

# Instituto de Computação – Unicamp

## MC426 Engenharia de Software

OF:S-5 T:002 P:002 L:000 O:000 D:000 HS:004 SL:004 C:004 AV:N EX:S FM:75%

**Pré-Req.:** MC302

**Ementa:** Paradigmas da Engenharia de Software. Levantamento de Dados. Técnicas e ferramentas de especificação dos requisitos do usuário e do sistema. Métodos de análise e projeto de sistemas de informação. Implementação de sistemas de informação.

---

1. semestre 2017

**Profa. Responsável:** Ariadne Maria Brito Rizzoni Carvalho

**PED voluntário:** José Valderlei da Silva (Jesus) [j163150@g.unicamp.br](mailto:j163150@g.unicamp.br)

### Programa:

Esta disciplina objetiva introduzir o aluno à Engenharia de Software apresentando conceitos e práticas necessários ao desenvolvimento de sistemas computacionais. O conteúdo envolve os seguintes tópicos, sem estar restrito a estes: Visão geral da área; Processos de Software; Modelos de Sistemas; Gerência de Projetos de Software; Engenharia de Requisitos de Software; Processos de Engenharia de Requisitos; Especificação de Requisitos; Projeto e Desenvolvimento Iterativos; Introdução à Análise e Projeto Orientados a Objeto; Processo Unificado Ágil.

**Detalhamento do Programa e Datas importantes:** ver Agenda da Disciplina (em Anexo)

### Método:

O programa será trabalhado de forma orientada ao desenvolvimento concomitante de um Sistema de Informação, que envolve a realização de atividades práticas em sala de aula e em laboratório. Portanto, teoria e prática serão articuladas ao longo do desenvolvimento da disciplina. Utilizaremos metodologia JiTT (Just in Time Teaching/Learning)<sup>1</sup>.

Atividades em grupo [máx. 4 alunos] envolverão diversas práticas que conduzirão ao desenvolvimento progressivo de um projeto de software, utilizando os artefatos metodológicos e ferramentas propostos. O trabalho em grupo é importante para a disciplina e será avaliado também pelos pares ao longo do semestre. Atividades individuais envolverão leitura e síntese de bibliografia indicada, realização de provas agendadas ao longo do semestre, apresentação e discussão de tarefas relativas ao grupo, participação em atividades de “aquecimento” com frequência semanal, e discussões no espaço da disciplina e em atividades de aulas. Todos os trabalhos produzidos individualmente e em grupo deverão ser postados na plataforma utilizada pela disciplina.

### Avaliação:

A nota final (NF) será calculada a partir da nota dos trabalhos em grupos (NG) e das notas individuais (NI). A nota dos trabalhos em grupo será dada pelos resultados apresentados para os entregáveis do projeto e outras tarefas em grupo (TG), mais um delta que poderá variar de -1 a 1 conforme a avaliação por pares de cada grupo ( $\Delta APG$ ). Já a nota individual será dada pela média aritmética simples das notas das provas (NP), mais um delta que poderá variar de -1 a 1 conforme as notas de Participação Presencial e Virtual (PPV). A PPV é determinada pela participação individual, tanto durante as aulas quanto por meio das atividades individuais postadas na plataforma da disciplina (ex. aquecimentos). As médias são calculadas pelas seguintes fórmulas:

---

<sup>1</sup>[http://en.wikipedia.org/wiki/Just\\_in\\_Time\\_Teaching](http://en.wikipedia.org/wiki/Just_in_Time_Teaching)

$$NG = TG + \Delta APG \text{ e } NI = NP + \Delta PPV$$

$$NF = \begin{cases} (NG + NI)/2, & \text{se } NG \geq 5 \text{ e } NI \geq 5 \\ \min\{NG, NI\}, & \text{se } NG < 5 \text{ ou } NI < 5 \end{cases}$$

Haverá Exame para aqueles com  $NF < 5$  e 75% de frequência às aulas é uma condição necessária para a aprovação. Após o exame,  $NFF = (NF + \text{Nota Exame}) / 2$ .

**Atendimento:**

Deverá ser usada a plataforma Google Classroom para agendar atendimento individual com o PED ou a professora.

