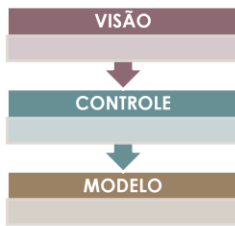


Arquitetura e Desenho de Software

AULA 01A



Profa. Milene Serrano



Agenda

Considerações Iniciais

Plano de Ensino

- Ementa
- Objetivos
- Conteúdo Programático
- Método de Ensino
- Avaliação
- Referências

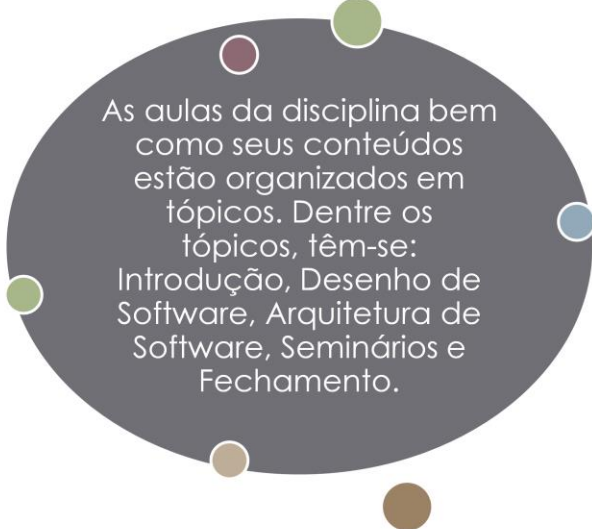
Considerações Finais



Considerações Iniciais



Considerações Iniciais



As aulas da disciplina bem como seus conteúdos estão organizados em tópicos. Dentre os tópicos, têm-se: Introdução, Desenho de Software, Arquitetura de Software, Seminários e Fechamento.

Plano de Ensino



Plano de Ensino: Módulo Inicial

Tópico 1: Introdução

Plano de Ensino

Introdução à Disciplina

Revisão de Requisitos

Plano de Ensino: Módulo Desenho de Software

Tópico 2: Desenho de Software

Desenho/Projeto Não Orientado a Abordagens Específicas

Metodologias

Orientação a Objetos

Desenho/Projeto Orientado às Abordagens Tradicionais

Desenho/Projeto Orientado às Abordagens Ágeis <<revisão>>

Desenho/Projeto Orientado às Abordagens Alternativas

Padrões de Projeto

Plano de Ensino: Módulo Arquitetura de Software

Tópico 3: Arquitetura de Software

Introdução à Arquitetura de Software

Estilos e Padrões Arquiteturais

Visões Arquiteturais

Reutilização de Software

- *Framework*
- Outras Soluções Reutilizáveis

Plano de Ensino: Módulo Seminários

Tópico 4: Seminários

Seminários

Tópico 5: Fechamento

Apresentação e Revisão de Menções

Fishbowl e outras atividades para debatermos sobre o conteúdo do curso

Plano de Ensino - Objetivos



Objetivo_01

- Apresentar ao aluno uma série de técnicas, modelos e conceitos relacionados ao projeto de software de modo que se espera, ao final do curso, que o aluno seja capaz de usá-los para a solução de algum problema através de um software.

Plano de Ensino - Objetivos



Objetivo_02

- Espera-se do aluno a capacidade de transformar requisitos de software (funcionais e não funcionais) em um desenho (projeto) de software.

Plano de Ensino - Objetivos



Objetivo_03

- Apresentar ao aluno conceitos de projeto, tais como abstração, coesão, acoplamento, decomposição, modularização, encapsulamento (dentre outros) e padrões de projeto (GRASP e GoF).

Plano de Ensino - Objetivos



Objetivo_04

- Apresentar ao aluno uma visão geral sobre arquitetura de software, estilos, padrões, visões e reutilização de software.

