

<b>Professor:</b> Estudos Autônomos - Sem Professor	<b>Disciplina / *Unidade Curricular:</b> Modelagem de software <b>Carga Horária:</b> 160	<b>Curso:</b> Ciência da Computação <b>Período Letivo:</b> 2023/1  <i>* Exclusivo para Unidades Curriculares</i>
<b>Objetivo Geral / *Tópico Gerador:</b> Como funciona um Sistema Gerenciador de Banco de Dados SGBDs Como os dados podem ser armazenados manipulados e recuperados em uma estrutura de banco de dados Como identificar requisitos de negócio no processo de planejamento do software através da análise de um problema real Qual a importância na escolha de um modelo de banco de dados para a modelagem de um software Determinando a elicitação de requisitos funcionais e não funcionais a partir da análise de requisitos Quais ferramentas podem ser utilizadas para criar modelos de softwares Como e quando diagramas devem ser utilizados Como a realização de prototipações pode contribuir com o desenvolvimento e modelagem de um software Modelagem orientada a objetos Modelagem e armazenamento de dados através de modelos conceituais lógicos e físicos		
<b>Objetivos Específicos / *Metas de compreensão:</b> Criar a especificação de software elicitando os requisitos funcionais e não funcionais do software em conformidade com os requisitos do usuário seguindo conceitos básicos de Engenharia de Software Criar modelos de sistemas de software utilizando o paradigma orientado a objetos a partir dos principais diagramas da UML com ênfase nos quatro mais utilizados casos de uso classes atividades e sequência Especificar modelos conceituais de banco de dados analisando aspectos do mundo real a serem tratados pelos sistemas de informação e representandoos corretamente de acordo com o metamodelo selecionado e integrandoos com as diretrizes de administração de dados de uma organização Analisar problemas avaliando as necessidades dos clientes Utilizar ferramentas de prototipagem de software e aplicar os tipos de prototipagem conforme o projeto Criar modelos conceituais lógicos e físicos de banco de dados de acordo com os propósitos das necessidades do sistema de informação especificando estruturas e mecanismos de armazenamento busca e recuperação de dados Meta máxima Realizar a modelagem do software e do armazenamento de dados de forma a atender às necessidades dos clientes Criar instruções de manipulação CRUD para armazenamento e recuperação de dados utilizando conexões com SGDBs e linguagem de consulta estruturada Compreender o funcionamento de Sistema Gerenciador de Banco de Dados SGBDs propriedades ACID e implicações de segurança		
<b>Plano Aula / Atividade Discente / *Desempenho de Compreensão:</b> Criar instruções de manipulação CRUD para armazenamento e recuperação de dados utilizando conexões com SGDBs e linguagem de consulta estruturada.		
<b>Avaliação Contínua:</b> Atividade Avaliativa 1 (A1) 30pts Avaliação Integrada (AIT) 30pts (Extra) Atividade Avaliativa 3 (A3) 40pts Atividade Avaliativa 2 (A2) 30pts  Exigência mínima para aprovação: 70 pontos e % de frequência nas disciplinas presenciais. Tipo de Avaliação: Avaliação de EAD		
Emitido por: DBADM		Rio de Janeiro, 02/06/2025

