

Formação de especialistas em ENGENHARIA DE SOFTWARE

2026



ENGENHARIA DE SOFTWARE

MODALIDADE: EXTENSÃO UNIVERSITÁRIA





INSTITUTO DE COMPUTAÇÃO
MODALIDADE - EXTENSÃO - UNIVERSITÁRIA



FORMAÇÃO DE ESPECIALISTAS EM ENGENHARIA DE SOFTWARE

O curso de Formação de Especialistas em Engenharia de Software do Instituto de Computação da Unicamp visa capacitar profissionais a atuarem de maneira estratégica e inovadora, por meio de uma formação avançada que integra teoria e prática, promovendo a excelência na análise, projeto, desenvolvimento, gestão e avaliação de sistemas de software complexos.

Para quem?

Profissionais que queiram se capacitar para desenvolvimento, teste e gestão de projetos de software.

Pré-requisitos

Nível superior completo. É esperado que os interessados tenham conhecimento prático em lógica de programação de computadores.

Certificação

Aqueles alunos aprovados nas 13 disciplinas receberão certificado de conclusão de curso pela Escola de Extensão da Unicamp – EXTECAMP.

Formato do curso

Em geral, o curso tem a duração de 10 meses, sendo realizado de fevereiro a dezembro de 2026, com aulas aos sábados e 100% à distância. A carga horária total é de 360 horas, divididas em 13 disciplinas que cobrem tópicos essenciais da engenharia de software.



HABILIDADE DESENVOLVIDAS

- **Análise e Modelagem:** Criar arquiteturas de software robustas usando práticas avançadas de modelagem e padrões de projeto.
- **Programação Orientada a Objetos:** Projetar e implementar soluções orientadas a objetos com foco em modularidade, reutilização e escalabilidade.
- **Design e Avaliação de Interfaces:** Desenvolver interfaces centradas no usuário e avaliar a experiência de uso.
- **Testes e Qualidade de Software:** Implementar testes automatizados, verificação de código e práticas de validação em ambientes ágeis.
- **Engenharia de Dados:** Modelar e gerenciar dados usando bancos relacionais e não relacionais, aplicando técnicas avançadas de consulta e estruturação.
- **Requisitos e Especificação de Software:** Levantar e definir requisitos de software, aplicando métodos ágeis e técnicas de especificação.
- **Componentização e Reuso:** Desenvolver e integrar componentes de software para maior eficiência e manutenção.
- **Gerenciamento de Projetos:** Planejar e gerenciar projetos de software com técnicas tradicionais e ágeis, incluindo estimativa de prazos e controle de riscos.
- **Tecnologias Emergentes:** Familiarizar-se com IoT, Machine Learning, Big Data e DevOps, integrando essas tecnologias no desenvolvimento de soluções de software.



CORPO DOCENTE

- Docentes da Unicamp:
 - André Santanchè
 - Bruno Barbieri de Pontes Cafeo
 - Cecília Mary Fischer Rubira
 - Eliane Martins
 - Julio Cesar dos Reis
 - Luiz Eduardo Buzato
 - Sandro Rigo
- Docentes Especialistas de empresas e outras universidades:
 - Bruno Teixeira de Abreu (Sofist)
 - Rosana Teresinha Vaccare Braga (USP)
 - Matheus Silva Mota (IFood)
 - Rodrigo Bonacin (CTI Renato Archer)
 - Mario Mancuso (Sensedia)
 - Patrícia Contiero (Ci&T)

DISCIPLINAS

- INF-0318 - Análise Orientada a Objetos e projeto arquitetural
- INF-0319 - Projeto e implementação Orientados a Objetos
- INF-0320 - Interfaces Homem / Computador
- INF-0321 - Verificação e validação de Software
- INF-0323 - Linguagens e ambientes para programação de software
- INF-0325 - Modelagem e projeto de Banco de Dados
- INF-0329 - Prática em Engenharia de Software
- INF-0330 - Requisitos de Software e Modelos de especificação
- INF-0331 - Componentização e reuso de Software: conceitos e práticas
- INF-0332 - Arquitetura Orientada a Serviços - SOA & Webservices: Conceitos e práticas
- INF-0333 - Gerenciamento de Projetos de Software: Aspectos Econômicos e Planejamento
- INF-0334 - Tópicos em Engenharia de Software I
- INF-0335 - Ambientes para Concepção de Software