Plano de Ensino – Conteúdo Programático

Conteúdo Programático – FOCO: Desenho de Software

1. Projeto/Desenho de Software

1.1. Projeto/Desenho de Software Não Orientado a Abordagens Específicas

1.2. Projeto/Desenho de Software Orientado a Abordagens Tradicionais

1.3. Projeto/Desenho de Software Orientado a Abordagens Ágeis <<revisão>>

1.4. Projeto/Desenho de Software Orientado a Abordagens Alternativas

Plano de Ensino – Conteúdo Programático

Conteúdo Programático - FOCO: Desenho de Software

2. Processos, Metodologias e Abordagens

3. Paradigmas e Boas Práticas

2.1. OpenUp, PUÁgil, Scrum, SAFE, Lean, XP
e Siimilares

3.1. Orientação a Objetos

Plano de Ensino – Conteúdo Programático

Conteúdo Programático – FOCO: Desenho de Software

4. Padrões de Projeto

4.1. Padrões GRASP

4.2. Padrões GoFs

Plano de Ensino – Conteúdo Programático

Conteúdo Programático – FOCO: Arquitetura de Software

5. Contextualização para Arquitetura de Software

5.1. Definições

Plano de Ensino – Conteúdo Programático

Conteúdo Programático – FOCO: Arquitetura de Software

6. Estilos Arquiteturais

6.1. Visão Geral

6.2. Stand-alone

6.3. Cliente-servidor

6.4. Peer to Peer

6.5. Objetos Distribuídos

6.6. N-camadas

6.7. Filtros e Dutos (Pipes and Filters)

6.8. Repositório

6.9. Orientado a Eventos (Publisher/Subscriber)

6.10. Orientado a Serviços

Plano de Ensino – Conteúdo Programático

Conteúdo Programático – FOCO: Arquitetura de Software

7. Padrões Arquiteturais

7.1. Visão Geral

7.2. Padrão Model-View-Controller (MVC)

7.3. Outros Emergentes

Plano de Ensino – Conteúdo Programático

Conteúdo Programático – FOCO: Arquitetura de Software

8. Documentação de Arquitetura de Software

8.1 Trabalhando com Múltiplas Visões (notação UML)

Plano de Ensino – Conteúdo Programático

Conteúdo Programático – FOCO: Arquitetura de Software

9. Framework

9.1 Noções Básicas

Plano de Ensino – Conteúdo Programático

Conteúdo Programático – FOCO: Arquitetura de Software

10. *Model Driven Architecture (MDA)*
<<se possível, no módulo desafio>>

10.1 Noções Básicas

Plano de Ensino – Método de Ensino



Aulas teóricas, expositivas e práticas, com atividades intra e extraclasse...

Moodle como canal de comunicação.

Como base, serão utilizados livros, artigos e outros materiais bibliográficos.

Plano de Ensino – Avaliação



- A menção final será constituída das seguintes avaliações:
 - **Nota 1** (**70%** da Nota Final), constituída por:
 - Prova (individual e sem consulta)
 - **Nota 2** (**30%** da Nota Final), constituída por:
 - Trabalho de Desenvolvimento de Software, avaliado ao longo de dinâmicas presenciais (vide cronograma). Avaliações de artefatos e implementações individualizadas.
 - Atividades Intra e Extraclasse (adicionais) também poderão ocorrer ao longo do semestre, a depender do período letivo. Ocorrendo essas avaliações adicionais, suas pontuações variam de semestre para semestre, e de atividade para atividade.
 - Detalhes quanto à prova SUB constam na versão do Plano de Ensino estendida.

Plano de Ensino – Avaliação

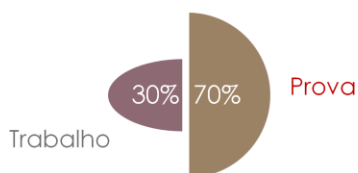
- A Nota Final será dada pela Média Final (MF), sendo:
 - Nota Final = MF = (Nota 1 * 70%) + (Nota 2 * 30%)
 - * Não considera as pontuações das atividades intra e extraclasses adicionais.
- O aluno será aprovado se MF ≥ 5 e Frequência $\geq 75\%$.

- **Exemplo:**

Nota 1 (8,0), Nota 2 (7,0) e Frequência $\geq 75\%$, temos:

- $$\begin{array}{rclclcl} \text{MF} & = & (8,0 * 70\%) & + & (7,0 * 30\%) & = \\ & & 5.6 & + & 2.1 & = 7.7 \end{array}$$

- Portanto, aprovado com **MS**.



Plano de Ensino – Avaliação



Apresentação do Trabalho – Projeto com capacidade de reutilização via Padrões de Projeto e/ou demais recursos que permitam flexibilizar o uso do software desenvolvido.

Demais observações quanto ao trabalho...

Dinâmicas, realizadas ao longo do semestre, de forma presencial, permitirão o acompanhamento constante do trabalho bem como resultarão em notas individuais. Faltas nas dinâmicas resultam em perdas de nota e, conseqüentemente, de menção. Caprichem nas participações! :)

Ao final da disciplina, principalmente, a título de divulgação dos trabalhos obtidos ao longo do semestre, serão realizados seminários com os grupos.



<<espaço aberto para debate sobre o trabalho da disciplina>>

Os macro temas dos trabalhos serão estabelecidos pela professora, variando de semestre para semestre <<a consultar>>.

