



UNIVERSIDADE FEDERAL DO CEARÁ
CAMPUS DE RUSSAS

PROJETO PEDAGÓGICO DO CURSO DE ENGENHARIA DE SOFTWARE

RUSSAS

2017

Reitor

Henry de Holanda Campos

Vice-Reitor

Custódio Luís Silva de Almeida

Pró-Reitor de Graduação

Cláudio de Albuquerque Marques

Pró-Reitora Adjunta

Simone da Silveira Sá Borges

Coordenadora da COPAC

Ana Paula de Medeiros Ribeiro

Diretor do Centro, Instituto ou Faculdade

Lindberg Lima Gonçalves

Vice-diretor

Aliny Abreu de Sousa Monteiro

Coordenador de Programas Acadêmicos

Aliny Abreu de Sousa Monteiro

Coordenador do curso

Markos Oliveira Freitas

Vice-Coordenador

Marcos Vinicius de Andrade Lima

Membros do Colegiado

Daniel Márcio Batista de Siqueira
Filipe Maciel Roberto
Jeferson da Silva Juliani
José Osvaldo Mesquita Chaves
Marcos Vinicius de Andrade Lima
Marília Soares Mendes
Markos Oliveira Freitas
Tatiane Fernandes Figueiredo
Valéria Lelli Leitão Dantas

Membros do NDE

Daniel Márcio Batista de Siqueira
José Osvaldo Mesquita Chaves
Marcos Vinicius de Andrade Lima
Marília Soares Mendes
Markos Oliveira Freitas
Valéria Lelli Leitão Dantas

Comissão de elaboração

Alexandre Matos Arruda
Daniel Márcio Batista de Siqueira
Dmontier Pinheiro Aragão Júnior
Eurinaldo Rodrigues Costa
Filipe Maciel Roberto
José Osvaldo Mesquita Chaves
Luiz Antônio Caetano Monte
Marcos Vinicius de Andrade Lima
Marília Soares Mendes
Markos Oliveira Freitas
Pablo Luiz Braga Soares
Rafael Fernandes Ivo
Tatiane Fernandes Figueiredo
Valéria Lelli Leitão Dantas

SUMÁRIO

1. APRESENTAÇÃO

- 1.1 Histórico da UFC
- 1.2 Histórico do curso
- 1.3 Contextualização nacional, regional e local

2. IDENTIFICAÇÃO DO CURSO

- 2.1 Nome do curso
- 2.2 Titulação conferida
- 2.3 Modalidade do curso
- 2.4 Duração do curso
- 2.5 Regime do curso
- 2.6 Número de vagas oferecidas por semestre/ano
- 2.7 Turnos previstos
- 2.8 Ano e semestre de início de funcionamento do curso
- 2.9 Ato de autorização
- 2.10 Processo de ingresso
- 2.11 Relação do curso com as políticas institucionais de ensino, pesquisa e extensão constantes no PDI
- 2.12 Princípios norteadores
- 2.13 Objetivos do curso
- 2.14 Perfil profissional do egresso
- 2.15 Áreas de atuação do futuro profissional

3. ORGANIZAÇÃO CURRICULAR

- 3.1 Conteúdos curriculares
- 3.2 Unidades e Componentes curriculares
- 3.3 Integralização curricular
- 3.4 Metodologias de ensino e de aprendizagem
- 3.5 Procedimentos de avaliação dos processos de ensino e aprendizagem
- 3.6 Estágio curricular supervisionado
- 3.7 Trabalho de Conclusão de Curso
- 3.8 Atividades complementares
- 3.9 Ementário e bibliografias

4. GESTÃO ACADÊMICA DO CURSO

- 4.1 Coordenação
- 4.2 Colegiado
- 4.3 Núcleo Docente Estruturante
- 4.4 Integração com as redes públicas de ensino
- 4.5 Apoio ao discente
- 4.6 Acompanhamento e avaliação do PPC

5. INFRAESTRUTURA DO CURSO

6. REFERÊNCIAS

7. ANEXOS

1. APRESENTAÇÃO

O presente documento visa apresentar e detalhar o projeto pedagógico do Curso de Engenharia de Software do Campus da Universidade Federal do Ceará em Russas, segundo as Diretrizes Curriculares Nacionais em vigor e a Lei de Diretrizes e Bases da Educação Nacional (LDB) de 1996 (Lei 9.394/96).

A proposta pedagógica do Curso de Engenharia de Software busca uma diretriz ideal, enfatizando o aspecto tecnológico no âmbito das técnicas computacionais, sem descuidar, naturalmente, do seu fundamento científico. Foram tomadas como base para a construção do plano pedagógico aqui apresentado os seguintes documentos:

- Currículo de referência para programas de graduação organizado pelas duas principais agremiações mundiais de profissionais da área de computação, ACM (Association for Computing Machinery) e IEEE Computer Society (Institute of Electrical and Electronics Engineers) [IEEE-CS/ACM 2014];
- Guia para o corpo de conhecimento em Engenharia de Software, organizado pela IEEE Computer Society (SWEBOK) [IEEE-CS 2004];
- Currículo de referência para cursos de pós-graduação em Engenharia de Software [SIT 2009];
- Projeto pedagógico confeccionado para o curso de Bacharelado em Engenharia de Software do Campus de Quixadá da Universidade Federal do Ceará;
- Projeto pedagógico confeccionado para o curso de Bacharelado em Ciência da Computação do Campus de Fortaleza da Universidade Federal do Ceará.

O projeto pedagógico do curso busca dosar, de forma racional, teoria e prática nas proporções adequadas, de modo a formar um profissional apto a atuar na indústria de software. O maior desafio da formação proposta é apresentar fundamentos teóricos de forma que o egresso seja capaz de manter-se continuamente atualizado diante do progresso incessante que é uma característica dessa área de atuação.

Não obstante, tem-se como objetivo complementar formar um profissional empreendedor, capaz de lidar com técnicas avançadas de gerenciamento de projetos, qualidade de processos e produtos e inovação tecnológica. Para tanto, propõe-se um modelo

pedagógico capaz de adaptar-se à dinâmica das demandas da sociedade, em que a graduação passa a constituir-se numa etapa de formação inicial em processo de educação permanente.

Este projeto pedagógico originou-se a partir de um esforço coletivo dos professores do curso de Engenharia de Software, em especial, do seu Núcleo Docente Estruturante e do seu Colegiado, como maneira de adequá-lo aos padrões de qualidade estabelecidos pela Pró-Reitoria de Graduação na Universidade Federal do Ceará.

Este projeto pedagógico está organizado da seguinte maneira. Esta seção apresenta a finalidade e a contextualização do curso. A segunda seção caracteriza melhor o curso, identificando seus objetivos e princípios norteadores, bem como o perfil profissional do egresso e suas áreas de atuação. A Seção 3 apresenta a organização curricular do curso, suas unidades curriculares e disciplinas, bem como descreve como são realizados o trabalho de conclusão, o estágio e as atividades complementares. A Seção 4 apresenta a gestão acadêmica do curso e suas ações, e a Seção 5 descreve a infra-estrutura disponível para o curso. A Seção 6 lista as referências utilizadas e a Seção 7 contém anexos a este projeto pedagógico.

1.1 Histórico da UFC

A ideia da criação de uma universidade, com sede em Fortaleza, foi ventilada pela primeira vez no ano de 1944, quando o médico cearense Dr. Antônio Xavier de Oliveira encaminhou ao Ministério da Educação e Saúde um relatório sobre a refederalização da Faculdade de Direito do Ceará. A partir daí, tal ideia passou a vigorar no pensamento dos cearenses, notadamente de alunos e professores das escolas superiores existentes.

Por ocasião da visita do então Ministro da Educação, prof. Clemente Mariani Bittencourt, à terra de Iracema, os alunos da Faculdade de Direito entregaram-lhe um documento, com quase 10 mil assinaturas, pleiteando uma Universidade para o Ceará. O discurso do Ministro naquela faculdade foi pautado na objetivação da criação da referida instituição e, ao finalizá-lo, o fez com o seguinte desfecho: “Teremos, então, a vossa universidade, para cujo advento contaremos comigo, como um leal companheiro nesta campanha, que juntos encetaremos”.

Inquieto e impressionado com as últimas palavras do titular da Pasta da Educação, o prof. Antônio Martins Filho solicitou audiência com o governador da época, o desembargador Faustino de Albuquerque, que o acolheu prontamente e à sua proposta, e o designou como um

de seus membros para, junto às autoridades competentes do Ministério da Educação e Saúde, estudarem as medidas cabíveis à criação da referida instituição.

Em 30 de setembro de 1953, o Presidente Getúlio Vargas envia ao Poder Legislativo a Mensagem no 391, de 1953, com o projeto de lei e demais documentos sobre a criação da Universidade do Ceará, com sede em Fortaleza, capital do Ceará. Logo em seguida, e dentro da tramitação legal, o Presidente enviou o referido projeto de lei, através do processo no 3713/53, ao Congresso Nacional. Da Câmara dos Deputados, a matéria foi encaminhada à Comissão de Educação e Cultura, cujo relator foi o deputado cearense João Otávio Lobo.

Antes de terminada a legislatura de 1954, o projeto de lei tão esperado, já finalmente aprovado nas duas Casas do Congresso, foi encaminhado à Comissão de Redação Final na forma do Regimento da Câmara. E, na presença do governador eleito, Paulo Sarasate, e de vários representantes cearenses no Congresso, o Presidente Café Filho sancionou a Lei no 2.373, criando a Universidade do Ceará, fato ocorrido em 16 de dezembro de 1954, tendo sido instalada no dia 25 de junho de 1955. Originalmente foi constituída pela união da Escola de Agronomia, Faculdade de Direito, Faculdade de Medicina e Faculdade de Farmácia e Odontologia.

Assim, a Universidade Federal do Ceará (UFC) nasceu da vontade e determinação de um grupo de intelectuais cearenses que vislumbrava o papel determinante de uma universidade pública como um elemento de mudanças e transformações culturais, sociais e econômicas do Estado do Ceará e da Região Nordeste.

De fato, ao longo de toda sua existência, a UFC vem contribuindo de forma decisiva para a evolução da educação superior do Ceará e do Nordeste. Grandes passos já foram dados rumo a sua consolidação como instituição de ensino superior “inserida entre as grandes universidades brasileiras”, e novos desafios se apresentam ao seu futuro. Cada vez mais o lema “O universal pelo regional”, do Reitor Martins Filho, fundador da UFC, reafirma-se como vocação desta instituição que, a partir de seu crescimento, principalmente rumo aos grotões do sertão cearense, leva soluções universais para os anseios de nosso povo. Neste sentido, a UFC foi tutora na criação da Universidade Federal do Cariri (UFCA) e da Universidade da Integração Internacional da Lusofonia Afro-Brasileira (UNILAB).

Em 2016, a UFC foi classificada na décima posição do Ranking Universitário da Folha (RUF) [RUF 2016], sendo a sétima entre as federais e a primeira do Nordeste. O RUF é um levantamento anual do jornal Folha de S. Paulo que, em 2016, mapeou o desempenho de 195

instituições em cinco dimensões: pesquisa, ensino, reconhecimento do mercado, internacionalização e inovação. No mesmo ano, a UFC foi classificada entre como a 87^a melhor universidade da América Latina, segundo o QS University Rankings - Latin America 2015, em divulgação feita no site da QS World University [QS 2016]. A avaliação é feita com base em sete indicadores: reputação acadêmica, reputação como empregador, relação com o corpo docente, citações por artigo, número de artigos por faculdade, proporção de corpo de servidores com PhD e impacto na web. Em 2014, a Universidade Federal do Ceará obteve mais uma vez destaque na avaliação do ensino superior realizada pelo Ministério da Educação (MEC). Com IGC, Índice Geral de Cursos, igual a 4 (bom) [UFC 2016], em uma escala que vai de 1 a 5, a UFC permanece como a melhor universidade do Ceará e segunda do Nordeste, ficando atrás somente da Universidade Federal de Pernambuco (UFPE). No geral, a UFC foi classificada na 21^a posição do ranking publicado pelo MEC, em um total de 2.042 instituições de ensino superior (IES) avaliadas em todo o País.

1.2 Histórico do Curso

O curso de Engenharia de Software foi o primeiro curso a entrar em funcionamento no campus de Russas, no segundo semestre do ano de 2014, nas dependências do Centro Vocacional Tecnológico (CVT) da cidade de Russas. A partir de 2015, com a construção da primeira unidade didática do campus, o curso passou a funcionar em dependências próprias, contando com a disponibilidade de oito salas de aula com capacidade para 60 alunos cada e com três laboratórios de informática com capacidade para 30 alunos cada, além de biblioteca, salas de professores, salas de administração e secretarias e de auditório.

Em 2015, foi criado o LINCE, o Laboratório Interdisciplinar de Computação e Engenharia de Software, o primeiro grupo de pesquisa do campus de Russas. O LINCE conta com um laboratório próprio, com capacidade para 15 alunos. Apesar de suas limitações físicas, o grupo conta atualmente com mais de 50 alunos, orientados por nove professores do campus de Russas e dois do campus de Quixadá. Apesar de a maioria dos alunos trabalhar de maneira voluntária, diversos recebem bolsa de Iniciação Científica, de Iniciação Acadêmica, de Iniciação à Docência ou de Monitoria. O grupo tem dois projetos financiados pela FUNCAP, e um deles resultou em um artigo publicado no XV Simpósio Brasileiro sobre Fatores Humanos em Sistemas Computacionais.

Nesses dois anos e meio de funcionamento do curso, aconteceram duas edições da SESCOMP – Semana de Engenharia de Software e Ciência da Computação, um evento anual organizado por alunos e professores do Campus de Russas que contaram com a participação de convidados externos a Russas e à UFC. Os alunos do curso organizaram ainda uma edição do Vale Web Livre e uma edição do FLISOL – Festival Latino-Americano de Instalação de Software Livre. Os alunos do curso de Engenharia de Software participaram ainda de dois Encontros Universitários da UFC, o primeiro em Fortaleza e o segundo no próprio Campus de Russas.

1.3 Contextualização nacional, regional e local

O campus de Russas foi idealizado para atender as regiões do Vale do Jaguaribe e do Litoral Leste do Ceará. As cidades abrangidas pelo campus de Russas vão desde Jaguaribe e Pereiro, ao sul, até Aracati, Fortim e Icapuí, no litoral.

O município de Russas está localizado a 165 km da capital Fortaleza, tendo como principal acesso a BR 116. A cidade constitui um dos mais importantes centros populacionais e econômicos do Vale do Jaguaribe. É um importante pólo econômico do Vale do Jaguaribe e do Estado do Ceará. Localizada numa região de solo fértil do Vale Jaguaribano, sempre foi ponto estratégico para o transporte de pessoas e mercadorias, por ali ter passado a Estrada Real do Jaguaribe no período colonial, depois a estrada Transnordestina e hoje a BR-116. Após o ciclo da Carne de Charque, a economia do município passou pelo ciclo do algodão, o ciclo da carnaúba, e o da laranja. Esta última lhe rendeu o título de “Terra da Laranja Doce”, pois se criou uma verdadeira marca para a laranja da região, a “Laranja de Russas” conhecida nacionalmente.

Além do campus da UFC de Russas, as regiões do Vale do Jaguaribe e do Litoral Leste do Ceará contam com seis campi do Instituto Federal de Educação, Ciência e Tecnologia do Ceará (IFCE), nas cidades de Aracati, Jaguaruana, Limoeiro do Norte, Morada Nova, Tabuleiro do Norte e Jaguaribe, e com um campus da Universidade Estadual do Ceará (UECE), na cidade de Limoeiro do Norte. O Centro Vocacional Tecnológico (CVT), unidade operacional vinculada ao Instituto Centro de Ensino Tecnológico (CENTEC), também está presente, com campi nas cidades de Fortim, Itaiçaba, Russas, Quixeré, Limoeiro do Norte, Tabuleiro do Norte e Jaguaribara. Juntos, o IFCE, a UECE e o CVT oferecem diversos cursos técnicos e superiores, principalmente nas áreas referentes às licenciaturas, às ciências agrárias, aos agronegócios, e

à tecnologia de alimentos. Além da UFC, apenas o IFCE oferece cursos relacionados a Tecnologia de Informação e Comunicação (TIC), como o curso técnico de Informática nos campi de Morada Nova e de Aracati, o curso superior tecnológico de Redes de Computadores no campus de Jaguaribe e o curso de bacharelado em Ciência da Computação, também no campus de Aracati.

O estado do Ceará conta, há mais de cinquenta anos, com a contribuição da UFC na formação de pessoal altamente qualificado, na geração e preservação de conhecimento, na inovação tecnológica e na integração com a sociedade através de atividades e projetos de extensão. Atualmente, a UFC conta com três campi em Fortaleza e quatro novos campi no interior do estado: Sobral, Quixadá, Russas e Crateús. O estado do Ceará conta também com a Universidade Federal do Cariri (UFCA), que nasceu campus da UFC e tornou-se uma universidade independente em 2013 e com a Universidade da Integração Internacional da Lusofonia Afro-Brasileira (UNILAB) que é uma autarquia vinculada ao Ministério da Educação com sede em Redenção.

Desde 1975 a UFC vem oferecendo cursos de graduação na área de Computação relacionados à Engenharia de Software e TIC, tendo iniciado com a formação de Tecnólogo em Processamento de Dados, posteriormente transformado em Bacharelado em Ciência da Computação (1985). O Departamento de Computação da UFC realizou esforços de formação de professores doutores e, em 1995, inaugurou a pós-graduação stricto sensu com o Mestrado em Ciência da Computação. Dez anos mais tarde, foi criado o Doutorado em Ciência da Computação.

Em 2006, foi criado o campus da UFC de Sobral, que oferece somente o curso de Engenharia da Computação relacionado a TIC. Em 2007, foi criado o campus da UFC de Quixadá, um polo avançado voltado à TIC, contando com o curso de tecnólogo em Redes de Computadores e os cursos de bacharelado em Sistemas de Informação, Engenharia de Software, Ciência da Computação, Engenharia da Computação e Design Digital. Tendo em vista o sucesso dos dois campi no interior do estado, em 2014, foram criados os campi de Russas e de Crateús. O campus de Crateús oferece os cursos de Sistemas de Informação e de Ciência da Computação, relacionado a TIC. O campus de Russas, além do curso de Engenharia de Software, conta com curso de Ciência da Computação e os cursos de Engenharia Civil, Engenharia Mecânica e Engenharia de Produção.

2. IDENTIFICAÇÃO DO CURSO

2.1 Nome do curso

Bacharelado em Engenharia de Software

2.2 Titulação conferida

Bacharel(a) em Engenharia de Software

2.3 Modalidade do curso

Presencial

2.4 Duração do curso

Integralização mínima em 8 semestres (4 anos) e máxima em 12 semestres (6 anos)

2.5 Regime do curso

Semestral

2.6 Número de vagas oferecidas

50 vagas por semestre

2.7 Turnos previstos

Manhã e Tarde

2.8 Ano e semestre de início de funcionamento do curso

2014.2

2.9 Ato de Autorização

Resolução N^o 11/CONSUNI, de 22 de julho de 2014

2.10 Processo de ingresso

Atualmente o acesso aos cursos de graduação presenciais da UFC se dá unicamente pelo processo MEC-SiSU. Outras formas de ingresso também são previstas pela legislação para a ocupação das vagas residuais: readmissão ao curso, transferência interna ou externa, portador de diploma de nível superior, continuidade de estudos, convênio resultante de acordo cultural e intercâmbio acadêmico, programa de mobilidade acadêmica entre as universidades federais.

2.11 Relação do curso com as políticas institucionais de ensino, pesquisa e extensão constantes no PDI

O Plano de Desenvolvimento Institucional (PDI), conforme Decreto n^o 5.773 de 2006, é um instrumento de planejamento e gestão que considera a identidade da IES, no que diz respeito à sua filosofia de trabalho, à missão que propõe, às estratégias para atingir suas metas e objetivos, à sua estrutura organizacional, ao Projeto Pedagógico Institucional (PPI) com as diretrizes pedagógicas, que orientam suas ações e as atividades acadêmicas e científicas, que desenvolve ou que pretende desenvolver, e os recursos financeiros que dispõe.

O presente Projeto Pedagógico articula-se naturalmente com o Plano de Desenvolvimento Institucional – PDI da UFC, na medida em que seus pressupostos refletem aqueles estabelecidos neste documento.

O PDI da UFC (2013-2017) [UFC 2013] organiza os objetivos, estratégias e ações que a comunidade percebeu como sendo necessários para que a UFC se eleve ao patamar desejado ao final do ano de 2017 em quatro grandes eixos: ensino e aprendizagem, pesquisa, extensão e gestão.

O Eixo Estratégico Ensino e Aprendizagem apresenta-se dividido em 2 programas: Melhoria da Qualidade do Ensino e Expansão da Oferta de Ensino.

O primeiro programa, Melhoria da Qualidade do Ensino, é dividido nos seguintes subprogramas: Avaliação; Metodologias de Ensino e Aprendizagem; Formação para a Docência no Ensino Superior; Protagonismo Estudantil; Assistência Estudantil; e Melhoria do Ensino no Âmbito dos Hospitais.

Para cada subprograma, foi definida uma série de objetivos e, para cada objetivo, foram propostas estratégias e ações. A seguir, discutiremos programas, subprogramas, objetivos, estratégias e ações constantes no PDI que influenciam o curso de Bacharelado em Engenharia de Software.

No programa “Melhoria da Qualidade do Ensino”, o curso de Engenharia de Software está associado aos subprogramas “Avaliação”, “Metodologias de Ensino e Aprendizagem”, “Assistência Estudantil”, além do “Protagonismo Estudantil”.

