

Estrutura de ensino/aprendizagem da Unidade Curricular Programação na Internet (PI)

Resumo

A Unidade Curricular (UC) de PI trata a componente de distribuição e actualização da UI em aplicações Web. Estudam-se as tecnologias de cliente mais comuns e normalizadas suportadas pelos Web Browsers: XHTML, CSS, ECMAScript, API DOM para XHTML e técnica Ajax. Descreve-se a identificação de recursos na Web através de URLs. Como protocolo de transporte estuda-se o HTTP. A tecnologia de servidor para criação de *endpoints* que produzem conteúdos dinamicamente é a infra-estrutura ASP.NET.

A avaliação da Unidade Curricular de PI é teórico/prática. A avaliação teórica consiste num exame escrito e a prática na realização de uma aplicação Web dividida por 5 fases. No final do semestre lectivo é realizada uma discussão com o respectivo docente, sobre o trabalho e a matéria leccionada na UC.

O período lectivo tem a duração aproximada de 14 semanas.

Avaliação

A avaliação da UC inclui uma componente teórica (CT) e outra prática (CP). A componente teórica consiste num teste/exame. A componente prática consiste na realização de um trabalho e respectivo relatório.

A nota final (NF) será atribuída após discussão final, onde serão validados os conhecimentos teórico-práticos dos temas tratados na Unidade Curricular.

A nota final tem a seguinte ponderação: $NF = 40\% CT + 60\% CP$.

Para obter aprovação à UC o aluno tem que obter avaliação positiva (nota mínima de 10 valores) em ambas as componentes e comprovar os conhecimentos resultantes dessas avaliações na discussão final.

Componente teórica da avaliação

A componente teórica d

uma e acrescentá-la ao trabalho comum desenvolvido até então. Não é permitido dois elementos do mesmo grupo implementarem a mesma funcionalidade. Se os alunos pretenderem implementar uma funcionalidade que faça sentido na aplicação mas que não conste nas propostas, poderão implementá-la desde que seja apresentada previamente ao docente e este a aprove.

Até ao prazo limite de entrega da 5ª parte, deve ser entregue o código da totalidade do trabalho (partes comuns e individuais) e um relatório global que descreve todo o trabalho. A componente que descreve a parte comum deve ser redigida pelos elementos do grupo (com um critério definido internamente ao grupo). A descrição das componentes opcionais deverá ser redigida, num capítulo ou secção distintos do relatório global, pelo aluno que implementou a respectiva funcionalidade, identificando devidamente a sua autoria. **Não serão aceites relatórios finais que sejam apenas a compilação dos documentos entregues em cada uma das partes.**

Os prazos de entrega do trabalho que deve resultar de cada uma das fases têm como base num sistema de créditos. A cada grupo são atribuídos 7 créditos iniciais para gerir ao longo do semestre. Cada crédito corresponde a um dia de atraso na entrega de cada uma das partes. Quando os créditos atingirem valor negativo, o grupo reprova à componente prática, reprovando assim à UC. Por cada dia de entrega antecipada será atribuído um crédito, tendo como valor máximo o valor inicial de 7 créditos.

Note-se que o trabalho final deve constituir uma aplicação coerente, nas vertentes funcionais, estruturais e de implementação. Não consiste apenas no acumulado das 5 partes, sendo necessários desenvolvimentos adicionais e, eventualmente, reestruturações parciais ou totais de componentes implementados anteriormente.

Trabalho prático a realizar por todos os grupos

O trabalho prático consiste na implementação de uma aplicação Web através da qual se joga o *Minesweeper Flags*. Este jogo é uma versão do clássico jogo *Minesweeper*, tendo como objectivo indicar a localização das bombas, em vez de as evitar. Não serão aqui dados detalhes sobre as regras básicas do jogo, constituindo a sua aprendizagem uma das competências que deve ser adquirida pelos alunos antes de iniciarem a realização da aplicação.

Antes de poderem jogar ao *Minesweeper Flags*, os utilizadores registam-se para criar um perfil. Neste, o utilizador insere informações pessoais através, de um conjunto de formulários. Cada utilizador tem uma rede de amizades (social) que é constituída através do convite e aceitação de convites de outros utilizadores.

Uma partida é jogada pelo menos por dois jogadores. Em cada momento um utilizador pode consultar os amigos que estão online, que manifestaram disponibilidade para jogar e desafiá-los para uma partida. O desafio provoca uma notificação ao utilizador convidado que pode aceitar ou recusar. A constituição dos intervenientes numa partida define-se antes de esta começar, não podendo ser adicionados novos intervenientes após o início do jogo. Durante um jogo é possível desistir. Quando há uma desistência o jogador que desiste perde, mantendo-se no tabuleiro as bombas por si indicadas e continuando o jogo para o próximo jogador, caso permaneçam pelo menos dois jogadores. Caso contrário, o jogo termina com a vitória do outro jogador. Quando o número de bombas por revelar for insuficiente para derrotar o jogador com mais bombas marcadas, o jogo termina dando a vitória a esse jogador e revelando todo o conteúdo do tabuleiro.

