

Plano Analítico: Plataformas para Aplicações Distribuídas na Internet

1. Identificação da Unidade Curricular

- **Curso:** Engenharia de Redes e Telecomunicações (ERT)
- **Ano:** 4^o | **Semestre:** 2^o
- **Créditos:** 6.0 UC
- **Carga Horária Total:** 90 Horas
- **Distribuição:**
 - **Teóricas (T):** 20h
 - **Teórico-Práticas (TP):** 20h
 - **Práticas (P):** 20h
 - **Trabalho Autónomo (TA):** 22h
 - **Orientação e Tutoria (OT):** 4h
 - **Avaliação (AV):** 4h

1. Fundamentação

As aplicações modernas não residem num único servidor; elas são distribuídas por múltiplos nós e datacenters. Para um engenheiro de ERT, compreender estas plataformas é essencial para projetar serviços de nuvem (Cloud), sistemas de streaming e aplicações de IoT. A disciplina foca em arquiteturas de microsserviços, protocolos de comunicação web (REST/gRPC) e a gestão de estado em ambientes distribuídos, garantindo alta disponibilidade e escalabilidade.

2. Objectivos Instrutivos e Educativos

- **Instrutivos:** Compreender os princípios de transparência e escalabilidade em sistemas distribuídos; dominar arquiteturas de software para a Web (MVC, Microsserviços); aprender sobre middleware e comunicação entre processos (RPC, Message Queues); estudar o ecossistema de Cloud Computing.
- **Educativos:** Fomentar a visão crítica sobre o compromisso entre consistência e disponibilidade (Teorema CAP); desenvolver autonomia na utilização de novas frameworks e promover a eficiência no consumo de recursos de rede por aplicações.

3. Resultado de Aprendizagem

O estudante será capaz de:

- Desenvolver e implementar APIs robustas utilizando o padrão RESTful.
- Projetar arquiteturas baseadas em microsserviços e contentores (**Docker**).
- Utilizar sistemas de mensageria (ex: RabbitMQ ou Kafka) para comunicação assíncrona.

- Implementar mecanismos de tolerância a falhas e balanceamento de carga.
- Configurar e utilizar serviços de armazenamento distribuído e bases de dados NoSQL.

4. Planejamento Temático (6 UC)

Tema	Horas (T+TP+P)	Conteúdo Programático
I. Arquiteturas Web e Cloud	10h	Evolução da Web (Client-Server, N-Tier); Cloud Computing (SaaS, PaaS, IaaS); Escalabilidade horizontal vs vertical.
II. Comunicação Distribuída	12h	Protocolos HTTP/HTTPS; REST vs gRPC; Serialização de dados (JSON, Protobuf); Invocação Remota de Métodos (RMI/RPC).
III. Microsserviços e Contentores	14h	Design de Microsserviços; Virtualização ao nível do SO (Docker); Orquestração básica (Introdução ao Kubernetes).
IV. Gestão de Dados e Estado	12h	Transações distribuídas; Teorema CAP; Bases de dados NoSQL (MongoDB, Redis); Consistência eventual.
V. Mensageria e Eventos	12h	Arquiteturas orientadas a eventos; Filas de mensagens; Publish-Subscribe; Middleware para sistemas distribuídos.

5. Recomendações Metodológicas

- **Aulas Práticas (20h):** Desenvolvimento de um projeto prático utilizando frameworks modernas (ex: **Node.js/Express, Spring Boot ou Python/FastAPI**).
- **Hands-on Cloud:** Utilização de instâncias gratuitas em provedores de nuvem (AWS, Google Cloud ou Azure) para o deploy das aplicações desenvolvidas.
- **Projetos com Docker:** Implementação de uma aplicação multi-contentor para compreender a rede interna e o isolamento de serviços.

6. Sistema de Avaliação

Conforme a alocação de **4h para AV**:

- **Avaliação Contínua (60%):** Desenvolvimento de um sistema distribuído funcional (40%) e laboratórios de integração de APIs (20%).
- **Avaliação Formal (40%):** Exame escrito focado nos conceitos teóricos de distribuição, consistência e protocolos.

7. Bibliografia Principal Indicada

1. **TANENBAUM, Andrew S. & VAN STEEN, Maarten.** *Distributed Systems: Principles and Paradigms.*
2. **NEWMAN, Sam.** *Building Microservices: Designing Fine-Grained Systems.* O'Reilly.
3. **KLEPPMANN, Martin.** *Designing Data-Intensive Applications.* O'Reilly.