No subprograma “Avaliação” destacam-se os seguintes objetivos:

- Objetivo 3 (no PDI): Realizar avaliações periódicas dos cursos de graduação. Para este objetivo, merece destaque a seguinte ação: Realizar a análise de dados oriundos dos sistemas de avaliação, de modo a que se possam gerar relatórios diagnósticos dos cursos de graduação e das Unidades Acadêmicas. O sistema de Avaliação Institucional da UFC fornece os dados brutos, que são analisados e interpretados pelo corpo docente do campus.
- Objetivo 4 (No PDI): Compartilhar os resultados diagnósticos com a comunidade vinculada ao curso de graduação. Para este objetivo, merecem destaque as seguintes ações: a) Proporcionar o efetivo emprego dos resultados das avaliações, de modo que o processo de ensino e aprendizagem possa ser objeto de reflexão coletiva e b) Realizar seminários setoriais para a divulgação dos resultados da avaliação. O corpo docente do curso reúne-se semestralmente com corpo discente e discute-se abertamente as práticas docentes que obtiveram sucesso durante o semestre, o que não funcionou, o que precisa ser ajustado, etc, bem como o resultado da Avaliação Institucional.

Já no subprograma “Metodologias de ensino e aprendizagem” destacam-se os seguintes objetivos:

- Objetivo 1 (No PDI): Estimular a inovação da prática de ensino superior através da flexibilização, da criatividade, de práticas interdisciplinares, da mudança e recriação contínua das práticas de ensinar/aprender em sala de aula. Para este objetivo, merece destaque a seguinte ação: Buscar uma forte articulação com o Programa de Formação para a Docência do Ensino Superior na UFC, a CASA - Comunidade de Cooperação e Aprendizagem Significativa. Essa estratégia tem sido posta em prática com a realização de diversos cursos da CASA, como o curso de Didática no Ensino Superior, de maneira presencial no Campus de Russas, e o curso de Docência Integrada às Tecnologias de Informação e Comunicação, na modalidade a distância.
- Objetivo 3 (No PDI): Ampliar o uso de multimídias, Tecnologias da Informação e Comunicação (TICs), do Ensino a Distância (EaD) nos cursos da UFC. Para esse objetivo, merecem destaque as seguintes ações: a) Disseminar a prática do uso de multimídia, de teleconferência e interação didática na aprendizagem presencial e a distância; b) Ampliar a capacitação dos docentes no uso das TICs no processo de ensino-aprendizagem; e c) Implantar salas de informática e videoconferências em todas as unidades acadêmicas da UFC. Essas ações têm sido alcançadas através da realização do curso de Docência Integrada às Tecnologias de Informação e Comunicação por boa parte dos professores do curso de Engenharia de Software, bem como a colocação em prática dos conceitos aprendidos em diversas disciplinas do curso. O Campus de Russas possui uma sala de videoconferência, que foi utilizada em eventos realizados no campus, como a SESCOMP, a Semana de Engenharia de Software e Ciência da Computação.

No subprograma “Protagonismo Estudantil” destaca-se o seguinte objetivo:

- Objetivo: Ampliar e consolidar as estratégias relativas ao protagonismo estudantil na UFC. Para este objetivo, merece destaque a seguinte ação: Desenvolver, junto com os professores, metodologias pautadas na aprendizagem cooperativa a serem utilizadas em sala de aula e em atividades extracurriculares. O Programa de Aprendizagem Cooperativa em Células Estudantis (PACCE) no Campus de Russas iniciou-se em 2015, um semestre após o início do funcionamento do campus, tem trazido cada vez mais alunos adeptos, e tem-se mostrado efetivo entre os participantes.

O segundo programa no Eixo Ensino e Aprendizagem, o programa de Expansão da Oferta de Ensino, é dividido nos seguintes subprogramas: Expansão dos Campi e das Unidades Acadêmicas Existentes; e Criação de Novos Campi e Novas Unidades Acadêmicas.

O Campus de Russas, um dos quatro campis no interior da UFC no interior do Ceará, é a realização direta do segundo subprograma do programa de Expansão da UFC. O campus visa levar mais oferta de cursos para formação de pessoal de nível superior

para as regiões do Vale do Jaguaribe e do Litoral Leste, aproveitando a expansão do mercado de TIC no interior do estado do Ceará.

2.12 Princípios norteadores

O presente projeto é destacado como um empenho político-pedagógico que visa uma formação em Engenharia de Software socialmente consciente e instigante, ultrapassando limites disciplinares e considerando o saber como uma construção social. Essa vertente orientadora reafirma como elementos fundantes, para atuar como profissional da Computação, princípios da ética democrática, dos quais se destaca: a dignidade humana, a justiça, o respeito mútuo, a participação, a responsabilidade, o diálogo e a solidariedade, na sua atuação como profissional e como cidadão.

Nesse sentido, o curso de Engenharia de Software defende os seguintes princípios norteadores:

- O ser humano seja o princípio e fim de todo processo formativo no qual haja comprometimento com a ética na busca da verdade e do conhecimento;
- A inclusão de pessoas com deficiências específicas e a promoção da acessibilidade, bem como o comprometimento com a superação de preconceitos;
- A prevalência da integração entre formação básica, diferenciada, garantindo a esta uma flexibilidade do pensamento e liberdade de expressão;
- O compromisso com o fortalecimento da cultura acadêmica, através da integração entre ensino, pesquisa e extensão;
- A reflexão e a articulação entre teoria e prática, técnica e humanismo;
- A capacidade de adaptação à evolução tecnológica.

Como os demais cursos criados no Campus de Russas, em termos curriculares, este projeto pedagógico assume como preocupação central, a formulação de uma proposta que seja capaz de formar cidadãos e profissionais que não se limitem apenas ao exercício técnico-profissional, mas que estejam aptos a uma atuação ético-política, comprometida com as transformações qualitativas do mundo em que vivemos, na perspectiva da promoção de uma sociedade democrática, plural e justa.

As orientações curriculares visam, nestes termos: à promoção de uma cultura acadêmica, de caráter não tecnicista, pautada na ampliação dos espaços de aprendizagem, na diversidade e integração crescente dos conteúdos científicos e artísticos e na capacidade de lidar com sua intensa mutação na sociedade contemporânea.

Sob esta ótica, defende-se como princípios curriculares:

- Flexibilização Curricular – recomenda-se a presença de uma proporção significativa de conteúdos de natureza optativa nos currículos e a redução das exigências de pré-requisitos, sempre que pertinente, de modo a permitir que o aluno participe do processo de definição do seu percurso acadêmico.

- Caráter inter e transdisciplinar – compreendido como a valorização da articulação dos diversos campos do saber, vinculando a formação técnica à formação humanística, promovendo a relação teoria/técnica/prática articulada a uma dimensão ético-estética, e dando ao currículo uma perspectiva de conjunto, que favoreça a superação da visão fragmentada do conhecimento.
- Atualização Permanente – recomenda-se que o PPC fomente um processo de atualização permanente, que permita estabelecer o aprimoramento e/ou a correção de trajetórias, a incorporação dos avanços científicos e tecnológicos, as inovações artísticas e as conquistas substantivas nos diversos campos do conhecimento.

Considerando os elementos em referência, o Projeto Pedagógico do Curso de Engenharia de Software busca a consolidação de uma identidade própria, orientado por princípios que compreendem que a formação profissional em Engenharia de Software, envolve uma prática específica, que pressupõe saberes e competências coerentes. Para isso, é preciso que o currículo seja flexível e possibilite não só a formação de competência técnica como também o compromisso da ciência com as transformações sociais.

2.13 Objetivos do curso

O Curso de Bacharelado em Engenharia de Software da UFC em Russas tem como objetivo formar profissionais aptos a introduzirem melhorias e a participarem efetivamente de empreendimentos de software voltado para os mercados local e global, oferecendo a base teórica suficiente para que os seus egressos possam manter-se constantemente atualizados. Além disso, é objetivo deste curso preparar profissionais para construir, usando as técnicas da Engenharia de Software, sistemas de software corretos, completos, seguros, amigáveis, usáveis, com qualidade, fáceis de manter e de baixo custo.

2.14 Perfil profissional do egresso

O egresso do curso de Engenharia de Software da UFC em Russas deverá ser um profissional com sólida formação básica de conhecimentos teóricos e práticos da Engenharia de Software. Deverá ainda ser capaz de desempenhar as atividades necessárias para a concepção, construção, e manutenção de sistemas de software de qualidade, pensando de forma rigorosa durante a atuação profissional e realizando escolhas bem fundamentadas ao executar suas tarefas. Possuir experiência prática em pelo menos um projeto de software significativo. Aplicar o conhecimento das várias áreas da computação para a solução de problemas reais com inovação tecnológica, reconhecendo as oportunidades de negócio provenientes destas soluções. Deverá atuar

com ética profissional, em equipes multidisciplinares, sempre atento às demandas sociais da região. Por fim, deverá manter-se constantemente atualizado e assimilar o permanente avanço tecnológico.

A coordenação do curso de Engenharia de Software manterá um cadastro virtual dos seus alunos egressos, através de um sistema on-line a ser desenvolvido futuramente e mantido no portal do curso na internet, também a ser criado. Os egressos de destaque profissional ou acadêmico serão convidados a ministrar palestras aos alunos do curso, propiciando uma maior interação entre estes, os docentes e os servidores técnico-administrativos do curso.

2.15 Áreas de atuação do futuro profissional

O egresso terá condições de assumir um papel de agente transformador do mercado, sendo capaz de provocar mudanças através da incorporação de novas tecnologias da informação na solução dos problemas e propiciando novos tipos de atividades, agregando:

- Domínio de novas tecnologias da informação e gestão da área de Engenharia de Software, visando melhores condições de trabalho e de vida;
- Conhecimento e emprego de modelos associados ao uso das novas tecnologias da informação e ferramentas que representem o estado da arte na área;
- Conhecimento e emprego de modelos associados ao diagnóstico, planejamento, implementação e avaliação de projetos de sistemas de software aplicados nas organizações;
- Uma visão humanística consistente e crítica do impacto de sua atuação profissional na sociedade e nas organizações.

Desta forma, não exclusivamente, o egresso do curso poderá atuar como:

- Analista de sistemas de software
- Desenvolvedor de sistemas de software
- Gerente/Analista de configuração
- Projetista de sistemas de software
- Arquiteto de software
- Gerente/Analista de qualidade de software
- Gerente/Analista de teste de software
- Gerente de projetos de software
- Consultor/Auditor de sistemas software
- Professor e/ou Pesquisador

3. ORGANIZAÇÃO CURRICULAR

As Diretrizes Curriculares Nacionais para cursos de Computação [CNE 2016] serviram como parâmetros na preparação e organização do currículo, que considera o

desenvolvimento de competências tecnológicas, a capacidade de aprender a aprender, acompanhar as mudanças no mundo contemporâneo, contemplando a flexibilidade, a interdisciplinaridade e a contextualização.

O currículo do curso oferece flexibilidade ao utilizar o critério de pré-requisitos mínimos para disciplinas, o que facilita para o aluno uma melhor oferta de disciplinas no curso, ao permitir ao aluno cursar disciplinas optativas ofertadas por outros cursos (optativa-livre) e ao ofertar atividades complementares.

As parcerias com empresas da área de Informática do estado permitirão que a coordenação do curso e os professores tenham uma avaliação permanente da demanda local e com isso uma informação que possibilita uma atualização constante do curso e seu currículo.

A proposta de formatação do curso foi elaborada a partir das necessidades regionais para formar profissionais aptos a atuarem nas diversas áreas relacionadas a informática no estado do Ceará, advindas da expansão do mercado de TIC. Através de parcerias mantidas com empresas e instituições que trabalham o desenvolvimento da informática no estado, como o Instituto do Software do Estado do Ceará, o projeto do curso procura priorizar o atendimento às demandas dos cidadãos, da sociedade e do mercado de trabalho.

Toda a estrutura curricular do curso de Bacharelado em Engenharia de Software foi elaborada de forma a contemplar os objetivos do curso e atingir o perfil profissional proposto. A organização do currículo permite a compreensão, o entendimento e o conhecimento para aplicar e desenvolver modelos, utilizando as novas tecnologias e metodologias, assegurando as inter-relações com outras áreas do conhecimento, contribuindo assim, com o processo de compreensão e transformação da realidade, desenvolvendo no discente não só competências, como também formando um cidadão consciente do seu papel na sociedade e alicerçado nos princípios da ética e da cidadania.

A matriz curricular compreenderá uma formação de 3.200 horas (equivalendo a 200 créditos, já que cada crédito equivale a 16 horas aula) com tempo ideal para conclusão do curso estimado em 4 (quatro) anos ou 8 (oito) semestres letivos. Sendo assim, o estudante do Curso de Engenharia de Software, grau Bacharelado, deverá observar o tempo máximo para a sua conclusão, estipulado em 6 (seis) anos ou 12 (doze) períodos letivos. Os alunos deverão cursar uma carga horária mínima por semestre de 4 créditos (64 horas) e, no máximo, 32 créditos (512 horas).

As disciplinas do Curso de Engenharia de Software serão de três tipos: obrigatórias, optativas (da integralização curricular do curso) e optativas-livres (de fora da integralização curricular do curso) e também atividades extraclasse: estágio supervisionado, trabalho de conclusão de curso e atividades complementares, como extensão, monitoria, iniciação científica, participação e organização de seminários e palestras etc. Eventualmente, respeitado o limite, algumas disciplinas poderão ser ministradas a distância.

3.1 Conteúdos curriculares

O desempenho das atividades inerentes às áreas de atuação em Engenharia de Software, aqui agrupadas em três pilares de formação, exige uma ação profissional fundamentada no conhecimento teórico-prático aprofundado da aplicação das soluções tecnológicas oferecidas pela ciência da computação a problemas existentes no mercado de software.

Para a formação do perfil do egresso pretendida pelo Curso de Bacharelado em Engenharia de Software, faz-se necessário o desenvolvimento de competências que, aliadas aos conhecimentos técnico-científicos adquiridos ao longo do Curso, permitam o desenvolvimento das habilidades necessárias à atuação do Bacharel em Engenharia de Software. O presente projeto prevê a ação integradora em disciplinas de outras áreas, visando fortalecer a interdisciplinaridade e a multidisciplinaridade para a formação do futuro profissional.

Inicialmente esta exigência implica em uma capacitação profissional que integre formação humanística e formação técnica com a busca incessante pela interdisciplinaridade. Além disso, a capacitação deve incluir o desenvolvimento de habilidades de relacionamento interpessoal, comunicação e trabalho em equipe, na medida em que são características cada vez mais importantes na atuação profissional. Assim, o profissional de Engenharia de Software deve dispor de uma sólida formação conceitual (conhecimento explícito) aliada a uma capacidade de aplicação destes conhecimentos científicos em sua área de atuação (conhecimento tácito) de forma a agregar valor econômico à organização e valor social ao indivíduo, bem como realizar seu trabalho de modo ético e consciente da sua realidade econômica, política e social.

Formação Humanística:

Esse elemento é imprescindível para a formação da postura profissional do egresso e é abordado em uma disciplina própria, dedicada especificamente ao assunto, e estimulado nas demais. Dessa forma, é esperado que o profissional de Engenharia de Software tenha as seguintes competências:

- Ser criativo e inovador na proposição de soluções para os problemas e oportunidades identificados nas organizações;
- Expressar ideias de forma clara, empregando técnicas de comunicação apropriadas para cada situação;
- Participar e conduzir processos de negociação para o alcance de objetivos;
- Trabalhar em equipe, somando as suas competências às daquelas de profissionais de outras áreas, em prol do desenvolvimento de projetos com objetivos comuns;

- Ter uma visão contextualizada da área de Engenharia de Software em termos políticos, sociais, econômicos e ambientais;
- Identificar oportunidades de negócio, criar e gerenciar empreendimentos para a concretização dessas oportunidades;
- Atuar social e profissionalmente de forma ética e cidadã, respeitando as políticas ambientais vigentes, os direitos humanos e as diferenças étnico-raciais dos diferentes povos que formam a nação brasileira.

Formação Técnica:

Tem como base as recomendações de importantes documentos da área, tais como o corpo de conhecimento em Engenharia de Software [IEEE-CS 2004] e o guia para currículos de graduação em Engenharia de Software [IEEE-CS/ACM 2014]. Dessa forma, é esperado que o profissional de Engenharia de Software desenvolva as seguintes competências:

- Compreender a dinâmica empresarial decorrente de mercados mais exigentes e conscientes de seus direitos e das novas necessidades sociais, ambientais e econômicas;
- Participar do desenvolvimento e implantação de novos modelos de competitividade e produtividade nas organizações no que tange à produção de sistemas de software;
- Diagnosticar e mapear, com base científica, problemas e pontos de melhoria nas organizações, propondo alternativas de soluções baseadas nos fundamentos da Engenharia de Software;
- Elicitar, especificar, rastrear e gerenciar requisitos de software, bem como projetar, especificar, implementar, implantar, verificar, validar e manter (evoluir ou corrigir) sistemas de software de maneira apropriada;
- Participar da implantação e monitoramento dos processos organizacionais de empresas de software, identificando as possíveis mudanças que podem surgir em função dos objetivos das organizações, das exigências do mercado, da legislação vigente e dos avanços nos modelos de qualidade de processo e produto no âmbito da Engenharia de Software;
- Criar, implantar, refinar e avaliar processos organizacionais, relacionados à construção de software, que visam garantir maior produtividade, qualidade, alinhamento da produção aos objetivos das organizações e conformidade com a legislação vigente nas esferas local, regional, nacional e mundial;
- Planejar e gerenciar projetos de software observando as necessidades dos clientes, a tríade tempo, escopo e custo, o modelo de processo adotado na organização e a legislação vigente.

Interdisciplinaridade:

Construir sistemas de software implica, necessariamente, no emprego de dois domínios: 1) o de Engenharia de Software ligado à computação e; 2) o domínio onde

está inserido o problema que motiva a construção do software. Dessa forma, é esperado que o profissional de Engenharia de Software desenvolva as seguintes competências:

- Deve ser capaz de trabalhar em equipe multidisciplinar (com alunos assumindo diferentes papéis na execução dos projetos das disciplinas);
- Ser capaz de investigar e entender problemas em diversificados domínios de aplicação (com o desenvolvimento de projetos que aplicam conhecimento de várias disciplinas);
- Capacidade de autoaprendizagem no que tange ao entendimento de outros domínios do conhecimento (tais como empreendedorismo, financeiro, design, educação).

Além disso, na formação profissional do aluno, busca-se orientá-lo no sentido de adaptar-se às mudanças e novidades da área de TIC. Para tanto, torna-se indispensável o desenvolvimento da inteligência emocional, tendo autoconhecimento, administração das emoções, automotivação, sociabilidade e liderança como algumas características a serem desenvolvidas durante a formação do aluno através do incentivo de atividades e trabalhos interessantes e em equipe.

3.2 Unidades e Componentes curriculares

As disciplinas optativas-livres podem ser escolhidas fora do elenco específico de disciplinas do curso. Nesse caso, o aluno pode ter acesso a disciplinas de qualquer outro curso, desde que autorizado pelo coordenador do seu curso, e devem ser contadas para a integralização curricular.

A tabela a seguir apresenta os conteúdos e as disciplinas ou atividades de cada unidade curricular.

Unidade Curricular	Conteúdo	Componente Curricular
Ciência da Computação	Programação	Fundamentos de Programação
		Laboratório de Programação
		Programação Orientada a Objetos
		Estruturas de Dados
		Estruturas de Dados Avançadas
		Linguagens de Programação
	Teoria da Computação e Algoritmos	Algoritmos em Grafos
		Projeto e Análise de Algoritmos
		Linguagens Formais e Autômatos
		Teoria da Computação
	Arquitetura de Computadores	Arquitetura de Computadores
Matemática Computacional	Matemática Computacional	

Unidade Curricular	Conteúdo	Componente Curricular
Matemática	Matemática	Matemática Básica
		Matemática Discreta
		Probabilidade e Estatística
		Lógica para Computação
		Pré-Cálculo
		Cálculo Diferencial e Integral I
		Álgebra Linear
Engenharia de Software	Engenharia de Software	Introdução à Engenharia de Software
		Introdução a Processos e Requisitos de Software
		Análise e Projeto de Sistemas
		Interação Humano-Computador
		Projeto Detalhado de Software
		Gerência de Projetos de Software
		Processos de Software
		Requisitos de Software
		Qualidade de Software
		Verificação e Validação
		Manutenção de Software
		Reuso de Software
Experimentação em Engenharia de Software		
Tecnologia da Informação	Sistemas Operacionais	Sistemas Operacionais
	Banco de Dados	Fundamentos de Banco de Dados
	Redes de Computadores, Sistemas Distribuídos e Internet	Redes de Computadores
		Desenvolvimento de Software para Web
		Sistemas Distribuídos
		Redes Sociais
	Computação em Nuvem	Computação em Nuvem
	Compiladores	Compiladores
	Inteligência Artificial	Inteligência Artificial
		Sistemas Multiagentes
Segurança	Segurança	
Jogos	Introdução ao Desenvolvimento de Jogos	
Engenharia de Software Aplicada	Engenharia de Software Aplicada	Arquitetura de Software
		Especificação Formal de Software
		Gerência de Configuração
		Integração de Aplicações

Unidade Curricular	Conteúdo	Componente Curricular
		Métodos e Ferramentas da Engenharia de Software
		Desenvolvimento de Software Concorrente
		Desenvolvimento de Software para Dispositivos Móveis
		Desenvolvimento de Software para Persistência
		Leitura de Software
		Trabalho Cooperativo Baseado em Computadores
		Estimativa de Custos em Projetos de Software
Formação Complementar e Humanística	Ética	Ética Profissional
	Administração	Empreendedorismo
		Fundamentos da Economia e da Administração
	Línguas	Inglês Instrumental I
		Inglês Instrumental II
		Língua Brasileira de Sinais
	Educação	Educação em Direitos Humanos
		Educação Ambiental
		Relações Étnico-Raciais e Africanidades
Formação Suplementar	Trabalho de Conclusão	Projeto de Pesquisa Científica e Tecnológica
		Trabalho de Conclusão de Curso I
		Trabalho de Conclusão de Curso II
	Estágio Profissional	Estágio Supervisionado I
		Estágio Supervisionado II

A área de formação básica é composta pelas unidades curriculares de Ciência da Computação e de Matemática, bem como a unidade de Engenharia de Software, que foi criada em virtude do objetivo do curso.