A descrição anterior inclui as funcionalidades mínimas que a aplicação deve ter no final, não impedindo que cada grupo decida implementar outras funcionalidades que considerem adequadas.

Segue-se uma descrição de cada uma das partes do trabalho.

Trabalho – Parte I

Tabuleiro de Jogo – Versão cliente

Descrição:

Criar uma página HTML que inclui o controlo que representa o tabuleiro do jogo e implementa a sua lógica no cliente. No final desta parte é possível dois (ou mais) jogadores jogarem sobre a mesma janela de browser.

Temas a usar no trabalho

HTML, CSS com Box Model, JavaScript, DOM

Requisitos

- O tabuleiro é gerado aleatoriamente no cliente.
- O número de bombas existente no tabuleiro deve ser ímpar.
- O clique numa célula não visível mostra a bomba (caso exista neste local) ou o número de bombas que lhe estão adjacentes.
- Os jogadores jogam à vez, sendo indicado o jogador actual (o que deve realizar a próxima jogada).
- Quando um dos jogadores ganhar é dada esta indicação e reveladas as bombas que ficaram por indicar (caso existam).

Trabalho – Parte II

Tabuleiro de Jogo – Versão servidor

Descrição:

Alteração do tabuleiro de jogo desenvolvido na parte I do trabalho, de modo a que a este seja criado e gerido no servidor. No final desta parte é possível dois (ou mais) jogadores jogarem em janelas de *browser* distintas.

Temas a usar no Trabalho

Ajax e *handlers* ASP.NET

Requisitos

- O tabuleiro é gerado aleatoriamente no servidor.
- O número de bombas existente no tabuleiro deve ser ímpar.
- O clique numa célula não visível tem o mesmo resultado visual que o apresentado na parte I do trabalho. No entanto, no tratamento do clique é realizado um pedido Ajax a um *handler* no servidor que retorna, assincronamente, a indicação de bomba nesse local ou o número de bombas que lhe estão adjacentes.

Trabalho – Parte III

Páginas de perfil e submissão de fotos

Descrição:

Formulários de preenchimento, edição e visualização da página de perfil e gestão jogos.

Temas a usar no Trabalho

ASP.NET MVC

Requisitos

- Formulário para inserção/edição dos dados do perfil;
- Visualização dos dados do perfil;
- Suportar o envio de desafios e aceitação de desafios.
- Modo *online/offline* para aceitar desafios. No modo *offline* o jogador não consta na lista de jogadores disponíveis para desafios.
- Apresentação da lista de amigos disponíveis para jogar, através da qual são realizados os desafios.
- Jogo entre dois ou mais utilizadores, com possibilidade de desistência.

Trabalho – Parte IV

Implementação do mecanismo de Login

Descrição:

Implementação do mecanismo de Login (autenticação e autorização) da aplicação.

Temas a usar no Trabalho

Módulos e/ou `HttpApplication`, acesso a ficheiros de configuração.

Requisitos

- Desenvolver controlo Web (Login) que disponibiliza a funcionalidade esperada de um controlo de login, ou seja, apresentar caixas de texto para introdução no nome de utilizador (*username*) e respectiva palavra-chave (*password*);
- Suportar gestão de utilizadores (criação e remoção);
- Módulo de autenticação e autorização;
- Autorização através da propriedade `HttpContext.User` e com acesso a ficheiro de configuração (`Web.config`).

Informação adicional

A funcionalidade de *login* deve ser implementada através de um módulo http. Este intersecta todos os pedidos e verifica se o utilizador corrente está autenticado. Em caso afirmativo viabiliza o processamento normal do pedido. Caso contrário, redirecciona o cliente para uma página de autenticação (configurável) que contém um formulário *login*, para inserção dos dados de autenticação. A implementação desta funcionalidade pode ser baseada no mecanismo de autenticação já existente em ASP.NET (`FormsAuthentication`). No entanto, salienta-se que este mecanismo apenas deve servir de inspiração e não ser utilizado directamente.

Trabalho – Parte V

Implementação de funcionalidades adicionais

Descrição:

Nesta parte, cada elemento do grupo deve implementar uma funcionalidade adicional ao trabalho, escolhida da lista apresentada a seguir, e acrescentá-la ao trabalho comum desenvolvido até então. Não é permitido a elementos do mesmo grupo implementarem a mesma funcionalidade. Se os alunos pretenderem incluir uma funcionalidade que faça sentido na aplicação mas que não conste nas propostas, poderão fazê-lo, desde que seja apresentada previamente ao docente e este a aprove.

