

# Instruções para o projeto final

MCTA016 - Paradigmas de Programação – 2018-Q2  
Prof. Diogo S. Martins e Emilio Francesquini

v. 12/06/2018

## Resumo dos prazos

- Parte 0: 19/06
- Parte 1: 26/06
- Parte 2: 17/07
- Parte 3: 16/08
- Parte 4: 21/08

## 1 Escopo

O projeto final consiste em desenvolver um sistema utilizando a linguagem Racket. O sistema deverá utilizar primariamente o paradigma funcional. Outros paradigmas podem ser combinados à programação funcional, a exemplo da programação concorrente e da programação orientada a objetos. Porém, é desincentivado o uso da programação imperativa.

O tema do projeto é livre e será analisado pelos professores quanto à originalidade, relevância, complexidade, e outros fatores, conforme os critérios documentos na seção 5.

Note que o sistema deverá incluir uma *aplicação*<sup>1</sup>, portanto deverá apoiar alguma atividade e trazer benefícios a um usuário ou grupo de usuários. Conseqüentemente, deverá prover algum componente *front-end* (i.e. interface ou *shell*), que poderá ser via linha de comando, interface gráfica, ou outra modalidade, a depender do problema.

Mesmo que a contribuição principal do projeto consista em desenvolver algum componente de software *back-end*, por exemplo, um algoritmo, uma biblioteca, uma API, um serviço (de acesso local ou remoto), o projeto deverá incluir uma aplicação que demonstre, de modo mais completo possível, o uso desse componente de software.

## 2 Restrições

- A linguagem de implementação deverá ser Racket;
- É permitido usar bibliotecas Racket, porém não é permitido que:

---

<sup>1</sup>[https://en.wikipedia.org/wiki/Application\\_software](https://en.wikipedia.org/wiki/Application_software)

- a biblioteca, ou conjunto de bibliotecas, pelo simples reúso, já resolva o problema principal proposto no projeto;
- a biblioteca force seu programa a adotar um estilo fortemente imperativo (analise a documentação da biblioteca para decidir).

Devido a essas restrições, é importante que se escolha um problema de complexidade adequada, com volume de trabalho compatível com o prazo disponível e com a quantidade de participantes envolvidos, limitando o reúso de bibliotecas prontas apenas para apoiar o desenvolvimento do sistema.

### 3 Estrutura

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.

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.

O projeto deverá ser realizado em grupos de até 4 integrantes.

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.

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 grupo<sup>2</sup>. No repositório poderão ser mantidos, além do código-fonte, opcionalmente, outros artefatos solicitados, como documentos, imagens, etc. Caso não saiba usar ferramentas de controle de versão, consulte a documentação de algum dos sites sugeridos, ou entre em contato com os professores, para que possamos indicar material técnico complementar.

O acompanhamento do projeto pelo professor ocorrerá também pela atividade observada no repositório. Embora 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 membros do grupo deverão contribuir com tarefas de programação (por exemplo, concentrar a contribuição somente na redação nos documentos, 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. 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.

---

<sup>2</sup>Caso 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:

- 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 (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)?

Os professores irão analisar a proposta e enviarão 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 complementação, reportando como os problemas identificados na Parte 0 foram solucionados;
- Documento de proposta, em formato PDF. O documento deverá apresentar, com clareza, todos os seguintes aspectos principais do projeto:
  - **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 da aplicação? Apresentar diagrama.
    - \* 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?
  - Lista dos **integrantes** do grupo
  - Link para o **repositório** do projeto

## Parte 2: Acompanhamento

### Objetivos:

- Avaliar o andamento do desenvolvimento da aplicação

### O que entregar:

- Documento de complementação, reportando como os problemas identificados na Parte 1 foram solucionados;
- Relatório descrevendo todas as atividades que foram executadas (novas funcionalidades, problemas, etc.).

## Parte 3: Final

### Objetivos:

- Entrega do protótipo final da aplicação;
- Apresentação dos resultados obtidos.

### O que entregar:

- Documento de complementação, reportando como os problemas identificados na Parte 2 foram solucionados;
- Vídeo de visão geral do sistema, de no máximo de 8 minutos, ilustrando os seguintes tópicos:
  - motivação (por quê a aplicação é útil);
  - definição do problema (qual problema a aplicação se propõe a resolver);
  - demonstração (como as funcionalidades da aplicação solucionam o problema).

A parte de demonstração pode deverá ser um tutorial do sistema, mostrando as principais funcionalidades, e apresentando os testes e respectivos resultados.

- 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, de acordo com o modelo da SBC (<http://tinyurl.com/sbc-template-artigos>). Deverá conter pelo menos as seguintes seções:
  - Introdução
  - Definição do problema
  - Arquitetura e implementação
  - Avaliação (testes realizados, com ou sem usuários, com os testes de execução e screenshots);
  - Discussão (e.g. reflexões, dificuldades, lições aprendidas, etc.);
  - Detalhamento das contribuições de cada integrante para o projeto;
  - Link para o repositório do projeto.

## Parte 4: Apresentação

Na data marcada, todos os participantes do grupo deverão apresentar o projeto para toda a turma. Cada apresentação será seguida de arguição por todos os presentes (discentes e docentes). A avaliação da apresentação considerará o desempenho do grupo e também o desempenho individual dos participantes.

## 5 Critérios de avaliação

O projeto será avaliado quanto aos seguintes critérios gerais:

- **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;
  - uso da memória primária;
  - política de acesso à memória secundária;
  - política de uso dos recursos de rede;
  - 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);

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.);
- **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 ( $T$ ):** atrasos na entrega dos componentes de cada fase, ou atraso em uma fase como um todo, implicará em descontos na nota final. Obviamente, se nada for entregue, esse desconto irá zerar a nota.

Com base nesses critérios, a nota final do projeto será obtida de acordo com a Equação 1.

$$N_{proj} = \frac{O + C + E + A + R}{5} \cdot I - T \quad (1)$$