

 <p>ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO</p>	<p>PAW- Programação em Ambiente Web</p> <p>2º Semestre ■ Docentes: FAS, EFE, JCarneiro, OAO</p> <p>Trabalho prático ■ Avaliação Contínua</p>
---	--

## Destinatários

Este trabalho destina-se a todos os estudantes inscritos na unidade curricular de Programação em Ambiente Web em Avaliação Contínua 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, 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

Considerando os tempos atuais, é necessário montar um centro de análises regional para a realização de testes despiste e imunização à Covid-19. Este centro é coordenado pelo poder político local, administrador da plataforma, que contratualizou técnicos laboratoriais para o centro de análises. Para agilizar todo o processo foi decidido que o centro de análises deve ser suportado por uma aplicação web que permita registar pedidos de diagnóstico ao centro de análises que deverá processar o pedido por um técnico do centro de análises e agendar o teste.

No pedido de teste de diagnóstico o utilizador deve poder anotar informação relevante se foi encaminhado pela linha Saúde24, se pertence a um grupo de risco ou se trabalha em locais de risco. Estes exemplos devem ter prioridade na marcação de testes.

No dia do teste o técnico do centro de análises deve verificar a identificação do utilizador e proceder ao teste de diagnóstico. A ficha do utilizador deve ser alterada para teste realizado. Quando obtidos os resultados deve ser possível registar o resultado clínico na ficha do pedido do utilizador anexando um ficheiro (pdf) com os resultados clínicos e adicionando o resultado final ao pedido.

Um utilizador sem histórico é considerado suspeito, caso exista histórico (ex: infetado) deve ser utilizado esse estado na marcação do pedido de diagnóstico. Após pelo menos 2 testes negativos, espaçados pelos menos 48h, para ser considerado não infetado. Se o primeiro teste for considerado negativo, o utilizador deve continuar com o estado suspeito/infetado e deve ser agendado automaticamente o segundo teste. Caso qualquer teste seja positivo o utilizador deve ser marcado como infetado.

Do lado dos técnicos do centro de análises apenas é necessário consultar pedidos de utilizadores pelo código do pedido, alterar o estado do pedido, anotar o resultado após os testes laboratoriais e agendar segundo teste caso seja necessário.

Por fim os administradores da plataforma, devem usar a plataforma para aceder a várias informações sobre o estado geral do centro de análises, nomeadamente:

- Número de testes realizados por dia
- Número de testes realizados por pessoa
- Números de pessoas infetadas

O administrador da plataforma tem a responsabilidade de registar os técnicos de laboratório na plataforma. Tem de ser garantido que todos os utilizadores da plataforma devem estar autenticados na plataforma para a usarem.

## 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 utilizador na plataforma
  - b. Criar/Editar/Remover técnicos de laboratório
  - c. Editar password administrador
2. Gestão de Pedidos
  - a. Criar pedido na plataforma:
    - i. Criar código aleatório de identificação
    - ii. Anotar informação relevante
    - iii. Indicar estado do utilizador
    - iv. Estado do teste
    - v. Resultado do teste
  - b. Apresentar *dashboard* para os diferentes utilizadores;
  - c. Listar pedidos de acordo com filtros específicos por cada tipo de utilizador;
3. Funcionalidades Extra para bonificação do trabalho
  - a. Uso de gráficos nos *dashboards*;
  - b. 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 e frameworks css (ex: Bootstrap) 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 de todos os serviços REST necessários à aplicação com *endpoints* desenvolvidos em *nodeJS/ExpressJS*. Os serviços devem poder ser testados usando a aplicação *postman*. Qualquer *mockup*, ou páginas experimentais em html desenvolvidas deverão ser valorizadas.

Serão também valorizados a presença de documentação da API REST e a existência de um mecanismo de autenticação de utilizadores nesta fase do projeto.

Por fim, a entrega dos elementos de avaliação no primeiro *milestone* não impede a sua modificação no segundo *milestone*.

## Milestone #2

O segundo milestone considera a entrega de um conjunto de recursos web que permita dar resposta ao enunciado integralmente. No entanto, algumas restrições adicionais são aplicadas. A qualidade da aplicação e das funcionalidades apresentadas serão alvo de avaliação.

Para a parte do servidor é obrigatório o uso de *endpoints* REST utilizando *nodeJS* e a *framework ExpressJS* para criar *webservices* de suporte à aplicação. A parte do cliente deve ser desenvolvida utilizando a *framework Angular*.

No caso de o grupo de trabalho decidir não usar a *framework Angular* e fornecer as páginas utilizando *nodeJS/ExpressJS* com *template engine*, existirá uma penalização de 3 valores na avaliação do trabalho prático. Neste caso é igualmente obrigada a utilização dos serviços REST.

## 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 Github/Gitlab

## 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)

## Formato das entregas

Os trabalhos entregues deverão evitar (se possível) utilizar 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 10 de Maio**.

O trabalho deve ser entregue até:

- às **23h55** (hora moodle) horas do **dia 16 de Maio** no primeiro milestone
- às **23h55** (hora moodle) horas do **dia 5 de Junho** no segundo milestone

As entregas devem ser feitas 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**.