DISCIPLINAS

- **INF-0318 - Análise Orientada a Objetos e Projeto Arquitetural**
 - **Descrição:** Ao final da disciplina o aluno deve ter aprendido os conceitos fundamentais de orientação a objetos e deve ser capaz de, partindo da especificação de um problema, criar um diagrama de classes que represente uma solução para o problema. Também o aluno deve ser capaz de definir a arquitetura de software do sistema alvo, de acordo com os requisitos de qualidade priorizados durante o projeto arquitetural.
 - **Ementa:** Conceitos fundamentais da análise orientada a objetos para a estruturação e modelagem de sistemas através da construção de diagramas de classes. Diagramas dinâmicos da UML, como por exemplo, diagramas de sequência e de colaboração. Modelagem estática e modelagem dinâmica. Identificação de objetos e sua classificação em classes, especificação de atributos e operações, identificação dos relacionamentos de generalização/especialização, agregação e associação entre as classes. Noções de tipos abstratos de dados, encapsulamento, polimorfismo, herança simples e múltipla, classes abstratas, interfaces, pacotes, metaclasses, delegação, e padrões de projeto. Projeto arquitetural. Conceitos básicos de arquitetura de software, definição de componentes, conectores e configurações arquiteturais, atributos de qualidade associadas à arquitetura de software e como ela deve ser usada de desenvolvimento de sistemas centrados na arquitetura.



DISCIPLINAS

- **INF-0319 - Projeto e Implementação Orientados a Objetos**
 - **Descrição:** Ao final da disciplina o aluno deve ser capaz de traduzir uma especificação de um problema para um sistema orientado a objetos e também deve ter adquirido uma visão completa do ciclo de desenvolvimento de software orientado a objetos: análise, projeto e implementação. Esta visão inclui um estudo comparativo das principais linguagens orientadas a objetos existentes atualmente.
 - **Ementa:** Introdução: projeto orientado a objetos. Projeto: de sistema. Organização de um sistema em subsistemas, estruturação. Identificação de concorrência inerente ao problema. Identificação de problemas de redes, banco de dados e interface com o usuário. Projeto de objetos. Definição completa das classes. Projeto dos algoritmos para implementar as operações. Particionamento em classes e relacionamentos em módulos. Iteração, verificação e refinamento do modelo. Ambiente de implementação. Escolha do ambiente alvo para a implementação. Linguagens de programação orientadas a objetos. Ambientes de programação orientados a objetos. Estudo de casos.



DISCIPLINAS

- **INF-0320 - Interfaces Homem Computador**
 - **Descrição:** Desenvolver uma noção dos conceitos básicos e das principais práticas em Interação Humano-Computador (IHC) e Design da Interação. Entender a qualidade da interface de usuário como fator de sucesso de um sistema interativo. Conscientizar-se da importância da clarificação de problema (análise de usuário, tarefas, contexto organizacional) para projetos centrados no usuário. Conhecer um processo de projeto de interface de usuário e suas técnicas subjacentes. Ser capaz de construir e avaliar um protótipo tanto em papel quanto por meio de uma ferramenta computacional. Relacionar as atividades de design em IHC com atividades de um modelo iterativo e incremental de desenvolvimento de software. Desenvolver o raciocínio crítico para o projeto de sistemas de qualidade que façam sentido aos seus usuários e demais stakeholders.
 - **Ementa:** Meta a ser alcançada na construção de uma interface de usuário: design de um sistema computacional que faça sentido aos seus usuários. Conceitos Básicos em IHC: Interface, Interação, Affordance. Qualidade em IHC: Usabilidade, Acessibilidade, Comunicabilidade, Fatores humanos, Princípios do Design Centrado no Usuário, Princípios do Design Universal. A importância de um processo para a construção de interfaces de usuário. IHC e Engenharia de Requisitos. Artefatos e métodos para apoiar o design de sistemas interativos. Prototipação: em papel, por meio de ferramentas. Storyboard: uso de ferramentas, análise de tarefas, Avaliação em IHC: teoria e prática. Redesign de interfaces.



