

Instruções para o projeto final

MCZA019 - Programação para Web – 2020-Q1

Prof. Diogo S. Martins

v. 11/02/2020

Resumo das datas de entrega

- Parte 0: 18/02
- Parte 1: 10/03
- Parte 2: 24/03
- Parte 3: 07/04
- Parte 4: 06/05
- Parte 5: 11/05

1 Objetivo

O projeto final consiste em desenvolver uma aplicação web que utilize os conceitos e as tecnologias trabalhados durante o curso, ou seja, a aplicação deverá ser desenvolvida necessariamente utilizando Node.js como servidor. Já os outros componentes da pilha MEAN (MongoDB, Express e Angular) podem ser substituídos/aumentados com outras tecnologias, desde que a substituição seja devidamente justificada.

2 Escopo do tema

A aplicação deverá se enquadrar em um dos seguintes domínios de aplicação:

- Saúde
- Esportes
- Educação
- Mobilidade urbana
- Meio ambiente
- Cidadania
- Acessibilidade
- Causas sociais

3 Dinâmica e estrutura do projeto

1. **Dinâmica geral.** A dinâmica do projeto prevê o desenvolvimento segundo um processo incremental e iterativo. O processo é incremental pois o sistema será refinado e construído continuamente, até a obtenção de um protótipo final. O processo é iterativo pois em cada fase haverá a possibilidade de solucionar os problemas apontados na fase anterior.
2. **Composição da equipe.** Devido ao tamanho da turma e à necessidade de haver apresentação oral dos projetos em data específica, o projeto precisa ser feito em grupos de três integrantes.
3. **Entregáveis.** Cada fase do projeto define um conjunto de artefatos (documentos, códigos-fonte, executáveis, etc.), cada um dos quais tem entrega obrigatória. Consequentemente, a ausência de algum dos artefatos solicitados implicará em desconto parcial ou total da nota, a depender da complexidade e importância do artefato. É importante que haja plena dedicação do grupo durante todo o desenvolvimento, pois a avaliação do projeto, e consequentemente a nota obtida, considerará todas as fases do projeto, e não apenas os resultados obtidos na fase final. A entrega dos artefatos de cada fase deverá ser feita no TIDIA-AE, por apenas um dos membros de cada grupo, o qual ficará responsável por comunicar o feedback do professor aos outros membros do grupo.
4. **Repositório de código.** Cada projeto deverá ser hospedado em um repositório de controle de versão online (e.g. GitHub, GitLab, Bitbucket, ou outro), o qual pode ser privado ou público, a critério do grupo¹. No repositório poderão ser mantidos, além do código-fonte, opcionalmente, outros artefatos solicitados, como documentos, imagens, etc. (por exemplo, caso não caibam no Tidia). Caso não saiba usar ferramentas de controle de versão, consulte a documentação de algum dos sites sugeridos, ou entre em contato com o professor, para que seja indicado material técnico complementar.
5. **Acompanhamento do desenvolvimento via o repositório.** O acompanhamento do projeto pelo professor ocorrerá também pela atividade observada no repositório. Mesmo que o trabalho seja em grupo, haverá avaliação da contribuição individual dos integrantes (vide seção 5). É preciso que todos os membros do grupo participem dos *commits*. Isso vai exigir uma boa coordenação da equipe, boa distribuição de tarefas, etc., fatores cruciais para qualquer projeto de desenvolvimento de software. Todos os integrantes deverão contribuir com tarefas de programação (por exemplo, concentrar a contribuição somente na redação dos documentos, dos relatórios, etc., não é suficiente).

Projetos bem executados costumam apresentar commits bem distribuídos no tempo (e não exclusivamente na véspera da entrega), indicando trabalho contínuo e consistente — isso é válido independentemente do número de integrantes da equipe. Projetos que, mesmo bem feitos, sejam publicados como um todo em cima do prazo implicarão em descontos de nota. Recomenda-se também usar o sistema de *issue tracking* do repositório, pois a interação que ocorrer nos tickets ajudará a identificar a contribuição dos integrantes do grupo.

¹Caso seja privado, o professor deverá ser incluído como usuário do projeto

4 Orientações

Parte 0: Proposta

Objetivos:

- Definir o tema do projeto

O que entregar: Título e resumo do projeto. No resumo é *obrigatório* especificar os seguintes itens explicitamente:

- *Contexto*: qual a motivação para desenvolver o sistema, o que/quem se beneficia e como?
- *Problema*: Defina o problema e como esse problema é tratado atualmente via tecnologias web (ou se não é tratado, por que não).
- *Proposta*: Como você planeja tratar o problema, qual será a abordagem?
- *Justificativa*: Por que sua abordagem pode ser considerada adequada e como se diferencia de abordagens competidoras (i.e. sistemas similares)? Pesquise outros sistemas que façam tarefas parecidas e analise as vantagens e desvantagens de cada um, e qual seria o diferencial do que você está propondo.
- Lista dos **integrantes** do grupo

Como será avaliado: O professor analisará a proposta e enviará feedback com orientações que deverão ser trabalhadas na próxima fase.

