

**Nome da disciplina (em Português):**

Laboratório de Sistemas Computacionais Complexos

**Nome da disciplina (em Inglês):**

*Laboratory of Complex Computational Systems*

**Objetivo (em Português):** Capacitar os estudantes para construir sistemas complexos e distribuídos, utilizando metodologias de desenvolvimento e tecnologias web / mobile atuais.

**Objetivo (em Inglês):** Train the students to build complex and distributed systems using current development methods and web / mobile technologies.

**Ementa / Programa resumido (em Português):**

Métodos ágeis, DevOps, sistemas distribuídos, padrões arquiteturais, princípios arquiteturais, tecnologias web fundamentais, desenvolvimento para *front-end* e modelos de páginas, desenvolvimento para *back-end* e padrões arquiteturas, autenticação, infraestrutura, banco de dados e persistência, plataformas de entrega.

**Ementa / Programa resumido (em Inglês):**

Agile methodologies, DevOps, distributed systems, architectural patterns, architectural principles, fundamental web technologies, front-end development and page models, back-end development and architectural patterns, authentication and authorization, databases, infrastructure, deployment platforms.

**Programa (em Português):**

Métodos ágeis (valores, princípios e práticas), DevOps (entrega contínua, versionamento, trunk-based development, automação de testes), sistemas distribuídos (teorema CAP, consistência forte e eventual, níveis de disponibilidade), padrões arquiteturais (camadas, microkernel, microsserviços, baseada em eventos, nuvem), princípios arquiteturais (aplicações com 12 fatores, sistemas reativos, sistemas nativos à nuvem), tecnologias web fundamentais (HTML, CSS, JavaScript), modelo requisição-resposta (ad hoc, REST, GraphQL), desenvolvimento para front-end e modelos de páginas (MPA, SPA, SSR, PWA), desenvolvimento para back-end e padrões arquiteturais (MVC, MVP, MVVM, JAM), autenticação (identificadores sequenciais, UUID, tokens, JWT) e autorização (protocolos de controle de acesso), banco de dados (relacionais, não relacionais, messageiros), infraestrutura (virtualização, containerização, orquestração e coreografia, malha de serviços) e plataformas de entrega (IaaS, PaaS, CDN, serverless).

**Programa (em Inglês):**

Agile methodologies (values, principles and practices), DevOps (continuous delivery, versioning, trunk-based development, test automation), distributed systems (CAP theorem, strong and eventual consistency, availability levels), architectural patterns (layered, microkernel, microservices, event-driven, cloud), architectural principles (12-factor apps, reactive systems, cloud-native systems), fundamental web technologies (HTML, CSS, JavaScript), request-response model (ad hoc, REST, GraphQL), front-end development and page models (MPA, SPA, SSR, PWA), back-end development and architectural patterns (MVC,

MVP, MVVM, JAM), authentication (sequence identifiers, UUID, tokens, JWT) and authorization (protocols for access control), databases (relational model, non-relational model), infrastruttura (virtualization, containerization, orchestration and choreography, service mesh) and deployment platforms (IaaS, PaaS, CDN, serverless).

## **Pré-requisitos:**

Ciência da Computação IME-USP:

- [MAC0350 \(Introdução ao Desenvolvimento de Sistemas de Software\)](#)
- **ou** [MAC0218 \(Técnicas de Programação II\)](#)
- **ou** [MAC0472 \(Laboratório de Métodos Ágeis\)](#)

Ciência da Computação ICMC-USP:

- [SSC0130 \(Engenharia de Software\)](#)

Ciência da Computação FFCLRP-USP:

- [5954015 \(Base de Dados II\)](#)
- **e** [5954017 \(Análise e Projeto de Software\)](#)

Sistemas da Informação ICMC-USP:

- [SSC0527 \(Engenharia de Software\)](#)

Sistemas da Informação EACH-USP:

- [ACH2006 \(Engenharia de Sistemas de Informação I\)](#)

Engenharia da Computação EP-USP:

- [PCS3213 \(Engenharia de Software\)](#)
- **ou** [PCS3413 \(Engenharia de Software e Banco de Dados\)](#)

Engenharia Mecatrônica EP-USP:

- [PMR3201 \(Computação para Automação\)](#)
- **e** [PMR3304 \(Sistemas de Informação\)](#)

Engenharia da Computação ICMC-USP / EESC-USP:

- [SSC0620 \(Engenharia de Software\)](#)

**Créditos aula:** 4

**Créditos trabalho:** 2

**Avaliação:**

- **Método:**

O método que será adotado nesta disciplina consiste em aulas teóricas / expositivas com aplicação prática dos conceitos abordados e a utilização de laboratório de informática para a realização de desenvolvimento em grupo ao longo do semestre letivo.

- **Critério:**

Média ponderada de notas em presença, comprometimento ao método, empenho, autoavaliação e qualidade do software desenvolvido.

- **Norma de Recuperação:**

Alunos que ficarem com média final entre 3,0 e 4,9 ficarão de recuperação e lhes será passado um trabalho adicional de implementação de software que será avaliado pelo docente responsável no período da recuperação.

