

MC437 – Projeto de Sistemas de Informação
2º Semestre/2015
PROGRAMA

Profª: Eliane Martins
Horários da disciplina: 6ª das 19-23h

- Local:
 - Teóricas: CC53
 - Lab: CC02/CC03

Onde me achar:
eliane_at_ic.unicamp.br
IC1 - sala 36

Monitores: Luiz Virginio (PED)
luiz.virginio.jr_at_gmail.com

Horário de atendimento:
Enviar email para marcar horário

Apresentação do curso

EMENTA

- Análise e Projeto de Sistemas de Informação usando técnicas de Bancos de Dados, Interface Humano-Computador e Engenharia de Software

PROGRAMA

- Levantamento de requisitos funcionais e não funcionais
- Projeto arquitetural
- Desenvolvimento e testes dos componentes do sistema **usando método ágil**
- Integração e testes dos componentes do sistema

Atividades do Curso

O objetivo da disciplina é aplicar os conceitos e boas práticas de Engenharia de Software aprendidos na disciplina teórica. Durante a disciplina, os alunos realizarão um projeto real, usando métodos ágeis, de forma que as boas práticas da Engenharia de Software sejam aplicadas sem perda de agilidade no processo de desenvolvimento. O projeto é realizado em grupo, e o número de membros de cada grupo será definido de acordo com o tamanho da turma. Os grupos serão criados pelos formadores, de acordo com as respostas dadas ao **Questionário para divisão dos grupos**. O objetivo é ter grupos uniformes, para que todos tenham as mesmas chances de realizar o projeto com sucesso.

O desenvolvimento é incremental, dividido em *sprints*. Serão dois *sprints* no total. A cada *sprint*, diversas iterações são realizadas para desenvolver o incremento. A cada aula a equipe desenvolverá um incremento do sistema, o qual deve ser colocado em um repositório do grupo. Uma aula = período de 2 horas. Cada aula será considerada, portanto, como uma iteração de um *sprint*.

SÓ SERÁ PERMITIDO DESENVOLVER CÓDIGO DURANTE A AULA. A equipe não pode trazer código pronto, nem inseri-lo no repositório em horário fora da aula.

Entregas

A cada iteração, diversos artefatos serão produzidos, e deverão ser entregues. Os entregáveis serão indicados no decorrer da disciplina.

Todas as entregas deverão ser depositadas no repositório do projeto ao final de cada aula. O repositório utilizado será indicado no decorrer da disciplina.

Haverá ainda um questionário de avaliação dos pares, a ser preenchido individualmente, ao final de cada *sprint*.

→ Atenção: o não preenchimento do questionário acarretará em um ônus na sua nota individual.

Ao final do projeto, cada grupo cada grupo deverá produzir um pacote do produto final (PAC) a ser entregue ao cliente.

As entregas deverão ser depositadas nas datas estipuladas. Caso contrário, haverá **10%** de desconto por dia de atraso.

- Arquivos com formato inválido, com problemas de leitura ou sem acesso pelos formadores serão considerados como não entregues e sujeitos a descontos por atraso.

Critério de Avaliação

As notas de cada aluno constarão das notas de exercícios e notas de projeto. Os exercícios são individuais incluem o questionário para divisão de grupos e exercícios para uso de tecnologias. As notas de exercícios (NE) valem 10% de peso, enquanto a nota de projeto (NP) vale 90% de peso.

As notas de projeto, conterão duas partes: notas individuais (I) e notas do grupo (G). As notas individuais serão dadas por *sprint*, e serão compostas por:

- Nota_Audio_diário (a cada aula)
- Nota_Avaliação_Pelos_Pares (ao final do *sprint*)
 - a não entrega do questionário de avaliação por pares ⇒ -1 ponto na nota individual (I)

As notas de cada grupo (G) serão dadas de acordo com os entregáveis por *sprint*, definidos durante a disciplina.

Portanto, ao final de um *sprint*, a nota de cada aluno é dada por:

$$NS = (2 \cdot I + G) / 3 \quad \text{Se } I \geq 5 \text{ e } G \geq 5, \text{ senão, vale a menor. (Eq1)}$$

Cálculo da média:

A média **M** do aluno será calculada como:
 $M = 10\%*NE + 90\%*NP$

em que NP, média do projeto, é dada por:
 $NP = 40\%*NS1 + 40\%*NS2 + 20\%*PAC$
em que

NSi corresponde à nota do sprint, calculada conforme Eq1 e PAC é a nota do empacotamento final do produto.