DISCIPLINAS

- **INF-0321 - Verificação e Validação de Software**
 - **Descrição:** Descrição: Ao final da disciplina espera-se que o aluno tenha adquirido consciência sobre a importância da Verificação e Validação para a qualidade do software que é produzido. Em especial, espera-se criar a conscientização de que testes são essenciais, mas não são a única forma de se garantir a qualidade de um sistema. Com relação aos testes, espera-se que o aluno tenha compreendido a importância dos mesmos, seu impacto nos custos de desenvolvimento do software e também que se trata de uma atividade a ser levada em conta desde cedo no desenvolvimento do software e que, como o desenvolvimento, o processo de testes também deve ser dividido em fases e deve ser documentado.
 - **Ementa:** Introdução: definição, importância da verificação e validação ao longo do ciclo de vida, classificação das técnicas. Revisões técnicas: passeio (walkthrough), inspeção do produto. Testes: fundamentos, os testes e o ciclo de vida. Técnicas de teste: caixa branca e caixa preta. Testes de unidade, de integração e de sistemas. Testes OO. Documentação de testes. Automação de testes. Metodologias de teste.



DISCIPLINAS

- **INF-0323 - Linguagens e Ambientes para Programação de Software**
 - **Descrição:** Ao final dessa disciplina o aluno dominará conceitos básicos de programação e um ambiente de projeto.
 - **Ementa:** Esta disciplina introduz conceitos básicos de programação orientada a objetos, como encapsulamento, tipos abstratos de dados, herança, classes abstratas, agregação, acoplamento dinâmico, polimorfismo e noções de interface, pacotes e tratamento de exceções. Para exemplificar os conceitos, a disciplina toma como base uma linguagem de programação moderna (Java), e um ambiente de projeto amplamente utilizado (Eclipse).
- **INF-0325 - Modelagem e Projeto de Banco de Dados**
 - **Descrição:** Ao final da disciplina o aluno deverá estar familiarizado com a Tecnologia de Banco de Dados de forma que possam desenvolver modelos de sistemas baseados nesta tecnologia.
 - **Ementa:** Conceitos básicos. Modelagem de dados (MER). Projeto lógico de dados (Modelo Relacional). Conceitos de normalização de dados relacionais. A Linguagem SQL - Teoria e Prática. Projeto e implementação de aplicações em Sistemas de Informação usando bancos de dados e Banco de dados objeto-relacional e persistência de objetos.



DISCIPLINAS

- **INF-0329 - Prática em Engenharia de Software**
 - **Descrição:** Ao final da disciplina o aluno terá praticado noções teóricas envolvidas na construção de módulos de software.
 - **Ementa:** Definição de um problema. Elaboração de uma proposta de desenvolvimento. Análise e projeto do sistema. Implementação e teste.
- **INF-0330 - Requisitos de Software e Modelos de especificação**
 - **Descrição:** Ao final da disciplina o aluno deve ter aprendido os conceitos fundamentais para especificação de requisitos de software.
 - **Ementa:** Introdução a engenharia de software. Paradigmas de desenvolvimento. Elicitação, especificação, documentação e validação de requisitos. Modelos para especificação de sistema de software.
- **INF-0331 - Componentização e Reuso de Software: Conceitos e Práticas**
 - **Descrição:** Ao final da disciplina o aluno deverá estar familiarizado com a Tecnologia de Banco de Dados de forma que possam desenvolver modelos de sistemas baseados nesta tecnologia.
 - **Ementa:** Conceitos básicos. Modelagem de dados (MER). Projeto lógico de dados (Modelo Relacional). Conceitos de normalização de dados relacionais. A Linguagem SQL - Teoria e Prática. Projeto e implementação de aplicações em Sistemas de Informação usando bancos de dados e Banco de dados objeto-relacional e persistência de objetos.