**Datas de Provas e Exame:**

**Provas: 20 de abril de 2017 e 06 de junho de 2017**

**Exame: 11 de julho de 2017**

**Bibliografia Principal:**

Sommerville, Ian (2012). Engenharia de Software 9a. Edição. Pearson/Addison Wesley.

Larman, Craig (2004). Applying UML and Patterns An Introduction to Object-Oriented Analysis and Design and the Unified Process. 3rd. Edition. Prentice-Hall, Inc.(2015) Portuguese Edition.

Chiossi, T., Carvalho, A.. Introdução à Engenharia de Software, Editora da UNICAMP, 2001.

Ambler, S.W. The Agile Unified Process, <http://www.ambysoft.com/unifiedprocess/agileUP.html> (em 05/02/2016)

Outras referências poderão ser indicadas no decorrer do curso.

# Instituto de Computação – Unicamp

MC426 Engenharia de Software  
1. semestre 2017

Profa. Responsável: Ariadne Maria Brito Rizzoni Carvalho

PED voluntário : José Valderlei da Silva (Jesus) [j163150@g.unicamp.br](mailto:j163150@g.unicamp.br)

## AGENDA PRELIMINAR DA DISCIPLINA MC426

[sujeita a alterações ao longo do período]

### MARÇO/2016

- 02/03 Apresentação e Administração da disciplina
- 07/03 Introdução - Iniciando em Engenharia de Software - FAQs [\[ch01CecIntro\] 24 slides](#)  
Processos e Modelos de Processos de Software [\[ch03Cec\] 39 slides](#)
- 09/03 Processos Ágeis [Ch03-IS] - (Em ajustes)
- 14/03 Lab. Apresentação do Projeto;  
Prática Participativa com sistema DSC para Clarificação do Problema
- 16/03 Análise e Projeto OO - Processo Unificado [\[Lar123\] 35 slides](#)
- 21/03 Agile UP Leituras Discussão Proposta
- 23/03 Modelos de Sistemas de Software [\[ch07Cec\] 41 slides](#)
- 28/03 Gerência de Projetos [\[ch04Cec\] 34 slides](#)
- 30/03 Entregável 1 – a. Apresentação dos Grupos para Clarificação do Problema  
[b. AUP\\* AUP customizado pelo grupo para o Projeto](#)

### ABRIL/2016

- 04/04 Escrevendo Requisitos – Modelo de Casos de Uso (a) [\[Lar456\] 36 slides](#)  
Outros Requisitos + Exemplo [\[Lar78\] 28 slides](#)
- 06/04 Lab. [com o KIT de desenvolvimento](#)
- 11/04 Diagrama de Sequência de Sistema (SSD) – Modelo de Casos de Uso (b) [\[Lar9\] 19 slides](#)  
+ Contrato de Operação [\[Lar13\] 13 slides](#)
- 13/04 **NÃO HAVERÁ AULA**
- 18/04 Revisão de conteúdo para prova, com questões do QUIZ
- 20/04 **Prova 1**

- 25/04 Entregável 2 - a. Apresentação dos Grupos para Modelo de Requisitos do sistema  
b. Arquitetura, requisitos priorizados, plano de entregas de acordo com AUP\*
- 27/04 Prototipação de IU ( inclui telas de interação e maquete do ambiente)

## MAIO/2016

- 02/05 Modelo de Classes de Domínio [\[Lar10\] 31 slides](#) Associação e Atributos [\[Lar1112\]34 slides](#)
- 04/05 Lab - Implementação de funcionalidade de hardware do kit, customizada para cada grupo
- 09/05 Design/Projeto [\[Lar1415\] 36 slides](#) e Padrões de Design de Software - GRASP [\[Lar16\] 46 slides](#)
- 11/05 Entregável 3 - a) prototipação de IU (digital); b) prototipação do espaço físico (maquete); c) arquitetura de componentes associando protótipo digital e físico.
- 16/05 Padrões [\[Lar17\]42 slides](#) e Modelo de Design [\[Lar1819\] 42 slides](#)
- 18/05 Lab Implementação Release 1 do sistema
- 23/05 Teste de Software [\[Somm 23\] 55 slides](#)
- 25/05 Entregável 4 - Resultado da iteração 1- Release 1 do projeto