A área de formação tecnológica é composta por um conjunto de conteúdos relacionados à Engenharia de Software e às Tecnologias da Informação e Comunicação utilizadas no desenvolvimento de sistemas de software. As unidades que tratam desses conteúdos são, respectivamente, Engenharia de Software Aplicada e Tecnologia da Informação.

A unidade de Formação Complementar e Humanística é composta por um conjunto de conhecimentos que visam à preparação do egresso para interação com profissionais de outras áreas. Para o Bacharelado em Engenharia de Software destacam-

se as matérias que visam dar ao egresso o embasamento sobre o empreendedorismo. A área de formação humanística é composta por um conjunto de conteúdos que objetivam subsidiar a discussão e a compreensão da dimensão humana em relação à Engenharia de Software. As disciplinas optativas-livres também poderão auxiliar na formação complementar e humanística.

A unidade de Formação Suplementar é composta por uma disciplina (Projeto de Pesquisa Científica e Tecnológica) e por quatro atividades (Estágio Supervisionado I e II e Trabalho de Conclusão de Curso I e II) que permitem a síntese e integração dos conhecimentos adquiridos ao longo do curso.

As unidades curriculares deverão formar o futuro Bacharel em Engenharia de Software para exercer seu papel de profissional e cidadão levando em conta o desempenho de atividades nas áreas de Engenharia de Software e sua responsabilidade social.

De forma mais específica, é possível traçar algumas recomendações em relação aos docentes, de acordo com a área em que atuarão no currículo:

- Recomenda-se que os professores que atuarão na Unidade Curricular de Matemática e na de Formação Complementar e Humanística tenham formação nas áreas específicas das disciplinas que lecionam. Além disso, é desejável que tenham conhecimentos e experiência profissional que os habilitem a promover a articulação entre os conteúdos desenvolvidos em suas disciplinas e a aplicação em Engenharia de Software;
- Recomenda-se que os professores da Unidade Curricular de Ciência da Computação tenham formação na área de Computação. É desejável que estes docentes tenham conhecimentos e experiência profissional que os habilitem a promover a articulação entre os conteúdos desenvolvidos em suas disciplinas e a aplicação em Engenharia de Software;
- Os professores de Tecnologia da Informação podem ter formação variada de acordo com a área de aplicação envolvida, sendo geralmente provenientes de Computação. Além disso, é desejável que disponham de experiência profissional relacionada à aplicação da tecnologia específica em Engenharia de Software;
- Os professores das áreas de Engenharia de Software e Engenharia de Software Aplicada podem ser formados nas áreas de Computação, sendo desejável que tenham cursado a graduação ou pós-graduação em uma destas áreas. Além disso, é desejável que estes docentes tenham experiência profissional e/ou de pesquisa na área da matéria lecionada.

Algumas disciplinas presenciais que possam precisar de professores de outros campi da UFC, como as disciplinas de Inglês Instrumental I e II, Educação em Direitos Humanos, Educação Ambiental, Relações Étnico-Raciais e Africanidades, e Língua Brasileira de Sinais, todas da Unidade Curricular de Formação Complementar e

Humanística, serão ofertadas em caráter modular. Os módulos funcionarão como disciplinas intensivas, com maior flexibilidade quanto à sua duração e, geralmente, terão sua carga horária distribuída em um período de até oito semanas.

Temas importantes para a sociedade, como acessibilidade, TI verde, dengue e outras doenças transmitidas pelo *Aedes Aegypti*, educação ambiental, direitos humanos, relações étnico-raciais e africanidades, estão sendo considerados em disciplinas introdutórias, obrigatórias para o primeiro semestre, como Ética Profissional e Introdução à Engenharia de Software. Por terem um caráter transversal, esses temas também são contemplados em programas institucionais, como o "Projeto Boas Vindas" (projeto de incentivo à permanência), a "SESCOMP" (programa de extensão), o "Combate ao *Aedes Aegypti*, agente transmissor da dengue, da zika, da chikungunya e da febre amarela" (projeto de iniciação acadêmica), e em atividades complementares, na forma de palestras e mini-cursos, no decorrer de todo o curso.

3.3 Integralização curricular

A integralização curricular do curso de Engenharia de Software do Campus de Russas pode ser encontrada na tabela abaixo.

Semes- tre	Nome do Componente Curricular	Carga Horária Teórica	Carga Horária Prática	Carga Horária Total	Pré-requisito(s)	Correquisi- tos	Equivalência(s)
1	Fundamentos de Programação	32	32	64			Fundamentos de Programação
1	Ética Profissional	64		64			Ética, Normas e Postura Profissional
1	Introdução à Engenharia de Software	64		64			Introdução à Computação e Engenharia de Software
1	Matemática Básica	64		64			
2	Arquitetura de Computadores	64		64			
2	Laboratório de Programação		64	64	Fundamentos de Programação		Fundamentos de Programação
2	Estruturas de Dados	32	32	64	Fundamentos de Programação		Estrutura de Dados
2	Matemática Discreta	64		64	Matemática Básica		
2	Introdução a Processos e Requisitos de Software	64		64	Introdução à Engenharia de Software		
3	Programação Orientada a Objetos	32	32	64	Laboratório de Programação		
3	Probabilidade e Estatística	64		64	Matemática Básica		
3	Linguagens de Programação	48	16	64	Fundamentos de Programação		
3	Algoritmos em Grafos	64		64	Estruturas de Dados		Projeto e Análise de Algoritmos
3	Requisitos de Software	64		64	Introdução a Processos e Requisitos de Software		

Semes- tre	Nome do Componente Curricular	Carga Horária Teórica	Carga Horária Prática	Carga Horária Total	Pré-requisito(s)	Correquisi- tos	Equivalência(s)
4	Fundamentos de Banco de Dados	32	32	64	Programação Orientada a Objetos		
4	Análise e Projeto de Sistemas	32	32	64	Programação Orientada a Objetos; Introdução a Processos e Requisitos de Software		Modelagem e Análise de Software
4	Processos de Software	64		64	Introdução a Processos e Requisitos de Software		
4	Lógica para Computação	48	16	64	Matemática Discreta		
4	Projeto e Análise de Algoritmos	64		64	Matemática Discreta; Algoritmos em Grafos		Projeto e Análise de Algoritmos
5	Interação Humano-Computador	32	32	64			Interação Humano-Computador; Interface Humano-Computador
5	Sistemas Operacionais	48	16	64	Arquitetura de Computadores		
5	Projeto Detalhado de Software	64		64	Análise e Projeto de Sistemas		
5	Redes de Computadores	64		64	Arquitetura de Computadores		Redes de Computadores; Redes e Sistemas Distribuídos
5	Gerência de Projetos de Software	64		64			Gerência de Projetos
6	Verificação e Validação	64		64	Requisitos de Software		
6	Qualidade de Software	64		64			
6	Manutenção de Software	64		64	Análise e Projeto de Sistemas		
6	Arquitetura de Software	64		64	Projeto Detalhado de Software		
6	Empreendedorismo	64		64			
7	Projeto de Pesquisa Científica e Tecnológica	16	16	32		Trabalho de Conclusão de Curso I	
7	Estágio Supervisionado I		160	160			
7	Trabalho de Conclusão de Curso I	32		32		Projeto de Pesquisa Científica e Tecnológica	
8	Estágio Supervisionado II		160	160	Estágio Supervisionado I		
8	Trabalho de Conclusão de Curso II		96	96	Trabalho de Conclusão de Curso I		
Optativa	Pré-Cálculo	32		32			
Optativa	Trabalho Cooperativo Baseado em Computadores	64		64			
Optativa	Álgebra Linear	48	16	64	Matemática Básica		
Optativa	Cálculo Diferencial e Integral I	64		64	Pré-Cálculo		
Optativa	Estruturas de Dados Avançadas	32	32	64	Estruturas de Dados		Estrutura de Dados Avançada
Optativa	Matemática Computacional	48	16	64	Álgebra Linear		

Semestre	Nome do Componente Curricular	Carga Horária Teórica	Carga Horária Prática	Carga Horária Total	Pré-requisito(s)	Correquisitos	Equivalência(s)
Optativa	Fundamentos da Economia e da Administração	64		64	Probabilidade e Estatística		
Optativa	Desenvolvimento de Software para Web	32	32	64	Programação Orientada a Objetos		
Optativa	Introdução ao Desenvolvimento de Jogos	32	32	64	Programação Orientada a Objetos		
Optativa	Desenvolvimento de Software para Dispositivos Móveis	32	32	64	Programação Orientada a Objetos		
Optativa	Linguagens Formais e Autômatos	64		64	Linguagens de Programação		
Optativa	Desenvolvimento de Software para Persistência	16	48	64	Fundamentos de Banco de Dados		
Optativa	Segurança	32	32	64	Matemática Discreta; Análise e Projeto de Sistemas		
Optativa	Gerência de Configuração	64		64	Processos de Software		
Optativa	Inteligência Artificial	48	16	64	Lógica para Computação		
Optativa	Especificação Formal de Software	48	16	64	Lógica para Computação		
Optativa	Redes Sociais	48	16	64	Desenvolvimento de Software para Web		
Optativa	Compiladores	64		64	Linguagens Formais e Autômatos		
Optativa	Teoria da Computação	64		64	Linguagens Formais e Autômatos		
Optativa	Reuso de Software	64		64	Projeto Detalhado de Software		
Optativa	Estimativa de Custos em Projetos de Software	32	32	64	Gerência de Projetos de Software		
Optativa	Sistemas Distribuídos	48	16	64	Redes de Computadores		
Optativa	Sistemas Multiagentes	64		64	Inteligência Artificial		
Optativa	Desenvolvimento de Software Concorrente	32	32	64	Programação Orientada a Objetos; Sistemas Operacionais		
Optativa	Leitura de Software	16	48	64	Arquitetura de Software		
Optativa	Métodos e Ferramentas da Engenharia de Software	16	48	64	Arquitetura de Software		
Optativa	Integração de Aplicações	32	32	64	Arquitetura de Software		
Optativa	Experimentação em Engenharia de Software	32	32	64	Qualidade de Software		
Optativa	Computação em Nuvem	32	32	64	Sistemas Distribuídos		
Optativa	Inglês Instrumental I	64		64			
Optativa	Inglês Instrumental II	64		64	Inglês Instrumental I		
Optativa	Educação em Direitos Humanos	64		64			
Optativa	Educação Ambiental	64		64			
Optativa	Relações Étnico-Raciais e Africanidades	64		64			
Optativa	Língua Brasileira de Sinais	64		64			

A tabela abaixo mostra distribuição da carga horária do curso.

Distribuição da Carga Horária				
Tipo do Componente	Componente Curricular	Número de Créditos	Carga Horária	
Componentes obrigatórios	Disciplinas obrigatórias	118	1.888	59%
Componentes optativos	Disciplinas optativas ¹	36	576	18%
	Disciplinas optativas-livres ²	20	320	10%
Estágio supervisionado		20	320	10%
Trabalho final de curso		8	128	4%
Atividades complementares		18	288	9%
Total		200	3.200	100%

A tabela abaixo mostra os prazos para conclusão do curso.

Prazos³	Número de semestres
Mínimo	8
Médio	10
Máximo	12

A carga horária por semestre pode ser vista na tabela abaixo.

Carga horária por semestre	Número de horas
Carga horária mínima	64
Carga horária média	334
Carga horária máxima	512

O acesso às Tecnologias de Informação e Comunicação se dão através do uso dos três laboratórios de informática do campus, com capacidade para 30 alunos cada.

1 Da carga horária total de optativas, 576 horas, até 320 horas podem ser cursadas como optativas-livres. Não é obrigatório ao aluno integralizar um valor específico de horas optativas-livres.

2 Disciplinas optativas-livres, como o próprio nome diz, são de livre escolha, são aquelas que o aluno pode escolher fora do elenco específico de disciplinas de seu curso, em outro curso da universidade. Portanto, qualquer código de componente que não seja parte do rol de obrigatórios e optativos do curso, ao ser cursado pelo aluno, será integralizado como "optativa-livre".

3 De acordo com os limites definidos pela Resolução CEPE/UFC n.º. 14, de 3 de dezembro de 2007 que dispõe sobre a regulamentação do tempo máximo para conclusão dos cursos de graduação.

3.4 Metodologias de ensino e de aprendizagem

O projeto do curso de Engenharia de Software visa estimular o desenvolvimento de um profissional versátil com fundamentos sólidos de matemática, computação, e engenharia de software. Busca-se a formação de um profissional ativo, criativo e possuidor de um raciocínio crítico.

Dessa forma, é imprescindível o uso de metodologias que promovam a capacidade de expressão oral e escrita, alcançada através de técnicas para envolvimento e participação ativa dos alunos ao longo dos componentes curriculares (“aprender fazendo”). A participação ativa dos alunos é promovida não só na interação docente-aluno, mas principalmente na interação aluno-aluno, visando tanto a consolidação dos conhecimentos e habilidades (“aprender ensinando”), quanto o desenvolvimento de habilidades sociais e colaborativas (“aprender em grupo”).

No âmbito das disciplinas, a metodologia de ensino e aprendizagem baseia-se no planejamento combinado de aulas expositivas, aulas práticas (laboratórios), trabalhos individuais e em equipe, a fim de articular da forma mais adequada a carga horária teórica e prática prevista para cada disciplina.

A carga horária teórica é tipicamente conduzida através de aulas expositivas e atividades de leitura e análise de textos. O Campus de Russas oferece salas de aula com projetor digital e quadro branco, além de várias ferramentas para compartilhamento de materiais didáticos. A carga horária prática é tipicamente conduzida através de atividades práticas em sala ou laboratório, e por trabalhos individuais e em equipe.

A carga teórica visa habilitar o aluno na definição, identificação, compreensão, descrição, e combinação dos fundamentos científico-tecnológicos relacionados com a ementa da disciplina. A carga horária prática visa habilitá-lo na análise, comparação, explicação, e aplicação dos conceitos teóricos em situações de cunho prático.

Os programas de ensino das disciplinas preveem o planejamento de carga horária para cada unidade e assunto das aulas. Esses programas são elaborados e mantidos pelas unidades curriculares, e quando sofrem alterações são enviados para aprovação pelo colegiado do curso. As unidades curriculares envolvem um conjunto de disciplinas afins e são compostas por docentes que ministram esse grupo de disciplinas. Dessa forma, permite-se uma visão mais abrangente da inter-relação das disciplinas com a integralização curricular.

As seguintes atividades são propostas de como conduzir as disciplinas para alcançar os objetivos de formação:

- Aulas presenciais expositivas;
- Apresentação de vídeos ou documentários;
- Grupos de discussão e debates;
- Aulas mescladas (expositivas e exercícios práticos);

- Preparação de resumos;
- Listas de exercícios;
- Trabalhos práticos individuais e em equipe;
- Preparação e apresentação de seminários;
- Resolução de problemas (PBL – Problem-based learning).

Essas atividades são conduzidas com apoio de:

- Espaços das salas de aula, laboratórios e bibliotecas;
- Listas de discussão, grupos em redes sociais;
- Ambientes virtuais de aprendizado;
- Sistemas de acompanhamento;
- Ferramentas e tecnologias do estado da prática.

O compartilhamento de materiais e abordagens de ensino entre os professores dos cursos permite uma unificação do conteúdo entre disciplinas lecionadas por professores diferentes.

Um aspecto importante é a preocupação dos docentes com alunos com deficiências, em especial, com deficiências auditivas e visuais. Algumas medidas incentivadas e/ou utilizadas pelos professores são:

- Preparar extenso material de apoio didático, como notas de aula e apresentações, com bastante conteúdo visual;
- Demonstrar e resolver exercícios passo a passo, permitindo que o aluno acompanhe o raciocínio em questão;
- Contar com um intérprete de Linguagem Brasileira de Sinais (LIBRAS) em sala de aula, quando a turma tiver aluno deficiente auditivo;
- Disponibilizar previamente o material de apoio, permitindo que o intérprete e os alunos se preparem antecipadamente para a aula;
- Ministrando a aula de maneira pausada, dando tempo ao intérprete traduzir o conteúdo;
- Minimizar ruídos em sala;
- Discutir assuntos em sala de forma que uma pessoa fale por vez, com o objetivo de facilitar a tradução do intérprete;
- Escrever grande parte do texto falado no quadro;
- Conversar constantemente com o intérprete de LIBRAS, com o objetivo de descobrir as melhores abordagens e práticas a serem adotadas em sala de aula;
- Utilizar um intérprete acompanhando os deficientes auditivos na realização de trabalhos de grupo, em horários agendados;
- Utilizar tecnologias de reconhecimento automático de texto falado, e projetar esse texto, quando a utilização de um intérprete for impossibilitada.
- Usar movimentos manuais com as mãos de um deficiente visual, com o intuito de exemplificar um conceito visual;

- Fazer uso de material físico, concreto e palpável para a explicação de conceitos tridimensionais;
- Usar áudio-descrição, quando necessário;
- Ler material em voz alta;
- Utilizar tecnologias de leitura automática de telas de computadores;
- Reservar assentos preferenciais aos deficientes, nas primeiras fileiras.

Além dessas medidas, os professores e a coordenação do curso têm mantido conversas frequentes com a Secretaria de Acessibilidade da Universidade Federal do Ceará, a fim de sanar eventuais dúvidas sobre como atender os alunos deficientes de maneira apropriada e efetiva.

3.5 Procedimento de avaliação dos processos de ensino e aprendizagem

De modo geral, os mecanismos de avaliação da aprendizagem do aluno em sala são muito particulares a cada professor que os determina no momento da elaboração do plano de ensino. Entretanto, entendendo que a aprendizagem não se dá através da simples memorização de conteúdos, mas sim, a partir da sua compreensão e contextualização, o curso de Engenharia de Software de Russas pretende incentivar a definição conjunta entre docentes e discentes das formas de avaliação e a utilização de instrumentos diversos, como:

- Provas objetivas e subjetivas, que avaliam o entendimento e aplicação dos conceitos;
- Trabalhos individuais ou em grupo que avaliam, a prática da aplicação dos conceitos e normalmente feitos fora do horário de aula;
- Avaliações práticas individuais, que avaliam a execução prática dos conhecimentos da disciplina, aplicados em laboratório, no horário de aula, especialmente nas disciplinas de programação.
- Realização de seminários, que avaliam a habilidade de transmitir ideias com clareza na forma verbal;
- Elaboração de relatórios, que avaliam a capacidade de escrever textos técnicos e científicos da área;
- Construção de protótipos e projetos, que avaliam a competência técnica para a realização das diversas etapas do desenvolvimento de um software, obedecendo seu cronograma em fases.

Para atender alunos com deficiência auditiva, as avaliações são disponibilizadas previamente aos intérpretes de LIBRAS, para que estes possam gravar a prova. Essa prova gravada é exibida somente para o aluno com deficiência auditiva, em local e horário separados dos demais alunos. Alunos com deficiência visual têm o acompanhamento de um monitor para a leitura da avaliação. Além disso, alunos deficientes têm mais tempo para resolver a avaliação.

Ao lado disso, pretende-se a cada início de ano, realizar alguns fóruns de avaliação dos resultados do desempenho dos alunos em relação aos objetivos de cada

disciplina e atividade, a fim de detectar dificuldades na aprendizagem, replanejar e tomar decisões que diminuam o represamento e evasão de alunos.

A avaliação dos alunos nas diversas disciplinas segue o regimento geral aprovado pelo Conselho Nacional de Educação, conforme Parecer no 218/82, de 4 de maio de 1982 e atualizado em junho de 2011, o Regimento Geral da UFC e a Resolução No 12/CEPE, de 19 de junho de 2008 da Universidade Federal do Ceará que trata das regras para aprovação e reprovação por nota ou por falta nas disciplinas. A avaliação do rendimento escolar por disciplina abrange a assiduidade e a eficiência, ambas eliminatórias por si mesmas. Na verificação da assiduidade, será aprovado o aluno que frequentar 75% (setenta e cinco por cento) ou mais da carga horária da disciplina, vedado o abono de faltas. Na verificação da eficiência, será aprovado por média o aluno que, em cada disciplina, apresentar média aritmética das notas resultantes das avaliações progressivas igual ou superior a 07 (sete). O aluno que apresentar a média igual ou superior a 04 (quatro) e inferior a 07 (sete), será submetido à avaliação final. Nesse caso, o aluno será aprovado se obtiver nota igual ou superior a 04 (quatro) na avaliação final e média final (a média aritmética entre a média das avaliações progressivas e a avaliação final) igual ou superior a 05 (cinco).

A resolução No 12/CEPE dispõe sobre as reprovações por assiduidade: a) o estudante de graduação que contrair duas reprovações por frequência na mesma disciplina ou atingir um total de quatro reprovações por frequência em disciplinas do curso terá sua matrícula do semestre subsequente bloqueada; b) Esta Resolução não se aplica às reprovações ocorridas em Monografia, Estágio e Projeto Final de Curso c) O desbloqueio da matrícula só poderá ser feito após assinatura de Termo de Compromisso no qual o estudante atestará que está ciente de que qualquer outra reprovação por frequência causará o cancelamento definitivo de sua matrícula.