Observações:

1. **Não** serão dados provas ou trabalhos substitutivos.
2. A codificação dos *sprints* será efetuada apenas no horário da aula, por isso é importante sua presença.
3. Por se tratar de disciplina prática, não tem exame previsto.

Frequência às aulas

- O limite de faltas é de 25% do total das aulas previstas. O aluno que tiver menos do que 75% de presença é **reprovado por falta**.
- **Única Exceção:** O aluno com problemas de saúde que impeçam o seu comparecimento às aulas por períodos mais longos, devem entrar com um requerimento de "regime especial" junto à Diretoria Acadêmica (DAC).
- A presença deve ser documentada com uma assinatura em uma lista de chamada para posterior contabilização. **A assinatura nas listas de chamada deve ser a mesma utilizada em documentos oficiais** junto à Diretoria Acadêmica (DAC). **Não são aceitáveis iniciais apenas ou rubricas.**

Conduta Ética

- Os trabalhos deverão ser de autoria do aluno ou do grupo UNICAMENTE. Discussões e troca de idéias com colegas, professor ou assistente são saudáveis e bem vindas, mas a solução final deve ser exclusivamente do autor, ou dos autores, quando for trabalho em grupo.
- Consultas a fontes externas (Web, artigos e livros) são válidas e altamente recomendadas desde que explicitamente referenciados no trabalho.
- Qualquer outro tipo de conduta será considerado como plágio, e implicará em pontuação **ZERO** para todos os envolvidos.

Referências

- Orit Hazzan and Yael Dubinsky. Agile Software Engineering (Undergraduate Topics in Computer Science). Springer London; 1 edition (February 28, 2009)
- Ken Schwaber and Mike Beedle. Agile Software Development with Scrum. Prentice Hall, Upper Saddle River, New Jersey, 2002.
- Mary Poppendieck and Tom Poppendieck. Lean Software Development: An Agile Toolkit. Addison-Wesley, 2003.

- Mike Cohn. User Stories Applied: For Agile Software Development. Addison-Wesley, 2004.
- John F. Smart. BDD In Action. Manning, 2013.
- Caelum Ensino e Inovação. Laboratório de MVC com Hibernate e JSF para Web, 2007.

Recursos na Internet

(Esta lista poderá ser atualizada ao longo do curso. Ela está longe de ser exaustiva)

- Sobre UML: www.uml.org
- Sobre arquitetura de software: <http://www.serc.nl/people/florijn/interests/arch.html>
- Padrões e Anti-padrões: http://www.dmoz.org/Computers/Programming/Methodologies/Patterns_and_Anti-Patterns/
- Agile Manifesto – <http://agilemanifesto.org/iso/ptbr/>
- Scrum Guide - versão de 2011 em português - [http://www.scrum.org/storage/Scrum Guide 2011 - PTBR.pdf](http://www.scrum.org/storage/Scrum%20Guide%202011-PTBR.pdf)
- Scrum Master in Under 10 Minutes video <http://www.axosoft.com/ontime/videos/scrum/#scrum-diagram>
- Scott Ambler. Introduction to TDD: <http://www.agiledata.org/essays/tdd.html>
- Ryan Greenhall. BDD by Examples: <http://www.ryangreenhall.com/articles/bdd-by-example.html>
- John Smart tem vários tutoriais: http://pt.slideshare.net/search/slideshow?searchfrom=header&q=john+smart%2C+bdd&ud=any&ft=all&lang=*&sort= Em especial:
 - sobre TDD: http://pt.slideshare.net/skillsmatter/real-developersdontneedunittests?qid=5a746836-4918-43ef-b7cc-09881e2160d4&v=default&b=&from_search=12
 - BDD para testes Web: http://pt.slideshare.net/wakaleo/bdd-atdd-page-objects-the-road-to-sustainable-web-testing?qid=5a746836-4918-43ef-b7cc-09881e2160d4&v=default&b=&from_search=8
- Manual do Thucydides: <http://thucydides.info/docs/thucydides-one-page/thucydides.html>
- Sobre Selenium Web Driver: http://docs.seleniumhq.org/docs/03_webdriver.jsp
- Sobre Java Web:
 - Java para Desenvolvimento Web: <http://www.caelum.com.br/apostila-java-web/>
 - Laboratório java com testes, JSF e design patterns: <http://www.caelum.com.br/apostila-java-testes-jsf-web-services-design-patterns/>

Datas importantes

- Serão disponibilizadas ao longo da disciplina.