



Universidade Federal de Goiás  
Instituto de Informática  
Engenharia de Software

Matriz Curricular: ENGSO-BN-2 - 2017.1  
Plano de Disciplina  
Ano Letivo: 2025 - 2º Semestre

#### Dados da Disciplina

<b>Código:</b> INF0483	<b>Nome:</b> SOFTWARE PARA SISTEMAS UBÍQUOS	<b>CH Teórica:</b> 32 h
<b>Turma:</b> A	<b>Prof:</b> Fabio Nogueira de Lucena	<b>CH Prática:</b> 32 h

#### Ementa

Sistemas de informação que fazem uso de dispositivos (ubíquos) (16h): smartphones, sensores, internet das coisas (IoT), stream analytics e aspectos de segurança (vulnerabilidades, criptografia, certificados digitais). Definição de arquiteturas para soluções móveis (16): conectar serviços, possivelmente de grande volume, fluxo e em tempo real, com a necessidade de analisá-los. Desenvolvimento de código para smartphone, sensor ou outro dispositivo capaz de alimentar/receber informações de sistema de informação (32h).

#### Objetivo Geral

Capacitar o estudante a compreender e aplicar os princípios e conceitos fundamentais do paradigma de computação ubíqua e pervasiva e suas aplicações em diferentes domínios e tipos de sistema, assim como as tecnologias, métodos e ferramentas utilizadas para construir aplicações e sistemas nesse contexto. O aluno deve ser capaz de reconhecer, em situações reais, os problemas da computação ubíqua e pervasiva, bem como as oportunidades para aplicar os conceitos estudados no curso.

#### Objetivos Específicos

Ao final desta disciplina, os alunos deverão ser capazes de:

- Compreender e caracterizar o paradigma de computação ubíqua e pervasiva e seu uso como parte de aplicações e sistemas de informação.
- Compreender e caracterizar os conceitos fundamentais envolvidos na construção de software para sistemas ubíquos: redes de sensores internet das coisas (IoT) e sistemas ciberfísicos; ambientes inteligentes (smart spaces); ciência de contexto; processamento de streams de dados; computação móvel; computação no contínuo da Internet (nuvem, névoa e borda); ambientes virtuais e metaverso.
- Compreender e avaliar o panorama geral de plataformas, middleware, frameworks e ferramentas utilizadas no desenvolvimento de software para sistemas ubíquos, desenvolvendo a capacidade de aplicação prática em cenários representativos.
- Compreender os principais aspectos não-funcionais de software para sistemas ubíquos, notadamente: mobilidade, persistência, segurança, desempenho e privacidade.

#### Relação com Outras Disciplinas

Sistemas ubíquos são essencialmente sistemas distribuídos, com forte aplicação dos conceitos estudados na disciplina de Software Concorrente e Distribuído. Por tratar da construção de software, são também necessários conhecimentos adquiridos nas disciplinas relacionadas, notadamente Projeto de Software, Construção de Software e Arquitetura de Software. Finalmente, as competências desenvolvidas nas disciplinas relacionadas com programação (Introdução a Programação, Algoritmos e Estruturas de Dados 1 e 2, e Programação Orientada a Objetos) são também importantes.

O uso desta disciplina se dará, principalmente, na disciplina de Prática em Engenharia de Software, onde o aluno poderá incorporar os conhecimentos adquiridos nos projetos práticos que desenvolver.

#### Programa

UNIDADE 1 – Introdução e Fundamentos (22h)

1 – Introdução: definições básicas, exemplos, objetivos de projeto e classificação de sistemas de computação ubíqua e pervasiva (04h).

2 - Princípios de arquitetura de software para sistemas de computação ubíqua. (02h)

3 – Conceitos fundamentais: computação móvel; redes de sensores; IoT; smart spaces; ciência de contexto; processamento de streams; computação no contínuo nuvem-névoa-borda; gêmeos digitais; metaverso. (16h, inclui avaliação escrita)

UNIDADE 2 – Construção de software para sistemas ubíquos (42h)

4 – Princípios do desenvolvimento de software para sistemas ubíquos: middleware, plataformas e frameworks.