3.6 Estágio Curricular Supervisionado

O estágio curricular supervisionado é um componente curricular sugerido pelas Diretrizes Curriculares e visa consolidar e articular as competências desenvolvidas ao longo do curso por meio das demais atividades formativas, de caráter teórico ou prático, permitindo assim que o egresso vivencie situações, contextos e organizações próprios da atuação profissional.

A carga horária do estágio é de 320 horas, distribuídas preferencialmente nos dois últimos semestres do curso. O estágio supervisionado será realizado em empresas conveniadas com a UFC através da sua Agência de Estágios. A integralização da carga horária será implementada através das atividades “Estágio Supervisionado I”, com 160 horas, e “Estágio Supervisionado II”, com 160 horas, realizadas fora do âmbito da UFC em estágios em empresas, ou sob a orientação do professor responsável pelo Núcleo de Soluções em Software.

As formas e condições de realização, bem como formas de acompanhamento e avaliação das atividades de estágio estão regulamentadas em portaria específica. Mais informações sobre o Estágio Supervisionado podem ser encontradas no Regulamento para o Estágio Supervisionado (Anexo I).

O Núcleo de de Soluções em Software (N2S) foi criado com o objetivo de suprir as necessidades de sistemas para uso interno do campus e como o provimento de estágio para estudantes dos cursos de graduação na área de TIC do Campus de Russas. Como principal motivação para a criação e o eventual crescimento do Núcleo de Soluções em Software, pode-se destacar a baixa absorção dos alunos de graduação no mercado de desenvolvimento de software da cidade de Russas, devido a poucas empresas de TIC instaladas, e o crescimento da demanda de software por parte dos parceiros do Núcleo.

O N2S é estruturado para funcionar nos moldes de uma fábrica de software com processo definido e projetos com clientes reais. Um processo de desenvolvimento foi elaborado e implantado como forma de padronizar as atividades dos alunos no desenvolvimento de software, e incorporar melhores práticas de Engenharia de Software de acordo com metodologias, métodos e modelos de maturidade de processo que já são largamente utilizados na indústria de software e academia.

3.7 Trabalho de Conclusão de Curso

O Curso de Bacharelado em Engenharia de Software possui um Trabalho de Conclusão de Curso (TCC), que envolva todos os procedimentos de uma investigação técnico-científica, a ser desenvolvida pelo estudante ao longo dos dois últimos semestres do curso. Procura-se a aplicação do método científico nos trabalhos TCC, que tem como objetivo principal a aplicação das tecnologias abordadas nas disciplinas aspectos específicas de engenharia de software. Considera-se sempre a evolução da área nas atividades.

O trabalho poderá ser desenvolvido nas diversas áreas de pesquisa em engenharia de software, tais como gerência de projetos, qualidade de software, reuso de software, entre outros temas que podem ser acordados com o professor orientador.

O TCC se desdobra em dois semestres, na forma de três componentes curriculares, denominados: Projeto de Pesquisa Científica e Tecnológica - PPCT (2 créditos), Trabalho de Conclusão de Curso I - TCC I (2 créditos, ofertados simultaneamente à disciplina PPCT, no sétimo semestre do curso) e Trabalho de Conclusão de Curso II - TCC II (6 créditos, ofertado no oitavo semestre do curso).

Enquanto o professor da disciplina PPCT contribui com técnicas para elaboração do projeto do trabalho de conclusão de curso, apresentando cada uma de suas etapas de forma conceitual e aplicada, cabe ao professor orientador (TCC I e TCC II) as responsabilidades relacionadas ao conteúdo do trabalho e acompanhamento das atividades do aluno.

No TCC I, o aluno elabora o Projeto de atividades a serem executadas e realiza estudos preliminares que são aprofundados no TCC II. O Projeto deve ser desenvolvido sob a supervisão de um professor do curso, que deverá ser seu orientador na elaboração do TCC I, objeto da atividade TCC II. Durante o TCC II, o aluno deve terminar de desenvolver o trabalho iniciado no TCC I e redigir o texto final da Monografia, devendo estar atento aos requisitos e prazos da Universidade.

O trabalho deverá ser formatado e entregue em formato acadêmico e defendido perante uma banca de três professores da área, de acordo com os critérios gerais da UFC. Mais informações sobre o TCC podem ser encontradas no Regulamento para o Trabalho de Conclusão de Curso (Anexo II).

3.8 Atividades complementares

As atividades complementares possibilitam o reconhecimento de habilidades e competências do aluno, inclusive adquiridas fora do ambiente escolar, contribuindo para a flexibilização do currículo com a contabilização no histórico escolar de vivências adquiridas fora da sala de aula. Trata-se, portanto, de componentes curriculares enriquecedores da formação.

Na UFC, essas atividades encontram-se normatizadas pela Resolução No. 07/CEPE, de 17 de Junho de 2005, segundo a qual:

Art. 2o. – São consideradas atividades complementares:

I – Atividades de iniciação à docência;

II – Atividades de iniciação à pesquisa;

III – Atividades de extensão;

IV - Atividades artístico-culturais e esportivas;

V – Atividades de participação e/ou organização de eventos;

VI – Experiências ligadas à formação profissional e/ou correlatas;

VII – Produção Técnica e/ou Científica;

VIII – Vivências de gestão;

IX – Outras atividades, estabelecidas de acordo com o Art. 3o. desta Resolução.

Art. 3º. – As Coordenações de Cursos de Graduação poderão aprovar normatizações específicas, incluindo estratégias pedagógico-didáticas não previstas no Art. 2o. Desta Resolução e estipulando carga horária mínima integralizada ou período cursado das Atividades Complementares.

Em função de oferecer aos alunos a experiência prática requisitada pelo mercado de Tecnologia da Informação e Comunicação, estes são incentivados a desenvolver atividades diversas nos laboratórios de ensino de informática, de forma a contemplar situações fictícias e reais e aprender a trabalhar em equipe. Os professores do curso são incentivados, dessa forma, a ofertar oportunidades de trabalho a serem desenvolvidos nos laboratórios do campus. As atividades complementares serão realizadas de acordo

com a resolução anterior que dispõe sobre estas atividades nos Cursos de Graduação da UFC.

As Atividades Complementares são parte obrigatória da integralização curricular do curso, sendo exigidas 288 horas. Cada categoria especificada na resolução anterior engloba uma série de diferentes atividades que podem ser aproveitadas, a descrição dessas atividades consta no Regulamento de Atividades Complementares, disponibilizado no website do curso. Também no website, após aprovação deste Projeto Pedagógico, será disponibilizado um tutorial que ressalta a importância das atividades complementares para a formação dos discentes, e demonstram exemplos de atividades que podem ser aproveitadas. No último semestre do curso, os alunos são orientados a protocolar a atividades complementares realizadas, a serem validadas pela Coordenação. Mais informações sobre as atividades complementares podem ser encontradas no Regulamento das Atividades Complementares (Anexo III).

Parte da carga horária curricular do curso (288 horas que representam 18 créditos) deve ser obtida através de atividades complementares. Essas atividades visam diversificar a formação do aluno através de projetos de graduação, atividades artístico-culturais, cursos e participação em eventos científicos, experiência em gestão, entre outras atividades.

Abaixo são descritos os objetivos e metodologia das atividades complementares fomentadas pela própria coordenação do curso, docentes e unidade acadêmica, que servem como complementação da formação.

Monitoria de Iniciação à Docência:

A atividade de monitoria de iniciação a docência engloba o acompanhamento de uma disciplina junto com o docente responsável, auxiliando em aulas de exercício, correção de trabalhos e listas de exercícios, bem como oferecendo plantão para esclarecimento de dúvidas.

As atividades de monitoria auxiliam os alunos com maior dificuldade, permitindo que esses alunos sejam acompanhados por alunos mais experientes e pelos professores do curso. Dentre as disciplinas contempladas por bolsistas de monitoria, duas disciplinas que formam a base de um curso de Engenharia de Software podem ser citadas: Matemática Básica e Fundamentos de Programação.

Projetos de iniciação à pesquisa e extensão:

Alunos com bom rendimento acadêmico têm a oportunidade de participar dos grupos de pesquisa e de trabalho, com a possibilidade de obtenção de bolsas de iniciação científica oferecidas por agências governamentais (PIBIC do CNPq, FUNCAP, etc.). O desenvolvimento de trabalhos de iniciação científica contribui tanto para o aprimoramento dos conhecimentos técnicos do aluno, como para a obtenção de experiência no desenvolvimento de pesquisas e no relacionamento com pesquisadores e com outros alunos. Projetos de extensão também promovem oportunidade interessante para os alunos atuarem em ações mais ligadas a sociedade e/ou a pesquisa aplicada e

também contam com a possibilidade de obtenção de bolsas de extensão em programas da Pró-Reitoria de Extensão.

Iniciação Acadêmica:

Alunos com situação de vulnerabilidade socioeconômica podem participar do programa de Iniciação Acadêmica, que visa principalmente evitar a evasão. Através desse programa, os alunos participantes desenvolvem atividades de iniciação científica, de extensão, docência, ou atuam em projetos e atividades acadêmicas de natureza técnica ou administrativa. Com a participação nesse programa, objetiva-se ao aluno obter uma percepção maior sobre o curso, grupos de pesquisa e projetos dentro da Universidade.

Projetos de Monitoria de Graduação:

Esses projetos visam a melhoria da oferta dos cursos de graduação da UFC. No Campus de Russas, os projetos de monitoria de graduação podem promover estudos pontuais para melhoria em disciplinas e atividades ofertadas, tais como adoção de ferramentas, melhores práticas, etc.

Programa de Aprendizagem Cooperativa em Células Estudantis:

O Programa de Aprendizagem Cooperativa em Células Estudantis (PACCE) tem como principal objetivo colaborar no combate à evasão discente e, conseqüentemente, melhorar a taxa de conclusão nos cursos de graduação da UFC. A principal estratégia utilizada é a difusão de células estudantis – grupos de estudo que utilizam a metodologia de aprendizagem cooperativa. A aprendizagem cooperativa promove uma maior interação entre os estudantes e a construção de relacionamentos positivos. Com isso, há um encorajamento mútuo entre os alunos da graduação no enfrentamento de problemas acadêmicos e extra-acadêmicos, contribuindo para um melhor rendimento e aprovação em disciplinas da graduação. O programa oferece bolsas para alunos da graduação.

Eventos Acadêmicos:

Anualmente, são promovidos no Campus de Russas e/ou em várias cidades do Vale do Jaguaribe e do Litoral Leste uma série de eventos acadêmicos, científicos e tecnológicos, que visam propiciar aos alunos o contato com tendências no mercado de trabalho, em pesquisas acadêmicas, e também de abrir espaço para que os alunos apresentem seus trabalhos universitários.

SESCOMP – Semana da Engenharia de Software e Ciência da Computação: evento que inclui palestras, minicursos e painéis. Esse evento é de grande interesse e se destaca, pois abrem espaço para discussões de assuntos que interessam à toda comunidade e para a apresentação de palestras e de trabalhos que vêm sendo desenvolvidos.

Encontros Universitários: evento anual de divulgação de projetos de iniciação científica, extensão e docência. Nesses encontros alunos, professores e a comunidade

realizam discussão e troca de ideias e experiências sobre assuntos acadêmicos e de mercado.

Vale Web Livre: maior evento de desenvolvimento web do interior do Ceará, realizado pela Comunidade Vale Livre e que conta com programadores, designers, professores, alunos e entusiastas de software livre com o propósito de compartilhar conhecimento.

FLISOL – Festival Latino-Americano de Instalação de Software Livre: evento internacional realizado anualmente e que ocorre de forma simultânea em diversas cidades da América Latina, organizado de maneira descentralizada, com o objetivo de promover o uso de software livre, apresentando sua filosofia, alcance, avanços e desenvolvimento ao público geral.

Maratona de programação:

A maratona de programação permite que os alunos aprofundem seus conhecimentos em programação, buscando a solução de problemas não triviais. Os alunos viajam e participam de competições, onde a troca de experiências entre as diversas equipes aprimora o aprendizado dos alunos em geral.

Atividades Conjuntas de Graduação e Pós-graduação:

O Departamento de Computação e o Departamento de Teleinformática da UFC em Fortaleza poderão contribuir fazendo com que suas atividades de pós-graduação, stricto e lato sensu, interajam com o curso de Engenharia de Software em Russas.

- Oferta de seminários de pesquisa abertos à participação de professores e alunos de Russas;
- Criação de projetos e grupos de pesquisa envolvendo as duas unidades;
- Oferta de palestras em Russas, Fortaleza e demais cidades;
- Oportunidade de qualificação acadêmica para professores e alunos em seus cursos de pós-graduação.

Mobilidade Acadêmica:

O MEC introduziu o programa de Mobilidade Acadêmica que permite o intercâmbio entre alunos de IFES. Este programa se mostra amplamente adequado para que alunos dos cursos de Russas possam por períodos determinados conhecer a realidade da formação de outros cursos, ampliando suas possibilidades de formação. Os alunos do curso de Engenharia de Software são incentivados a participar desse tipo de programa para manter contato com outros sistemas educacionais e possam consolidar o conhecimento adquirido.

3.9 Ementário e bibliografias

Fundamentos de Programação

Ementa:

Algoritmos, Conceitos Fundamentais de Programação, Expressões, Controles de Fluxo, Funções e

Procedimentos, Vetores e Matrizes, Cadeias de Caracteres, Tipos Estruturados e Arquivos.

Bibliografia Básica:

1. MEDINA, M.; FERTIG, C. Algoritmos e programação: teoria e prática 2ed. Novatec, 2004. ISBN: 9788575220733/857522073X.
2. ASCENCIO, A. F. G.; CAMPOS, E. A. V. Fundamentos da programação de computadores: algoritmos, Pascal, C/C++ e Java. 2 ed. Prentice Hall, 2007. ISBN: 978576051480.
3. CELES, W.; CERQUEIRA, R.; RANGEL, J. L. Introdução à estrutura de dados: com técnica de programação em C. Elsevier, 2004. ISBN: 8535212280.

Bibliografia Complementar:

1. FORBELLONE, A. L. V. ; EBERSPACHER, H.F. Lógica de programação: a construção de algoritmos. 3 ed. Prentice Hall, 2005.
2. Fundamentos de Programação - 3ª Ed. Joyanes, Luis Aguilar; Joyanes, Luis Aguilar. Amgh Editora.
3. Fundamentos De Programação Usando C - 4ª Ed. De Sá, Marques, Lidel – Zamboni.
4. Lógica de Programação - 3ª Edição. Forbellone, André L. V. Makron Books.
5. Algoritmos - Lógica para Desenvolvimento de Programação de Computadores. Oliveira, Jayr Figueiredo de; Manzano, José Augusto N. G.. Editora Érica

Ética Profissional

Ementa:

Conceituação e fundamentos da ética. Ética e Direitos Humanos. Responsabilidade social, relações étnico-raciais e africanidades. Responsabilidade Ambiental. Ética profissional, deveres profissionais. Propriedade intelectual e industrial. Licenças de Softwares. O direito aplicado à Computação: legislação e procedimentos. Código de Ética e Práticas Profissionais na Computação.

Bibliografia Básica:

1. Ética e direito: uma perspectiva integrada, ALMEIDA, G. A.; CHRISTMANN, M. O., 3a edição 2009 Atlas.
2. Ética em computação, MASIERO, P. C., 2008, EDUSP, 9788531405754.
3. Ética na Computação: uma abordagem baseada em casos, BARGER, R. N., 20011, LTC, 9788521617761.

Bibliografia Complementar:

1. As pessoas na organização, LIMONGI-FRANÇA, A.C., 2002, Gente.
2. O direito à intimidade na era da informática, LIMBERGER, T., 2007, Livraria do Advogado.
3. O direito do autor na multimídia, CARBONI, G. C., 2003, Quartier Latim, 8588813297
4. O direito na sociedade da informação, PAESANI, L., 2007, Atlas.
5. Tecnologias da inteligência: futuro do pensamento, LEVY, P., 1995, Prentice Hall, 9788585490157.
6. The Case of the Killer Robot: Stories about the Professional, Ethical and Societal Dimensions of Computing, EPSTEIN, R. G., 1996, John Willey, 9780471138235.

Introdução à Engenharia de Software

Ementa:

Fundamentos de computação, arquiteturas de computadores e sistemas operacionais. Ambientes de processamento automatizado de informações. Evolução das profissões e características do profissional de engenharia de software. Conceitos básicos: dado, informação e conhecimento. Operações básicas com números binários. Mudança de base. Computador e seus elementos básicos. Acessibilidade Digital. Tecnologia da Informação Verde. Conceitos básicos de Engenharia de Software, programas e documentação. SWEBOK e as áreas de conhecimento, Etapas de desenvolvimento de um Software.

Bibliografia Básica:

1. FEDELI, Ricardo Daniel.; POLLONI, Enrico Giulio Franco; PERES, Fernando Eduardo. Introdução à ciência da computação. 2. ed. atual. São Paulo, SP: Cengage Learning, 2010. 250 p. ISBN 139788522108459 (broch.).
2. SOMMERVILLE, Ian, |d 1951-; OLIVEIRA, Kalinka; BOSNIC, Ivan. Engenharia de software. 9. ed. São Paulo, SP: Pearson/ Prentice Hall, 2011. 529 p. ISBN 9788579361081 (broch.).

3. PRESSMAN, R., Engenharia de software. 7. ed. São Paulo: Pearson, 2011. ISBN 9788563308337

Bibliografia Complementar:

1. PFLEEGER, Shari Lawrence. Engenharia de software: teoria e prática. 2. ed. São Paulo, SP: Pearson/Prentice Hall, 2007. 537 p. ISBN 9788587918314 (broch.).

2. GUIMARÃES, Ângelo de Moura; LAGES, Newton Alberto de Castilho. Introdução a ciência da computação. Rio de Janeiro: Livros Técnicos e Científicos, 1984. 165p. (Ciência da computação) ISBN 852160372X.

3. VELLOSO, Fernando de Castro. Informática: conceitos básicos . 8. ed. rev. atual. Rio de Janeiro, RJ: Campus, 2011. ISBN 9788535243970.

4. HENNESSY, John L; PATTERSON, David A. Arquitetura de computadores: uma abordagem quantitativa. 4. ed. Rio de Janeiro, RJ: Elsevier, 2008. 494 p. ISBN 9788535223552 (broch.).

5. TURBAN, Efraim. Introdução a sistemas de informação: uma abordagem gerencial. Rio de Janeiro, RJ: Elsevier, 2007. 364 p. ISBN 978853522067 (broch.).

Matemática Básica

Ementa:

Lógica: conectivos lógicos, tabela verdade, fórmulas equivalentes. Conjuntos: notação, operações, propriedades das operações, diagramas de Venn, partição, cardinalidade, conjunto das partes, produto cartesiano. Relações: definições, terminologia, propriedades. Funções (polinomiais, exponenciais e logarítmicas, entre outras): definições, terminologia, propriedades.

Bibliografia Básica:

1. GERSTING, Judith L. Fundamentos matemáticos para a ciência da computação: um tratamento moderno de matemática discreta . 5. ed. Rio de Janeiro: Livros Técnicos e Científicos, 2004. 597 p. ISBN 8521614225 (broch.).

2. SILVA, Sebastião Medeiros da; SILVA, Elio Medeiros da; SILVA, Ermes Medeiros da. Matemática básica para cursos superiores. São Paulo: Atlas, 2002. 227 p. ISBN 8522430357 (broch.).

3. IEZZI, Gelson; MURAKAMI, Carlos. Fundamentos de matemática elementar: 1 : conjuntos, funções . 8. ed., São Paulo, SP: Atual, 2004. 374 p. ISBN 8570562705 (broch.).

Bibliografia Complementar:

1. ROSEN, Kenneth H. Matemática discreta e suas aplicações. 6. ed. São Paulo, SP: McGraw-Hill, 2009. xxi, 982 p. ISBN 9788577260362 (broch.).

2. ALENCAR FILHO, Edgard de. Iniciação a lógica matemática. 21. ed. São Paulo: Nobel, 1990. 203p ISBN 9788521304036.

3. MENEZES, Paulo Blauth; UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL. Matemática discreta para computação e informática. 3. ed. Porto Alegre, RS: Bookman, 2010. 350 p (Livros didáticos. 16).

4. DEMANA, Franklin D. Pré-cálculo. São Paulo: Addison-Wesley, 2009. 380 p. ISBN 9788588639379 (broch.).

5. MENEZES, Paulo Blauth; TOSCANI, Laira V.; GARCÍA LÓPEZ, Javier. Aprendendo matemática discreta com exercícios. Porto Alegre, RS: Bookman, 2009. 356p. (Livros didáticos informática ufrgs; v. 19) ISBN 9788577804719 (broch.).

6. SCHEINERMAN, Edward R. Matemática discreta: uma introdução . São Paulo: Cengage Learning, 2011. 573 p. ISBN 9788522107964 (broch.).

Arquitetura de Computadores

Ementa:

Sistemas numéricos. Aritmética binária: ponto fixo e ponto flutuante. Organização de computadores: memórias, unidade central de processamento, unidades de entrada e unidades de saída. Linguagens de montagem. Modos de endereçamento, conjunto de instruções. Mecanismos de interrupção e de exceção. Barramento, comunicações, interfaces e periféricos. Organização de memória. Memória auxiliar. Arquiteturas RISC e CISC. Pipeline.

Bibliografia Básica:

1. STALLINGS, W. Arquitetura e organização de computadores. 8. ed. São Paulo: Prentice Hall, 2010.

2. TANENBAUM, A. Organização estruturada de computadores. 5. ed. São Paulo: Pearson Prentice Hall, 2007.
3. MURDOCCA, Miles J.; HEURING, Vincent P. Introdução à arquitetura de computadores. Rio de Janeiro: Campus, 2000. 512 p. ISBN 8535206841.

