


**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](mailto: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:	2023-1	
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 congrega 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 e dois protótipos ao longo da disciplina. O processo de desenvolvimento usará ideias de "Hackathon" e focará na concepção, detalhamento da solução e implementação do protótipo para validar a solução.

O domínio do software a ser construído será proposto pelo professor. Serão apresentados seis 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.

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

**Horários das atividades presenciais práticas:** Segunda-feira de 20:50 às 22:30.

**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. Os alunos serão incluídos manualmente na equipe do Microsoft Teams. Além disso, um email de boas vindas com as instruções iniciais do curso também será enviado.

**Horário de atendimento:** os atendimentos poderão acontecer de três formas: i) durante as aulas presenciais, ii) toda segunda entre 18:00 e 19:00 ou iii) de forma remota usando o Microsoft Teams.

Semana	Aula	Conteúdo	Carga horária teórica	Carga horária prática	TDE	Entregas
1	31/jul	Apresentação da disciplina	4			
2	07/ago	R1 - Apresentação dos problemas, discussão dos problemas, apresentação da proposta (arquitetura, tecnologias, principais funcionalidades), feedback (professor e alunos)	4			
3	14/ago	R2 - Análise do feedback, primeira rodada de implementação do protótipo (repositório de código, persistência, organização de tarefas, início da codificação), apresentação do que foi feito	4			E1
4	21/ago	R3 - Segunda rodada de implementação do protótipo, início da documentação,		4		

		apresentação do que foi feito				
5	28/ago	R4 - Terceira rodada de implementação do protótipo, documentação, apresentação do que foi feito		4		
6	04/set	Reposição de quinta-feira. Não haverá aula. Atividade extra-curricular - quarta rodada de implementação do protótipo.			4	
7	11/set	R6 - Quinta rodada de implementação do protótipo, documentação, apresentação do que foi feito	4			
8	18/set	Desenvolvimento da apresentação - Dúvidas		4		
9	25/set	Apresentação do protótipo 1 (AP1 - meio de semestre)	4			E2
10	02/out	R7 - Discussão dos resultados obtidos até o momento.	4			
11	09/out	R8 - Sexta rodada de implementação do protótipo, documentação, apresentação do que foi feito	4			
12	16/out	R9 - Sétima rodada de implementação do protótipo, documentação, apresentação do que foi feito		4		
13	23/out	R10 - Oitava rodada de implementação do protótipo, documentação, apresentação do que foi feito		4		
14	30/out	R11 - Nona rodada de implementação do protótipo, documentação, apresentação do que foi feito		4		
15	06/nov	R12 - Décima rodada de implementação do protótipo, documentação, apresentação do que foi feito	4			
16	13/nov	Apresentação do protótipo 2 (AP2 - fim de semestre)	4			E3
17	20/nov	Atividade extra-curricular - finalização do desenvolvimento e entrega do protótipo			4	
18	27/nov	Finalização do desenvolvimento e entrega do protótipo		4		

Carga horária teórica total (hora-aula) = 36

Carga horária prática total (hora-aula) = 28

Carga horária assíncrona (TDE) = 8

Carga horária total (teórica+prática+TDE) = 72

## 7. AVALIAÇÃO

A disciplina será dividida em duas partes:

1. Apresentação da disciplina, apresentação do problema, discussão, elaboração da primeira versão do protótipo e apresentação parcial;
2. Discussão dos resultados obtidos até o momento, refinamento e implementação da versão "final" do protótipo e apresentação final semanas;

As aulas serão organizadas em reuniões. Cada reunião irá produzir um breve relatório (P). Além disso, são esperadas três entregas em forma de documento (E1, E2 e E3) e duas apresentações do protótipo (AP1 e AP2). Portanto, a nota final da disciplina será dada por:

$$NF = (0,15)*E1 + (0,15)*E2 + (0,15)*E3 + (0,2)*AP1 + (0,25)*AP2 + (0,1)*P$$

Cada uma das entregas está relacionada a documentação do protótipo e incluirá diagramas, manuais e código-fonte. As datas previstas das entregas e das apresentações estão dispostas no cronograma (seção 6 - Metodologia).

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

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

Cada relatório deverá ser produzido e enviado em até um dia após os encontros (aulas presenciais). O envio deverá ser feito usando a plataforma Microsoft Teams.

#### **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.

**c) Validação da assiduidade dos discentes:** a assiduidade dos discentes será aferida via chamada.

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

O discente terá direito a **Atividade de recuperação de aprendizagem**, se e somente se, **não obtiver o rendimento mínimo para aprovação e com frequência mínima de 75% (setenta e cinco por cento)**, de acordo com o Art 141. das novas Normas Gerais de Graduação (Resolução CONGRAD Nº 46/2022). A frequência será aferida por chamada feita em sala de aula. Nesse caso, ele poderá fazer uma prova sobre o conteúdo de toda a disciplina;

O aluno será aprovado caso  $(NA+NR)/2 \geq 60$ , onde NA = Nota atual do aluno e NR = Nota da recuperação do aluno;

Em caso de recuperação, a nota máxima do aluno será de 60 pontos. Caso necessário, a recuperação será aplicada no dia 27/11/2023.

## **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.

Valente, Marco Tulio. Engenharia de software moderna. Princípios e Práticas para Desenvolvimento de Software com Produtividade, v. 1, p. 24, 2020. Disponível em: <https://engsoftmoderna.info/>

### **Complementar**

FURLAN, José Davi. Modelagem de Objetos através UML. Makron Books, 1998.

LAIRMAN, Craig. Utilizando UML e Padrões. Ed. Bookman, 2007.

OESTEREICH, Bernd; Weikiens, 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; Rumbaugh, 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 25/09/2023, às 16:07, 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?](https://www.sei.ufu.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0)

[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 **4732196** e o código CRC **COE3CD23**.