

Relatório do Trabalho Prático – Programação em Ambiente Web

Licenciatura em Engenharia Informática

2020/2021

Grupo 16

- 8190340, Tiago Machado Coelho
- 8190614, Tiago Miguel Moreira Da Silva
- 8190709, Maria da Conceição Pereira Tavares

Índice

Introdução	3
Identificação e Caracterização do Projeto	4
1º Milestone	4
2º Milestone	4
Especificação geral do Software a desenvolver	6
Requisitos principais:	6
Requisitos extra considerados:	6
Decisões de implementação:	7
Desenvolvimento:	8
Bibliotecas Importantes usadas:	8

Introdução

O trabalho realizado no âmbito da U.C. de Programação em Ambiente Web consiste na criação de uma WebPage com backend e frontend preparada para servir o cliente através de um servidor eficiente, em que o objetivo principal consiste na criação de eventos e venda de bilhetes.

Como ferramentas ao longo deste projeto usamos Node.js, Express.js, Angular e MongoDB.

Na primeira Milestone o projeto foi desenvolvido à base de Express.js.

Na segunda tratou-se de passar o projeto todo para Angular e desenvolver o resto também em Angular, e comunicar com o backend apenas quando pedido utilizando pedidos REST.

Utilizamos o VSCode como IDE para todo o desenvolvimento do projeto.

Identificação e Caracterização do Projeto

Como já referido na introdução o objetivo essencial deste projeto era criar uma plataforma que fosse semelhante a uma ticketline com a possibilidade de comprar bilhetes, criar eventos e geri-los, entre outras possibilidades.

Neste projeto, começando pela parte do cliente projetamos que seria utilizado por 4 tipos diferentes de utilizador.

Estes tipos de utilizadores são nomeadamente os não registados na plataforma com acesso bastante limitado ao que a mesma contém tendo apenas a opção de visualizar um certo conteúdo, os clientes que se referem a utilizadores comuns registados, podendo estes realizar a compra de bilhetes e visualizar, editar ou apagar a sua conta e ainda a possibilidade de fazer um pedido para ser promotor, este mesmo promotor tem acesso às mesmas áreas que o cliente e ainda a possibilidade de criar eventos, gerir esses mesmos e ainda criar localizações.

Por último o administrador do site tem também a hipótese de comprar bilhetes e criar localizações e ainda consegue ter o registo de todos os utilizadores podendo apagar um a qualquer altura, e visualizar os pedidos para promotor tendo liberdade de os aceitar ou recusar, pode também ter acesso a uma dashboard com charts ilustrativos como forma de gestão da aplicação.

1º Milestone

Tarefa	Membro(s) Encarregue	Tempo Previsto	Observações
Design EJS	Tiago Silva e Maria Tavares	4 dias	
Rotas	Tiago Coelho	1 semana	
Modelos	Todos	1 dia	
Controladores	Tiago Coelho e Tiago Silva	4 dias	
Swagger	Maria Tavares	1 dia	

2º Milestone

Tarefa	Membro(s) Encarregue	Tempo Previsto	Observações
Conversão para Angular	Tiago Coelho e Maria Tavares	1 semana	
REST API	Maria Tavares	2 dias	
Modelos Angular	Todos	1 dia	

Componentes/Design e TS	Tiago Coelho e Tiago Silva	2 semanas	Desenvolvido do início ao fim
Serviços	Tiago Coelho	2 dias	
Autenticação	Tiago Coelho	4 dias	
Rotas	Todos	1 dia	
Extras	Tiago Silva	2 dias	

O tempo previsto foi uma estimativa feita, pelo que é normal haver discrepâncias ou irregularidades tendo em conta a execução real do projeto e os objetivos que o mesmo proporcionou.

A atribuição das tarefas estava prevista para certas pessoas do grupo, mas todos os membros trabalharam em conjunto nelas, sendo então que em todas as tarefas, subentende-se que o trabalho lá apresentado reflete esforços de todos os membros, apesar de um deles ser o verdadeiro responsável, desta forma todo o desenvolvimento do projeto até ao produto final foi acompanhado, implementado, verificado e corrigido por todos os pertencentes ao grupo.

Temos então neste projeto a identificação de 3 entidades essenciais e dentro de algumas delas objetos que funcionariam como sub-entidades:

- Users, que contém os dados do user e uma lista de objetos tickets que funciona como sub-entidade;
- Events, que contém os dados do evento, incluindo a sua localização como referência à entidade Localização;
- Locals, que contém a morada, latitude e longitude da localização que será referente a um determinado evento.

A partir destas entidades tratamos de definir como iria ser estruturado e dividido o nosso trabalho, as suas funcionalidades base, possíveis extras ou funcionalidades não requisitadas que nos fariam sentido como grupo implementar, sendo que tudo isto encaixaria num conjunto de requisitos estruturados.

Especificação geral do Software a desenvolver

Requisitos principais:

- Gestão de acessos (registo e login e gestão de utilizadores);
- Editar dados;
- Apagar conta;
- Comprar bilhetes;
- Cancelar bilhetes;
- Visualizar bilhetes;
- Introdução de documentos/imagens;
- Visualizar eventos;
- Criar Eventos;
- Criar Locais;
- Editar/Apagar Eventos;
- Gerir Pedidos;

Requisitos extra considerados:

- Visualização de uma dashboard com charts de dados específicos;
- Download de documentos;

Decisões de implementação:

Era de conhecimento geral que não se podia fechar a página para clientes que não estivessem registados, no entanto mantendo a página visível a utilizadores não autenticados na página, estes terão as suas funcionalidades restringidas unicamente à visualização dos eventos disponíveis.