Bibliografia Complementar:

1. DELGADO, José; RIBEIRO, Carlos. Arquitetura de computadores. 4. ed. rev. atual. Lisboa: FCA, 2010.
2. HENNESSY, John L.; PATTERSON, David A. Arquitetura de computadores: uma abordagem quantitativa. 4. ed. Rio de Janeiro: Elsevier, 2008.
3. WEBER, Raul Fernando. Fundamentos de arquitetura de computadores. 3. ed. Porto Alegre, RS: Sagra Luzzatto, 2008.
4. FEDELLI, Ricardo Daniel; POLLONI, Enrico Giulio Franco; PERES, Fernando Eduardo. Introdução à ciência da computação. 2. ed. São Paulo, SP: Cengage Learning, 2009.
5. MONTEIRO, M. A. Introdução à Organização de Computadores. 5. ed. LTC, Rio de Janeiro, 2008.

Laboratório de Programação

Ementa:

Linguagem de Programação C: Introdução, Tipos de Dados, Operadores e Expressões, Entrada e Saída, Fluxo de Controle, Funções e Estrutura de Programa, Apontadores e Vetores, Strings, Tipos Estruturados, Alocação Dinâmica, Arquivos. Código Limpo (Boas Práticas de Programação). Resolução de Problemas de Programação. Depuração. Versionamento.

Bibliografia Básica:

1. KERNIGHAN, B. W.; RITCHIE, D. M. C: A Linguagem de Programação Padrão ANSI. 2ed. Rio de Janeiro. Elsevier. 1989.
2. CELES, W.; CERQUEIRA, R.; RANGEL, J. L. Introdução a estrutura de dados: com técnicas de programação em C. 2ed. Rio de Janeiro. Campus. 2004
3. PRINZ, P; CRAWFORD, T. C In a Nutshell: The Definitive Reference. 2ed. O'Reilly. 2016.
4. SCHILDT, H. C – Completo e Total. 3ed. São Paulo. Makron Books. 1996.

Bibliografia Complementar:

1. SKIENA, S. S.; REVILLA, M. A. Programming Challenges: The Programming Contest Training Manual. New York. Springer. 2003.
2. MARTIN, R. C. Código Limpo: Habilidades Práticas do Agile Software. Rio de Janeiro. Alta Books. 2011.
3. MEDINA, M.; FERTIG, C. Algoritmos e programação: teoria e prática 2ed. Novatec, 2004. ISBN: 9788575220733/857522073X.
4. ASCENCIO, A. F. G.; CAMPOS, E. A. V. Fundamentos da programação de computadores: algoritmos, Pascal, C/C++ e Java. 2 ed. Prentice Hall, 2007. ISBN: 978576051480.
5. FORBELLONE, A. L. V. ; EBERSPACHER, H.F. Lógica de programação: a construção de algoritmos. 3 ed. Prentice Hall, 2005.

Estruturas de Dados

Ementa:

Noções de análise de algoritmos, recursividade, tipos abstratos de dados, algoritmos de ordenação, listas sequenciais e encadeadas, pilhas, filas e árvores.

Bibliografia Básica:

1. FEOFILOFF, Paulo. . Algoritmos em linguagem C. Rio de Janeiro: Campus, 2008. 208p. ISBN 9788535232493 (broch.).
2. DROZDEK, Adam. Estrutura de dados e algoritmos em C++. São Paulo: Thomson, 2002. 579p ISBN 852210295 (broch.).
3. ZIVIANI, Nivio; BOTELHO, Fabiano Cupertino. Projeto de algoritmos: com implementações em java e C++. São Paulo, SP: Thomson Learning, 2007. 620 p. ISBN 8522105251 (broch.).

Bibliografia Complementar:

1. CORMEN, Thomas H. Algoritmos: teoria e prática. Rio de Janeiro: Elsevier, 2002. 916 p. ISBN 8535209263 (broch.).

2. SZWARCFITER, Jayme Luiz; MARKENZON, Lilian. Estrutura de dados e seus algoritmos. Rio de Janeiro: LTC, 1994. 320 p. ISBN 8521610149.
3. ASCENCIO, Ana Fernanda Gomes; CAMPOS, Edilene Aparecida Veneruchi de. Programação de computadores: algoritmos, Pascal, C/C++ e java. 2. ed. São Paulo, SP: Prentice Hall, 2007. 434 p. ISBN 8576051480 (broch.).
4. JOYANES AGUILAR, Luis. Fundamentos de programação: algoritmos, estrutura de dados e objetos. São Paulo, SP: McGraw-Hill, 2008. 690 p. ISBN 9788586804960 (broch.).
5. CELES, Waldemar; CERQUEIRA, Renato; RANGEL, José Lucas. Introdução a estrutura de dados: com técnicas de programação em C. Rio de Janeiro: Elsevier, 2004. 294 p. (Campus, Sociedade Brasileira de Computação) ISBN 8535212280 (broch.).

Matemática Discreta

Ementa:

Técnicas de demonstração: exaustiva, direta, contraposição, absurdo, indução (fraca e forte). Somatórios: notação, propriedades, séries aritméticas, geométricas e harmônicas, algumas fórmulas de somatório úteis. Teoria dos números: divisibilidade, primos, teorema fundamental da aritmética, aritmética modular, aplicações. Contagem: princípios da multiplicação e da adição, princípio da inclusão e exclusão, princípio das casas dos pombos, permutações, combinações, teorema binomial, triângulo de pascal, permutações e combinações com repetições. Relações: fechos, ordem parcial e total, relações e classes de equivalência.

Bibliografia Básica:

1. GERSTING, Judith L. Fundamentos matemáticos para a ciência da computação: um tratamento moderno de matemática discreta . 5. ed. Rio de Janeiro: Livros Técnicos e Científicos, 2004. 597 p. ISBN 8521614225 (broch.).
2. MENEZES, Paulo Blauth; UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL. Matemática discreta para computação e informática. 3. ed. Porto Alegre, RS: Bookman, 2010. 350 p (Livros didáticos. 16).
3. ROSEN, Kenneth H. Matemática discreta e suas aplicações. 6. ed. São Paulo, SP: McGraw-Hill, 2009. 982 p. ISBN 9788577260362 (broch.).

Bibliografia Complementar:

1. ALENCAR FILHO, E. Iniciação à lógica matemática. 21. ed. São Paulo: Nobel, 2008. ISBN:9788521304036.
2. SILVA, Sebastião Medeiros da; SILVA, Elio Medeiros da; SILVA, Ermes Medeiros da. Matemática básica para cursos superiores. São Paulo: Atlas, 2002. 227 p. ISBN 8522430357 (broch.).
3. MENEZES, Paulo Blauth; TOSCANI, Laura V.; GARCÍA LÓPEZ, Javier. Aprendendo matemática discreta com exercícios. Porto Alegre, RS: Bookman, 2009. 356p. (Livros didáticos informática ufrgs ; ; v. 19) ISBN 9788577804719 (broch.).
4. SCHEINERMAN, Edward R. Matemática discreta: uma introdução . São Paulo: Cengage Learning, 2011. 573 p. ISBN 9788522107964 (broch.).
5. HUNTER, David J. Fundamentos da matemática discreta. Rio de Janeiro, RJ: LTC, 2011. 235 p. ISBN 9788521618102 (broch.).

Introdução a Processos e Requisitos de Software

Ementa:

Conceitos básicos de Processo de Software. Fases de um Processo de Software e geração de artefatos a cada fase. Tipos de Processo de Software: tradicionais e ágeis. Definição e introdução à elicitação e análise de requisitos de software.

Bibliografia Básica:

1. PFLEEGER, Shari Lawrence. Engenharia de software: teoria e prática. 2. ed. São Paulo, SP: Pearson/ Prentice Hall, 2007. 537 p. ISBN 9788587918314 (broch.).
2. PRESSMAN, Roger S. Engenharia de software. 7. ed. São Paulo: McGraw-Hill, 2011. 720 p. ISBN 9788563308337(broch.).
3. SOMMERVILLE, Ian; OLIVEIRA, Kalinka; BOSNIC, Ivan. Engenharia de software. 9. ed. São Paulo, SP:

Pearson/ Prentice Hall, 2011. 529 p. ISBN 9788579361081 (broch.).

Bibliografia Complementar:

1. SCOTT, Kendall. O processo unificado explicado. Porto Alegre: Bookman, 2003. 160 p. ISBN 8536302313 (broch.).
2. KOSCIANSKI, André; SOARES, Michel dos Santos. Qualidade de software. 2. ed. São Paulo, SP: Novatec, 2007. 395p. ISBN 8575221129 (broch.).
3. WIEGERS, Karl Eugene. Software requirements: practical techniques for gathering and managing requirements throughout the product development cycle . 2. ed. Redmond: Microsoft, 2003. 516 p. ISBN 0735618798.
4. WITHALL, Stephen. Software requirement patterns. Redmond, Wash.: Microsoft Press, 2007. 366 p. (Best practices) ISBN 9780735623989 (broch.).
5. THAYER, Richard H. Software requirements engineering. 2. ed. California: IEEE Computer Society, 2000. 528 p.

Programação Orientada a Objetos

Ementa:

Introduzir o paradigma de Programação Orientada a Objetos (OO), juntamente com seus conceitos de classes, objetos, herança, encapsulamento e polimorfismo, além dos conceitos de Interfaces e exceções que são inerentes às linguagens de programação orientadas a objetos. Desenvolvimento de um pequeno sistema baseados no paradigma de programação OO.

Bibliografia Básica:

1. DEITEL, H. M. Java: como programar. 8 ed. Prentice Hall, 2010. ISBN: 9788576055631
2. HORSTMANN, Cay S. Core Java: volume I - fundamentos. 8. ed. São Paulo, SP: Pearson, 2009. xiii, 383 p. ISBN 9788576053576
3. MANZANO, José Augusto N. G.; COSTA Jr., Roberto Affonso da. Java 7 - Programação de Computadores - Guia Prático de Introdução, Orientação e Desenvolvimento. 1. ed. Editora Érica, 2011. ISBN: 9788536503745

Bibliografia Complementar:

1. BARNES, D.J.; KOLLING, M. Programação orientada a objetos com Java: uma introdução prática usando BLUEJ. 4 ed. Prentice Hall Brasil, 2009. ISBN: 9788576051879
2. SIERRA, K.; BATES, B., USE A CABEÇA! JAVA, 2 ed, 2007, ALTA BOOKS
3. DEITEL, H. M. C++ como programar. 5 ed. Prentice Hall, 2006. ISBN: 8576050560
4. MEYER, Bertrand. Object-Oriented Software Construction, Ed. Prentice Hall PTR, 1997.
5. RUMBAUGH, James; BLAHA, Michael. Modelagem e Projetos Baseados em Objetos, Editora Campus, 2006.

Probabilidade e Estatística

Ementa:

Fundamentos de análise combinatória. Conceito de probabilidade e seus teoremas fundamentais. Variáveis aleatórias. Distribuições de probabilidade. Estatística descritiva. Noções de amostragem. Distribuições amostrais: discreta e contínua. Inferência estatística: teoria da estimação e testes de hipóteses. Regressão linear simples. Correlação. Análise de variância.

Bibliografia Básica:

1. Larson, R.; Farber, B. Estatística aplicada.
2. William J. Stevenson. Estatística aplicada à Administração.. Harbra. 2001.
3. BARBETTA, Pedro Alberto; REIS, Marcelo Menezes; BORNIA, Antonio Cezar . Estatística para cursos de engenharia e informática.. 3. ed. São Paulo, SP.
4. HAZZAN, Samuel . Fundamentos de matemática elementar, 5 : combinatória, probabilidade.. 7. ed. São Paulo, SP: Atual, 2004.

Bibliografia Complementar:

1. Dancey, C.P.; Reidy, J..Estatística sem matemática para a psicologia.
2. Triola, Mário F. Introdução à estatística...

3. Meriam, J.L.; Kraige, L.G. Solving statistics problem in Matlab...
4. WALPOLE, Ronald E. Probabilidade e estatística : para engenharia e ciências.. 8. ed. São Paulo, SP: Pearson/ Prentice Hall, 2009.
5. Smith, Peter J.; Into Statistics;. Springer. 2001.
6. MAGALHÃES, Marcos Nascimento Noções de probabilidade e estatística.. 6. ed. rev. São Paulo: Editora da Universidade de São Paulo, 2005.

Linguagens de Programação

Ementa:

Conceitos básicos de LP: domínios de aplicação, influências no projeto, paradigmas, métodos de implementação, critérios de avaliação, evolução das linguagens. Análise léxica e sintática. Variáveis: identificadores, vinculações, verificação de tipos, escopo. Tipos de dados. Expressões e a declaração de atribuição. Abstração de processos: subprogramas. Abstração de dados e orientação à objetos. Noções de programação funcional. Noções de programação lógica.

Bibliografia Básica:

1. SEBESTA, Robert W. Conceitos de linguagens de programação. 9. ed. - Porto Alegre, RS: Bookman, 2011. 792 p. ISBN 9788577807918 (broch.).
2. TUCKER, Allen B. |; NOONAN, Robert. Linguagens de programação: princípios e paradigmas. 2. ed. São Paulo, SP: McGraw-Hill, 2008. 599 p. ISBN 9788577260447 (broch.).
3. WATT, David Anthony; FINDLAY, William; HUGHES, John. Programming language: concepts and paradigms. New York : Prentice Hall, 1990. 322p. ISBN 0137288743 (broch.).

Bibliografia Complementar:

1. DEITEL, Harvey M.; DEITEL, Paul J. C++: como programar. 5. ed. São Paulo, SP: Prentice Hall, 2006. 1163 p. ISBN 8576050560 (broch.).
2. AHO, Alfred V.; SETHI, Ravi; ULLMAN, Jeffrey D. Compiladores: princípios, técnicas e ferramentas. 2. ed. São Paulo, SP: Pearson/Addison Wesley, 2008. 634 p. ISBN 9788588639249 (broch.).
3. HOPCROFT, John E.; ULLMAN, Jeffrey D.; MOTWANI, Rajeev. Introdução à teoria dos autômatos, linguagens e computação. Rio de Janeiro: Campus, 2002. ISBN 8535210725.
4. BARWISE, Jon; ETCEMENDY, John; ALLWEIN, Gerard; BARKER-PLUMMER, Dave; LIU, Albert. Language, proof and logic. Stanford: CSLI, 2008. 587 p. ; (CSLI lecture notes ;v 23) ISBN 1 57586 374 X (broch.).
5. MENEZES, Paulo Blauth. Linguagens formais e autômatos. 5. ed. Porto Alegre: Bookman, 2008. 215 p. (Livros didáticos ; n.3 Série Livros Didáticos 3) ISBN 9788577802661 (broch.).
6. BARNES, David J.; KÖLLING, Michael. Programação orientada a objetos com java: uma introdução prática usando o blueJ. 4. ed. São Paulo, SP: Pearson Prentice Hall, 2009. xxii , 444 p. ISBN 9788576051879 (broch.).
7. URUBATAN, Rodrigo. Ruby on rails: desenvolvimento fácil e rápido de aplicações Web. São Paulo, SP: Novatec, 2009. 285 p. ISBN 9788575221846 (broch.).

Algoritmos em Grafos

Ementa:

Conceitos e definições de grafos: isomorfismo, conectividade, árvores, grafos direcionados e não-direcionados. Representação de grafos: matriz e listas de adjacências. Algoritmos de percurso em grafos. Ordenação topológica. Árvore geradora mínima. Caminhos mínimos. Fluxo máximo e multifluxo.

Bibliografia Básica:

1. CORMEN, T.; LEISERSON, C.; RIVEST, R.; STEIN, C. Algoritmos - Teoria e Prática. 3a edição, Editora Campus, 2012. ISBN-13: 978-8535236996.
2. DASGUPTA, S.; PAPADIMITRIOU, C.; VAZIRANI, U. Algoritmos. McGraw Hill, 2009. ISBN-13: 978-8577260324.
3. KLEINBERG, J.; TARDOS, E. Algorithm Design, Addison Wesley, 2005.

Bibliografia Complementar:

1. GOLDBARG, E.; GOLDBARG, M. Grafos – Conceitos, algoritmos e aplicações. Elsevier Acadêmico, 2012.

ISBN-13: 978-8535257168.

2. ZIVIANI, N. Projeto de Algoritmos com Implementações em Java e C++, Editora Cengage Learning, 2006.
3. TOSCANI, L. V.; VELOSO, P.A.S. Complexidade de Algoritmos.
4. GERSTING, J. L. Fundamentos matemáticos para a ciência da computação: um tratamento moderno de matemática discreta.
5. MENEZES, P.B. Matemática discreta para computação e informática.

Requisitos de Software

Ementa:

Definição de requisitos de produto, projeto, restrições, fronteiras de um sistema. Níveis de requisitos (necessidades, objetivos, requisitos dos usuários, requisitos de sistema). Fontes e técnicas de elicitação. Atributos de qualidade (Completeness, consistência, robustez, FURPS, SMART). Características de requisitos (testáveis, verificáveis e outras). Tipos (segurança, safety, usabilidade, desempenho). Especificação de requisitos. Documentação de requisitos (normas, tipos, audiência, estrutura, qualidade). Processo de requisitos. Gerência de requisitos. Modelagem de processos de negócios. Construção de protótipos para validar requisitos. Relação com testes de aceitação. Processos fundamentais (análise estática, simulação, verificação de modelos, análise de causa/efeito, priorização, análise de impacto, rastreabilidade). Padrões de análise. Interação entre requisitos e arquitetura. Revisões e inspeções.

Bibliografia Básica:

1. MACHADO, Felipe Nery Rodrigues. Análise e gestão de requisitos de software: onde nascem os sistemas. São Paulo: Érica, 2011. 286 p. ISBN 9788536503622.
2. COHN, Mike. User stories applied: for agile software development. Boston, Estados Unidos: Addison Wesley, 2004. 268 p. (Addison-Wesley signature series) ISBN 9780321205681.
3. SOMMERVILLE, Ian. Requirements Engineering: A good practice guide. 1. ed. Wiley, 1997. ISBN: 9780471974444

Bibliografia Complementar:

1. COCKBURN, Alistair. Escrevendo casos de uso eficazes: um guia prático para desenvolvedores de software. Porto Alegre: Bookman, 2005. 254 p. ISBN 9788536304571 (broch.).
2. WITHALL, Stephen. Software requirement patterns. Redmond, Wash.: Microsoft Press, 2007. 366 p. (Best practices) ISBN 9780735623989 (broch.).
3. SOMMERVILLE, Ian. Engenharia de software. 9. ed. São Paulo, SP: Pearson/ Prentice Hall, 2011. 529 p. ISBN 9788579361081 (broch.).
4. ROBERTSON, Suzanne; ROBERTSON, James. Mastering the requirements process. 2. ed. Upper Saddle River, NJ: Addison-Wesley, 2006. 560 p. : ISBN 0321419499.
5. WIEGERS, Karl Eugene. Software requirements: practical techniques for gathering and managing requirements throughout the product development cycle. 2. ed. Redmond, Wash.: Microsoft Press, 2003. 516 p. ISBN 0735618798.
6. THAYER, Richard H.; DORFMAN, M.; BAILIN, Sidney C. Software requirements engineering. 2. ed. Los Alamitos, Calif. : IEEE Computer Society Press, 2000. 483 p. ISBN 0818677384 (broch.).

Fundamentos de Banco de Dados

Ementa:

Visão geral do gerenciamento de banco de dados. Arquitetura de um Sistema Gerenciador de Banco de Dados. Modelagem e projeto de banco de dados: Modelo Entidade-Relacionamento, Modelo Relacional e Projeto de Bancos de Dados Relacionais. Noções de álgebra e cálculo relacional. SQL. Projeto Avançado: Restrições de Integridade e Normalização.

Bibliografia Básica:

1. SILBERSCHATZ, A.; SUDARSHAN, S. Sistema de banco de dados. Campus, 2006. ISBN: 9788535211078.
2. ELMASRI, R.; NAVATHE, S. B. Sistemas de banco de dados. 6 ed. Pearson/Addison-Wesley, 2011.

ISBN: 9788579360855.

3. HEUSER, C. A. Projeto de banco de dados. 6 ed. Bookman, 2009. ISBN: 9788577803828.

Bibliografia Complementar:

1. RAMAKRISHNAN, R.; GEHRKE, J. Sistemas de gerenciamento de banco de dados. McGraw-Hill, 2008.

2. DATE, C.J. Introdução a Sistemas de Banco de Dados. 8 ed. Campus, 2004. ISBN: 9788535212730.

3. OLIVEIRA, C.H. SQL: Curso prático. Novatec, 2002. ISBN: 9788575220245 .

4. LYNN, B. Use a cabeça! SQL. Alta Books, 2008. ISBN: 9788576022101.

5. KROENKE, D. M. Database Processing. 8. ed. Prentice Hall, 2001. ISBN: 9780130648396.

Análise e Projeto de Sistemas

Ementa:

Teorias, métodos, técnicas e ferramentas associadas ao projeto de software enquanto atividade sistemática. Técnicas orientadas a objeto para análise e projeto de sistemas. Diagramas UML de Análise e Projeto. Visão Geral de Padrões de Projeto.

Bibliografia Básica:

1. BEZERRA, E. Princípios de análise e projeto de sistemas com UML. 2. ed. rev. e atual. Rio de Janeiro: Campus, 2007. ISBN:9788535216967.

2. FOWLER, M. UML essencial: um breve guia para a linguagem-padrão de modelagem de objetos. 3. ed. Porto Alegre: Bookman, 2005. ISBN: 8536304545.

3. BOOCH, Grady; RUMBAUGH, James; JACOBSON, Ivar. UML: guia do usuário. 1. ed. Rio de Janeiro, RJ: Campus, 2006. 9788535217841.