DISCIPLINAS

- **INF-0332 - Arquitetura Orientada a Serviços - SOA & WEBSERVICES: Conceitos e Práticas**
 - **Descrição:** O objetivo dessa disciplina é apresentar aos alunos a forma de construção de aplicações através da utilização de arquitetura orientada a serviços (SOA) e das principais tecnologias WebServices. Trata-se de uma abordagem que vem atraindo muita atenção do mercado pois permite a aplicação dos conceitos de componentização e reuso reduzindo custo e prazo de novos projetos de desenvolvimento. Os alunos serão capacitados nos principais conceitos, benefícios, técnicas, boas práticas, aplicações em estudo de casos e exercícios. O objetivo é que os alunos estejam capacitados a planejar a implantação do conceito SOA dentro de suas empresas e a projetar aplicações orientadas a serviços lançando mão de um amplo leque de tecnologias e produtos.
 - **Ementa:** Introdução SOA: motivadores, definições, benefícios e desafios. Principais características: provider, consumer & registry. Estudo de casos nacionais e internacionais. Identificação e modelagem de serviços. Principais tecnologias de web services: XML, WSDL, SOAP e UDDI. Interoperabilidade em web services, especificações emergentes e ferramentas. Governança SOA e métricas de reuso. Infraestrutura SOA (ESB, Registries and Repositories etc.). Roadmap de adoção nas empresas e tendências de mercado.



DISCIPLINAS

- **INF-0333 - Gerenciamento de Projetos de Software: Aspectos Econômicos e Planejamento**
 - **Descrição:** Ao final da disciplina os alunos deverão ser capazes de: elaborar medição do produto de software utilizando Pontos de Função e Pontos de Caso de Uso, elaborar estimativas de custo do produto baseando-se nas medidas realizadas e produzir um plano de desenvolvimento do produto de software baseado nas estimativas.
 - **Ementa:** Conceitos de Gerenciamento de Projetos de Software. Métricas de processo e de produto. Modelos de previsão de custo/duração. Planejamento: estudo de viabilidade, organização de projetos, análise de risco, acompanhamento e controle de projetos.
- **INF-0334 - Tópicos em Engenharia de Software I**
 - **Descrição:** Ao final dessa disciplina o aluno dominará conceitos básicos de engenharia de software.
 - **Ementa:** Temas atuais e emergentes em Engenharia de Software, envolvendo tanto tecnologias quanto metodologias. Temas atuais transversais à Engenharia de Software. Ambientes e ferramentas para Engenharia de Software.




DISCIPLINAS

- **INF-0335 - Ambientes para Concepção de Software**
 - **Descrição:** Ao final da disciplina o aluno deverá estar familiarizado com as principais ferramentas que auxiliam a concepção e desenvolvimento de software.
 - **Ementa:** Análise completa de um arcabouço de ferramentas usadas no processo de desenvolvimento e entrega de software. Aspectos de versionamento de software, como resolução de conflitos, mesclagem, ramificações, etc. Integração contínua. Acesso a banco de dados NoSQL.


CRONOGRAMA

DISCIPLINAS	INSCRIÇÕES		AULAS	
	Início	Fim	Início	Fim
MATRÍCULAS	16/12/2025	16/01/2026	-	-
INF-0323 (PARTE 1)	19/01/2026	30/01/2026	07/02/2026	28/02/2026
INF-0330	19/01/2026	30/01/2026	07/02/2026	28/03/2026
INF-0320	16/02/2026	27/02/2026	07/03/2026	18/04/2026
INF-0318	23/03/2026	03/04/2026	11/04/2026	23/05/2026
INF-0325	06/04/2026	17/04/2026	25/04/2026	30/05/2026
INF-0331	11/05/2026	22/05/2026	30/05/2026	04/07/2026
INF-0333	18/05/2026	29/05/2026	06/06/2026	18/07/2026
INF-0335	29/06/2026	10/07/2026	11/07/2026	15/08/2026
INF-0332	27/07/2026	07/08/2026	25/07/2026	29/08/2025
INF-0319	03/08/2026	14/08/2026	22/08/2026	03/10/2026
INF-0323 (PARTE 2)	-	-	05/09/2026	19/09/2026
INF-0321	07/09/2026	18/09/2026	26/09/2026	07/11/2026
INF-0329	21/09/2026	02/10/2026	10/10/2026	21/11/2026
INF-0334	26/10/2026	06/11/2026	14/11/2026	05/12/2026

INFORMAÇÕES

 ic.unicamp.br/extensao/cursos/engenharia-de-software/

 icext@unicamp.br

 (19) 3521-5883 | (19) 3521-5888



ENGENHARIA DE SOFTWARE
MODALIDADE: EXTENSÃO UNIVERSITÁRIA