Independente do tema, será necessário o desenvolvimento de uma solução computacional cobrindo os artefatos de cada módulo da disciplina.

Além disso, escolham um(a) processo/metodologia adequada para conduzir o desenvolvimento.

Complementando, irão modelar artefatos, trabalhar padrões de projeto, usar repositório, organizar a documentação na Wiki, dentre outros aspectos, de forma a facilitar a apresentação dos resultados ao longo das dinâmicas presenciais.

Visando uma solução computacional extensível, será necessário generalizar essa solução de forma que mesma possa ser reutilizada por terceiros.

Normalmente, os pontos flexíveis de uma solução reutilizável são obtidos modelando e implementando padrões de projeto. Por exemplo: *Template Method*, *Strategy*, *State* e *Composite*. Além disso, podem ser obtidos utilizando o conceito de serviços e outros níveis de reutilização.

Reparem que, até o momento, os esforços concentraram-se em tópicos mais associados a Desenho de Software.

Entretanto, a tendência é essa solução computacional concretizar-se como uma biblioteca, uma API, um *plugin*, uma *gem*, uma composição de serviços, um *framework* ou algum outro nível de reutilização. Mas, a solução apenas se moldará dessa forma ao longo do semestre...

Por fim, nas últimas dinâmicas, perceberão que a solução será refatorada, em termos de documentação e implementação, com base em um estilo/padrão arquitetural. Assim, tem-se a cobertura dos tópicos mais associados à Arquitetura de Software.

Tudo ocorrerá passo a passo, orientando-se pelas aulas, dinâmicas e atividades extras. Ok?

As dinâmicas estão organizadas em fases, sendo:

Fase de Planejamento:

Os 15 minutos iniciais da dinâmica serão consumidos para planejamento dos artefatos por parte das equipes.

Nesse período, podem solicitar orientação à professora e/ou aos monitores presentes.

Aqui, cabe debate entre os membros sobre os artefatos que serão realizados bem como o escopo de atuação de cada membro ou de cada dupla.

Portanto, podem usar "pair design", ou seja, artefatos sendo modelados dois a dois...

* se envolver implementação na dinâmica, podem usar pair programming. :)

Fase de Execução:

Serão dedicados 25 minutos da aula para elaboração/modelagem do(s) artefato(s), conforme planejado pelas equipes.

* se envolver implementação na dinâmica, caberá desenvolvimento também (pelo menos um esboço da solução). :)

Fase de Avaliação:

Essa avaliação será realizada a partir de 40 minutos de aula, ou seja, nos 70 minutos finais da aula.

Serão 10 minutos, no máximo, por grupo, para mostrar, bem brevemente, o que fizeram...

Demais grupos que não estiverem, naquele momento, sendo avaliados, poderão continuar elaborando/modelando o(s) artefatos.

Lembrando que essas avaliações são tanto por equipe, quanto individuais...

<<validar tema com a professora, por gentileza>>

Plano de Ensino – Avaliação

- Demais observações...
 - O trabalho deve ser disponibilizado via Wiki do Git, organizando essa Wiki de forma a demonstrar a evolução do trabalho bem como contemplando no trabalho os conteúdos do programa da disciplina (pertinentes às escolhas do projeto). A Wiki será construída ao longo da disciplina, visando facilitar a exposição dos artefatos e demais esforços nas dinâmicas presenciais e no seminário final. Há um conjunto de diretrizes base para auxiliar nessa organização...
 - Será cobrado o uso do Git, como repositório, visando acompanhar os esforços individuais dos membros, principalmente, nas atividades de implementação. Portanto, caprichem na distribuição dos esforços, visando bons gráficos de participação dos membros. Usem nomes que viabilizem a identificação de cada membro do grupo por parte da professora.



Template base disponível no Moodle.

Plano de Ensino – Avaliação

- Demais observações...
 - Entregas, se necessárias, serão realizadas via Moodle da disciplina, em atividades abertas para esse fim. Arquivo único, sem re-submissão.
 - Qualquer atividade, trabalho e/ou questão de prova que seja identificada como cópia receberá nota zero (todos os envolvidos).
 - Presença em sala de aula será um critério considerado. Portanto, sejam pontuais e procurem não faltar! Tolerância: 10 min (não repetidas vezes). :)

Plano de Ensino – Avaliação

Brincadeira...