Bibliografia Complementar:

1. SOMMERVILLE, Ian. Engenharia de software. 8. ed. São Paulo: Addison-Wesley, 2007.

2. PRESSMAN, Roger S. Engenharia de software. 6. ed. São Paulo: Makron Books, 2006.

3. ERIKSSON, Hans-Erik. UML 2 toolkit. New York: Wiley, 2004.

4. GUEDES, Gilleanes T. A. UML 2: uma abordagem prática. São Paulo: Novatec, 2009.

5. LARMAN, Craig. Utilizando o UML e padrões: uma introdução à análise e ao projeto orientados a objetos e ao desenvolvimento iterativo. 3. ed. Porto Alegre: Bookman, 2007. 695 p. ISBN 9788560031528.

Processos de Software

Ementa:

Conceitos e terminologia. Infraestrutura de processos (pessoas, ferramentas, treinamentos e outros). Modelagem e especificação de processos de software. Medição e análise de processos de software. Melhoria de processos de software (individual e equipe). Análise e controle de qualidade (prevenção de defeitos, revisão de processos, métricas de qualidade, análise de causa e outros). Níveis de definição de processos. Modelos de ciclo de vida (ágil, cascata, RUP e outros). Modelos de processos e normas técnicas de qualidade (IEEE, ISO e outros). Modelo, definição, medida, análise e melhoria tanto de processo de software individual quanto de equipe. Personalização de processo. Requisitos para processos de software (ISO/IEEE 12207). Visão geral do CMMI e ITIL. Detalhada apresentação do MSP.BR (guias).

Bibliografia Básica:

1. André Koscianski e Michel dos Santos Soares. Qualidade de Software. Editora Novatec. 2a Edição. ISBN 978-85-7522-112-9

2. Wazlawick, Raul Sidnei; Engenharia de Software - Conceitos e Práticas. Campus. Edição : 1 / 2013. ISBN 9788535260847

3. CMMI for Development®: Guidelines for Process Integration and Product Improvement. CHRISSIS, M. B.; KONRAD, M.; SHRUM, S. 3/2011 Addison Wesley

Bibliografia Complementar:

1. ZAHARAN, S., 1998, Software Process Improvement – Practical Guidelines for Business Success, Addison-Wesley.

2. SCOTT, Kendall. O processo unificado explicado. Porto Alegre: Bookman, 2003. 160 p. ISBN

8536302313 (broch.).

3. SOMMERVILLE, I. Engenharia de Software. 7. Ed. São Paulo: Pearson Addison-Wesley, 2007.

4. CMMI - Integração dos Modelos de Capacitação e Maturidade de Sistemas. Ana Brasil Couto. Editora Ciência Moderna. 1ª edição, 2007.

5. -Guia MPS-BR: Melhoria do processo de software brasileiro. Disponível em www.softex.br/mpsbr

6. PRESSMAN, Roger S. Engenharia de software. 7. ed. São Paulo: Makron Books, 2011. ISBN: 9788563308337

Lógica para Computação

Ementa:

Lógica proposicional e de Primeira Ordem; Formalização de problemas; Sistemas dedutivos: axiomático, natural e tableaux; Correção e Completude.

Bibliografia Básica:

1. SOUZA, João Nunes de. Lógica para ciência da computação. Rio de Janeiro: Campus, 2ª ed, 2008. ISBN 9788535229615.

2. SILVA, Flávio Soares Corrêa da; FINGER, Marcelo; MELO, Ana Cristina Vieira de. Lógica para computação. São Paulo, SP: Thomson Learning, 2006. 234 p. ISBN 8522105170 (broch.).

3. HUTH, Michael; RYAN, Mark. . Lógica em ciência da computação: modelagem e argumentação sobre sistemas . 2. ed. Rio de Janeiro: LTC, 2008. 322 p. ISBN 9788521616108 (broch.).

Bibliografia Complementar:

1. ENDERTON, Herbert B. A mathematical introduction to logic. 2nd ed. San Diego, California: Harcourt/Academic Press, c2001. 317 p. ; ISBN 0122384520 (enc.).

2. GERSTING, Judith L. Fundamentos matemáticos para a ciência da computação: um tratamento moderno de matemática discreta . 5. ed. Rio de Janeiro: Livros Técnicos e Científicos, 2004. 597 p. ISBN 8521614225 (broch.).

3. DALEN, D. van. Logic and structure. 4. ed. Berlin: Springer-Verlag, 2004. 263 p. (Universitext) ISBN 3540208798 (broch.).

4. CLARKE, E. M. Model checking. Cambridge: MIT Press, 1999. 314 p. ISBN 9780262032704 (enc.).

5. BAIER, Christel; KATOEN, Joost-Pieter. Principles of model checking. Cambridge, Massachusetts: The MIT Press, 2008. 975 p. ISBN 9780262026499 (broch.).

6. BARWISE, Jon; ETCEMENDY, John; ALLWEIN, Gerard; BARKER-PLUMMER, Dave; LIU, Albert. Language, proof and logic. Stanford: CSLI, 2008. 587 p. ; (CSLI lecture notes ;v 23) ISBN 1 57586 374 X (broch.).

7. BLACKBURN, Patrick; RIJKE, Maarten de; VENEMA, Yde. Modal logic. Cambridge: Cambridge Univ. Press, c2001. 554 p. (Cambridge tracts in theoretical computer science ; 53) ISBN 9780521527149 (broch.).

8. VASCONCELOS, Davi Romero de; HAEUSLER, Edward Hermann. PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO DE JANEIRO Departamento de Informática. Lógica modal de primeira ordem para raciocinar sobre jogos. 2007. 241 f. Tese (Doutorado em Informática)-Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, 2007.

Projeto e Análise de Algoritmos

Ementa:

Noções de análise de algoritmos: análise assintótica de pior caso e caso médio; notação big-O, little-o, ômega e teta; principais classes de complexidade; medida empírica de performance; análise de algoritmos recursivos utilizando relações de recorrência. Projeto de algoritmos: força bruta; gulosos; divisão e conquista; programação dinâmica. NP-completude: definição das classes P e NP; teorema de Cook; principais problemas NP-completos; técnicas de redução.

Bibliografia Básica:

1. LEISERSON, C.E., STEIN, C., RIVEST, R.L. & CORMEN T.H. Algoritmos: teoria e prática. Editora Campus, 3ª. Edição, 2012.

2. VAZIRANI, U.; DASGUPTA, S.; PAPADIMITRIOU, C. Algoritmos, Mcgraw-Hill, 2009.

3. KLEINBERG, J.; TARDOS, E. Algorithm Design, Addison Wesley, 2005.

Bibliografia Complementar:

1. ZIVIANI, N. Projeto de Algoritmos com Implementações em Java e C++, Editora Cengage Learning, 2006.
2. TOSCANI, L. V.; VELOSO, P.A.S. Complexidade de Algoritmos.
3. GERSTING, J. L. Fundamentos matemáticos para a ciência da computação: um tratamento moderno de matemática discreta.
4. MENEZES, P.B. Matemática discreta para computação e informática.
5. ROSEN, K. Matemática discreta e suas aplicações.

Interação Humano-Computador

Ementa:

Conceitos básicos de Interação Humano-Computador; Estilos e paradigmas de interação (interfaces gráficas, manipulação direta, ícones e linguagens visuais); Teorias de IHC (Engenharia cognitiva, Engenharia semiótica); Princípios e recomendações ergonômicas para IHC; Projeto da interface e da interação (prototipação de interfaces, modelagem de usuários, contexto de uso, tarefas e ferramentas de apoio); Avaliação de sistemas interativos (métodos de investigação, de observação de uso e de inspeção, aspectos éticos na relação com os usuários); Acessibilidade (conceitos, Lei Nacional de Acessibilidade, recomendações W3C para um site acessível, ferramentas de apoio ao design de sistemas acessíveis, avaliação de acessibilidade).

Bibliografia Básica:

1. PREECE, J.; ROGERS, Y. Design de interação: além da interação homem-computador. Bookman, 2005. ISBN: 9788536304946
2. BARBOSA, S.; SILVA. Interação Humano-computador. Campus, 2010. ISBN: 9788535234183
3. CYBIS, Walter; BETIOL, Adriana H.; FAUST, Richard. Ergonomia e usabilidade: conhecimentos, métodos e aplicações. São Paulo: Novatec, 3ª Edição. 2015. ISBN: 9788575224595

Bibliografia Complementar:

1. SHNEIDERMAN, B. Designing the user interface. 4 ed. Addison Wesley, 2005.
2. SOMMERVILLE, I. Engenharia de software. 8 ed. Addison Wesley, 2007.
3. KRUG, Steve. Não me faça pensar, atualizado: uma abordagem do bom senso à usabilidade na WEB e Mobile. 1. ed. Rio de Janeiro: Alta Books, 2014. ISBN 9788576088509
4. NIELSEN, J. Usabilidade na web: projetando websites com usabilidade. Campus, 2007. ISBN: 9788535221909
5. BUTOW, E. User interface design for mere mortals: a hands-on guide to user interface design software-independent approach. Addison Wesley, 2007.

Sistemas Operacionais

Ementa:

O histórico, o conceito e os tipos de sistemas operacionais. A estrutura de sistemas operacionais. Conceito de processo. Gerência de processador: escalonamento de processos, Concorrência e sincronização de processos. Alocação de recursos e deadlocks. Gerenciamento de memória. Memória virtual. Gerenciamento de arquivos. Gerenciamento de dispositivos de entrada/saída.

Bibliografia Básica:

1. TANENBAUM, A. S. Sistemas Operacionais Modernos. São Paulo: 3ª Edição, Prentice Hall, 2010.
2. SILBERSCHATZ, Abraham; GAGNE, Greg; GALVIN, Peter; Fundamentos de Sistemas Operacionais. Rio de Janeiro. 8ª Edição. LTC. 2010.
3. FERREIRA, R. E. Linux: guia do administrador do sistema. 2 ed. Novatec, 2008. ISBN: 9788575221778.

Bibliografia Complementar:

1. MACHADO, F.B. Arquitetura de sistemas operacionais. 4 ed. LTC, 2007.
2. SILBERSCHATZ, A.; GALVIN, P.B.; GAGNE, G. Sistemas Operacionais com Java. 7 ed. Campus, 2008. ISBN: 9788535224061.

3. TANENBAUM, A. S. Sistemas operacionais: projeto e implementação. 3 ed. Prentice Hall, 2008.
4. CARISSIMI, A.; TOSCANI, S.; OLIVEIRA, R. S. Sistemas Operacionais. 4 ed. Bookman, 2010. ISBN: 9788577805211.
5. FEDELI, Ricardo Daniel; POLLONI, Enrico Giulio Franco; PERES, Fernando Eduardo. Introdução à ciência da computação. 2. ed. São Paulo, SP: Cengage Learning, 2009.

Projeto Detalhado de Software

Ementa:

Definição de projeto. Princípios de projeto (separação de interesses, encapsulamento de informações, coesão e acoplamento). Projeto orientado por responsabilidade. Arquitetura de software. Projeto arquitetural. Reutilização de software. Padrões de projeto. Projeto de componentes. Projeto de interfaces entre componentes e sistemas. Métricas e ferramentas de suporte a projeto (análise estática e outras). Maus cheiros de projeto (design smells). Refatoração. Medidas de projeto.

Bibliografia Básica:

1. LARMAN, Craig. Utilizando UML e padrões: uma introdução à análise e ao projeto orientados a objetos e ao desenvolvimento iterativo. 3. ed. Porto Alegre: Bookman, 2007. 695 p. ISBN 9788560031528 (broch.).
2. GAMMA, Erich. Padrões de projeto: soluções reutilizáveis de software orientado a objetos. Porto Alegre: Bookman, 2005. ISBN: 9788573076103.
3. MARTIN, Robert C.; MARTIN, Micah. Princípios, padrões e práticas ágeis em C#. Porto Alegre: Bookman, 2011. 735 p. ISBN 9788577808410 (broch.).

Bibliografia Complementar:

1. BEZERRA, E. Princípios de análise e projeto de sistemas com UML. 3. ed. rev. e atual. Rio de Janeiro: Campus, 2015. ISBN:9788535226263.
2. MCCONNELL, Steve. Code Complete: um guia prático para a construção de software . 2. ed. Porto Alegre, RS: Bookman, 2005. 928 p. ISBN 0735619670 (broch.).
3. KERIEVSKY, Joshua. Refatoração para padrões. Porto Alegre: Bookman, 2008. 400 p. ISBN 9788577802449 (broch.).
4. FREEMAN, Eric; FREEMAN, Elisabeth; SIERRA, Kathy; BATES, But. Use a cabeça!: padrões e projetos. 2. ed. rev. Rio de Janeiro, RJ: Alta Books, 2007. 478 p. ISBN 9788576081746 (broch.).
5. SOMMERVILLE, Ian. Engenharia de software. 9. ed. São Paulo: Addison-Wesley, 2011. ISBN: 97879361081
6. EVANS, Eric. Domain-driven design: atacando as complexidades no coração do software. Rio de Janeiro, RJ: Alta Books, 2009. 499 p.

Redes de Computadores

Ementa:

Organização das redes de computadores. Modelos de referência OSI e TCP/IP. Padrões de rede. Meios físicos de transmissão. Protocolos de acesso ao meio. Interconexão de redes. Algoritmos e protocolos de roteamento. Protocolos de redes. Protocolos de transporte TCP e UDP. Protocolos de aplicação. Projeto e Dimensionamento de Redes.

Bibliografia Básica:

1. COMER, D. Redes de computadores e a internet. 4 ed. Bookman, 2007. ISBN: 9788560031368.
2. TANENBAUM, A. S. Redes de Computadores. 4 ed, Campus, 2003. ISBN: 9788535211856.
3. KUROSE, J.; ROSS, K. W. Redes de computadores e a internet: uma abordagem top-down, 5 ed. Addison Wesley, 2010. ISBN: 9788588639973.

Bibliografia Complementar:

1. COMER, D. Interligação de redes com TCP/IP.1. 5 ed.(vol.1). Campus, 2006. ISBN: 9788535220179.
2. OLIVEIRA, Gorki Starlin da Costa. Redes de computadores comunicações de dados TCP/IP : conceitos, protocolos e usos. Alta Books, 2004. ISBN : 8576080567
3. KAUFMANN, M. Computer Networks: A SYSTEMS APPROACH. 4 ed. Elsevier, 2007. ISBN: 9780123705488.

4. FOROUZAN, B. A. Comunicação de dados e redes de computadores. Bookman, 2008. ISBN: 9788577260881.
5. TORRES, G. Redes de computadores. Nova Terra, 2009.
6. MORIMOTO, C.E. Redes: guia prático. GDH Press, 2008. ISBN: 9788599593110.

Gerência de Projetos de Software

Ementa:

Conceitos, terminologia e contexto de gerência de projetos. Ciclo de vida de produto e projeto. Interessados (stakeholders). Organização de empresas (funcionais, matriciais e baseadas em projetos). Estratégias para seleção de projetos. Processos de gerência de projetos. Gerência de escopo. Gerência de tempo (definição de atividades, sequenciamento de atividades, estimativa de recursos, estimativa de duração, desenvolvimento de cronograma e controle de cronograma). Gerência de custos (estimativas, orçamento e controle). Gerência de qualidade. Gerência de recursos humanos. Gerência de comunicação. Gerência de riscos. Gerência de aquisições. Gerência de integração (desenvolver carta de projeto, desenvolver escopo preliminar, desenvolver plano de gerência de projeto, dirigir e gerenciar a execução de projetos, monitorar e controlar atividades de projeto, controle de mudanças e fechamento do projeto). Gerenciamento de projetos ágil (SCRUM).

Bibliografia Básica:

1. DINSMORE, Paul Campbell; BARBOSA, Adriane Monteiro Cavaliere. Como se tornar um profissional em gerenciamento de projetos: livro-base de 'Preparação para certificação PMP® - Project management professional'. 4. ed., rev. e ampl. Rio de Janeiro: Qualitymark, 2011. 383p. ISBN 9788573039788 (Broch.).
2. GREENE, Jennifer; STELLMAN, Andrew. Use a cabeça! PMP. Rio de Janeiro: Alta Books, 2010. 794 p. ISBN 9788576084983 (broch.).
3. COHN, Mike; SILVA, Aldir José Coelho da. Desenvolvimento de software com scrum: aplicando métodos ágeis com sucesso. Porto Alegre: Bookman, 2011. 496 p. ISBN 9788577808076 (broch.).

Bibliografia Complementar:

1. APPELO, J. Management 3.0: Leading agile developers, developing agile leaders. New York: Addison Wesley, 2010. ISBN 9780321712479.
2. A GUIDE to the project management body of knowledge (PMBOK GUIDE). 4. ed. Newtown Square, Pa: Project Management Institute, 2008. 467 p. ISBN 9781933890517 (broch.).
3. DINSMORE, Paul C.; CABANIS-BREWING, Jeannette. AMA: manual de gerenciamento de projetos. Rio de Janeiro, RJ: Brasport, 2009. 498p. ISBN 9788574523237 (broch.).
4. COHN, Mike; User stories applied: for agile software development. 2004, Addison-Wesley. ISBN: 9780321205681
5. MARTINS, José Carlos Cordeiro. Gerenciando projetos de desenvolvimento de software com PMI, RUP E UML. 5. ed. Rio de Janeiro: Brasport, 2010. 290 p. ISBN 9788574524511

Verificação e Validação

Ementa:

Objetivos e restrições de V&V (Verificação e Validação). Planejamento de V&V. Documentação de estratégias de V&V, testes e outros artefatos. Medidas e Métricas. Análise estática de código. Atividades de V&V ao longo do ciclo de vida de um produto. Revisão de software. Testes de unidade. Análise de cobertura. Técnicas de teste funcional (caixa preta). Testes de integração. Desenvolvimento de casos de teste baseados em casos de uso e histórias de usuários. Testes de sistema. Testes de aceitação. Testes de atributos de qualidade. Testes de regressão. Ferramentas de teste (combinação com ferramentas de integração contínua). Análise de relatórios de falha. Técnicas para isolamento e falhas (depuração). Análise de defeitos. Acompanhamento de problemas (tracking). IEEE Std 1012.

Bibliografia Básica:

1. COPELAND, Lee. A practitioner's guide to software test design. Boston, Mass.; London: Artech House, 2004. 294 p. ISBN 9781580537919
2. MYERS, Glenford J. The Art of software testing. New York : J. Wiley, 2004. 177p. ISBN

9780471678359.

3. DELAMARO, Márcio; MALDONADO, José Carlos. Introdução ao teste de software. Rio de Janeiro, RJ: Elsevier: Campus, 2007. 394 p. (Sociedade brasileira de computação) ISBN 9788535226348 (broch.)

Bibliografia Complementar:

1. SOMMERVILLE, Ian, |d 1951-. Engenharia de software. 8. ed. São Paulo, SP: Pearson/ Prentice Hall, 2007. 552 p. ISBN 9788588639287 (broch.).

2. BASTOS, Aderson et al. Base de conhecimento em teste de software. 3. ed. São Paulo: Martins Fontes, 2012. 263 p. ISBN 9788580630534.

3. MOLINARI, Leonardo. Inovação e Automação de Testes de Software. Érica, 2010. 144 p. ISBN 9788536502694 (broch)

4. MOLINARI, Leonardo. Teste de Software - produzindo sistemas melhores e mais confiáveis. 4. ed. Érica, 2010. 232 p. ISBN 9788571949591 (broch)

5. PRESSMAN, Roger S. Engenharia de software. 7.ed. São Paulo: McGraw-Hill, 2011. ISBN 9788563308337.

6. LOPES, CAMILO. TDD na prática. 1ª edição. Editora Ciência Moderna, 2012. 160p. ISBN 978-8539903276

Qualidade de Software

Ementa:

Definições e terminologia de qualidade de software. Custos e impactos de baixa qualidade. Custo de um modelo de qualidade. Terminologia para características de qualidade de software (ISO 9126). Papel de pessoas, processos, métodos, ferramentas e tecnologias em qualidade. Padrões de qualidade (ISO 9001, ISO 9003, IEEE Std 1028, IEEE Std 1465, IEEE Std 12207, ITIL).

Revisões, auditoria e inspeções. Modelos e métricas de qualidade de software. Aspectos relacionados à qualidade de modelos de processos de software. Visão geral do CMMI. MPS.BR. Planejamento de qualidade. Garantia da qualidade. Análise de causa e prevenção de defeitos. Avaliação de atributos de qualidade. Métricas e medidas de qualidade de software. Desenvolver planos de qualidade de software em conformidade com o padrão IEEE Std 730.

Bibliografia Básica:

1. KOSCIANSKI, André; SOARES, Michel dos Santos. Qualidade de software. 2. ed. São Paulo, SP: Novatec, 2007. 395p. ISBN 8575221129 (broch.).

2. SOMMERVILLE, Ian. Engenharia de software. 8. ed. São Paulo, SP: Pearson/ Prentice Hall, 2007. xiv, 552 p. ISBN 9788588639287 (broch.).

3. CHRISSIS, M. B.; KONRAD, M.; SHRUM, S. CMMI for Development®: Guidelines for Process Integration and Product Improvement. 3. Ed. Addison Wesley, 2011

Bibliografia Complementar:

1. COUTO, Ana Brasil. CMMI: integração dos modelos de capacitação e maturidade de sistemas. Rio de Janeiro: Ciência Moderna, 2007. xvi, 276 p. ISBN 9788573935707 (broch.).

2. DELAMARO, Márcio; MALDONADO, José Carlos. Introdução ao teste de software. Rio de Janeiro, RJ: Elsevier: Campus, 2007. 394 p. (Sociedade brasileira de computação) ISBN 9788535226348 (broch.).

