir o utilizador pôr uma fotografia e escrever um pequeno “about me” no seu perfil.
- Permitir que os utilizadores possam identificar numa publicação sua o perfil de outros autores que tenham sido co-autores do trabalho e que também façam parte da plataforma (deixando um link para o seu perfil).
- Criar uma aba na barra de navegação de eventos. Nessa parte da plataforma os utilizadores poderiam criar eventos relativos a palestras/workshops e convidar pessoas.
- Fazer com que os eventos fossem automaticamente sugeridos a utilizadores que o pudessem achar interessantes (através do tipo de publicações que o utilizador cria e vê mais).

- No fim da página de visualização de um trabalho específico adicionar uma pequena lista de trabalhos parecidos como sugestão.
- Permitir visualizar no site a localização de eventos através do google maps.
- Envio automático de emails.
- Melhorar a estética da plataforma.
- Melhorar a eficiência do código.
- Melhorar a documentação do código.

Para além das acima mencionadas, haveriam ainda muitas outras funcionalidades que poderiam ser adicionadas/melhoradas. Gostaria de ter implementado mais delas até à entrega final, mas um projeto desta magnitude requer muito tempo a uma pessoa.

Contudo, considero que a realização deste projeto foi para mim uma experiência imensamente enriquecedora tanto a nível pessoal, como profissional, não só aumentando os meus conhecimentos teóricos e práticos de desenvolvimento web, como melhorando a minha capacidade de gerir e organizar projetos mais complexos.