



## UNIVERSIDADE FEDERAL DE UBERLÂNDIA

### Faculdade de Computação

Av. João Naves de Ávila, nº 2121, Bloco 1A - Bairro Santa Mônica, Uberlândia-MG, CEP 38400-902

Telefone: (34) 3239-4144 - <http://www.portal.facom.ufu.br/> facom@ufu.br



## PLANO DE ENSINO

### 1. IDENTIFICAÇÃO

Componente Curricular:	Projeto e Desenvolvimento de Sistemas de Informação 1						
Unidade Ofertante:	FACOM						
Código:	GSI034	Período/Série:	7	Turma:	S		
Carga Horária:				Natureza:			
Teórica:	30	Prática:	30	Total:	60	Obrigatória(x)	Optativa: ( )
Professor(A):	Rodrigo Sanches Miani				Ano/Semestre:	2021-2	
Observações:							

### 2. EMENTA

Apresentação de um problema a ser solucionado através do uso do computador; Especificação do software a ser implementado; Implementação do software especificado. Problemas e práticas recomendadas no desenvolvimento de software; visão geral do processo de desenvolvimento; planejamento e elaboração; análise e projeto; implementação.

### 3. JUSTIFICATIVA

A disciplina busca congregiar todo o conhecimento obtido durante os períodos anteriores. Isto é feito através da construção de um sistema de informação. Os estudantes terão o papel de desenvolvedores enquanto que o docente (ou outros envolvidos) fará o papel de cliente. A ideia é simular o ambiente real de desenvolvimento de software encontrado no mercado.

### 4. OBJETIVO

#### Objetivo Geral:

Identificar problemas do mundo real implementáveis computacionalmente, realizar a análise e projeto de soluções em software, de forma padronizada e eficiente, para resolvê-los.

#### Objetivos Específicos:

Utilizar os diferentes conceitos obtidos durante os períodos anteriores e utiliza-los para o desenvolvimento de um sistema de informação

Vivenciar um processo de desenvolvimento de software completo, buscando simular diversas situações reais encontradas durante o desenvolvimento no mundo real

### 5. PROGRAMA

1. Identificação de um problema a ser implementado

2. Análise

- o Problemas e práticas recomendadas
- o Levantamento de requisitos
- o Custos relacionados
- o Metodologias de análise
- o Modelagem de banco de dados
- o Diagramas para análise
- o Visão geral das ferramentas de análise

### 3. Projeto

- o Técnicas de projeto
- o Projeto de telas
- o Projeto de banco de dados
- o Escolha de ferramentas de desenvolvimento
- o Modelos de construção de software
- o Camadas de software
  - Componentes / reutilização de software

### 4. Criação de protótipos

## 6. METODOLOGIA

O objetivo da disciplina é o desenvolvimento de um sistema de software. A fim de aproximar de um ambiente real, a turma é dividida em grupos. O trabalho em grupo aproxima-se de uma situação real, onde um software é desenvolvido por equipes.

O processo de desenvolvimento baseia-se em processos de desenvolvimento de software como o Rational Unified Process (RUP) e cada grupo deve preparar um conjunto de artefatos ao longo da disciplina. O processo de desenvolvimento foca principalmente nas fases de concepção e elaboração do processo. Entretanto o grupo deve construir um protótipo englobando dois ou três casos de uso visando validar a solução técnica elaborada ao longo do curso.

O domínio do software a ser construído será proposto pelo professor. Serão apresentados quatro problemas para a turma. O professor nesse caso, assumindo o papel semelhante a um cliente, pode propor requisitos funcionais e não funcionais para o problema escolhida. As aulas expõem as atividades e artefatos associados a cada disciplina e fornece informações sobre a execução destas atividades por cada grupo. Ao longo do semestre, a cada aula, os grupos apresentam o andamento das atividades e indicam quais serão as atividades seguintes. O grupo é também avaliado por cumprir os compromissos assumidos a cada aula. Cada grupo apresenta os artefatos à medida que são criados e recebem uma avaliação sobre pontos a serem melhorados e/ou corrigidos.

Dessa forma cada grupo pode perceber o que está incorreto, melhorar o entendimento de cada atividade e ajustar o seu trabalho. Esse processo permite que um grupo com rendimento mais baixo em uma etapa possa recuperar-se ao longo do semestre.