Parte 1: Concepção

Objetivos:

- Refinar o escopo da aplicação;
- Detalhar o problema a ser resolvido pela aplicação.

O que entregar:

- Documento de concepção, em formato PDF. O documento deverá apresentar, com clareza, todos os seguintes aspectos principais do projeto (cada aspecto é uma seção/subseção):
 - **Introdução** (contextualização e detalhamento) do problema a ser resolvido pelo sistema. Incluir como título o **nome** do sistema (mesmo que seja preliminar)
 - **Metodologia** a ser empregada. Responder:
 - * Quais são os principais componentes previstos para a arquitetura do sistema? Apresentar diagrama (e.g. blocos, componentes, UML, etc.).
 - * Que tipos de dados serão processados pela aplicação?
 - * Quais são as aplicações similares ou relacionadas com a que você está propondo? Citar adequadamente, especificar vantagens e desvantagens.
 - Como será a **avaliação** da aplicação? Quais conjuntos de dados serão utilizados? Será necessário testar com usuários?
 - Seção de complementação: reportar como os problemas identificados na Parte 0 foram solucionados.

Como será avaliado: O professor analisará o documento de concepção e enviará feedback com orientações que deverão ser trabalhadas na próxima fase.

Parte 2: Prototipação em baixa fidelidade

Objetivos:

- Decomposição das funcionalidades da aplicação, do ponto de vista do *front-end*;
- Projeto da interação com a aplicação.

O que entregar: Para essa fase, apoiar-se em material complementar sobre prototipação que será fornecido pelo professor com antecedência.

- Documentação do protótipo, em PDF, com as seguintes seções:
 1. Cenário textual (caso de uso global) ilustrando como um usuário típico utilizaria o sistema;
 2. Proposta de logomarca para o sistema;
 3. Lista de todas as tarefas realizadas pelo sistema, com detalhamento;
 4. *Storyboard* para cada tarefa realizada no sistema (vide documentação com exemplos de storyboard);
 5. Seção de complementação, reportando como os problemas identificados na Parte 1 foram solucionados.

Como será avaliado: O professor analisará a documentação do protótipo e enviará feedback com orientações que deverão ser trabalhadas na próxima fase.

Parte 3: Prototipação em média fidelidade

Objetivo:

- Gerar um protótipo preliminar, com ênfase no *front-end*, apresentando navegabilidade e interatividade semi-funcional.

O que entregar:

- Documentação do protótipo (em PDF), contendo:
 - Informações relevantes para compreender o protótipo, por exemplo: modificações que o grupo achou necessário desde a última versão, telas que se mostraram desnecessárias em relação ao proposto nos storyboards, etc.
 - Seção de complementação, reportando como os problemas identificados na Parte 2 foram solucionados;
 - Vídeo (screencast), demonstrando as telas/páginas que foram implementadas e a navegabilidade do sistema (basear-se no cenário formulado na parte 2, por exemplo);
 - Código-fonte das páginas do front-end (no repositório do grupo)
 - Link para o **repositório** do projeto
- Nota: o professor clonará o repositório e navegará no sistema para testar, além de analisar o código-fonte. Portanto, garanta que o sistema tenha todas as instruções e dados necessários para a adequada execução.

Como será avaliado: O professor analisará o protótipo e enviará feedback com orientações que deverão ser trabalhadas na próxima fase.

Parte 4 (final): Prototipação em alta fidelidade

Objetivos:

- Desenvolvimento de um protótipo funcional do sistema, incluindo backend;
- Apresentação dos resultados obtidos.

O que entregar:

- Relatório final, em formato PDF, de no máximo 6 páginas, reportando o projeto e especificando a contribuição de cada participante. O relatório deverá ser apresentado em formato de artigo, obrigatoriamente usando o modelo da SBC (<http://tinyurl.com/sbc-template-artigos>). Deverá conter pelo menos as seguintes seções (é possível propor outras, porém as listadas abaixo são as obrigatórias):

1. Introdução

Contexto do problema, definição do problema, ideia geral da solução, visão geral do que o grupo conseguiu realizar como resultado;

2. Arquitetura do sistema

Definir diagramas que ilustrem adequadamente a arquitetura, de acordo com as boas práticas de documentação em engenharia de software. Pode ser diagramas de blocos, diagramas de classes, diagramas de componentes, etc. O importante é que os aspectos estáticos e dinâmicos do sistema estejam bem documentados.