**Docente Responsável:** Alfredo Goldman

**Cursos para os quais será oferecida e obrigatoriedade para cada curso:** Optativa Eletiva - Bacharelado em Ciência da Computação e Bacharelado em Sistemas da Informação

**Justificativa** (da criação/alteração. NO CASO DE ALTERAÇÃO, JUSTIFIQUE CLARAMENTE CADA ITEM ACIMA QUE TIVER SIDO ALTERADO):

Na última década, houve um grande aumento no número de usuários utilizando serviços disponíveis online. Como consequência, certas aplicações web e mobile passaram a ter de milhões a bilhões de acessos diariamente. Essa dimensão sem precedentes gerou uma série de desafios para a criação, expansão e manutenção de sistemas modernos, que são desenvolvidos por múltiplos times, possuem domínios de negócio complexos e são provisionados em infraestruturas distribuídas para atender seus requisitos de escalabilidade. Essa disciplina abordará os desafios de construir tais sistemas em três dimensões: arquitetura, implementação e operações. O curso começará com aulas teóricas que apresentarão as tecnologias e metodologias essenciais para o desenvolvimento de um sistema complexo que será proposto para a turma. Para começar, os alunos serão divididos em times que farão, cada um, a implementação dos serviços que irão compor esse sistema. Os estudantes receberão uma especificação com arquitetura de componentes, modelagem de dados, modelagem de processos e esboço das telas da aplicação. No restante do curso, aplicarão práticas de métodos ágeis e DevOps para coordenar o trabalho dentro das equipes e entre os diferentes times, com o objetivo de produzir um MVP (Produto Mínimo Viável, do inglês *Minimum Viable Product*) do sistema. Essa disciplina é baseada na experiência do curso de extensão homônimo “Desenvolvimento de Sistemas Complexos”, ministrada pelos alunos de pós-graduação Renato Cordeiro Ferreira e Thatiane de Oliveira Rosa e realizada durante as férias de verão e inverno de 2018 e 2019 como parte das atividades do grupo de extensão CodeLab.

### Bibliografia:

1. BONÉR, Jonas. **Reactive Microsystems: The Evolution of Microservices at Scale.** . Sebastopol: O'Reilly Media, Inc, 2017.
2. GARRISON, Justin, NOVA, Kris. **Cloud Native Infrastructure: Patterns for Scalable Infrastructure and Applications in a Dynamic Environment.** 1. ed. Sebastopol: O'Reilly Media, Inc, 2017.
3. NEWMAN, Sam. **Building Microservices: Designing Fine-Grained Systems.** Sebastopol: O'Reilly Media, Inc, 2015.
4. SADALAGE, Pramod J., FOWLER, Martin. **NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence.** Upper Saddle River, NJ: Addison-Wesley, 2013.

### Bibliografia complementar:

1. BONÉR, Jonas. **Reactive Microservices Architecture: Design Principles for Distributed Systems.** Sebastopol: O'Reilly Media, Inc, 2016.
2. ECKSTEIN, Jutta. **Agile Software Development in the Large: Diving Into the Deep.** New York: Dorset House Publishing, 2004.
3. EVANS, Eric. **Domain-Driven Design: Tackling Complexity in the Heart of Software.** 1. ed. Boston: Addison-Wesley Professional, 2003.
4. FORSGREN, Nicole, HUMBLE, Jez, KIM, Gene. **Accelerate: The Science of Lean Software and Devops: Building and Scaling High Performing Technology Organizations.** Revolution Press, 2018.
5. GARBAJOSA, Juan, WANG, Xiaofeng, AGUIAR, Ademar. (Org.). **Agile Processes in Software Engineering and Extreme Programming.** XP 2018. Lecture Notes in Business Information Processing, vol 314. Springer, Cham.
6. GEERS, Michael. **Micro Frontends: Extending the Microservice idea to frontend development.** 2017. Disponível em: <<https://micro-frontends.org/>>.
7. HUMBLE, Jez, FARLEY, David. **Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation.** 1. ed. Boston: Addison-Wesley Professional, 2010.
8. KNIBERG, Henrik, IVARSSON, Anders. **Scaling Agile @ Spotify: With Tribes, Squads, Chapters & Guilds.** 2012. Disponível em: <[blog.crisp.se/wp-content/uploads/2012/11/SpotifyScaling.pdf](http://blog.crisp.se/wp-content/uploads/2012/11/SpotifyScaling.pdf)>
9. LEFFINGWELL, Dean. **Scaling Software Agility: Best Practices for Large Enterprises.** 1. ed. Boston: Pearson Education, 2007.
10. RICHARDS, Mark. **Software Architecture Patterns.** Sebastopol: O'Reilly Media, Inc, 2015.
11. RICHARDSON, Chris. **Microservice Patterns.** Manning Publications, 2017.
12. TRUNK BASED DEVELOPMENT. **Trunk Based Development.** Disponível em: <[trunkbaseddevelopment.com/](http://trunkbaseddevelopment.com/)>.
13. VERNON, Vaughn. **Implementing Domain-Driven Design.** Addison-Wesley Professional, 2013.
14. WAMPLER, Dean. **Fast Data Architectures for Streaming Applications.** Sebastopol: O'Reilly Media, Inc, 2016.
15. YANAGA, Edson. **Migrating to Microservice Databases: From Relational Monolith to Distributed Data.** Sebastopol: O'Reilly Media, Inc, 2017.