(02h)  
 5 – Arquiteturas de comunicação e integração para sistemas ubíquos: cliente-servidor; publish-subscribe; messaging; streaming. (08h)  
 6 – Construção de aplicações para sistemas ubíquos usando dispositivos móveis: adaptação da interface de usuário entre dispositivos heterogêneos; offloading de funções no contínuo da computação; mobile cloud/edge computing. (12h)  
 7 – Construção de software para sistemas ubíquos: projeto e implementação de sistemas ciberfísicos; processamento de streams em tempo real; ciência de contexto; projeto e implementação de gêmeos digitais; construção de aplicações ubíquas; segurança e privacidade. (20h, inclui acompanhamento do trabalho e avaliação escrita)

#### Procedimentos Didáticos

Legenda	Descrição	Objetivo
AEX	Aula teórica	Transmitir conhecimento utilizando quadro ou slides.
AP	Aula prática	Proporcionar ao aluno a aplicação prática do conteúdo ministrado em aula teórica.
ED	Estudo dirigido	Desenvolver a capacidade analítica, capacidade de síntese, de avaliação crítica e de análise.
OTR	Outros	Transmitir conhecimento utilizando quadro ou slides.
RE	AEX com resolução de	Desenvolver o raciocínio lógico, criatividade e capacidade de abstração e a capacidade de identificar, analisar e projetar soluções de
SE	Seminários	Desenvolver o raciocínio lógico, criatividade, capacidade de abstração, capacidade para identificar, analisar, projetar soluções de problemas, a capacidade de comunicação oral e a capacidade de trabalhar em grupo.
TG	Trabalho em grupo	Desenvolver a capacidade de comunicação oral e escrita. Capacidade de trabalhar em grupo.

#### Conteúdo Programático / Cronograma

Início	Proc.	Tópico	# Aul.
15/08/25	AEX	Apresentação da disciplina e do Plano de Ensino. Criação de conta no github. Criação de grupos. Criação do projeto do grupo em uma das contas de um dos estudantes do grupo. Acrescentar demais membros como colaboradores.	1
15/08/25	AEX, ED	Introdução: definições básicas, exemplos, objetivos de projeto e classificação de sistemas de computação ubíqua e pervasiva. Atividade supervisionada: Resumo sobre os princípios da computação ubíqua. Apresentação do trabalho prático (Sistema de Monitoramento de Hemogramas). Atividade supervisionada: resolução de exercícios.	3
22/08/25	AEX, RE, AP, ED	Conceitos fundamentais: redes de sensores; IoT; smart spaces; ciência de contexto; processamento de streams; computação no contínuo nuvem-névoa-borda; gêmeos digitais; metaverso e Web 3.0. Atividades supervisionadas: Estudo de capítulos de livros, artigos e documentação técnica sobre os conceitos e ferramentas estudadas. Compreensão e início do trabalho prático. Marco 1. Atividade supervisionada: resolução de exercícios.	14
12/09/25	AEX, RE, ED	Princípios do desenvolvimento de software para sistemas ubíquos: middleware, plataformas e frameworks. Atividade supervisionada: Estudo das plataformas, middleware e frameworks utilizados na disciplina (Android, Apache Kafka, MQTT, gRPC). Marco 2. Atividade supervisionada: resolução de exercícios.	2
19/09/25	AEX, RE, AP, TG	Arquiteturas de comunicação e integração para sistemas ubíquos: cliente-servidor; publish-subscribe; messaging; streaming. Atividade supervisionada: Atividade prática no ambiente do AWS / Google Cloud (configuração, implantação e operação de aplicações na nuvem). Marco 3. Atividade supervisionada: resolução de exercícios.	8
03/10/25	AEX, RE, AP, TG	Construção de software para sistemas ubíquos: projeto e implementação de sistemas ciberfísicos; processamento de streams em tempo real; ciência de contexto; projeto e implementação de gêmeos digitais; construção de aplicações ubíquas; segurança e privacidade. Atividade supervisionada: resolução de exercícios. Marco 4. Atividade supervisionada: resolução de exercícios.	10
17/10/25	AEX, RE, AP, TG	Desenvolvimento de software do trabalho prático. Atividade supervisionada: resolução de exercícios.	12
07/11/25	AEX	Princípios de arquitetura de software para sistemas de computação ubíqua. Atividade supervisionada: resolução de exercícios.	2
07/11/25	OTR	Revisão do trabalho prático. Atividade supervisionada: resolução de exercícios.	2
14/11/25	TG	Apresentação do Trabalho Prático.	8
28/11/25	OTR	Últimos ajustes no trabalho prático.	2
Total			64



## **Critério de Avaliação**

A avaliação da disciplina será baseada na metodologia de aula invertida, onde os estudantes conduzem o desenvolvimento do trabalho prático e o professor atua como mediador. A nota final será composta por:

### **### 1. Desenvolvimento Incremental do Sistema (60%)**

#### **\*\*1.1 Entrega dos Marcos Técnicos (40%)\*\***

- **\*\*Marco 1 - Recepção FHIR (10%)\*\***: Implementação funcional do receptor de mensagens FHIR via subscription, com parsing correto dos recursos Observation
- **\*\*Marco 2 - Análise Individual (10%)\*\***: Componente de análise individual de hemogramas com detecção de desvios nos parâmetros hematológicos
- **\*\*Marco 3 - Base Consolidada (10%)\*\***: Sistema de armazenamento local operacional com persistência dos hemogramas recebidos
- **\*\*Marco 4 - Análise Coletiva (10%)\*\***: Implementação da detecção de padrões coletivos em janelas deslizantes com os indicadores especificados

#### **\*\*1.2 Integração e Funcionalidades Avançadas (20%)\*\***

- **\*\*API REST (5%)\*\***: Endpoints funcionais para consulta de alertas com documentação adequada
- **\*\*Aplicativo Móvel (10%)\*\***: App Android funcional com recebimento de notificações e interface para consulta de alertas
- **\*\*Testes e Qualidade (5%)\*\***: Cobertura de testes automatizados e qualidade do código (segundo boas práticas)

### **### 2. Processo de Desenvolvimento e Colaboração (25%)**

#### **\*\*2.1 Gestão de Projeto (10%)\*\***

- Uso adequado de controle de versão (Git) com commits organizados e mensagens descritivas
- Organização de backlog e sprints com divisão clara de responsabilidades entre membros da equipe
- Documentação técnica atualizada (README, arquitetura, APIs)

#### **\*\*2.2 Apresentações e Demonstrações (15%)\*\***

- **\*\*Apresentações Intermediárias (10%)\*\***: Demonstrações funcionais nos marcos 1, 2, 3 e 4 (2,5% cada)
- **\*\*Apresentação Final (5%)\*\***: Demonstração completa do sistema integrado com cenários de uso realistas

### **### 3. Competências Técnicas e Conceituais (15%)**

#### **\*\*3.1 Aplicação de Conceitos da Disciplina (10%)\*\***

- Uso adequado de padrões de sistemas ubíquos: publish-subscribe, streaming, processamento em tempo real
- Implementação de aspectos não-funcionais: segurança (HTTPS, mTLS), performance, escalabilidade
- Aplicação correta do padrão HL7 FHIR e interoperabilidade

#### **\*\*3.2 Inovação e Solução de Problemas (5%)\*\***

- Capacidade de resolver problemas técnicos encontrados durante o desenvolvimento
- Implementação de melhorias ou funcionalidades adicionais relevantes ao contexto de saúde pública

### **### Critérios de Aprovação**

- **\*\*Nota mínima\*\***: 6,0 (sessenta por cento)
- **\*\*Frequência mínima\*\***: 75%
- **\*\*Entrega obrigatória\*\***: Todos os 4 marcos técnicos devem ser entregues (mesmo que parcialmente funcionais)
- **\*\*Trabalho em equipe\*\***: Equipes de 3-5 estudantes, com avaliação individual baseada na contribuição identificada no Git

## **Critérios Regimentais e/ou Observações INF/UFG**

- (a) Estará aprovado o(a) aluno(a) que atingir nota final (NF) igual ou superior a 6,0 (seis pontos) e frequência às atividades acadêmicas igual ou superior a 75% da carga horária da disciplina, sendo que a chamada será feita em todas as aulas, podendo acontecer a qualquer momento a critério do(a) docente;
- (b) O período para solicitar avaliação de segunda chamada é de até 07 (sete) dias após a data da aplicação da atividade avaliativa. O pedido de segunda chamada deverá ser protocolado conforme condições estipuladas na Resolução CONSUNI específica (RGCG) em vigor;
- (c) O período para solicitar revisão de nota é de até 07 (sete) dias após a data da entrega da nota ao(a) aluno(a);
- (d) Será atribuída a nota 0,0 (zero) a qualquer atividade ou trabalho acadêmico não realizado ou não entregue na data estipulada. Casos excepcionais serão tratados em comum acordo com o(a) docente;
- (e) Todas as atividades são supervisionadas. As atividades supervisionadas referem-se às atividades práticas e devem ser desenvolvidas segundo Resolução CNE/CES 03/2007 de 2 de julho de 2007, a qual considera que curso do período diurno divide a cada hora de atividade acadêmica em 50 minutos de preleções e/ou aulas expositivas e 10 minutos de atividades práticas supervisionadas que podem ser realizadas a distância ou não, mas com supervisão do professor. Para os cursos do período noturno a proporção entre as atividades é de 45 minutos e 15 minutos, respectivamente;
- (f) Os(As) alunos(as) que se envolverem em plágio, desvios de conduta [seja como facilitador(a) ou como beneficiário(a)] receberão nota 0 (zero) para a atividade correspondente. O caso poderá ser levado ao conhecimento da Coordenação e/ou Núcleo Docente Estruturante (NDE) do Curso, do Conselho Diretor do INF/UFG para as providências administrativas e legais cabíveis.
- (j) É obrigatório o uso de e-mail institucional (@discente.ufg.br) em comunicações relacionadas à disciplina,