3. CMMI for Development®: Guidelines for Process Integration and Product Improvement. CHRISSIS, M. B.; KONRAD, M.; SHRUM, S. 3/2011 Addison Wesley. ISBN 9780321711502

4. BARTIÉ, Alexandre. Garantia de Qualidade de Software. Editora: Elsevier. ISBN: 978-8535211245. 1 edição, 2002

5. RIOS, Emerson; BASTOS, Aderson; CRISTALLI, Ricardo; MOREIRA, Trayahú. Base de Conhecimento em Teste de Software. Editora: Martins Editora. I.S.B.N.: 8599102893. 2 edição, 2007.

6. Guia MPS-BR: Melhoria do processo de software brasileiro. Disponível em <<http://www.softex.br/mpsbr>>

Manutenção de Software

Ementa:

Conceitos e terminologia. Categorias (tipos) de manutenção. Questões técnicas e gerenciais de manutenção. Estimativa de custo de manutenção. Métricas/medidas para manutenção. Processos e

atividades de manutenção. Compreensão de programas. Reengenharia. Engenharia reversa. Norma IEEE Std 14764. Refatoração. Transformação de programas.

Bibliografia Básica:

1. Refatoração para padrões. KERIEVSKY, J.
2. Refatoração: Aperfeiçoando O Projeto De Código Existente. FOWLER, MARTIN
3. Software Maintenance Management: Evaluation And Continuous Improvement. APRIL, ALAIN; ABRAN, ALAIN

Bibliografia Complementar:

1. Code Complete: Um Guia Prático Para A Construção De Software, MCCONNELL, STEVE.
2. Software Maintenance: Concepts And Practice. PENNY GRUBB E ARMSTRONG A. TAKANG.
3. IEEE Std 14764-2006, Software Engineering – Software Life Cycle Processes Maintenance.
4. Frank Tsui, Orlando Karam , Barbara Berna; Essentials Of Software Engineering [Paperback]. Jones & Bartlett Learning; 3 edition (February 7, 2013). ISBN-13: 978-1449691998.
5. Thomas M. Pigoski; Practical Software Maintenance: Best Practices for Managing Your Software Investment. Wiley; 1 edition (November 1, 1996). ISBN-13: 978-0471170013

Arquitetura de Software

Ementa:

Definição de arquitetura de software. A importância e o impacto em um empreendimento de software. Estilos arquiteturais (pipes-and-filters, camadas, publish-subscribe, baseado em eventos, cliente-servidor, dentre outros). Relação custo/benefício entre vários atributos arquitetônicos. Questões de hardware em projeto de software. Rastreabilidade de requisitos e arquitetura de software. Arquiteturas específicas de um domínio e linhas de produtos de software. Notações arquiteturais (ex., visões, representações e diagramas de componentes). Reutilização em nível arquitetural.

Bibliografia Básica:

1. TAYLOR, R. N.; MEDVIDOVIC, N.; DASHOFTY, E. M. Software architecture: Foundations, Theory, and Practice. Wiley, 2009. 750 p.
2. CLEMENTS, Paul et al. Documenting software architectures: views and beyond. 2. ed. Massachusetts: Addison-Wesley Professional. 2010. 592 p.
3. BASS, Len; CLEMENTS, Paul; KAZMAN, Rick. Software architecture in practice. 2. ed. Boston, Massachusetts: Addison-Wesley Professional, 2003. 560 p.

Bibliografia Complementar:

1. SHAW, Mary; GARLAN, David. Software architecture: perspectives on an emerging discipline. São Paulo: Prentice Hall. 1996. 242 p.
2. SOMMERVILLE, I. Engenharia de software. 7. ed. São Paulo: Pearson Addison-Wesley, 2007.
3. GORTON, Ian. Essential software architecture. Berlin: Springer, 2006. 283 p. ISBN 3540287132 (enc.).
4. REEKIE, John. A software architecture primer. Sydney, Australia: Angophora Press, 2006. 179 p. ISBN 0646458418 (broch.).
5. BEZERRA, E. Princípios de análise e projeto de sistemas com UML. 2. ed. Rio de Janeiro: Elsevier, 2007.

Empreendedorismo

Ementa:

Conceito de empreendedorismo. A formação da personalidade. O processo comportamental. Fatores de sucesso, o perfil do empreendedor. Desenvolvimento de habilidades empreendedoras. Lições e práticas internacionais. Empreendedorismo no Brasil. Importância das MPEs na economia. Globalização dos mercados, dos negócios e das oportunidades. Pesquisas Tecnológicas. Propriedade Intelectual. Transferência de Tecnologia. Papel da inovação. Ambientes de pré-incubação e incubação de ideias. Incubadoras de empresas. Parques Tecnológicos. Capital de Risco. Recursos de Fomento. Fontes de Financiamento. Fundos Setoriais. Programas governamentais. Plano de Negócio. Ferramentas de Plano de Negócios. Projetos.

Bibliografia Básica:

1. SALIM, César Simões. Construindo Planos De Empreendimentos. Rio de Janeiro, RJ: Campus, 2010, ISBN 9788535234688 (broch.).
2. DORNELAS, José Carlos Assis. Empreendedorismo: transformando ideias em negócios. 3. ed. rev. atual. Rio de Janeiro: Elsevier, 2008. 232 p. ISBN 9788535232707 (broch.).
3. DOLABELA, Fernando. O segredo de Luísa: uma idéia, uma paixão e um plano de negócios: como nasce o empreendedor e se cria uma empresa. Rio de Janeiro: Sextante, 2008. 299 p. ISBN 9788575423387 (broch.).

Bibliografia Complementar:

1. HISRIC, Robert D. Empreendedorismo. 7. ed. Porto Alegre, RS: Bookman, 2009. 662 p. ISBN 9788577803460 (broch.).
2. FARAH, Osvaldo Elias. Empreendedorismo estratégico. São Paulo: Cengage Learning, 2008. 251 p. ISBN 9788522106080 (broch.).
3. FERRARI, Roberto. Empreendedorismo para computação: criando negócios em tecnologia. Rio de Janeiro, RJ: Elsevier, 2010. 164 p. ISBN 9788535234176 (broch.).
4. CHIAVENATO, Idalberto. Empreendedorismo: dando asas ao espírito empreendedor: empreendedorismo e viabilização de novas empresas, um guia eficiente para iniciar e tocar seu próprio negócio. 3. ed. rev. e atual. São Paulo, SP: Saraiva, 2008. 281 p. ISBN 9788502067448 (broch.).
5. FRIEDMAN, Thomas L. O mundo é plano: uma breve história do século XXI. 2. ed., rev. e atual. Rio de Janeiro: Objetiva, 2007. 557 p. ISBN 9788573028638.

Projeto de Pesquisa Científica e Tecnológica**Ementa:**

O problema da pesquisa e sua formulação. Métodos e Técnicas de Pesquisa. O planejamento da pesquisa. Elaboração de projeto de pesquisa referente ao Trabalho de Conclusão de Curso.

Bibliografia Básica:

1. WAZLAWICK, Raul Sidnei. Metodologia de pesquisa para ciência da computação. Rio de Janeiro : Elsevier. 2008.
2. LAVILLE, Christian; Dionne, Jean. A Construção do Saber: Manual de Metodologia da Pesquisa em Ciências Humanas. Porto Alegre 1ªed Artmed.
3. MARCONI, Marina de Andrade; Lakatos, Eva Maria. Fundamentos de Metodologia Científica. São Paulo 6ªed Atlas.

Bibliografia Complementar:

1. YIN, Robert K. Estudo de Caso - Planejamento e Métodos. 4ªed, Porto Alegre : Bookman, 2010.
2. CERVO, A.; BERVIAN, P.A.; SILVA, R. Metodologia Científica. 6ª ed, Pearson, 2006.
3. COOPER, D. R.; Schindler, Pamela S. Métodos de Pesquisa em Administração. Porto Alegre 7ª ed Bookman.
4. FOWLER, FJ. Pesquisa de Levantamento. Porto Alegre: Pearson, 2011.
5. FREIRE, P. Extensão ou Comunicação. 12ª ed. Paz e Terra.
6. CHAUÍ, Marilena. Convite à Filosofia. 13ª ed. São Paulo: Ática, 2003.

Estágio Supervisionado I**Bibliografia Básica:**

1. Resolução nº 19/CEPE, de 17 de junho de 1992;
2. Lei nº. 11.788/2008.
3. LOPES, V. K. M. Estágio Supervisionado Em Computação: Reflexões e Relatos. 1ª Edição. Appris Editora, 2016, 171p. ISBN 9788547300029.

Bibliografia Complementar:

1. ZABALZA, M. A. O Estágio e as práticas em contextos profissionais na formação universitária. Cortez. São Paulo, 2014. 313p. ISBN 9788524924118.
2. DE OLIVEIRA, E.S.; DE ALMEIDA, W.A.; GHEDIN, E. Estágio com pesquisa. Cortez Editora, 2015. 272 p. ISBN 9788524923913.
3. FREEDMAN, Jeri. Careers in Computer Science and Programming. The Rosen Publishing Group, 2011.

80 p. ISBN 9781448823420

4. PORTELA, K. C. A.; SCHUMACHER, A. J. Estágio Supervisionado - Teoria E Prática. Alexandre Schumacher. 2007. 191p. ISBN: 8537101087.

5. WAZLAWICK, Raul Sidnei. Metodologia de pesquisa para ciência da computação. 2. ed. Rio de Janeiro: Elsevier, 2014. ISBN: 9788535277821.

Trabalho de Conclusão de Curso I

Bibliografia Básica:

1. Severino, A. J. (1999) Metodologia do Trabalho Científico. 20ª edição. Cortez São Paulo.

2. ZOBEL, J. (2015). Writing For Computer Science. 3ª Edição. Springer Verlag Ny. Estados Unidos Da América.

3. DOS SANTOS, C.R e Noronha R.T.S. (2010) Monografias Científicas. 2º edição. Avercamp Importado.

Bibliografia Complementar:

1. ACEVEDO, C. R.; Nohara, J. J. (2013). Como Fazer Monografias - Tcc - Dissertações – Teses. 4ª Edição. Atlas.

2. ALMEIDA, M. S. (2011). Elaboração de Projeto, TCC, Dissertação e Tese. Atlas.

3. FIALHO, F. A. P.; Otani, N. (2011). Tcc-Métodos e Técnicas. Visual Books.

4. MEDEIROS, J B. (2010). Técnicas de Elaboração de Trabalho de Conclusão de Curso (TCC). Atlas.

5. BERTUCCI, J. L. O. (2008). Metodologia Básica para Elaboração de Trabalhos de Conclusão de Cursos (TCC). Atlas.

Estágio Supervisionado II

Bibliografia Básica:

1. Resolução nº 19/CEPE, de 17 de junho de 1992;

2. Lei nº. 11.788/2008.

3. LOPES, V. K. M. Estágio Supervisionado Em Computação: Reflexões e Relatos. 1ª Edição. Appris Editora, 2016, 171p. ISBN 9788547300029.

Bibliografia Complementar:

1. ZABALZA, M. A. O Estágio e as práticas em contextos profissionais na formação universitária. Cortez. São Paulo, 2014. 313p. ISBN 9788524924118.

2. DE OLIVEIRA, E.S.; DE ALMEIDA, W.A.; GHEDIN, E. Estágio com pesquisa. Cortez Editora, 2015. 272 p. ISBN 9788524923913.

3. FREEDMAN, Jeri. Careers in Computer Science and Programming. The Rosen Publishing Group, 2011. 80 p. ISBN 9781448823420

4. PORTELA, K. C. A.; SCHUMACHER, A. J. Estágio Supervisionado - Teoria E Prática. Alexandre Schumacher. 2007. 191p. ISBN: 8537101087.

5. WAZLAWICK, Raul Sidnei. Metodologia de pesquisa para ciência da computação. 2. ed. Rio de Janeiro: Elsevier, 2014. ISBN: 9788535277821.

Trabalho de Conclusão de Curso II

Bibliografia Básica:

1. Severino, A. J. (1999) Metodologia do Trabalho Científico. 20ª edição. Cortez São Paulo.

2. ZOBEL, J. (2015). Writing For Computer Science. 3ª Edição. Springer Verlag Ny. Estados Unidos Da América.

3. DOS SANTOS, C.R e Noronha R.T.S. (2010) Monografias Científicas. 2º edição. Avercamp Importado.

Bibliografia Complementar:

1. ACEVEDO, C. R.; Nohara, J. J. (2013). Como Fazer Monografias - Tcc - Dissertações – Teses. 4ª Edição. Atlas.

2. ALMEIDA, M. S. (2011). Elaboração de Projeto, TCC, Dissertação e Tese. Atlas.

3. FIALHO, F. A. P.; Otani, N. (2011). Tcc-Métodos e Técnicas. Visual Books.

4. MEDEIROS, J B. (2010). Técnicas de Elaboração de Trabalho de Conclusão de Curso (TCC). Atlas.

5. BERTUCCI, J. L. O. (2008). Metodologia Básica para Elaboração de Trabalhos de Conclusão de Cursos (TCC). Atlas.

Pré-Cálculo

Ementa:

Funções no espaço contínuo: estudo de sinal, raízes, polinomiais, racionais, exponenciais, logarítmicas, trigonométricas.

Bibliografia Básica:

1. IEZZI, GELSON et al; Fundamentos de matemática elementar v.1: conjuntos e funções, 8 ed., 2004, Saraiva.
2. IEZZI, GELSON et al; Fundamentos de matemática elementar v.2: Logaritmos, 8 ed., 2004, Saraiva.
3. IEZZI, GELSON et al; Fundamentos de matemática elementar v.3: Trigonometria, 8 ed., 2004, Saraiva.

Bibliografia Complementar:

1. IEZZI, GELSON et al; Fundamentos de matemática elementar v.6: Números Complexos polinômios e equações , 8 ed., 2004, Saraiva.
2. Valéria Zuma Medeiros, André Machado Caldeira, Luiza Maria Oliveira da Silva e Maria Augusta Soares Machado; PRÉ-CÁLCULO; 2ª edição revista e atualizada, 2010, Editora Cengage.
3. Elon Lages Lima, A Matemática no Ensino Médio, SBM.
4. SILVA, Sebastião Medeiros da, et al.; Matemática Básica para Cursos Superiores, 2006, Atlas.
5. DEMANA, Franklin et al. Pré-cálculo Vol. Único. 7a Ed. São Paulo 2009.

Trabalho Cooperativo Baseado em Computadores

Ementa:

Colaboração e Cooperação. Sociedade do conhecimento. Modelos de gestão e organização baseados em conhecimento. Organizações de aprendizagem. Modelos para ambientes de trabalho cooperativo baseado em computador. Tecnologias de suporte à comunicação e cooperação. Planejamento de processos organizacionais cooperativos.

Bibliografia Básica:

1. PROBST, G. Gestão do conhecimento: os elementos construtivos do sucesso. Bookman , 2002. ISBN: 9788573079784.
2. FREITAS, R. A. Portais corporativos: uma ferramenta estratégica. Brasport, 2004. ISBN: 9788574521916.
3. PIMENTEL, M.; Fuks, Hugo. SISTEMAS COLABORATIVOS. EISEVIER-CAMPUS-SBC, 2011. ISBN: 9788535246698.

Bibliografia Complementar:

1. LIMONGI-FRANÇA, A.C. As pessoas na organização. Gente, 2002.
2. NONAKA, I. Criação de conhecimento na empresa. 19 ed. Campus, 1997.
3. MCAFEE, A. Empresas 2.0: A Força Das Mídias Colaborativas. Campus, 2010. ISBN: 9788535238990.
4. SALEMBIER, P. ; RANDALL, D. From Cscw To Web 2.0. Springer, 2010. ISBN: 9781848829640.
5. O'BRIEN, James A. Sistemas de informação e as decisões gerenciais na era da internet. 2 ed. Saraiva, 2004.

Álgebra Linear

Ementa:

Matrizes. Sistemas de equações lineares. Vetores. Produto interno. Dependência e independência linear. Espaços vetoriais. Bases de espaços vetoriais. Transformações lineares. Autovalores e autovetores. Diagonalização de operadores.

Bibliografia Básica:

1. Álgebra Linear; Jose Luiz Boldrini; 3. Edição; Editora Harbra;
2. Alfredo Steinbruch; Introdução à Álgebra Linear; Makron Books Editora; 1990;
3. Seymour Lipschutz; Marc Lipson; Álgebra Linear; Coleção Schaum; Editora Bookman; 2011;

Bibliografia Complementar:

1. Barbieri Filho, Plínio; Biscolla, Laura M. Da Cunha C. O.; Espinosa, Isabel C. O. N.; Fundamentos de

Informática - Álgebra Linear para Computação; Editora LTC;
2. Howard Anton, Chris Rorres; Álgebra Linear com Aplicações; Editora Bookman 2002;
3. IEZZI, GELSON et al; Fundamentos de matemática elementar v.4: Sequências, matrizes, determinantes e sistemas, 8 ed., 2004, Saraiva.
4. HAMILTON PRADO BUENO; Álgebra Linear; Sociedade Brasileira de Matemática-SBM;
5. POOLE, DAVID. LINEAR ALGEBRA A MODERN INTRODUCTION. CENGAGE LEARNING INT, 2010. (ISBN: 0538735457)

Cálculo Diferencial e Integral I

Ementa:

Derivada; Aplicações; Integrais.

Bibliografia Básica:

1. IEZZI, G. Fundamentos de matemática elementar v.8: limites, derivadas, noções de integral. 6 ed. Atual
2. LEITHOLD, L. O cálculo com geometria analítica v.1, 3ed. Harbra, 1994
3. DEMANA, F. D. Pré-cálculo. Addison-Wesley, 2009

Bibliografia Complementar:

1. IEZZI, G. Fundamentos de matemática elementar v.10: Geometria Espacial. 6 ed. Atual, 2005.
2. Fundamentos de matemática elementar v.6: Complexos, polinômios, equações. 7 ed. Atual, 2005.
3. Fundamentos de matemática elementar v.7: Geometria Analítica. 7ed. Atual, 2005.
4. MAJEWSKI, M. Mupad pro computing essential. 2ed.Springer, 2004.
5. GILAT, A.; SUBRAMANIAM, V.Numerical Methods With Matlab. E-Wiley,2007. ISBN: 9780471734406
6. DEMANA, F. D. Precalculus graphical, numerical, algebraic: media update. 7 ed. Addison-Wesley, 2010. ISBN: 9780321356932

Estruturas de Dados Avançadas

Ementa:

Balanceamento de árvores de busca. Filas de prioridade (heaps). Estruturas de dados para conjuntos disjuntos (union-find). Grafos: representação e caminhamento. Tabelas hash e tratamento de colisões.

Bibliografia Básica:

1. SZWARCFITER, Jayme; MARKENZON, Lilian. Estruturas de Dados e Seus Algoritmos. LTC, 3a edição, 2010. (ISBN 9788521617501).
2. RIVEST, Ronald L.; LEISERSON, Charles E.; CORMEN, Thomas H.. Algoritmos: Teoria e Prática. Elsevier, 3ª. Edição, 2012. (ISBN 8535209263).
3. SEDGEWICK, Robert, WAYNE, Kevin. Algorithms (4th Edition). Addison-Wesley Professional; 4 edition, 2011. ISBN: 032157351X.

Bibliografia Complementar:

1. GOODRICH, Michael T.; TAMASSIA, Roberto. Data structures and algorithms in Java - 5th ed. / 2010 ISBN 9780470383261.
2. CELES, Waldemar; CERQUEIRA, Renato; RANGEL, José Lucas. Introdução a estruturas de dados: com técnicas de programação em C. Rio de Janeiro, RJ: Elsevier: Campus, 2004. ISBN 8535212280.
3. KARUMANCHI, NARASIMHA. DATA STRUCTURES AND ALGORITHMS MADE EASY. CREATSPACE PUB, 2011. (ISBN: 1468108867).
4. DASGUPTA, Sanjoy; PAPADIMITRIOU, Christos H.; VAZIRANI, Umesh. Algoritmos. São Paulo, SP: McGraw-Hill, 2009. ISBN 9788577260324.
5. ROSEN, Kenneth H. Matemática discreta e suas aplicações. 6. ed. São Paulo, SP: McGraw-Hill, 2009. ISBN 9788577260362.

Matemática Computacional

Ementa:

Modelagem matemática. Programação matemática: teoria da programação linear e o método simplex.

Programação inteira. Otimização. Uso de pacotes computacionais de otimização.

Bibliografia Básica:

1. HILLIER, FREDERICK S.; LIEBERMAN, GERALD J. INTRODUÇÃO À PESQUISA OPERACIONAL. MCGRAW HILL. 9a edição. (ISBN: 8580551188)
2. COLIN, EMERSON CARLOS. PESQUISA OPERACIONAL - 170 APLICAÇÕES EM ESTRATÉGIA, FINANÇAS, LOGÍSTICA, PRODUÇÃO, MARKETING E VENDAS. LTC, 2007. (ISBN: 8521615590)
3. PASSOS, EDUARDO JOSÉ PEDREIRA FRANCO DOS. PROGRAMAÇÃO LINEAR COMO INSTRUMENTO DA PESQUISA OPERACIONAL. ATLAS, 2008. (ISBN: 8522448396)

Bibliografia Complementar:

1. Christos H. Papadimitriou, Kenneth Steiglitz. Combinatorial Optimization: Algorithms and Complexity. Dover Publications, 1998. ISBN: 0486402584
2. Ravindra K. Ahuja, Thomas L. Magnanti, James B. Orlin. Network Flows: Theory, Algorithms, and Applications. Prentice Hall; 1 ed. 1993.(ISBN: 013617549X)
3. Laurence A. Wolsey, George L. Nemhauser. Integer and Combinatorial Optimization. Wiley-Interscience; 1 edition, 1999. ISBN: 0471359432
4. Anurag Sharma; Application of Graph Theory in Operational Research Problems: Basic concepts, Formulation of the problems, Algorithms Paperback – May 5, 2012. LAP LAMBERT Academic Publishing (May 5, 2012). ISBN-13: 978-3848449453.
5. Bernhard Korte, Jens Vygen; Combinatorial Optimization: Theory and Algorithms (Algorithms and Combinatorics) [Hardcover]. Springer; 5th ed. 2012 edition (January 13, 2012). ISBN-13: 978-3642244872.
6. Jorge Nocedal, Stephen Wright. Numerical Optimization. Springer; 2nd edition, 2006. (ISBN: 0387303030)

Fundamentos da Economia e da Administração

Ementa:

Conceitos Básicos de Economia. Os recursos econômicos e o processo de produção. As questões-chave da Economia: eficiência produtiva. Eficácia alocativa, justiça distributiva e ordenamento institucional. Fundamentos da Macroeconomia. As organizações e a administração. Os primórdios da administração. Abordagens da administração. O desempenho das organizações e o Modelo japonês de administração. Processo de administração. Administração de pessoas.

