

 <p>ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO</p>	<p>PAW- Programação em Ambiente Web</p> <p>2º Semestre ■ Docentes: FAS, JRMR, PJS, DMRC</p> <p>Trabalho prático ■ Avaliação em Período Letivo</p>
---	---

Destinatários

Este trabalho destina-se a todos os estudantes inscritos na unidade curricular de Programação em Ambiente Web em Avaliação em período letivo que pretendam obter aprovação à Unidade Curricular.

Objetivos

Com a realização do trabalho prático, pretende-se que os alunos ponham em prática todos os conhecimentos adquiridos na Unidade Curricular, demonstrando as suas apetências em:

- Conhecer e aplicar o modelo Cliente-Servidor e as principais tecnologias/protocolos associados para o desenvolvimento de aplicações Web;
- Compreender e implementar o padrão MVC;
- Implementar aplicações Web através da manipulação de linguagens e frameworks para o desenvolvimento das componentes de cliente e servidor em aplicações Web
- Desenvolver uma aplicação Web utilizando as linguagens, bibliotecas e *frameworks* lecionadas;
- Compreender o desenvolvimento de aplicações como *fullstack developer*.

Enunciado

Atualmente, devido à pandemia provocada pelo Sars-Cov-2, a realização de eventos culturais está significativamente condicionada pelos requisitos impostos pela Direção Geral de Saúde. Os promotores de eventos culturais deparam-se com grandes problemas logísticos associados à restrições necessárias como, por exemplo, redução da lotação dos espaços de espetáculo, obrigatoriedade de testes aos expectadores, devolução de bilhetes e cancelamento de espetáculos.

Tendo estas condicionantes em mente, é pedido a elaboração de uma plataforma web para ajudar os promotores de eventos culturais a gerir este novo fluxo de trabalho.

Especificamente, a plataforma web deve permitir a um promotor inserir locais de espetáculo, devidamente configurados, com a respetiva lotação, morada e limitação de lotação em percentagem. Deverá também permitir ao promotor criar espetáculos devidamente configurados, com pelo menos o nome do espetáculo, descrição e poster promocional.

Com esta informação o promotor deverá configurar os diferentes eventos culturais na plataforma e, de seguida, disponibilizar na parte pública da plataforma para que os utilizadores possam visualizar os eventos culturais com interesse para o seu caso. Apenas utilizadores autenticados deverão poder adquirir bilhetes.

Para assistir a um espetáculo, a aquisição de um bilhete não será suficiente. No contexto da atual pandemia é necessário assegurar que cada cliente tem um teste negativo à covid-19 antes da data do espetáculo. Para o efeito, um cliente deverá efetuar o upload da digitalização do comprovativo de teste negativo e o promotor deverá verificar o teste negativo na plataforma, caso contrário o bilhete será cancelado. Um utilizador pode cancelar um bilhete a qualquer momento, no entanto se o cliente cancelar mais de 5 reservas num mês deverá ser banido da plataforma.

Por fim, pode assumir-se que o pagamento dos bilhetes pode ser feito no local do evento na bilheteira usando o código do bilhete e um documento de identificação, cujos dados devem ficar associados à reserva do bilhete na plataforma.

Deve adicionar requisitos e funcionalidades que completem este problema tornando-o mais rico em funcionalidades. Devem discutir a implementação de funcionalidades adicionais com os respetivos docentes das aulas práticas.

Requisitos Gerais

Tendo em conta a descrição do enunciado, a plataforma deve **no mínimo** suportar:

1. Gestão de utilizadores
 - a. Criar/Editar/Ver perfil de cliente na plataforma
 - b. Criar/Editar/Remover promotores de espetáculo
 - c. Editar password administrador da plataforma
2. Gestão de eventos e locais de eventos culturais
 - a. Criar/Editar/Visualizar
 - b. Adicionar/Editar lotação dinamicamente
 - c. Apresentar *dashboard* com eventos a decorrer e ocupação de eventos;
 - d. Listar pedidos de acordo com filtros específicos por cada tipo de utilizador;
3. Gestão de bilhetes
 - a. Criar/Alterar/Cancelar
 - b. Adicionar comprovativo teste Covid
4. Funcionalidades **Extra** para bonificação do trabalho
 - a. Uso de gráficos nos *dashboards*;
 - b. Permitir pagamento no site usando uma API de pagamento externa (ex: Stripe API: <https://stripe.com/docs/checkout/integration-builder>)
 - c. Permitir verificar o estado meteorológico de eventos ao ar livre através de APIs (ex: <https://openweathermap.org/api>)
 - d. Adicionar o agendamento de testes de forma automática

Requisitos adicionais ao problema devem ser analisados pelo grupo de trabalho e acompanhados durante as aulas práticas pelos respetivos docentes. Será objetivo de avaliação a capacidade de interpretação e utilidade da aplicação final para o problema em questão.

É necessário lembrar que devem utilizar design e princípios de design responsivo. É autorizado o uso de ferramentas, bibliotecas e frameworks adicionais que complementem a matéria lecionada (ex.: framework CSS Bootstrap, Google Charts, etc.) para o desenvolvimento do trabalho prático.