Plano de Ensino – Referências



- Bibliografia Básica:

1. **(BIBLIOTECA FGA – 15 unidades)** Pfleeger, Shari Lawrence. Engenharia de Software: Teoria e Prática. 2ª. Edição. São Paulo: Prentice Hall, c2004. xix, 535 p. ISBN 9788587918314. [OPEN ACCESS] Slides Prentice Hall Education http://wps.prenhall.com/br_pfleeger_engsoftware_2/ (último acesso: 2017)
2. **(BIBLIOTECA FGA – 16 unidades)** Larman, Craig. Utilizando UML e Padrões: Uma Introdução à Análise e ao Projeto Orientado a Objetos e ao Desenvolvimento Iterativo. 3ª. Edição. Porto Alegre, RS: Bookman, 2007. xiv, 695 p. ISBN 9788560031528 OU Larman, Craig. Utilizando UML e Padrões: Uma Introdução à Análise e ao Projeto Orientado a Objetos. Porto alegre: Bookman, 2000. 492p. ISBN 85-7307-651-8 OU [OPEN ACCESS] Larman, Craig. Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Unified Process. 2ª Edition. Free Book for Everyone.
3. **(EBRARY)** Zhu, H. Software Design Methodology: From Principles to Architectural Styles. Butterworth-Heinemann. 2005.

Plano de Ensino – Referências



- Bibliografia Complementar:

1. **(BIBLIOTECA FGA – 11 unidades)** Pressman, Roger S. Engenharia de Software: Uma Abordagem Profissional. 7ª. Edição. Porto Alegre: AMGH, 2011. 780 p. ISBN 9788563308337 OU Pressman, Roger S. Engenharia de Software: Uma Abordagem Profissional. 5ª. Edição. Rio de Janeiro: McGraw-Hill, c2002. ISBN 8586804258.
2. **(BIBLIOTECA FGA – 7 unidades)** Page-Jones, Meilir. Fundamentals of Object-Oriented Design in UML. New York: Dorset, The Addison-Wesley object technology series. c2000. xxi, 458 p. ISBN 020169946X.
3. **(BIBLIOTECA FGA – 4 unidades)** Sommerville, Ian. Engenharia de Software. 8ª. Edição. São Paulo: Pearson Addison Wesley. 2007. 552 p. ISBN 978-85-88639-28-7. [OPEN ACCESS] Slides Pearson Education http://wps.aw.com/br_sommer_engen_6/8/2096/536701.cw/index.html (último acesso: 2017)
4. **(EBRARY)** Coplien, James and Bjørnvig, Gertrud. Learn Architecture: For Agile Software Development. John Wiley & Sons. 2010.

Plano de Ensino – Referências



- Bibliografia Complementar (continuação):

5. **(EBRARY)** Duggan, Dominic. Quantitative Software Engineering Series : Enterprise Software Architecture and Design : Entities, Services, and Resources (1). Hoboken, US: Wiley-IEEE Computer Society Pr, 2012.
6. **(EBRARY)** Klimczak, Erik. Design for Software: A Playbook for Developers (1). Somerset, GB: Wiley, 2013.
7. **(Open Access)** Portal sobre Arquitetura de Software. Disponível em: www.softwarearchitectureportal.org/ (último acesso: 2017)
8. **(Open Access)** Portal sobre MDA. Disponível em: <http://www.omg.org/mda/> (último acesso: 2017)

* Recomenda-se ainda o livro do Gamma, por ser uma literatura clássica para padrões GoFs.

Considerações Finais



Considerações Finais

- Nessa aula, foi apresentado o plano de ensino da disciplina de Desenho de Software, com detalhamento sobre:
 - Ementa;
 - Objetivos;
 - Conteúdo Programático;
 - Método de Ensino;
 - Avaliação, e
 - Referências.

ÓTIMO
SEMESTRE
PARA TODOS!

Considerações Finais



LEMBREM-SE...

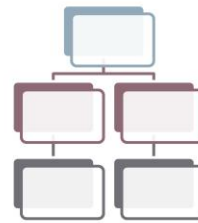
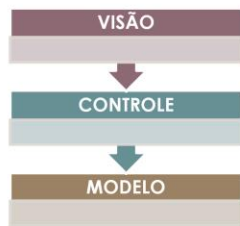
O plano de ensino poderá sofrer alterações, as quais
serão acordadas em sala de aula e/ou via Moodle.

Será evitado ao máximo!

MANTENHAM OS ESTUDOS EM DIA!
SÓ SE APRENDE PRATICANDO!

DICA: Procurem trazer as máquinas pessoais,
instalem os ambientes e as ferramentas para um
acompanhamento pleno da disciplina...

:)



FIM

Dúvidas?

CONTATO:
mileneserrano@unb.br
OU
mileneserrano@gmail.com

sugestões?

