

**UNIVERSIDADE FEDERAL DE UBERLÂNDIA****Faculdade de Computação**

Av. João Naves de Ávila, 2121, Bloco 1B - Bairro Santa Mônica, Uberlândia-MG, CEP 38400-902
Telefone: (34) 3239-4393 - www.facom.ufu.br - facom@ufu.br

**PLANO DE ENSINO****1. IDENTIFICAÇÃO**

Componente Curricular:	Processo de Desenvolvimento de Software								
Unidade Ofertante:	Faculdade de Computação								
Código:	FACOM32403		Período/Série:		4		Turma:	S	
Carga Horária:						Natureza:			
Teórica:	60	Prática:	0	Total:	60	Obrigat ória:	(X)	Optativ a:	()
Professor(A):	Fabíola S. F. Pereira					Ano/Semestre:		2026/1	
Observações:									

2. EMENTA

Visão geral sobre software. Requisitos de software. Processo de construção de software. Modelagem de software. Arquitetura de software. Abordagens para implementação de uma arquitetura de software. Confiança e proteção.

3. JUSTIFICATIVA

A disciplina proporciona aos futuros profissionais uma compreensão estruturada e sistemática das etapas envolvidas na construção de um software. Ao entenderem os processos, os estudantes se tornam mais preparados para trabalhar em equipe, gerenciar projetos e tomar decisões técnicas alinhadas com os objetivos de negócio, aumentando sua empregabilidade e capacidade de inovar no mercado de tecnologia da informação.

4. OBJETIVO**Objetivo Geral:**

Discutir e aplicar os principais conceitos relacionados à construção de software.

Objetivos Específicos:

- Discutir sobre a produção de software, especialmente sobre os critérios a serem considerados na construção de um produto de software (voltado para vários e diferentes usuários).
- Especificar e gerenciar requisitos de software.
- Discutir processos de construção de software, distinguindo processo clássico e processo ágil. Criar modelo de software.
- Discutir sobre a aplicabilidade de diferentes tipos de arquitetura de software.
- Aplicar padrão de projeto apropriado ao software em construção.
- Discutir sobre segurança e proteção em software.

5. PROGRAMA

I. Software:

1. O que é software
2. Histórico da produção de software
3. Software para uso particular vs. software para uso em grande escala (produto de software)
4. O que é engenharia de software

II. Requisitos de software:

1. Requisitos funcionais
2. Requisitos não funcionais
3. Técnicas para capturar requisitos
4. Técnicas para gerenciar requisitos
5. Técnicas para rastrear requisitos no software

III. Processo de construção de software:

1. Ciclo de vida de um software (fases canônicas): concepção (análise), planejamento (projeto), implementação, validação, implantação e evolução (manutenção)
2. Processos clássicos para a construção de software: cascata, incremental, reuso, processo unificado
3. Processos ágeis:
 1. O que é um processo ágil: qual é a diferença entre um processo ágil e um processo clássico
 2. Manifesto em favor do desenvolvimento ágil ([hp://agilemanifesto.org/](http://agilemanifesto.org/))
 3. Propostas de processo ágil: programação extrema e Scrum

IV. Modelagem de software:

1. O que é um modelo de software: em qual fase do ciclo de vida a modelagem se insere
2. UML:
 1. Histórico da UML
 2. Quais são os diagramas da UML e como desenhá-los
 1. Diagramas estruturais: classes, objetos, pacotes e demais
 2. Diagramas comportamentais: casos de uso, atividades, sequência e demais

V. Arquitetura de Software:

1. O que é a arquitetura de um software: em qual fase do ciclo de vida a arquitetura se insere
2. Critérios para a seleção de uma arquitetura
3. Tipos de arquitetura: Camada, Cliente-Servidor, Repositório

VI. Padrões de Projeto (ou abordagens para implementação de uma arquitetura):

1. O que é padrão de projeto
2. Tipos de padrões: MVC, Fábrica

VII. Confiança e proteção:

1. Propriedades da confiança
2. Disponibilidade e confiabilidade
3. Segurança
4. Proteção

6. METODOLOGIA

O curso será ministrado através de aulas expositivas sobre o tema, às quartas-feiras, de 19:00 até 20:40; e sextas-feiras, de 20:50 até 22:30. Para a exposição, serão usados slides, disponibilizados via página da disciplina, no grupo *pds-2026-1* do MS Teams (inscrição: bit.ly/pds20261), em conjunto com a exposição oral da professora. A apresentação será complementada, sempre que necessário, com anotações e demonstrações no quadro da sala. Serão ao todo 60 horas-aula presenciais. As 12 horas-aula faltantes serão contabilizadas por meio da realização de Atividades extraclasse distribuídas ao longo do semestre.

A) Cronograma de atividades

A carga horária presencial será distribuída ao longo das semanas letivas previstas no calendário acadêmico da UFU, conforme cronograma de atividades a seguir.

