UTFPR UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ	Universidade Tecnológica Federal do Paraná Campus Guarapuava Curso Superior de Tecnologia em Sistemas para Internet Disciplina: Desenvolvimento para Web 3

Instruções gerais

- A avaliação é individual.
- Cada aluno deve escolher um tema e avisar o professor no início do semestre.
- Cada tema pode ser escolhido por até 5 alunos.
- Os temas disponíveis estão listados neste documento. O aluno dependente ou que já faz TCC pode escolher qualquer tema, mesmo que não esteja listado neste documento.

Avaliação 1: Protótipo do Cliente

- A avaliação valerá 0,5 ponto extra.
- O aluno deverá entregar todas as telas do protótipo, feitas em HTML e CSS.
- É recomendado usar *framework* ou biblioteca CSS.
- JS é opcional e não valerá nota.
- Se a aplicação não funciona por causa do JS, o aluno perderá nota.
- Não deve ser implementado código no lado servidor.
- Não deve ser usado banco de dados.
- Não deve ser usado servidor web.
- Todos os arquivos devem estar armazenados localmente. Exemplo: não use um link para um arquivo CSS ou JS hospedado externamente. Isso torna o desenvolvimento lento e pode ser que o arquivo não esteja disponível quando o professor for corrigir o protótipo.
- As telas devem ter o visual final da aplicação, com possível navegação em todas as funcionalidades.
- Deve ser possível simular um cliente usando a aplicação.
- As telas devem conter dados falsos, como se já existissem dados cadastrados.
- Os dados falsos são estáticos.
- O visual deve ser bonito, sem erros e lembrar um aplicação real. Procure sistemas reais na internet para que o visual não fique ruim ou muito fora do padrão.
- O visual deve ser para uma tela horizontal (tela de computador).
- A partir da tela inicial, deverá ser possível que o cliente use o protótipo como se fosse um produto final. Assim, ele deverá conseguir acessar qualquer página a partir da inicial.
- Não pode haver o erro 404 “página não encontrada”.
- Sempre que possível, todo link deve levar o usuário para alguma página.

Dicas:

- Não crie campos desnecessários nos formulários.
- Não coloque muitos campos por formulário, nem crie várias funcionalidades extras, porque isso dificultará o desenvolvimento da aplicação final.
- É mais importante satisfazer os requisitos exigidos no projeto do que implementar funcionalidades extras.
- Baseie o visual do protótipo em sistemas reais que tenham um visual agradável.

- Não crie telas “sem saída”. Quando o usuário chegar em uma tela, deve ser possível a ele navegar para outras telas. Isso vale para todas as telas da aplicação. Exemplo: na tela de cadastro de usuário, deixe um botão para ir para a tela de login.

Avaliação 2: Aplicação Final

- O aluno deve desenvolver o código do tema escolhido, baseando-se nas telas projetadas.
- A aplicação deve ter testes automatizados. Se a aplicação tiver a funcionalidade de upload, especificamente a parte de upload não precisará ser testada.
- A entrega da aplicação completa será pelo Moodle.
- O limite de tamanho da aplicação é 20MB.
- Use PHP versão 8.1.x.
- Use MariaDB versão 10.4 ou superior.
- Use Apache Server 2.4.53 ou superior.
- Para Windows, aconselho usar o ambiente XAMPP: https://www.apachefriends.org/pt_br/index.html
- Para Linux, aconselho instalar tudo separadamente via terminal, usando tutoriais da internet para saber o que precisa ser instalado e onde é salva a instalação. Além disso, antes de instalar, é necessário verificar o que já vem instalado na distribuição, porque muitas já vêm com programas de desenvolvimento instalados.
- É opcional usar JS. Ele não valerá nota. Erros no JS descontará nota.
- É recomendado usar *framework* ou biblioteca CSS. Será descontada nota caso o visual seja ruim ou tenha problemas.
- É obrigatório usar o framework DW3: https://github.com/guilhermedacsilva/web3