O utilizador não autenticado para ter acesso a mais funcionalidades, deve, portanto, fazer login na sua conta ou caso não tenha, criar uma conta nova. Cada nova conta criada começa com o cargo de “cliente”, e essa conta terá apenas acesso às funcionalidades de cliente que são elas: a possibilidade de comprar bilhete, gerir os seus bilhetes e cancelar caso pretenda, ou fazer download do teste de covid que usou nessa compra, ou ainda editar os seus dados de perfil.

O cliente que quiser ser promotor deve na mesma página onde edita os seus dados, fazer um pedido para promotor através de um botão lá disponível, à entrada na área do seu perfil, todos os utilizadores terão de introduzir de novo a sua palavra-passe num pop up como forma de segurança.

Os promotores têm acesso a todas as funcionalidades do cliente e ainda a novas funcionalidades tais como, adicionar evento, adicionar localização e gerir os seus eventos.

A gestão de eventos por parte do utilizador está implementada numa dinâmica que foi criada e aprovada em grupo. Baseia-se no sentido que um promotor quando cria um evento, este mesmo vai para a lista dos seus eventos, mas numa área diferente disponível num botão de notificação chamado “Eventos não públicos”, isto consiste na ideia que todo o evento criado fica no estado de não publicado e o promotor só poderá editar e/ou eliminar eventos que não estejam publicados, esta abordagem serviu como solução para o caso infeliz de um cliente ter comprado bilhete para um evento que mudou de data ou outro atributo que impossibilite o cliente de poder assistir ao evento tendo este já realizado a compra sem que tenha de gastar os 5 avisos dele de cancelamento de bilhete. Desta forma o produtor pode editar e/ou eliminar apenas eventos não públicos e tem a possibilidade de os publicar, caso tenha certezas de que não o precisará de alterar, pois após a publicação, não poderão ser feitas alterações.

Todos os formulários, quer de adicionar evento, registo ou qualquer outro, seguem regras específicas de input baseadas em Regex, da qual o utilizador terá de ter atenção.

O administrador também tem acesso a todas as outras funcionalidades e ainda a possibilidade de aceitar ou recusar pedidos de utilizadores para promotores, consideramos que o administrador apenas pode eliminar o utilizador e não editar

diretamente os seus dados, essa edição ficaria ao encargo do próprio utilizador, promotor que queira voltar a ser cliente não precisará necessariamente de mudar a sua role, visto que as funcionalidades são as mesmas, assim evita-se os constantes pedidos de mudança caso o utilizador esteja sempre a mudar de ideias.

O administrador da página tem ainda acesso na sua área de gestão a um conjunto de gráficos disponibilizados na dashboard com dados específicos da aplicação.

Para a criação dos charts utilizamos o Mongo Atlas.

Os gráficos atualizam os seus dados a cada 5 minutos.

Desenvolvimento:

Toda a parte cliente foi desenvolvida em Angular da qual usamos serviços para fazer pedidos REST ao nosso backend para aceder a certas funções e obter certos dados, desenvolvemos serviços para trabalhar com os utilizadores, com os eventos e ainda serviços para gerir os tokens e com isso a autenticação.

Sempre que um pedido é feito o interceptor envia pelos headers o token com a informação do utilizador que depois é verificado.

Já a parte servidor foi transformado completamente para segunda Milestone para tratar dos pedidos REST, é aqui que é controlada toda a execução das funcionalidades da página, as funções estão aqui especificadas e responde quando pedido.

Bibliotecas Importantes usadas:

- Jwt – para gerar e controlar os tokens;
- Mongoose – para comunicar com a base de dados em mongo;
- Moment.js – para formatar datas;
- Bcrypt – encriptação de bases de dados;
- Bootstrap – para o uso de ícones, botões e tabelas;
- Material – para o uso de botões e outros elementos de design;
- Multer – para lidar com ficheiros
- Swagger – como forma de documentação;

4. Análise dos Principais pontos do trabalho

Em geral conseguimos abranger todos os pontos pedidos do trabalho, e conseguimos ser críticos ao ponto de analisarmos o que era pedido no enunciado e implementar e modificar de acordo com a nossa interpretação do que seria melhor para a aplicação.

Tendo em conta que concluímos todos os objetivos pedidos, a nossa capacidade de avaliação e análise levou-nos a pensar a fundo no problema de forma a conseguirmos desenvolver uma página consistente e funcional.

No entanto reparamos que a segunda Milestone foi mais exigente e onde tivemos mais dificuldades, devido à quantidade de tarefas a fazer e a nossa vontade em implementar com qualidade e com bastantes extras vimos nos forçados a cortar em muitos dos planos e ficar pela base do pedido, também com a descoberta de bugs na aplicação que reforçaram os cortes que tivemos de fazer no planeamento. Para além do sentimento que podíamos fazer muito mais, também sentimos que podíamos ter embelezado a página mais a nível aspetual.

Os bugs que fomos encontrando foram possivelmente a maior dificuldade e onde mais tempo foi gasto no desenvolvimento do projeto.

Analisando todos os aspetos falados, concluímos que desenvolvemos uma aplicação segura e eficiente, poderia, no entanto, ter mais extras e em certo ponto um design mais agradável, mas cumpre todos os requisitos, e de acordo com o realizado o grupo autoavalia o trabalho com [17](#).