

MC437–Projeto de Sistemas de Informação
Instituto de Computação
UNICAMP

Prof. Luiz Eduardo Buzato

Primeiro Semestre de 2015

Descrição da Disciplina

Horário das aulas

Turma	Dia	Horário	Sala
A,B	4 ^a	14–16h	316 (IC3.5)
	6 ^a	14–16h	302/303 (IC3.5)

Monitor

- Nome: Elias Ximenes
- Email: expn.pi@gmail.com

Ementa (DAC) Análise e projeto de Sistemas de Informação usando técnicas de Bancos de Dados, Interface Humano-Computador e Engenharia de Software.

OF:S-5 T:00 P:02 L:02 O:00 D:00 E:00 HS:04 SL:04 C:04 EX:N

Pré-Requisitos: MC426 MC536 / MC426 MC750

Ementa Alternativa

Nesta disciplina você realizará um projeto prático de desenvolvimento de software, de acordo com um processo ágil simples, com o objetivo principal de desenvolver as habilidades adquiridas até o momento em ciência da computação, com foco em engenharia de software, medidas de desempenho e testes. Você trabalhará como parte de um time de

alunos que será responsável por definir seu próprio roteiro, procurando responder perguntas que considera relevantes, interessantes. Eu ajudarei fornecendo guias, tarefas e conversas para auxiliá-lo a alcançar os resultados desejados. A disciplina deve mantê-lo motivado, engajado. Se não se vir motivado, venha conversar comigo e tentaremos encontrar novos aspectos motivadores para auxiliá-lo a trabalhar bem individualmente e em grupo.

Preparem-se para uma experiência de desenvolvimento de software em que vocês irão, com a minha ajuda, identificar as problemas no TPC-W e, em seguida, projetar, implementar, executar e avaliar o desenvolvimento da solução que propuserem para o problema, produzindo assim uma nova e melhor versão do TPC-W (Transaction Processing Council-Web Benchmark). A ênfase sempre será em uma solução mais robusta (tolerante a falhas) e de melhor desempenho.

Objetivos de Aprendizado Esta disciplina deve ajudá-lo no progresso em direção aos seguintes objetivos:

- Ser capaz de analisar o desempenho de sistemas computacionais, definindo critérios e encontrando a melhor alternativa;
- Compreender como funciona uma aplicação cliente-servidor típica, com persistência de dados realizada em banco de dados relacional;
- Compreender como os componentes de uma aplicação cliente-servidor se relacionam, propiciando a compreensão detalhada da arquitetura de software da aplicação;
- Entender e programar uma quantidade razoável de código em Java;
- Entender e utilizar várias ferramentas comumente adotadas na implementação de processos ágeis de software: controle de versão (git/hg), integração contínua (jenkins), ferramentas de avaliação de qualidade de software (sonarqube), ferramentas de automação de atividades de configuração e programação (ant, maven, make, etc);
- Trabalhar efetivamente como parte de um time;
- Utilizar comunicação escrita e oral para descrever métodos (procedimentos), resultados e conclusões;
- Demonstrar capacidade de aprendizado auto-suficiente e autônomo (auto-guiado), individual e em grupo, inclusive escolhendo for-

mas de aprendizado, gerenciamento de tempo e esforço, reflexão e auto-avaliação.

Programa

1. Especificação e explicação de um método ágil baseado em scrum. As ferramentas de apoio utilizadas para a execução do processo ágil de engenharia de software e da programação serão:
 - linguagem de programação: java;
 - interface humano-computador: HTML, javascript;
 - sistemas de gerência de banco de dados: postgresql <http://www.postgresql.org/> ou mysql www.mysql.com;
 - Programação das atividades (tarefas): kanbanflow <https://kanbanflow.com/>;
 - Integração contínua: jenkins <http://jenkins-ci.org/>;
 - Gerência de código (versionamento): git <http://git-scm.com/>;
 - Qualidade de software: sonarqube <http://www.sonarqube.org/>.
2. Desenvolvimento, utilizando o processo ágil, de uma versão aprimorada do TPC-W <http://www.tpc.org/tpcw/>, de acordo com os requisitos sugeridos pelo docente durante o semestre.