C
a
r
g
a
H
o
r
á
r
i
a
(
D
b
e
e
n
t
e
s
A
p
r
e
s
e
n
t
a
ç
ã
o
d
a
d
i
s
c
i
p
l
i
n
a
r
e
:
c
o
n
c
e
i
t
o
s
b
á
s
i

c
o
s
e
m
o
t
i
v
a
ç
ã
o
p
r
o
c
e
s
s
o
d
e
c
o
n
s
t
r
u
ç
ã
o
d
e
s
o
f
t
w
a
r
e
a
p
l
i
c
a
ç
õ
e
s
o
d
e
c
o
n
s
t
r
u
ç
ã
o
d
e
s
o
f
t

war Missão no exterior
Missão no exterior
Processos de construção de softwares

Week Requirements
Resolução de Exercícios
Prova Final

hopper requirements - kick off do a ball of fire

adesoftwareArquiteturadesoftware
L
s
s
P
a
d
r
õ
e
s
d
e
p
r
o
p
r
e
s
e
n
t
a
õ
d
e
u
ç
ã
o
d
e
e
x
e

r
c
i
c
i
o
s
P
r
o
v
e
n
i
e
n
c
i
a
s
P
a
d
r
õ
e
s
d
e
p
r
o
p
r
i
e
t
a
r
i
a
s
P
a
d
r
õ
e
s
d
e
p
r
o
p
r
i
e
t
a
r
i
a
s
A
n
t
e
s
e
n
t
a
ç
ã
o
d
e
t
r
a
b
a
l
h

o s A p r e s e n t a ç ã o d e t r a b a l h o s P r o v a s E n t r e g a d e r e s u l t a d o s C a r g a h o r á r i a p r e

S
e
n
c
i
a
l
t
o
t
a
l
:

A carga horária de atividades extraclasse será distribuída ao longo das semanas letivas previstas no calendário acadêmico da UFU, conforme cronograma de atividades a seguir.

C
a
r
g
a
H
o
r
á
r
i
a
(
h
o
c
o
n
t
e
n
t
e
d
o
V
i
s
t
a
d
a
P
r
o
f
e
s
s
o
r
a
P
r
o
v
a

B) Atendimento ao discente

O atendimento aos alunos ocorrerá semanalmente às segundas-feiras, entre 15:00 e 16:00. É necessário agendamento prévio por e-mail, fabiola.pereira@ufu.br, ou pelo chat do MS Teams. O atendimento será realizado na sala da professora, 1B132.

7. AVALIAÇÃO

A) Atividades avaliativas

A avaliação será feita por meio de provas individuais e dissertativas sobre o tema, a serem realizadas de forma presencial em sala de aula, com duração de 1h40min cada. Serão 2 provas ao todo (P1 e P2), sendo cada uma no valor de 30 pontos. Também haverá um trabalho final no valor de 40 pontos, a ser entregue e apresentado conforme especificação a ser apresentada aos discentes. A nota final será dada pelo somatório das provas P1 e P2 com o trabalho final.

B) Atividade avaliativa de recuperação

De acordo com o Art. 141 das Normas de Graduação (Res. CONDIR Nº 46/2022), haverá uma avaliação de recuperação de aprendizagem, que terá valor de 60 pontos. A mesma tem caráter substitutivo, ou seja, a pontuação obtida pelo aluno na avaliação de recuperação substituirá as notas das provas P1 e P2, caso seja maior que a soma das mesmas. A prova substitutiva abrangerá todo o conteúdo visto no semestre. Ainda, de acordo com o Art. 141, somente fará jus ao direito de realizar a avaliação de recuperação substitutiva o(a) discente que não obtiver o rendimento mínimo de aprovação (60 pontos) e que possuir no mínimo 75% de frequência na disciplina.

C) Controle de frequência

A assiduidade será computada através da chamada em sala durante as aulas, em um horário aleatório após 10 minutos do início de cada encontro diário. A professora poderá adotar, a seu critério, caso haja demasiada desistência de continuidade em sala, uma segunda chamada ao final do segundo horário de aula.

D) Cronograma das atividades avaliativas

Nro	Data	Hora	Descrição	Pontos
1	27/05/2026	19:00 – 20:40	Prova teórica 1	30
2	10/07/2026	20:50 – 22:30	Prova teórica 2	30
3	Entregas no decorrer do semestre	-	Trabalho final	40
Recuperação	29/07/2026	19:00 – 20:40	Prova de Recuperação	60
TOTAL:				100

8. BIBLIOGRAFIA

Básica

- 1) IAN, S. Engenharia de software. 9a ed. São Paulo: Prentice Hall, 2011.
- 2) PAULA FILHO, W. P. Engenharia de software: fundamentos, métodos e padrões. 3a ed. Rio de Janeiro: LTC, 2009.
- 3) PRESSMAN, R. S. Engenharia de software: uma abordagem profissional. 7a ed. Porto Alegre: McGraw-Hill, 2011.

Complementar

- 1) Marco Tulio Valente. Engenharia de Software Moderna: Princípios e Práticas para Desenvolvimento de Software com Produtividade, Editora: Independente, 2020.
- 2) FOWLER, M. UML essencial: um breve guia para a linguagem-padrão de modelagem de objetos. 3a ed. Porto Alegre: Bookman; 2005.

- 3) GAMMA, E. et al. Padrões de projeto: soluções reutilizáveis de software orientado a objetos. Porto Alegre: Bookman, 2000.
- 4) HORSTMANN, C. Padrões e projeto orientados a objetos. Porto Alegre: Bookman, 2007.
- 5) LIMA, A. S. UML 2.0: do requisito à solução. 4a ed. São Paulo: Érica; 2010.
- 6) MEDEIROS, E. Desenvolvendo software com UML 2.0: definitivo. São Paulo: Pearson Makron Books, 2004.

9. **APROVAÇÃO**

Aprovado em reunião do Colegiado realizada em: ____/____/____

Coordenação do Curso de Graduação: _____