**a) Atividades presenciais teóricas:** 28 horas/aula - as atividades presenciais teóricas envolvem, além da discussão de cada uma das etapas de desenvolvimento de software (investigação, análise, projeto e desenvolvimento) reuniões com os grupos para tirar dúvidas relacionadas ao andamento do projeto.

**Horários das atividades presenciais teóricas:** Quinta-feira de 19:00 às 20:40.

**b) Atividades presenciais práticas:** 28 horas/aula - as atividades presenciais práticas estão relacionadas ao desenvolvimento do projeto.

**Horários das atividades presenciais teóricas:** Segunda-feira de 19:00 às 20:40.

**c) Atividades assíncronas (Art 1º da Resolução CONSUN nº 30/2022):** 16 horas/aula - as atividades assíncronas estão relacionadas a construção do protótipo.

**Plataforma de T.I./softwares que serão utilizados:** Microsoft Teams

**d) Demais atividades letivas:** 0 horas;

Semana	Aula	Carga horária presencial	Carga horária assíncrona	Conteúdo	Entregas
1	02/mai	2		Apresentação da disciplina	
1	05/mai	2		Apresentação dos tópicos e organização dos grupos	
2	09/mai	2		Investigação do sistema - Realização do projeto	

2	12/mai	2		Investigação do sistema - R1	
3	16/mai	2		Investigação do sistema - Realização do projeto	
3	19/mai	2		Investigação do sistema - R2	
4	23/mai	2		Análise do sistema - Realização do projeto	
4	26/mai	2		Análise do sistema - R1	Documento de investigação do sistema
5	30/mai	2		Análise do sistema - Realização do projeto	
5	02/jun	2		Análise do sistema - R2	
6	06/jun	2		Análise do sistema - Realização do projeto	
6	09/jun	2		Projeto do sistema - R1	
7	13/jun	2		Projeto do sistema - Realização do projeto	Documento de análise do sistema
7	16/jun	-	-	Feriado - Corpus Christi	
8	20/jun	2		Projeto do sistema - Realização do projeto	
8	23/jun	2		Projeto do sistema - R2	
9	04/jul	2		Projeto do sistema - Realização do projeto	
9	07/jul	2		Desenvolvimento do sistema - R1	Documento do projeto do sistema
10	11/jul	2		Desenvolvimento do sistema - Realização do projeto	
10	14/jul	2	2	Desenvolvimento do sistema - R2	
11	18/jul	2	2	Desenvolvimento do sistema - Realização do projeto	
11	21/jul	2	2	Desenvolvimento do sistema - R3	
12	25/jul	2	2	Desenvolvimento do sistema - Realização do projeto	
12	28/jul	2	2	Desenvolvimento do sistema - R4	
13	01/ago	2	2	Desenvolvimento do sistema - Realização do projeto	
13	04/ago	2	2	Desenvolvimento do sistema - Realização do projeto	
14	08/ago	2	2	Desenvolvimento do sistema - Realização do projeto	
14	11/ago	2		Apresentação do sistema	Entrega do protótipo (código e documentação)
15	15/ago	-	-	Feriado - Nossa Senhora de Abadia	
15	18/ago	2		Apresentação do sistema	

Carga horária presencial total (hora-aula) = 56

Carga horária assíncrona total (hora-aula) = 16

Carga horária total (presencial + assíncrona) = 72

## 7. AVALIAÇÃO

Cada etapa (investigação, análise, projeto e desenvolvimento) terá UMA entrega associada. Além disso, será dada uma nota a apresentação final do sistema. Portanto:

$$NF = 15 * N_{Inv} + 30 * N_{An} + 30 * N_{Pro} + 15 * N_{Des} + 10 * N_{Apr}$$

**Importante 1:** cada integrante do grupo que não esteja presente nas reuniões de quinta-feira representará uma dedução de 0,5 ponto na nota de tal etapa. Ou seja, não pode vacilar! Somente ausências devidamente justificadas e com antecedência serão aceitas.

**Importante 2:** entregas com atraso serão penalizadas. 1 dia de atraso = dedução de 10%, 2 dias de atraso = dedução de 15% por cento, 3 dias de atraso = dedução de 20% por cento, 4 dias de atraso ou mais = dedução de 40% por cento.

