

MCZA019 - Programação para Web

Plano de ensino

Prof. Diogo S. Martins
Centro de Matemática, Computação e Cognição
Universidade Federal do ABC

Q1 2021 (Suplementar)
v. 01/02

1 Informações básicas

- TPI: 2-2-4
- Horários oficiais:
 - Diurno seg 10-12h semanal
 qua 08-10h semanal
 - Noturno seg 21-23h semanal
 qua 19-21h semanal
- Ferramentas de comunicação:
 - Discord
Link da sala: <https://discord.gg/EXDfag94F8>
Atenção: mudar o nickname do server para o seu nome real (tutorial: <https://bit.ly/3pAOXhz>)
 - Comunicação com o professor
 - * Prioritariamente via Discord, por mensagem pública ou privada
 - * Por email (caso Discord esteja indisponível): santana.martins@ufabc.edu.br
- Plantão de dúvidas:
 - qua 11-12h Discord
 - qua 18-19h Discord

Os plantões são nossos únicos eventos síncronos. O objetivo é esclarecer dúvidas e/ou reforçar temas vistos nas videoaulas e outros materiais. Caso haja interesse em participar do plantão, agendar slot no Google Calendar (<https://bit.ly/3owxJAc>).

As comunicações com o professor, exceto no horário de plantão, serão assíncronas, em geral com resposta dentro de 24h em dias úteis.
- Sala no Moodle: pw-2021.1
<https://moodle.ufabc.edu.br/course/view.php?id=1228>
Todos já foram convidados para a sala, em caso de dificuldade de acesso, entrar em contato com o professor.

2 Descrição da disciplina

A disciplina aborda os principais aspectos do desenvolvimento de aplicações web full-stack. Trata de temas relevantes ao desenvolvimento de clientes, servidores, documentos estruturados, *scripting*, programação assíncrona e integração com sistemas de banco de dados. Em profundidade, aplica os conceitos por meio de pilha de tecnologias baseadas em Javascript/Typescript.

3 Requisitos recomendados

Para participar dessa disciplina é recomendação oficial ter cursado e sido aprovado em:

- BC1501 - Programação Orientada a Objetos
- MC3310 - Banco de Dados

A disciplina também demanda conhecimentos de sistemas distribuídos e engenharia de software, que podem ser compreendidos/reforçados no contexto das aulas.

4 Objetivos

- Familiarizar-se com os fundamentos do desenvolvimento de aplicações Web;
- Ser capaz de projetar arquiteturas de aplicações Web utilizando e refinando padrões de projeto;
- Ser capaz de desenvolver aplicações Web seguindo os princípios de alta coesão e baixo acoplamento.

Ao final do curso, espera-se que o aluno, aprovado com conceito satisfatório, possua habilidades que permitam, a partir de um conjunto de requisitos, projetar aplicações Web que sejam eficientes, confiáveis, seguras e de fácil manutenção.

5 Bibliografia

1. Connolly, R. e Hoar, R. *Fundamentals of Web Development*, 1st edition. Pearson, 2014.
2. Deitel, Paul J., Deitel, Harvey M. e Deitel, A. *Internet & World Wide Web: How to Program*, 5th edition. Prentice Hall, 2011.
3. Sebesta, Robert W. *Programming the World Wide Web*, 8th edition. Pearson Addison Wesley, 2014.
4. Materiais online (tutoriais, guias, referências, etc.) trazidos no contexto de cada aula.

6 Metodologia de ensino-aprendizagem

Formato das aulas. As aulas serão assíncronas, em formato de vídeo, publicadas duas vezes por semana. Cada aula estará disponível nos respectivos dia e horário oficiais da turma. O conteúdo englobará tutoriais, os quais, idealmente, devem ser seguidos de forma síncrona, ou seja, que os alunos tentem implementar os exemplos ao mesmo tempo em que assistem ao vídeo. Essa abordagem tende a tornar mais explícitas as eventuais dificuldades e dúvidas.

Interações. Em duas modalidades: i) assíncrona, via Discord na modalidade texto, com prazo de 24 horas para resposta (em dias úteis); e ii) síncrona, via Discord nas modalidades texto ou videoconferência, no horário de plantão de dúvidas (vide seção 1). Além da interação professor-alunos é incentivada a interação alunos-alunos, para discussões, esclarecimentos de dúvidas, etc.