## Horários de atendimento docente

## Local de Divulgação dos Resultados das Avaliações

SIGAA

## Data da Realização das Provas

Semana	Atividade	Peso
4	Marco 1 - Apresentação + Código	10%
8	Marco 2 - Apresentação + Código	10%
12	Marco 3 - Apresentação + Código	10%
15	Marco 4 - Apresentação + Código	10%
16	Entrega API REST + Testes	5%
17	Entrega App Móvel	10%
18	Apresentação Final + Documentação	20%
Contínua	Gestão de Projeto (Git, Sprints)	10%
Contínua	Qualidade de Código e Testes	5%
Contínua	Aplicação de Conceitos	10%

## Bibliografia Básica

MCGRAW, G. Software Security: Build Security In. Addison-Wesley, 2006.  
GREENGARD, S. The Internet of Things. The MIT Press, 2015.  
WILMSHURST, T. An introduction to the design of small-scale embedded systems. Palgrave, 2001. ISBN 0333929942.

## Bibliografia Complementar

MANICO, J.; DETLEFSEN, A. Iron-Clad Java: Building Secure Web Applications. McGraw-Hill, 2014.  
HOWARD, M. 19 deadly sins of software security programming flaws and how to fix them. McGraw-Hill/Osborne, 2005. ISBN 0072260858.  
MARK, D. Dominando o desenvolvimento no iPhone: explorando o SDK do iPhone. Alta Books, 2009. ISBN 97885760833757.  
VAHID, F. Embedded system design: a unified hardware/software introduction. John Wiley & Sons, 2002. ISBN 0471386782.  
KERN, C.; KESAVAN, A.; DASWANI, N. Foundations of security: what every programmer needs to know. Apress, 2007. ISBN 9781590597842.

## Bibliografia Sugerida

BUYA, Rajkumar; DASTJERDI, Amir Vahid. Internet of Things: Principles and Paradigms. Morgan Kaufmann, 2016.  
CIRANI, Simone; FERRARI, Gianluigi; PICONE, Marco; VELTRI, Luca. Internet of Things: Architectures, Protocols and Standards. Wiley, 2018.  
HUANG, Dijiang; WU, Huijun. Mobile Cloud Computing: Foundations and Service Models, Morgan Kaufmann, 2018.  
DOBRE, Ciprian; XHAFIA, Fatos (Eds.). Pervasive Computing: Next Generation Platforms for Intelligent Data Collection. Elsevier, 2016.  
DE, Debashis. Mobile Cloud Computing: Architectures, Algorithms and Applications. CRC Press, 2016.  
COLINA, Antonio L.; VIVES, Alvaro; ZENNARO, Marco; BEGULA, Antoine; PIETROSEMOLI, Ermanno. Internet of Things in 5 days. 2016 (Creative Commons)

Plataformas para dispositivos móveis e IoT:  
Documentation for app developers (Android): <https://developer.android.com/docs>  
MQTT: <https://mqtt.org>  
CoAP (Constrained Application Protocol): <https://coap.technology>  
Apache Kafka: <https://kafka.apache.org>  
gRPC: <https://grpc.io>  
Google Cloud: <https://cloud.google.com/docs/get-started>  
Amazon Web Services documentation: <https://docs.aws.amazon.com/>

Leitura complementar: Artigos e relatórios técnicos sugeridos pelo professor (serão postados na Plataforma Turing).

Termo de Entrega	Termo de Aprovação
<p>Apresentado à Coordenação em</p> <p>Goiânia ____ de ____ de 20____.</p> <p>_____  Prof(a) Fabio Nogueira de Lucena  Professor</p>	<p>Aprovado em Reunião de CD no dia</p> <p>Goiânia ____ de ____ de 20____.</p> <p>_____  Prof. Dr. Sérgio Teixeira de Carvalho  Diretor do Instituto de Informática</p>
<p align="center"><b>Termo de Homologação</b></p>	
<p align="center">Data de Expedição: Goiânia, ____ de ____ de ____.</p>	