3. Avaliação

Detalhar os testes que foram realizados, com ou sem usuários.

Ilustrar exemplos de execução com screenshots.

4. Discussão

Refletir sobre as contribuições do projeto para o problema escolhido e os usuários-alvo, comentar sobre as dificuldades de concretização da ideia, lições aprendidas durante o processo, etc.

5. Distribuição das contribuições

Detalhar as contribuições de cada integrante do grupo. Que atividades executou, quais módulos desenvolveu ou ajudou a desenvolver.

6. Link para o repositório do projeto.

Garanta que no repositório haja instruções e documentação de todos os pré-requisitos para execução da aplicação. Caso o professor não consiga testar o sistema, a nota ficará pendente até o grupo mostre que o aplicativo está funcionando satisfatoriamente.

- Vídeo de visão geral do sistema, de no máximo de 4 minutos, estruturando informações sobre os seguintes tópicos:
 - Motivação (por quê a aplicação é útil);
 - Definição do problema (qual problema a aplicação se propõe a resolver);
 - Solução (como as funcionalidades da aplicação solucionam o problema).

Ao fazer o vídeo, imagine que você está apresentando a ideia e o protótipo a algum investidor, com conhecimento técnico em tecnologias web, que irá financiar o desenvolvimento de uma versão final do sistema. Portanto, precisa ser objetivo, convincente, tecnicamente correto e sem enrolações.

- *Sobre avaliação do código-fonte:* só será avaliado o código-fonte resultante de commits até a data de entrega, commits posteriores serão ignorados.

Parte 5: Apresentação

Na data marcada, todos os integrantes deverão apresentar o projeto para toda a turma. Após cada apresentação, haverá arguição por todos os presentes (discentes e docente). A avaliação da apresentação considerará o desempenho do grupo e também o desempenho individual dos participantes.

O objetivo da apresentação é, principalmente, socializar os resultados dos projetos com a turma, assim todos ficam sabendo os que os outros grupos fizeram.

5 Critérios de avaliação

A nota final do projeto será obtida de acordo com a Equação 1.

$$N_{proj} = [(O + C + E + A + R) \cdot 0.75 + (P + G) \cdot 0.25] \cdot I + T \quad (1)$$

A equação inclui critérios gerais e critérios específicos. Entre os critérios gerais temos:

- **Originalidade (O):** criatividade na proposição do problema; emprego de recursos adicionais não vistos no curso (bônus);
- **Complexidade (C):** englobam-se nesse critério sistemas que resolvam problemas desafiadores e de solução não-trivial;
- **Eficiência (E):** a solução proposta deverá ter bom desempenho, o que pode englobar o tratamento adequado dos seguintes fatores:
 - complexidade dos algoritmos empregados;
 - política de uso da memória primária, principalmente no cliente, mas também no servidor;
 - política de uso da rede para comunicação cliente-servidor;
 - entre outros.
- **Acurácia (A):** a solução deverá atender adequadamente aos requisitos enunciados para a aplicação (e.g. deve-se atender completamente ao que foi proposto — caso não atenda, deve ser muito bem justificado);

Além dos critérios gerais, a avaliação considera também os seguintes critérios específicos:

- **Qualidade do relatório final (R):** boa redação, apresentação, conteúdo técnico e clareza do texto; qualidade dos materiais complementares solicitados (e.g. imagens, vídeos, etc.);
- **Estrutura e organização da apresentação (P):** material de apresentação (e.g. slides, ou outro) com boas organização e estrutura; apresentação oral clara e dentro do intervalo de tempo estipulado;
- **Desempenho na arguição (G):** conhecimento em abrangência sobre o projeto como um todo e em profundidade sobre módulos/partes que cada participante tenha contribuído em maior parte; conhecimento dos conceitos envolvidos e suas aplicações.
- **Fator de contribuição individual ($I \in [0, 1]$):** consiste no fator de participação de cada membro da equipe de desenvolvimento, a ser aferido pela atividade no repositório do projeto e na apresentação. Nesse intervalo, $I = 1$ se a participação for forte, e $I = 0$ se a participação for muito fraca ou nula. Contribuição nula no repositório, ou contribuição nula na apresentação, implica $I = 0$;

- **Pontualidade** (T): a nota máxima de cada fase do projeto só será obtida caso a entrega ocorra no prazo estipulado. Logo, $T < 0$ se entregue fora do prazo, cumulativamente para todas as fases.