Técnicas de ensino aprendizagem:

- **Aprendizagem ativa**¹. As atividades das videoaulas enfatizam a análise, a síntese e a avaliação dos conteúdos, ou seja, haverá menor ênfase no consumo passivo de conteúdos, por isso é importante realizar os tutoriais sincronamente, como se estivesse no laboratório presencial;

¹https://en.wikipedia.org/wiki/Active_learning

- **Aprendizagem baseada em problemas².** O conteúdo das aulas é estruturado fundamentalmente em problemas-base concretos, os quais servem de contexto para abordar os temas conceituais/abstratos da ementa. A aquisição do conhecimento teórico-conceitual é abordado via estudo prévio ou posterior com os *materiais complementares* (referências de estudo, que podem ser textos ou vídeos) indicados no roteiro da respectiva aula.

7 Avaliação

A avaliação somativa consiste nos componentes dados pela Equação 1, onde:

$$N_F = 0.55 \cdot N_{atv} + 0.45 \cdot N_p \quad (1)$$

- N_{atv} é a média de 75% das atividades somativas com as melhores notas. Como total para calcular a porcentagem, consideramos todas as atividades semanais enunciadas durante o quadrimestre;
- N_p é a nota da prova.

O conceito final será obtido de acordo com a Equação 2.

$$C_F = \begin{cases} A, & \text{se } N_F \in [8.5, 10.0] \\ B, & \text{se } N_F \in [7.0, 8.5) \\ C, & \text{se } N_F \in [5.5, 7.0) \\ D, & \text{se } N_F \in [5.0, 5.5) \\ F, & \text{se } N_F \in [0.0, 5.0) \\ O, & \text{se ausência exceder 25\%} \end{cases} \quad (2)$$

A frequência na disciplina será mensurada via:

- Frequência de entrega das avaliações (atividades e prova).
- Frequência de acesso aos conteúdos no Moodle.

Sobre as atividades

Teremos por volta de uma atividade por semana, com prazo de entrega de 7 dias. Todas as atividades são somativas, ou seja, valem nota. As atividades serão entregues no ambiente Github Classroom. O conteúdo do ciclo 1 engloba treinamento sobre o uso da plataforma. Caberá ao aluno demonstrar disciplina e organização de tempo para executá-las, visto que são planejadas para preencher o componente I (4 horas/sem.) previsto oficialmente para a disciplina.

Sobre a prova

A prova será remota, via Moodle. Seguirá a metodologia de avaliação baseada em projetos. Consiste na construção de uma aplicação, completa ou parcialmente, com base em um conjunto de requisitos enunciados. A prova será organizada em duas fases:

- Na **parte 1**, com duração de 1 semana, ocorrerá a familiarização com o código-base do sistema, e possivelmente a implementação de algumas partes faltantes;
- Na **parte 2**, com duração de 3 horas, na semana agendada para a prova (vide cronograma), em dia e horário a escolha do aluno, serão implementadas funcionalidades adicionais, com base no código da parte 1.

No final, será feita apenas uma entrega, na data prevista no cronograma, correspondente aos resultados das partes 1 e 2. O prazo será controlado automaticamente pelo sistema de submissão. Não serão aceitas provas atrasadas ou entregues por meios de submissão alternativos. A dinâmica específica da prova será divulgada junto aos respectivos enunciados, no momento adequado.

²https://en.wikipedia.org/wiki/Problem-based_learning

Critérios de avaliação

Todas as avaliações somativas, isto é, atividades e prova, qualificam-se como atividades de programação. A nota máxima de cada avaliação será obtida apenas se a mesma for entregue no prazo e executada correta e completamente. Os programas solicitados em atividades de avaliação serão submetidos aos seguintes tipos de verificações:

- **Verificações automáticas.** Testes de corretude, de estilo de programação, de erros comuns e de detecção de plágio.
- **Verificações manuais.** O professor inspecionará os programas para verificar os seguintes critérios gerais:
 - **Eficiência:** os programas desenvolvidos deverão ter bom desempenho, o que pode englobar o tratamento adequado dos seguintes fatores:
 - * ler e escrever dados nas quantidades mínimas necessárias para resolver o problema;
 - * não desperdiçar memória primária (RAM), tanto no cliente como no servidor;
 - * acessar memória secundária (disco, banco de dados, etc.) somente quando necessário e sem redundância;
 - * minimizar comunicação em rede, tanto em tempo de operação quanto em quantidade de dados transmitidos (i.e. não baixar os mesmos dados múltiplas vezes, fazer o máximo de trabalho possível/viável no cliente ou no servidor antes de transmitir, etc.);
 - * entre outros.
 - **Acurácia:** o programa deverá atender adequadamente a todos os requisitos enunciados para a atividade;
 - **Corretude:** o programa deverá atender a todos os requisitos enunciados e a solução deverá estar funcionalmente correta;
 - **Estrutura e organização do código:** atentar principalmente aos seguintes aspectos:
 - * **Auto-documentação:** nomes intuitivos para variáveis e métodos/funções;
 - * **Modularização:** funções/métodos/classes com alta concisão e baixo acoplamento, isto é, que sejam em sua maioria curtos, e que realizem preferencialmente uma única tarefa;
 - * **Comentários:** documentação completa porém ao mesmo tempo concisa (sem poluição visual, apenas nos lugares adequados e necessários).
 - **Autenticidade:** o código é original e não foi copiado de outras fontes (i.e. web, livros, terceiros, etc.), sob pena das sanções previstas no código de honra.