Bibliografia

1. Processos de Engenharia de Software Ágeis
 - K. Beck, eXtreme Programming: Explained, Addison-Wesley, 2000.
 - K. Beck, Test-Driven Development: By Example, Addison-Wesley, 2002.
 - K. Beck and C. Andrés, Extreme Programming Explained: Embrace Change, 2nd Edition, Addison-Wesley Professional, 2004.
 - K. Beck and M. Fowler, Planning Extreme Programming, Addison-Wesley, 2000.
 - A. Cockburn, Agile Software Development, Addison-Wesley Longman, 2002.

- M. Cohn, *Succeeding with Agile: Software Development Using Scrum*, Addison-Wesley, 2009.
 - E. Derby and D. Larsen, *Agile Retrospectives Making Good Teams Great*, Pragmatic Bookshelf, 2006.
 - M. Fowler, *Refactoring: Improving the Design of Existing Code*, Addison-Wesley, 2000.
 - E. Gamma, R. Helm, R. Johnson, and J. Vlissides, *Design Patterns: Elements of Reusable Object-Oriented Software*, Addison-Wesley, 1995.
 - B.W. Kernighan and R. Pike, *The Practice of Programming*, Addison-Wesley, 1998.
2. TPC-W: documentos disponíveis em <http://www.tpc.org/tpcw/>. Separatas fornecidas pelo docente.

Dinâmica do Curso

O curso será composto de: (i) uma semana de aula teórica que cobrirá os fundamentos mínimos sobre métodos ágeis. Quatorze semanas de aulas práticas e *mentoring* para evolução do TPC-W. O código fonte de uma primeira versão do TPC-W, completamente operacional, será fornecido aos alunos.

Tempo de Dedicção à Disciplina Esta é uma disciplina de 4 créditos, então você deve se dedicar, em média, 8 horas por semana realizando todas as atividades associadas ao seu aprendizado. Contacte-me se estiver gastando muito menos ou muito mais tempo que o previsto para procurarmos formas de ajuste do trabalho.

Presença A presença às aulas e a participação nas atividades do time é parte fundamental para o desenvolvimento da disciplina.

Conduta na Disciplina Sua conduta nesta disciplina será considerada como um reflexo de sua atitude e conduta na vida real. Em especial, é esperado que cada um dos componentes do time contribua de forma significativa para o resultado do(s) projeto(s). Sempre que um membro do time não estiver em condições de contribuir com o time, ele ou algum outro membro do grupo deve me comunicar a dificuldade de contribuição imediatamente.

Atendimento aos Alunos O atendimento aos alunos acontecerá preferencialmente às quartas-feiras, das 16:00h às 16:30h no laboratório/sala de aula.

Avaliação

Serão avaliados através de uma média ponderada das notas—entre 0.0 e 10.0—atribuídas aos seguintes aspectos da atividade do aluno no curso:

- frequência às aulas (f);
- empenho (adesão ao método, empenho individual, empenho no grupo) (e);
- qualidade do software desenvolvido (sonar e avaliação pelo docente) (q).

O aproveitamento final do aluno (AF) será calculado da seguinte maneira:

$$AF = (50q + 25e + 25f)/100$$

Se $AF \geq 5.0 \wedge f \geq 75\%$ então o aluno se aprovou. Caso contrário, o aluno se reprovou.

Observações importantes:

- **Este curso não tem exame.**
- Qualquer tentativa de fraude durante o desenvolvimento do projeto implicará em **zero na disciplina** para todos os envolvidos, sem prejuízo de outras sanções.

Material Didático

Todo o material didático considerado relevante para os alunos, incluindo as listas de exercícios, estará disponível no ambiente Moodle do Grupo Gestor de Tecnologias Educacionais <http://www.ggte.unicamp.br/moodle/> e, portanto, não será distribuído de forma impressa em sala de aula.