Além da implementação da funcionalidade adicional, o aluno deve fazer uma análise crítica ao trabalho desenvolvido e alterá-lo, de modo a melhorar estrutural e tecnologicamente o que já foi implementado. Nesta fase, onde toda a matéria da disciplina já foi tratada, é expectável que os alunos tenham adquirido conhecimento de outras soluções tecnológicas que melhor se adequam à implementação de determinadas componentes da aplicação, desenvolvidas em fases anteriores. Essas alterações devem ser devidamente identificadas e justificadas no relatório.

De seguida apresentam-se exemplos de funcionalidades adicionais.

Funcionalidades adicionais propostas:

- Gestão da rede social do utilizador através de lista de amigos;
- Envio/recepção de mensagens privadas para/de outros utilizadores;
- Gravação do estado dos jogos para posterior continuação (por utilizador)
- Jogo em equipas;

- Jogo em que um (ou mais) dos intervenientes joga automaticamente (CPU). Esta funcionalidade permite um jogo com apenas um jogador humano.
- Criação de salas de jogo, públicas e privadas;
- Chat;
- Fórum;
- Etc... Usem a vossa imaginação ☺!

Componentes tecnológicas para as funcionalidades adicionais e/ou reestruturações:

Algumas tecnologias que não são tema da UC, mas que são relevantes para o desenvolvimento de aplicações *web*, podem ser usadas de modo a enriquecer o trabalho. A sua utilização implica um estudo por parte do aluno, de modo a que esteja em condições de as utilizar na aplicação e responder a perguntas sobre esse tema na discussão. Segue-se uma lista dos tópicos sugeridos, não invalidando a utilização de outros que aqui não constam.

- *Master Pages*;
- Temas;
- JQueryUI;
- Mecanismo de autorização das ASP.NET (1.1 ou 2.0).

Distribuição das matérias/trabalhos pelas semanas

A tabela seguinte apresenta a distribuição dos temas, dos slides de suporte, respectivas demos e dos trabalhos práticos para a disciplina de PI. Este é apenas um planeamento indicativo, podendo ser alterado por cada um dos docentes por motivos inerentes à dinâmica do semestre lectivo, ponderados ou imponderados.

Semana	Temas	Slides	Trabalho
1	Apresentação; WWW e Sistemas Distribuídos	PI - (01) Apresentação PI - (02) WWW e Sistemas Distribuídos	
2	XHTML (sem Forms); CSS	PI - (03) XHTML PI - (04) CSS	
3	JavaScript & DOM	PI - (05) Javascript PI - (06) DOM	
4	HTTP	PI - (07) HTTP	Parte I
5	Arquitecturas de extensão do servidor HTTP; ASP.NET: <i>HTTP Pipeline</i> – Handlers	PI - (08) Arq. extensão... PI - (09) HTTP Pipeline	
6	Técnica AJAX	PI - (10) AJAX	
7	ASP.NET.MVC	PI - (11) ASP.NET.MVC	Parte II
8	C# 3.0 e Linq	PI - (17) C# 3.0	
9	ASP.NET.MVC: Controllers	PI - (12) ASP.NET.MVC - Controllers	
10	ASP.NET.MVC: Views	PI - (13) ASP.NET.MVC - Views	
11	ASP.NET: Model	PI - (14) ASP.NET.MVC – Model	Parte III
12	ASP.NET: StateManagement	PI - (15) StateManagement	
13	ASP.NET: Http Pipeline – Http Application e Módulos	PI - (09) HTTP Pipeline	
14	ASP.NET.MVC: Unobtrusive JavaScript and Hijacking *	PI - (16) Unobtrusive JavaScript and Hijacking	Parte IV

Datas importantes

A tabela seguinte apresenta as datas importantes previstas, a ter em consideração durante o semestre. Algumas destas datas poderão ser alteradas, dependendo da evolução da matéria dada em cada turma durante o semestre. Qualquer alteração será atempadamente transmitida aos alunos.

Datas importantes	Descrição
08-10-2009	Publicação do enunciado do trabalho de PI
28-09-2008	Limite para entrega de propostas de trabalho alternativas
Semana 19 a 23 de Outubro	Entrega da 1ª parte do trabalho
Semana 09 a 13 de Novembro	Entrega da 2ª parte do trabalho
Semana 07 a 11 de Dezembro	Entrega da 3ª parte do trabalho
Semana 04 a 07 de Janeiro	Entrega da 4ª parte do trabalho
A definir	Entrega do trabalho e respectivo relatório
<i>Fevereiro/Março, após aprovação na componente teórica.</i>	Discussão do trabalho

ISEL, Outubro de 2009.

Autores,
Luís Falcão, Carlos Guedes e Miguel Carvalho