O cronograma de entrega das avaliações está disponível junto do cronograma de aulas (item 6 do presente plano de ensino).

### **Informações de acordo com a Resolução nº 7/2020 do Conselho de Graduação:**

#### **a) Datas e horários da avaliação:**

A disciplina não apresenta um conjunto de avaliações mas em um trabalho que é realizado ao longo do curso. À medida que a cada atividade é conduzida, periodicamente cada grupo receberá uma avaliação parcial do trabalho realizado e de indicações de como o mesmo pode ser melhorado antes da entrega final.

#### **b) Critérios para a realização e correção das avaliações:**

Durante a apresentação da atividade, os alunos receberão as correções realizadas utilizando os artefatos enviados por cada grupo. O trabalho final será apresentado por videoconferência em horário que será previamente combinado com cada grupo.

**c) Validação da assiduidade dos discentes:** durante as aulas presenciais, a assiduidade dos discentes será aferida via chamada. Durante as aulas assíncronas, a assiduidade dos discentes será aferida pela entrega do protótipo do sistema.

**d) Especificação das formas de envio das avaliações pelos discentes, por meio eletrônico:** Microsoft Teams

## **8. BIBLIOGRAFIA**

### **Básica**

Blaha, Rumbaugh. Modelagem e projetos baseados em objetos com UML 2. Elsevier: Campus. 2006  
Maldonado, Delamaro, Jino. Introdução ao Teste de Software. Campus, 2007.  
Guedes. UML 2 – Uma abordagem prática. Novatec. 2009  
Lowe, Pressman. Engenharia Web; LTC, 2009.

### **Complementar**

FURLAN, José Davi. Modelagem de Objetos através UML. Makron Books, 1998.  
LAIRMAN, Craig. Utilizando UML e Padrões. Ed. Bookman, 2007.  
OESTEREICH, Bernd; Weilkiens, Tim. UML 2 Certification Guide. MORGAN KAUFMANN, 2006.  
PENDER, Tom, UML – A Bíblia. Elsevier: Campus. 2004.  
Pressman. Engenharia de Software, 7ª. Edição.  
Sommerville. Engenharia de Software, 9ª. Edição

SOMMERVILLE, Ian. Engenharia de Software. Editora Pearson / Addison Wesley, 2003  
BOOCH, Grady; Jacobson, Ivar; Rumbauch, James. UML: Guia do Usuário. Campus, 2006

## Online

### Engenharia de Software Moderna

<https://engsoftmoderna.info>

### Rational Unified Process

<https://wiki.c2.com/?RationalUnifiedProcess>

### Documento de Visão

[https://pt.wikipedia.org/wiki/Documento\\_de\\_vis%C3%A3o](https://pt.wikipedia.org/wiki/Documento_de_vis%C3%A3o)

<https://www.ibm.com/docs/pt-br/elm/6.0.5?topic=requirements-vision-document>

### Documento de Especificação de Requisitos

<https://www.ibm.com/docs/pt-br/elm/6.0?topic=generation-managing-composing-requirements>

### Documento de Arquitetura de Software

[https://www.cin.ufpe.br/~gta/rup-](https://www.cin.ufpe.br/~gta/rup-vc/core.base_rup/workproducts/rup_software_architecture_document_C367485C.html)

[vc/core.base\\_rup/workproducts/rup\\_software\\_architecture\\_document\\_C367485C.html](https://www.cin.ufpe.br/~gta/rup-vc/core.base_rup/workproducts/rup_software_architecture_document_C367485C.html)

## 9. APROVAÇÃO

Aprovado em reunião do Colegiado realizada em: \_\_\_\_/\_\_\_\_/\_\_\_\_

Coordenação do Curso de Graduação: \_\_\_\_\_



Documento assinado eletronicamente por **Rodrigo Sanches Miani, Professor(a) do Magistério Superior**, em 03/06/2022, às 16:44, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



A autenticidade deste documento pode ser conferida no site [https://www.sei.ufu.br/sei/controlador\\_externo.php?acao=documento\\_conferir&id\\_orgao\\_acesso\\_externo=0](https://www.sei.ufu.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0), informando o código verificador **3516341** e o código CRC **48B17589**.