<b>Ementa:</b> Introdução ao processo de desenvolvimento software Fundamentos de requisitos Tipos de requisitos (funcionais e nãofuncionais) Modelagem e notação UML (Unified Modeling Language): diagramas estruturais e comportamentais (diagramas de casos de uso, de atividades e de classes) Noções básicas dos demais diagramas UML Análise e projeto de banco de dados: relacional e nãorelacional Modelo entidade-relacionamento Modelo relacional e normalização Modelo lógico e físico de banco de dados Sistemas Gerenciadores de Banco de Dados (SGBDs) Implementação de transações de banco de dados (CRUD) Linguagem SQL
<b>Metodologia (Não se aplica a Unidades Curriculares, exceto para o curso de Medicina):</b>
<b>Programa:</b> <ul style="list-style-type: none"><li>• Visão geral do conceito de processo de software. <i>¿</i> Sistemas Gerenciadores de Banco de Dados (SGBDs) <i>¿</i> Elicitação de requisitos funcionais e não-funcionais a partir da análise de requisitos. Prototipações. Model elicitação de requisitos. <i>¿</i> Tipos de requisitos: funcionais e não funcionais. <i>¿</i> Visão geral do conceito de processo de software <i>¿</i> Tipos de requisitos: funcionais e não funcionais. <i>¿</i> Prototipação não funcional. <i>¿</i> Notação UML (U e normalização <i>¿</i> Teoria de conjuntos. <i>¿</i> Operações matemáticas básicas com vetores e matrizes <i>¿</i> Raciocínio lógico booleano. <i>¿</i> Geometria <i>¿</i> Álgebra</li></ul>
<b>*Certificação:</b> Qualificação Profissional em Modelagem de Software
<b>*Competências:</b> Compreender o processo de desenvolvimento de software Identificar e criar especificações de requisitos para problemas computacionais Interpretar analisar avaliar e projetar documentação retirar a palavra documentação soluções por meio de modelos orientados a objetos utilizando notação UML com enfoque em análise e projeto de sistemas trocar sistemas por software Interpretar trocar Interpretar por Compreender analisar avaliar e criar modelos de dados utilizando técnicas de modelagem de dados relacional Compreender a necessidade de interagir com pessoas de diferentes perfis possivelmente de diversas áreas do conhecimento
<b>Bibliografia Básica:</b> ELMASRI, Ramez. Sistemas de banco de dados. 6. ed. São Paulo: Pearson, 2011. Disponível em: <a href="http://unibh.bv3.digitalpages.com.br/users/publications/9788579360855">http://unibh.bv3.digitalpages.com.br/users/publications/9788579360855</a> PRESSMAN, Roger; MAXIM, Bruce. Engenharia de Software. Uma abordagem profissional. 8a. Ed. Bookman, 2016. <a href="https://integrada.minhabiblioteca.com.br/#/books/9788580555349/cfi/31/4/2@100:0:00">https://integrada.minhabiblioteca.com.br/#/books/9788580555349/cfi/31/4/2@100:0:00</a> MEDEIROS, Emani. Desenvolvendo Software com UML 2.0. São Paulo: Pearson Education, 2004. <a href="https://bv4.digitalpages.com.br/?term=uml&amp;searchpage=1&amp;filtro=todos&amp;from=busca&amp;page=-20&amp;ion=0#/legacy/2921">https://bv4.digitalpages.com.br/?term=uml&amp;searchpage=1&amp;filtro=todos&amp;from=busca&amp;page=-20&amp;ion=0#/legacy/2921</a> RANGEL, Pablo. Sistemas orientados a objetos. Rio de Janeiro: Brasport, 2021. E-book. Disponível em: <a href="https://plataforma.bvirtual.com.br/Leitor/Publicacao/197367/epub/0">https://plataforma.bvirtual.com.br/Leitor/Publicacao/197367/epub/0</a> .

Emitido por: DBADM

Rio de Janeiro, 02/06/2025

---

**Bibliografia Complementar:**

PFLEEGER, Shari Lawrence. Engenharia de software: teoria e prática. 2. ed. São Paulo: Prentice Hall, 2004. <https://bv4.digitalpages.com.br/?term=engenharia%2520de%2520software&searchpage=1&filtro=todos&from=busca#/legacy/476>

HEUSER, Carlos Alberto. Projeto de banco de dados. 6. ed. Porto Alegre: Bookman, 2011. E-book. Disponível em: <https://integrada.minhabiblioteca.com.br/books/9788577804528>

LARMAN, Craig. Utilizando UML e padrões: uma introdução à análise e ao projeto orientados a objetos e desenvolvimento iterativo. 3. ed Porto Alegre: Bookman, 2007. <https://integrada.minhabiblioteca.com.br/#/books/9788577800476/cfi/01/4/2@100:0.00>

FOWLER, Martin; SCOTT, Kendall. UML essencial: um breve guia para a linguagem-padrão de modelagem de objetos. 3ª. ed. Porto Alegre: Bookman, 2004. <https://integrada.minhabiblioteca.com.br/#/books/9788560031382/cfi/6/2/4/2@0:0.131>

SOMMERVILLE, Ian. Engenharia de Software. 9. ed. São Paulo: Pearson Prentice Hall, 2011. [https://bv4.digitalpages.com.br/?term=engenharia%2520de%2520software&searchpage=1&filtro=todos&from=busca&page=\\_14\\$ion=0#/legacy/276](https://bv4.digitalpages.com.br/?term=engenharia%2520de%2520software&searchpage=1&filtro=todos&from=busca&page=_14$ion=0#/legacy/276)