

INF2102 Projeto Final de Programação

Período: 2015/2

Coordenador: Prof. Arndt von Staa

Informações gerais

Objetivo da disciplina

É objetivo do projeto final de programação averiguar se o aluno sabe empregar técnicas eficazes para especificar, projetar, desenvolver, controlar a qualidade e documentar programas, que tenham um nível de complexidade pelo menos mediano. Os programas devem ser confiáveis, úteis e utilizáveis e devem possuir elevada qualidade de engenharia. Na medida do possível, o trabalho final de programação deve estar relacionado com a dissertação de mestrado ou tese de doutorado do aluno.

Natureza do trabalho

O **resultado** do projeto de programação pode ser *qualquer programa* que sirva para demonstrar a capacidade do aluno de *especificar, desenvolver e testar racionalmente* programas satisfazendo os objetivos acima.

O programa será **útil** na medida em que realize um serviço de interesse a alguém que não seja exclusivamente o aluno. A utilidade do programa será avaliada pelo *professor orientador* (ver mais adiante). Sugere-se que o trabalho tenha alguma relação com a dissertação de mestrado ou tese de doutorado a ser elaborada. São exemplos de aplicações:

- aplicações para engenharia
- componentes de compiladores
- ferramentas de apoio ao desenvolvimento de software
- frameworks
- interfaces gráficas animadas
- jogos eletrônicos
- produtos de software
- programas re-engenheirados
- sistemas “e-qualquer coisa”
- sistemas de supervisão e controle
- sistemas web
- sistemas big data
- sistemas vestíveis

A **complexidade** envolve *tamanho* e *natureza* do problema resolvido pelo programa. Cabe ao professor orientador (ver mais adiante) assegurar a satisfação deste requisito.

A **especificação** deve deixar claro qual é a finalidade do programa. A tecnologia usada ao especificar pode ser qualquer. A especificação mínima que deve estar escrita antes de se redigir o programa e deve responder as perguntas a seguir. Qual é a finalidade do programa? Quais são as interações entre os usuários (pessoas, sensores, atuadores, outros programas ou serviços) e o programa? A maneira mais simples (e fraca) para responder essa pergunta seria empregar *user stories*. Caso o programa se relacione com a tese ou dissertação, qual seria o título tentativo dessas? Como será feita a aceitação do programa ao final do desenvolvimento?

A **garantia da qualidade** depende da especificação, arquitetura, projeto, programação e teste empregados. A presença e forma de redigir esses artefatos dependem do paradigma usado pela solução (p.ex. programação lógica, procedural, OO, dirigida por tabela, etc.). A documentação entregue ao final do projeto deve conter os artefatos listados adaptados ao paradigma usado. Deve também deixar claro que foi empregado um *controle da qualidade* (teste, inspeção) sistemático. O ideal é ter sido utilizada alguma forma de teste automatizado (ex. JUnit, CPPUnit, LuaUnit, Python unit testing framework, ou similares).

A **linguagem de programação** utilizada pode ser qualquer uma (ex. Lua, Lisp, Modula, C, C++, Java, Prolog, Python, Ruby, etc.). Também pode ser qualquer uma a plataforma utilizada (ex. Windows, Mac, Unix, iOS, celulares, tablets, etc.) ou o framework usado. Sendo de interesse do aluno, o trabalho pode ser desenvolvido utilizando alguma ferramenta CASE.

Como iniciar

Antes de mais nada, encontre um *professor orientador* do projeto de programação. Este é **necessariamente** um professor do Departamento de Informática da PUC-Rio. Não precisa ser o coordenador da disciplina. Sugere-se que seja o orientador da tese ou dissertação.

Junto com o professor orientador defina o objetivo do trabalho. O ideal é que isto seja feito antes do início do semestre e que se relacione com a dissertação ou tese.

Envie ao professor coordenador da disciplina, prof. Arndt von Staa, até o dia **21/setembro/2015**, uma mensagem, contendo:

- o número de matrícula e o nome do aluno
- o nome do professor orientador
- o nome do projeto
- um resumo curto (um parágrafo) descrevendo o objetivo do projeto

O título da mensagem deve ser INF2102-Proposta-**NomeAluno**

- obviamente a parte **NomeAluno** deve ser substituída pelo respectivo nome, **sem** utilizar caracteres em branco ou diacríticos (acentos, cedilha).

Como terminar

Entregue ao coordenador da disciplina, até o dia **09/dezembro/2015**:

1. uma folha contendo (ver apêndice):
 - o número de matrícula e o nome do aluno
 - o nome do professor orientador
 - a **avaliação do trabalho** feita pelo professor orientador (tipicamente um parágrafo de texto). Esta avaliação deve dizer se:
 - o programa satisfaz os requisitos de complexidade assumidos para projetos finais de programação
 - o programa é confiável e utilizável
 - o programa atende aos objetivos funcionais traçados
 - se os testes foram abrangentes e satisfatórios
 - eventuais comentários a respeito do trabalho
 - assinatura do professor orientador
2. a documentação final do projeto. Recomenda-se a entrega de tudo em um CD ou DVD, **neste caso não será necessário entregar qualquer documentação impressa**. Os arquivos contidos no CD / DVD **não** devem estar compactados (zip, tar, etc.). Não se esqueça de identificar o autor no CD / DVD. Tampouco se esqueça de anexar a folha de avaliação preenchida pelo orientador. A documentação deve conter:

- especificação do programa
 - objetivos, requisitos
 - especificação, por exemplo *use-cases*.
- projeto do programa
 - arquitetura
 - critérios de projeto utilizados
 - diagramas de arquitetura e/ou segmentação do programa, por exemplo UML.
 - organização do programa (componentes, módulos, classes,...), por exemplo diagramas de classe UML.
 - diagramas de organização dos dados, por exemplo diagramas de modelagem de dados, ou entidade e relacionamentos.
- código fonte **cuidadosamente comentado**
 - comentário inicial de cada módulo, identificando o autor
 - comentários cabeçalho de módulos, classes e funções
 - assertivas para dados e procedimentos, procure utilizar *design by contract*
 - pseudo instruções
 - procure estabelecer e/ou adotar padrões de programação. Os apêndices do livro Staa, A.v.; *Programação Modular*, Campus 2000; contêm uma extensa lista de padrões de programação que pode servir de exemplo para a adaptação às características específicas do trabalho.
- roteiro de teste efetuado, composto de:
 - critérios de teste utilizados
 - descrição dos casos de teste
 - na medida do possível procure utilizar testes automatizados
 - scripts de teste automatizado
 - logs gerados pelo teste automatizado
- documentação para o usuário

O formato da documentação técnica varia em função do domínio do problema e da linguagem de programação utilizada. Não obstante, o programa deverá ser especificado e projetado antes de ser implementado. Ou seja, a especificação, a organização e a composição do programa deverão estar definidas antes de iniciar a codificação. Da mesma forma os testes dependem da linguagem e do ambiente de execução. Apesar disto, deve ser projetado um conjunto de experimentos sistemáticos que sirvam para atestar que o programa é confiável e atinge os objetivos especificados.

Critério de avaliação

Não serão aceitos trabalhos sem a folha de avaliação produzida pelo professor orientador.

As notas máximas serão:

- 10 trabalho satisfatório e entregue no prazo, ou seja até, no máximo, **09/dezembro/2015**
- 8 trabalho satisfatório e entregue após a matrícula do semestre subsequente e antes da data limite imposta pela PUC: **18/janeiro/2016**. Esta data está em sintonia com a data limite estabelecida pela PUC-Rio para preenchimento de graus IN do semestre anterior. Já prevê uma folga para a correção dos inúmeros trabalhos.
- zero se não for entregue até no máximo dia: **18/janeiro/2016**, ou se o CD/DVD contiver um vírus.

Atentem para a regra da CCPG: O aluno tem **60 dias corridos a partir do final do semestre** (Término das atividades acadêmicas do semestre) para completar o incompleto. Como são necessárias as avaliações dos professores orientador e coordenador, a data limite 18/janeiro/2016 é estabelecida de modo que se tenha folga para as avaliações necessárias.

Se você acha que não vai conseguir terminar no prazo, cancele a disciplina. Não se matricule em *Projeto Final de Programação* sem saber quem será o orientador e qual será o trabalho!

Trabalhos que não forem entregues até **09/dezembro/2015**, recebem **automaticamente** IN (incompleto). Trabalhos que não forem entregues até a **18/janeiro/2016** recebem automaticamente zero, com todas as implicações que isto possa ter, como, por exemplo, jubramento por falta de média no período. Note que, se for o caso, o jubramento ocorrerá no semestre subsequente.

Causam a perda de pontos:

- entrega atrasada. Para evitar a perda de pontos só se matricule depois que souber quem é o orientador, qual é o trabalho, e se você tem condições de terminar o trabalho no semestre. Afinal saber planejar faz parte da vida profissional do Informata... Se durante o semestre você perceber que não conseguirá terminar, cancele a disciplina. Não se esqueça que o calendário da PUC determina um prazo para o cancelamento de disciplinas.
- inexistência de avaliação feita pelo orientador, **o trabalho não será aceito sem a avaliação do professor.**
- inexistência de especificação.
- inexistência de projeto.
- inexistência de um roteiro de teste para fins de teste sistemático. Deve estar explicitado o método de teste utilizado. Dê preferência a testes automatizados.
- inexistência de evidência de que foram realizados testes segundo o roteiro. Procure produzir um laudo de teste. O teste automatizado deve produzir arquivos log.
- inexistência de documentação para o usuário.
- código mal organizado e/ou mal comentado.



PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO DE JANEIRO
DEPARTAMENTO DE INFORMÁTICA
INF 2102 - PROJETO FINAL DE PROGRAMAÇÃO

FOLHA DE AVALIAÇÃO

NOME DO ALUNO(A): _____

Nº MATRÍCULA: _____

NOME DO ORIENTADOR: _____

TÍTULO DO PROJETO: _____

PALAVRAS-CHAVE: _____

AVALIAÇÃO DO TRABALHO

- | | | |
|------------------------------|------------------------------|--|
| <input type="checkbox"/> Sim | <input type="checkbox"/> Não | Este trabalho satisfaz os requisitos de complexidade assumidos para projetos finais de programação |
| <input type="checkbox"/> Sim | <input type="checkbox"/> Não | O programa é confiável e utilizável |
| <input type="checkbox"/> Sim | <input type="checkbox"/> Não | O programa atende aos objetivos funcionais traçados |
| <input type="checkbox"/> Sim | <input type="checkbox"/> Não | Os testes foram abrangentes e satisfatórios |

OBSERVAÇÕES

NOTA SUGERIDA:

Rio de Janeiro,

Professor Orientador