Requisitos gerais:

- Verifique se a SQL de criação do seu banco de dados está correta. Não use gerador de SQLs. Treine escrever a SQL manualmente. A SQL será rodada no computador do professor, direto no PHPMyAdmin. Se houver algum erro nela, o aluno terá nota zero, porque não será possível criar o banco de dados da aplicação.
- Cada *framework* e linguagem possui padronizações diferentes. Siga a padronização do banco de dados ensinada em aula. Ela se baseou na padronização do Laravel.
- Deve existir um login de usuário. Deve ser utilizada a autenticação com criptografia ensinada em aula.
- Sempre que o usuário cadastrar, editar ou deletar algo, mostre uma mensagem dizendo se funcionou ou não.
- Todos os formulários precisam ser validados no lado servidor. Todos os campos precisam ser validados, nem que seja pelo tamanho mínimo.
- A validação no lado cliente por JS é opcional e não vale nota.
- Os testes automatizados devem testar todas as rotas e todos os métodos dos modelos. Não precisam ser testados os métodos GETs e SETs dos modelos.

A seguir estão os temas disponíveis.

PROJETO A: VENDA ONLINE

Nota	Logado	Funcionalidade
10		O sistema somente pode ser acessado por um usuário logado. Cadastro de usuário e login.
15	X	Cadastro de oferta de produto. Informação mínima: descrição do produto, preço e imagem.
15	X	Listagem de ofertas disponíveis. Ordenar as mais antigas em primeiro. Filtrar por descrição. Mostrar somente as ofertas disponíveis.
15	X	Fechar negócio. Um usuário pode clicar em "comprar" e assim a oferta não estará mais disponível. O dono da oferta não pode vender para ele mesmo. Somente uma pessoa pode comprar em uma oferta.
15	X	Relatório de compras do próprio usuário. Relatório de vendas do próprio usuário.
10	X	Deletar oferta. Uma oferta já finalizada não pode ser deletada. Só o dono da oferta pode deletá-la.
20		Testes automatizados.

PROJETO B: RH++

Nota	Logado	Funcionalidade
10		O sistema somente pode ser acessado por um usuário logado. Cadastro de usuário e login. Existem três tipos de usuários: programador, chefe e RH. O programador começa com a situação "disponível". Essa aplicação será utilizada só por uma única empresa.
15	X	O chefe pode acessar a listagem com todos os programadores e a situação deles (disponível, convidado, aceite ou contratado). Ordenar pelo nome. Filtrar por nome.
15	X	O chefe pode convidar ou desconvidar um programador para ser contratado. Se o chefe desconvida, o programador volta para "disponível".
15	X	O programador pode acessar somente seu perfil. Lá aparecerá se ele tem ou não um convite para ser contratado. O programador pode aceitar ou rejeitar o convite. Se ele rejeitar, voltará para "disponível".
15	X	O RH pode acessar uma listagem com os programadores que aceitaram o convite para ser contratados. O RH poderá marcar o programador como contratado.
10	X	O RH pode acessar uma listagem de programadores contratados.
20		Testes automatizados.

PROJETO C: ROLÊS

Nota	Logado	Funcionalidade
5		Cadastrar usuário
5	X	Cadastrar cidade.
20	X	Cadastrar rolê. Informações mínimas: nome, descrição, foto, nota, horário, cidade (selecionada do cadastro de cidades). A nota é um inteiro de 0 a 5. O horário pode ser "dia" ou "noite".
20		Listagem de rolês. Não precisa estar logado para acessar a listagem. Ordenar por nota (maior) e por nome (alfabeticamente). Filtrar por cidade.

10	X	Deletar rolê. Só o dono do rolê pode deletá-lo.
20		Relatório: total de cidades cadastradas, total de rolês cadastrados, quantidade de rolês divididos por nota (exemplo: 30 rolês com nota 5, 10 rolês com nota 4...), total de rolês de dia cadastrados e total de rolês de noite cadastrados.
20		Testes automatizados.