## MAIO/JUNHO/2016

- 30/05 Teste de Software [\[Somm 23\] 55 slides](#)
- 01/06 Revisão de conteúdo para prova, com questões do QUIZ e Lab Iteração 2
- 06/06 **Prova 2**
- 08/06 Lab Iteração 2 - Implementação e teste
- 13/06 Teste de Aceitação (black box) do Release 2 (por pares)
- 15/06 **NÃO HAVERÁ AULA**
- 20/06 Lab Iteração 3 - Implementação e teste
- 22/06 Apresentação final dos projetos - Iteração 3
- 27/06
- 29/06

## JULHO/2016

- 02/07 Reposição eventual
- 03 a 08/07 Semana de Estudos
- 11/07 EXAME DE MC426

18/07 Prazo final para entrada de Médias e Freqüências do 1º período letivo de 2017

**Obs:** As atividades participativas em sala e/ou em laboratório fazem parte do desenvolvimento do Projeto.  
As apresentações dos Grupos devem ter a presença de todos os componentes do referido Grupo.

## Sistema de Software para Internet das Coisas (SSIC)

Nesta disciplina, além de estudarmos as principais técnicas da Engenharia de Software para construir software vamos aplicá-las em um projeto real que possa ajudar as pessoas.

O software escolhido deverá contemplar o conceito de **Internet das Coisas**<sup>1</sup>. Basicamente a ideia é pensar em objetos que possamos conectar de alguma forma (fio, sem fio, sensores, interação direta, etc) a uma rede de computadores, usando protocolo web, de forma que estes objetos façam parte de um sistema em que pessoas possam controlá-lo ou mesmo permitir que o sistema assuma o controle de atividades que são executadas.

O que se percebe é que novos dispositivos, melhoria na comunicação e a miniaturização dos componentes de sistemas computacionais são elementos para a criação de softwares “inteligentes” para ajudar o homem em suas atividades diárias.

Assim, todas as equipes trabalharão em uma arquitetura de hardware montada (processadores e sensores) no qual farão o deploy do seu projeto e testes. Os requisitos mínimos serão anunciados durante as aulas.

Começaremos com a concepção de um projeto pelas equipes. Para nivelar a avaliação, todos deverão trabalhar com uma base comum de requisitos iniciais.

Para o projeto, utilizaremos uma arquitetura pronta com sensores e atuadores para serem controlados (lâmpadas, abertura de portões, etc.). Todas as equipes podem inovar, trazendo elementos a mais para o seu projeto (sem alterar a arquitetura de hardware), como por exemplo dispositivos vestíveis, integrar celulares e outros dispositivos móveis ao projeto. Deixaremos livre a escolha de linguagens e outras tecnologias para o projeto.

Sugestão: Converse com pessoas de seu convívio diário, seus pares, ou com sua família, sobre objetos que poderiam ser colocados numa rede de sensores e fazer parte da Internet das Coisas e que funções poderiam ser automatizadas e ou ficar sob responsabilidade do sistema. Anote em seu projeto estas observações.

### 1. Requisitos iniciais comuns a todas as equipes

Estes requisitos estão em alto nível e cada equipe deve adequá-los à sua visão do projeto.

Configuração mínima:

Cadastro das pessoas que convivem na mesma casa;

Cadastro de objetos;

Cadastro de permissões de acesso a objetos;

Permitir o compartilhamento de objetos com vizinhos/parentes/etc. (Pensar em quais e porque).

Ações mínimas:

Acionar objetos via o sistema.

Receber informação de objetos do sistema.

---

1 IoT: Internet of Things, WoT: Web of Things

Gerenciamento:

- O compartilhamento de código e controle de versão deve ser feito no github (<https://github.com/>). As equipes devem adicionar o usuário jvander ([vander.vander@gamil.com](mailto:vander.vander@gamil.com)) como membro do time.

Etapas do Projeto e Entregáveis seguem Agenda da Disciplina