Milestone #1

Para o desenvolvimento do primeiro *milestone* do trabalho deve ser no mínimo considerada a especificação e elaboração da plataforma para registo de promotores, adicionar e visualizar a definição de eventos e espetáculos. O desenvolvimento deve ser feito obrigatoriamente recorrendo à framework ExpressJS para node.js usando o template engine EJS. A persistência em base de dados deve ser assegurada utilizando uma base de dados MongoDB para o efeito.

Adicionalmente será valorizada a apresentação de trabalho adicional que não seja considerada obrigatória nesta fase como, por exemplo, gestão de bilhetes, autenticação de utilizadores.

O primeiro *milestone* não terá apresentação, contudo o conteúdo entregue pelos alunos será avaliado e será fornecido um feedback durante as aulas práticas com o resultado da avaliação específica de cada grupo.

Milestone #2

O segundo milestone considera a entrega de um conjunto de recursos web que permita dar resposta completa ao enunciado. Neste caso, será avaliado todo o trabalho, incluindo as modificações e recomendações propostas pelos docentes após o milestone #1. A qualidade da aplicação e das funcionalidades apresentadas serão alvo de avaliação.

No segundo milestone existirá a implementação de restrições adicionais. A página do cliente deverá ser implementada recorrendo à plataforma Angular, utilizando serviços web para a gestão do estado das reservas.

Na parte do servidor é obrigatório o uso de *endpoints* REST utilizando *nodeJS* e a *framework ExpressJS* para criar *webservices* de suporte à aplicação. Todos os serviços REST desenvolvidos devem estar devidamente documentados utilizando a plataforma Swagger.

No caso extremo do grupo de trabalho optar pela não utilização da *framework Angular* para a página de clientes e implementar a plataforma utilizando *nodeJS/ExpressJS* com *template engines*, existirá uma **penalização de 3,5 valores** na avaliação do trabalho prático. Neste caso é igualmente **obrigatório** a utilização dos serviços REST (mesmo que não sejam utilizados).

Ferramentas

No desenvolvimento do trabalho prático os grupos de trabalho devem usar:

- NodeJS e a *framework ExpressJS*
- Angular
- IDE de desenvolvimento (ex: VSCode)
- Git e Gitlab (gitlab.estg.ipp.pt)

Relatório Final

O trabalho deverá ser acompanhado de um **relatório** de projeto, tendo os seguintes tópicos:

- **Identificação e caracterização do projeto**, justificando a abordagem seguida de acordo com o processo de negócio da empresa. Deve ainda ser apresentado um diagrama de desenvolvimento identificando as principais tarefas, o membro do grupo responsável por essa tarefa e o tempo previsto;
- **Especificação geral do software a desenvolver**, mapeando os requisitos do negócio com as componentes de software a desenvolver. Pode utilizar *mockups* e/ou diagramas de UML para justificar as decisões tomadas ao nível de implementação para as decisões mais relevantes.
- **Análise dos principais pontos do trabalho**. Discussão técnica sobre as opções de desenvolvimento e a sua avaliação (pode ser realizada uma apreciação crítica relativamente ao plano inicialmente estabelecido).

Avaliação

Ponderação das componentes:

- 30% primeiro milestone (nota mínima 7,5 valores)
- 70% segundo milestone (nota mínima 7,5 valores)

Em cada um dos componentes será reservado um total de 3 valores para funcionalidades de bonificação ou requisitos adicionais. É encorajado o espírito crítico e a implementação de funcionalidades adicionais e/ou de bonificação que deve ser coordenado com os docentes da UC.

Formato das entregas

Os trabalhos entregues deverão evitar (se possível) a utilização de caminhos absolutos ou endereços específicos, de modo a que possam ser facilmente utilizados em qualquer máquina. Para além disso, e no sentido de facilitar a receção dos vários trabalhos recebidos, estes deverão cumprir as seguintes regras:

- **Todos os elementos do grupo** deverão submeter o trabalho no link respetivo;
- Na submissão dos projetos deve ser omitida a pasta node_modules;
- O trabalho desenvolvido deverá ser entregue através do moodle, através da submissão de um ficheiro com o nome PAW_<nr_do_grupo>_<nr_do_aluno>_<nr_do_aluno2>_<nr_do_aluno3>.zip.

Datas e considerações

Os alunos devem comunicar atempadamente o seu grupo de trabalho na plataforma moodle, **até dia 20 de Abril**. Os grupos deverão ser constituídos por, no máximo, 3 elementos.

O trabalho deve ser entregue até:

- às **23h55** (hora moodle) do **dia 30 de Abril** no primeiro milestone
- às **23h55** (hora moodle) do **dia 5 de Junho** no segundo milestone

As entregas devem ser realizadas através da página da unidade curricular em <http://moodle.estg.ipp.pt>.

A defesa do trabalho será no horário respetivo para cada turma e apenas para a entrega do segundo milestone. Os alunos envolvidos no trabalho devem preparar a apresentação de forma a demonstrar o trabalho desenvolvido em cerca de 10 minutos. A apresentação poderá ser realizada no computador pessoal do aluno ou no computador disponível na sala de exame. Em ambos os casos, o aluno deverá preparar todo o conteúdo necessário para que possa demonstrar as funcionalidades desenvolvidas.

Considera-se por **defesa satisfatória**, quando o aluno **prova** que realizou o trabalho submetido e que **domina todos os conceitos aplicados** na resolução do trabalho. Tentativas de **fraude**, resultarão na avaliação do trabalho como: **Fraude Académica**.