7.1 Mecanismos de avaliação substitutivos

Teremos dois mecanismos de avaliação substitutivos:

- Para as atividades, como consideraremos 75% das maiores notas, basta entregar atividades com atraso até atingir no mínimo 75% de entregas;
- No caso da prova, que é assíncrona com janela de uma semana, é desnecessário o mecanismo de substituição, visto que o aluno pode escolher o melhor dia/horário para efetuar a prova.

7.2 Mecanismo de recuperação

A recuperação será aplicada apenas aos alunos que tiverem conceito final D ou F. Ocorrerá no início do próximo quadrimestre, em formato similar ao estabelecido para as outras provas.

A nota obtida na prova de recuperação (N_R) será usada para obter a nota final com recuperação (N_{FR}), que consiste na média estabelecida pela Equação 3.

$$N_{FR} = \frac{N_F + N_R}{2} \quad (3)$$

O conceito final obtido na recuperação (C_{FR}) é o conceito que entrará no histórico, obtido de acordo com os limiares para a nota final de recuperação (N_{FR}) dados pela Equação 4.

$$C_{FR} = \begin{cases} C, & \text{se } N_{FR} \geq 5.5 \\ D, & \text{se } N_{FR} \in (5.0, 5.5) \\ F, & \text{se } N_{FR} \leq 5.0 \end{cases} \quad (4)$$

No caso dos alunos que não participarem da recuperação, $C_{FR} = C_F$.

8 Cronograma de aulas

Como todas as atividades são assíncronas, as datas servem apenas como referência para liberação dos respectivos materiais.

O cronograma a seguir pode variar de acordo com o aproveitamento aferido nas turmas durante o quadrimestre.

Aula #	Data	Tema
1	01/02	Introdução ao desenvolvimento Web
2	03/02	HTML
3	08/02	CSS
4	10/02	Design responsivo
5	15/02	Javascript
6	17/02	Javascript
7	22/02	Typescript
8	24/02	Typescript
9	01/03	Programação assíncrona
10	03/03	Ajax
11	08/03	Single Page Applications (SPA)
12	10/03	Single Page Applications (SPA)
13	15/03	Servidores com Node.js
14	17/03	Servidores com Node.js
15	22/03	Server-side rendering (SSR) com Express
16	24/03	Server-side rendering (SSR) com Express
17	29/03	Templates para SSR com Handlebars
18	31/03	Model-View-Controller (MVC)
19	05/04	Bancos de dados com MongoDB
20	07/04	Bancos de dados com MongoDB
21	12/04	Sessões Web
22	14/04	Sessões Web e autenticação
23	19/04	Revisão + Prova
24	21/04	Prova

9 Código de honra

A aprovação na disciplina é baseada exclusivamente no esforço e trabalho pessoal do discente, ao qual cabe garantir que não ajudará ou receberá ajuda não-permitida em qualquer atividade usada pela equipe docente para fins de avaliação (e.g. provas, trabalhos, listas, etc.).

Exemplos de violação do código de honra incluem:

- Copiar atividades avaliativas (e.g. listas, trabalhos, provas, etc.) ou permitir que outros discentes copiem suas atividades avaliativas;
- Colaboração não-permitida entre indivíduos ou grupos (e.g. oferecer vantagens em troca de soluções prontas, doar trechos para o trabalho de outro grupo, etc.);

- Permitir que outros assumam sua identidade em atividades avaliativas (e.g. entregar trabalho que não fez ou permitir que outros façam provas por você);
- Plágio (i.e. aplicável a textos, programas de computador, etc.), o que envolve copiar porções significativas de textos ou programas de terceiros, sem atribuição de autoria;
- Receber ou conceder ajuda em atividades avaliativas quando o contexto mostra que não é sensato receber tal ajuda.

Como consequências de violação do código de honra tem-se:

- Reprovação automática na disciplina, com conceito F;
- Denúncia na Comissão de Transgressões Disciplinares Discentes da Graduação, a qual decidirá sobre a punição adequada à violação, o que pode levar a advertência, suspensão ou desligamento, de acordo com os arts. 78-82 do Regimento Geral da UFABC.