Bibliografia Básica:

1. CHIAVENATO, Idalberto. Introdução à teoria Geral da Administração: edição compacta. São Paulo: Editora Elsevier, 2004.
2. MAXIMIANO, Antônio C. A. Fundamentos de Administração: manual compacto para cursos de formação tecnológicas e sequenciais. São Paulo: Atlas, 2004.
3. ROSSETTI, José P. Introdução à Economia. São Paulo: Ed. Atlas. 2001.

Bibliografia Complementar:

1. CHIAVENATO, Idalberto. Introdução à Teoria Geral da Administração. São Paulo: Editora Campus, 2000.
2. CHIAVENATO, Idalberto. Administração: teoria, processo e prática. São Paulo: Pearson Education do Brasil Ltda, 1999.
3. MAXIMIANO, Antônio C. A. Teoria Geral da Administração. 3 Ed. São Paulo: Atlas, 2003.
4. LEITE, Antônio Dias.; A Economia Brasileira: de onde viemos e aonde estamos. Rio de Janeiro: Ed. Campus, 2004.
5. KUPFER, David. Economia Industrial. Rio de Janeiro: Ed. Campus, 2002.

Desenvolvimento de Software para Web

Ementa:

Introdução à Web (HTML, CSS, XML e outros), Programação Web com Java, Servlets e JSP. Padrão Arquitetural MVC/DAO. Introdução aos principais Frameworks para desenvolvimento Web (Java Server Faces, Hibernate/iBatis, PrimeFaces, Spring, entre outros).

Bibliografia Básica:

1. LUCKOW, Décio Heinzelmann; MELO, Alexandre Altair. Programação Java para a web. São Paulo: Novatec Editora, 2010. ISBN 9788575222386 .
2. KURNIAWAN, Budi. Java para a Web com Servlets, JSP e EJB. Rio de Janeiro: Ciência Moderna, 2002. 807p. ISBN 8573932104.
3. GEARY, David; HORSTMANN, Cay. Core JavaServer Faces. 3. ed. Rio de Janeiro, RJ: Alta Books, 2012: ISBN: 9788576086420.

Bibliografia Complementar:

1. URUBATAN, Rodrigo. Ruby on rails: desenvolvimento fácil e rápido de aplicações Web. São Paulo, SP: Novatec, 2009. 285 p. ISBN 9788575221846 (broch.).
2. GONÇALVES, Edson. Desenvolvendo aplicações Web com NetBeans IDE 6. Rio de Janeiro: Ciência Moderna, 2008. 581 p. : CD-ROM ISBN 97885739366742.
3. NIEDERAUER, Juliano. Desenvolvendo Websites com PHP: aprenda a criar Websites dinâmicos e interativos com PHP e bancos de dados. São Paulo: Novatec, 2008. 269 p. ISBN 8575220500.
4. BASHAM, Bryan. Use a cabeça!: Servlets & JSP. 2. ed. Rio de Janeiro, RJ: Alta Books, 2008. ISBN 9788576082941.
5. MOREIRA NETO, Oziel. Entendendo e dominando o Java para internet. 2. ed. São Paulo: Digerati Books, 2009. 318 p. ISBN 978-85-7873-087-1.
6. KRUG, Steve. Não me faça pensar: uma abordagem do bom senso à usabilidade na WEB. 2. ed. Rio de Janeiro: Alta Books, 2006. 201 p. ISBN 9788576082713.

Introdução ao Desenvolvimento de Jogos**Ementa:**

História dos jogos. Categorias de jogos. Conceitos de jogos. Gamedesign. Processos de desenvolvimento de jogos digitais. Ferramentas e técnicas para desenvolvimento de jogos.

Bibliografia Básica:

1. Paul Schuyttema. Design de Games: Uma Abordagem Prática. Cengage. 2008. ISBN 8522106150
2. Jeannie Novak. Desenvolvimento de Games. Cengage.2010. ISBN 8522106320
3. Brenda Brathwaite e Ian Schreiber. Challenges for Game Designers. Charles River Media. 2008. ISBN 158450580X

Bibliografia Complementar:

1. Raph Koster. A Theory of Fun for Game Design. Paraglyph Press. 2004. ISBN 1932111972
2. Jesse Schell. The Art of Game Design: A book of lenses. CRC Press. 2008. ISBN 0123694965
3. Mark DeLoura. Best of Game Programming Gems. Charles River Media. 2008. ISBN 1584505710
4. Jason Gregory, Jeff Lander e Matt Whiting. Game Engine Architecture. A K Peters. 2009. ISBN 1568814135
5. Al Sweigart. Invent Your Own Computer Games with Python, 2nd Edition. Creative Commons licenced. ISBN 0982106017. Disponível para download gratuito em <http://inventwithpython.com>

Desenvolvimento de Software para Dispositivos Móveis**Ementa:**

Visão geral sobre dispositivos móveis: Comparação entre dispositivos de sensoriamento, celulares, tablets e computadores convencionais; Visão geral sobre as plataformas de desenvolvimento mais utilizadas, como Android SDK, Iphone SDK e Windows Mobile. Requisitos e desafios para computação móvel. Arquitetura de Software Móvel. Comunicação para Software móvel. Middleware e frameworks para Computação Móvel. Sensibilidade ao contexto e adaptação. Plataforma Android. Activities e Intents. Interfaces e Layouts. Services. Localização e Mapas. Sensores disponíveis.

Bibliografia Básica:

1. SALMRE, Ivo. Writing mobile code: essential software engineering for building mobile applications. New Jersey: Addison-Wesley, 2005. xviii, 771p. ISBN 9780321269317 (broch.).
2. COULOURIS, George F.; DOLLIMORE, Jean; KINDBERG, Tim. Distributed systems: concepts and

design . 4th ed. Harlow: Addison-Wesley, 2005. xiv, 927 p. : : ISBN 0-321-26354-5

3. ROGERS, Rick; LOMBARDO, John; MEDNIEKS, Zigurd R.; MEIKE, Blake. Desenvolvimento de aplicações Android. São Paulo, SP: Novatec, 2009. xvi, 376 p. ISBN 9788575222034 (broch.).

Bibliografia Complementar:

1. Ubiquitous Computing: Smart Devices, Environments and Interactions (1st ed.). Stefan Poslad. 2009. Wiley Publishing.

2. Head First Android Development. SIMON, J. O'REILLY MEDIA. 1/2012. 9781449393304.

3. LECHETA, Ricardo R. Google android: aprenda a criar aplicações para dispositivos móveis com o Android SDK. 2. ed. rev. ampl. São Paulo, SP: Novatec, 2010. 608 p. ISBN 9788575222447.

4. MEDNIEKS, Zigurd; DORNIN, Laird; MEIKE, G. Blake; NAKAMURA, Masumi. Programming Android. O'Reilly Media, 2012. 542 p. ISBN 9781449316648.

5. ANNUZZI, Joseph; DARCEY, Lauren; CONDER, Shane. Advanced Android Application Development. Pearson Education, 2014. 624 p. ISBN 9780133892451.

Linguagens Formais e Autômatos

Ementa:

Introdução. Linguagens, gramáticas e expressões regulares, autômatos finitos. Linguagens e gramáticas livre-do-contexto e autômatos de pilha. Linguagens sensíveis ao contexto. Hierarquia de classes de linguagens. Tópicos especiais e aplicações das linguagens formais e autômatos.

Bibliografia Básica:

1. HOPCROFT, J. E.; ULLMAN, J. D.; MOTWANI, R., Introdução à teoria dos autômatos: linguagens e computação, 1 ed, 2002, 10a tiragem Campus.

2. MENEZES, P.B. Linguagens formais e autômatos. 5 ed. Sagra Luzzatto, 2008. ISBN: 9788577807659.

3. RAMOS, M. V.; NETO, J.J.; VEGA, I.S. Linguagens Formais: teoria, modelagem e implementação. Bookman, 2009. ISBN: 9788577804535.

Bibliografia Complementar:

1. AHO, A. V.; SETHI, R.; ULLMAN, J. D. Compiladores: princípios, técnicas e ferramentas. 2 ed. Pearson/Addison-Wesley, 2008. ISBN: 9788588639249.

2. CARNIELLI, W.; EPSTEIN, R. L. Computabilidade, Funções Computáveis, Lógica e os Fundamentos da Matemática. UNESP, 2009. ISBN: 9788571398979.

3. SEBESTA, R.W. Conceitos de linguagens de programação. 5 ed. Bookman, 2003. ISBN: 9788577807918.

4. SIPSER, M. Introdução a teoria da computação. 2 ed. Thomson Learning, 2007. ISBN: 9788522104994.

5. TUCKER, A. B.; NOONAN, R. Linguagens de programação: princípios e paradigmas. 2 ed. McGraw-Hill, 2008. ISBN: 9788577260447.

6. WATT, D.A. Programming language design concept. IE-Wiley, 2004. ISBN: 0470853204.

Desenvolvimento de Software para Persistência

Ementa:

Definição de persistência. Persistência empregando documentos XML, objetos serializáveis, SGBDs. Tecnologias para persistência de informações. Persistência de objetos usando base relacional. Persistência usando outros modelos de dados (orientado a documento, chave-valor, orientado a coluna).

Bibliografia Básica:

1. ULLMAN, J.D.; WIDOM, J. First Course in database systems. 3. ed. São Paulo: Prentice Hall, 2007.

2. HAMBRICK, G. et al. Persistence in the enterprise: a guide to persistence technologies. Boston: IBM Press, 2008.

3. FOWLER, M. NoSQL Distilled: a brief guide to the emerging world of polyglot persistence. São Paulo: Prentice-Hall, 2012.

Bibliografia Complementar:

1. WHITE, Tom. Hadoop: the definitive guide. California: O'Reilly, 2009. 501 p. ISBN 9780596521974

(broch.).

2. AMBLER, Scott W.; SADALAGE, Pramod J. Refactoring databases: evolutionary database design. New Jersey: Addison - Wesley, 2011. 350 p. (The Addison Wesley signature series) ISBN 321293533 2011.
3. ELMASRI, Ramez; NAVATHE, Shamkant B. Sistemas de banco de dados. 4. ed. São Paulo, SP: Pearson/Addison Wesley, 2009. 724 p. ISBN 9788588639171 (broch.).
4. BEIGHLEY, Lynn. Use a cabeça SQL. Rio de Janeiro, RJ: Alta Books, 2008. 454 p. ISBN 9788576022101 (broch.).
5. SILBERSCHATZ, Abraham; KORTH, Henry F.; SUDARSHAN, S. Sistema de banco de dados. Rio de Janeiro, RJ: Elsevier, 2006. 781 p. ISBN 9788535211078 (broch.).

Segurança

Ementa:

Ameaças. Segurança como atributo qualitativo de projeto de software. Autenticação. Autorização. Integridade. Confidencialidade. Criptografia (chaves simétricas e assimétricas). Infraestrutura de chaves públicas brasileiras (ICP-Brasil). Certificados digitais. Assinaturas digitais. Desenvolvimento de software seguro. Noções de auditoria de sistemas. Norma NBR 27002.

Bibliografia Básica:

1. IMONIANA, Joshua Onome. Auditoria de sistemas de informação. 2. ed. São Paulo: Atlas, 2008. 207 p. ISBN 9788522450022 (broch.).
2. STALLINGS, William. Criptografia e segurança de redes: princípios e práticas. 4. ed. São Paulo: Pearson/ Prentice Hall, 2008. 492 p. ISBN 9788576051190 (broch.).
3. BEAL, Adriana. Segurança da informação: princípios e melhores práticas para a proteção dos ativos de informação nas organizações. São Paulo, SP: Atlas, 2008. 175 p. ISBN 9788522440856 (broch.).

Bibliografia Complementar:

1. DASWANI, Neil; KERN, Christoph; KESAVAN, Anita. Foundations of security: what every programmer needs to know . Berkeley, Ca: Apress, 2007. 290 p. (The Expert's voice in security) ISBN 9781590597842 (broch.).
2. ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. ABNT NBR ISO/IEC 27001- Tecnologia da informação - técnicas de segurança - sistemas de gestão de segurança da informação - requisitos. Rio de Janeiro, RJ, 2006. 34 p.
3. ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. ABNT NBR ISO/IEC 27002- Tecnologia da informação - técnicas de segurança - código de prática para a gestão da segurança da informação. Rio de Janeiro, RJ, 2005. 120 p. ISBN 9788507006480.
4. KUROSE, James F.; ROSS, Keith W. Redes de computadores e a Internet: uma abordagem top-down. 5. ed. São Paulo: Pearson Addison Wesley, 2010. xxii, 614 p. ISBN 9788588639973 (broch.).
5. NAKAMURA, Emilio Tissato; GEUS, Paulo Lício de. Segurança de redes em ambientes cooperativos. São Paulo: Novatec, c2007. ISBN 9788575221365 (broch.).
6. STATO FILHO, André. Linux: controle de redes. Florianópolis: Visual Books, 2009. 352 p. ISBN 9788575022443 (broch.).
7. ULBRICH, Henrique Cesar; DELLA VALLE, James. Universidade H4CK3R: desvende todos os segredos do submundo dos hackers . 6. ed. São Paulo: Digerati Books, 2009. 348p. (Série Universidade) ISBN 9788578730529 (broch.).

Gerência de Configuração

Ementa:

Conceitos e terminologia. Processos de gerência de configuração. Identificação de itens de configuração. Atributos a serem registrados para cada item de configuração. Armazenamento. Controle de mudanças. Relatórios de status. Controle de versões e linhas base ou de referência (baselines). Gerência de configuração segundo o MPS.BR. Papéis em gerência de configuração. Normas (IEEE 828). Princípios de gerência de configuração e relação com atividades de desenvolvimento de software. Gerência de configuração segundo desenvolvimento ágil, técnica de builds frequentes e desenvolvimento iterativo. Gerência de configuração para diferentes tipos de produtos (compostos, multiplataforma, múltiplas variantes, críticos, pequenos, médios e grandes). Gerência de configuração para desenvolvimento de

software distribuído geograficamente, múltiplos interessados e desenvolvimento paralelo. Melhoria de gerência de configuração. Considerações práticas acerca de gerência de configuração de software. Ferramentas.

Bibliografia Básica:

1. HASS, Anne Mette Jonassen. Configuration management: principles and practice. Boston, Massachusetts: Addison-Wesley, 2003. 370 p. (The Agile software development series) ISBN 9780321117663 (broch.).
2. AIELLO, R.; SACHS, L. Configuration management best practices: practical methods that work in the real world. Upper Saddle River, NJ: Addison-Wesley, 2011. 229 p.
3. MOLINARI, Leonardo. Gerência de configuração: técnicas e práticas no desenvolvimento do software. Florianópolis: Visual Books, 2007. 208 p. ISBN 8575022105 (broch.) 9788575022108.

Bibliografia Complementar:

1. DUVALL, Paul M. Continuous integration: improving software quality and reducing risk. Boston, MA: Addison-Wesley, 2007. 283 p. ISBN 9780321336385
2. MOREIRA, Mario E. Adapting configuration management for agile teams: balancing sustainability and speed. New York: John Wiley & Sons, 2009.
3. HUMBLE, Jez.; FARLEY, David. Continuous delivery: reliable software releases through build, test, and deployment automation. Upper Saddle River, NJ: Addison-Wesley, 2011. 463 p. (Addison-Wesley Signature Series.) ISBN 9780321601919.
4. SOFTEX Brasil. Guias MPS.BR. Disponível em: <http://www.softex.br/mpsbr/_home/default.asp>. Acesso em: 23 jan. 2013.
5. CHRISSIS, M. B.; KONRAD, M.; SHRUM, S. CMMI for Development®: guidelines for process integration and product improvement. 3. ed. New York: Addison Wesley, Addison Wesley, 2011.

Inteligência Artificial

Ementa:

Conceito de IA, Histórico e Metas. Agentes Inteligentes. Solução de Problemas, Busca e Jogos. Sistemas Lógicos, Conhecimento e Raciocínio. Sistemas Baseados em Conhecimento. Planejamento. Incerteza, Probabilidade e Teoria da Decisão. Aprendizado. Linguagem e Comunicação. Percepção. Robótica. Questões Filosóficas.

Bibliografia Básica:

1. RUSSELL, S.; NORVIG, P. Inteligência artificial. 2 ed. Campus, 2004. ISBN: 8535211772
2. COPPIN, B; Inteligência artificial. LTC, 2010. ISBN: 9788521617297.
3. SHOHAM, Y. Multiagent systems: algorithms, game theoretic. Cambridge University, 2009. ISBN: 9780521899437.

Bibliografia Complementar:

1. OSBORNE, M.J. A course in game theory. MIT Press, 2009. ISBN: 0262650401/9780262650403
2. EFRAIM, T. Decision support and business. 8 ed. Prentice Hall, 2007.
3. WOOLDRIDGE, M. Introduction to Multiagent Systems. Wiley, 2009. ISBN: 9780470519462
4. BRACHMAN, R. Knowledge representation and reasoning. Morgan-Kaufmann, 2004.
5. BRATKO, I. Prolog Programming for Artificial Intelligence. 4 ed. Addison Wesley, 2011. ISBN 0321417461
6. HAYKIN, S. Redes neurais: princípios e prática. 2 ed. Bookman, 2001.

Especificação Formal de Software

Ementa:

Importância da Especificação Formal na Engenharia de Software; Visão geral de modelos matemáticos e linguagens de especificação; Especificação e Verificação de Sistemas Sequenciais utilizando Lógica de Primeira-Ordem e Notação Z; Especificação e Verificação de Sistemas Concorrentes utilizando Álgebra de Processo (CSP), Redes de Petri e Verificação de Modelos.

Bibliografia Básica:

1. Da Silva, Flávio Soares Corrêa; Finger, Marcelo; De Melo, Ana Cristina Vieira, Lógica para Computação, São Paulo, 1a. Edição, Editora Thomson Learning.
2. Christel Baier and Joost-Pieter Katoen, Principles of model checking, MIT Press, 2008.
3. C. A. R. Hoare. Communicating Sequential Processes. 2004. Disponível Online em: <http://www.usingcsp.com/cspbook.pdf>.
4. Jim Davies and Jim Woodcock (1996). Using Z: Specification, Refinement and Proof. Prentice Hall International Series in Computer Science. ISBN 0-13-948472-8. Disponível Online em: <http://www.usingz.com/text/online/>

Bibliografia Complementar:

1. Holzmann, G.J., The Spin Model Checker: Primer and Reference Manual (2003), Addison-Wesley.
2. Engenharia de software. SOMMERVILLE, I.
3. Engenharia De Software. PRESSMAN, R.
4. Knowledge representation and reasoning. BRACHMAN, R.
5. Model checking / Edmund M. Clarke, Jr., Orna Grumberg, Doron A. Peled, ISBN : 9780262032704

Redes Sociais

Ementa:

Redes Sociais; Gerações de Redes Sociais; Análise de Redes Sociais; Diferentes Usos de Redes Sociais; Máquinas Sociais; Capital Social e Gestão do Conhecimento; Aplicações de Redes Sociais; Desenvolvimento de Redes Sociais.

Bibliografia Básica:

1. PIMENTEL, M.; FUKS, Hugo. Sistemas Colaborativos. Rio de Janeiro: Elsevier-Campus-SBC, 2011. ISBN 9788535246698.
2. MCAFEE, Andrew. Empresas 2.0: a força das mídias colaborativas para superar grandes desafios empresariais. Rio de Janeiro, RJ: Elsevier, 2010. 216 p. ISBN 9788535238990 (broch.).
3. NONAKA, Ikujiro; TAKEUCHI, Hirotaka. Criação de conhecimento na empresa : como as empresas japonesas geram a dinâmica da inovação . 19. ed. Rio de Janeiro: Campus; Elsevier, 1997. 358p. : ISBN 8535201772 (broch.).

Bibliografia Complementar:

1. CROSS, Robert L.; PARKER, Andrew. The hidden power of social networks: understanding how work really gets done in organizations. Boston: Harvard Business School, 2004. 213 p. ISBN 1591392705.
2. PRELL, Christina. Social network analysis: history, theory and methodology. California: Sage Publications Ltd, 2011. ISBN 9781412947152.
3. CROSS, Robert L.; SINGER et al. The organizational network fieldbook: best practices, techniques and exercises to drive organizational innovation and performance. New York: John Wiley, 2010. ISBN 9780470542200.
4. BARABASI, Albert-Laszlo. Linked: how everything is connected to everything else and what it means for business, science, and everyday life. New York,: Plume Book, 2003. 294 p. ISBN 0452284392.
5. INTERNATIONAL Network for Social Network Analysis. Disponível em: <<http://www.insna.org>>. Acesso em: 23 jan. 2013.
6. MEIRA, Silvio R. L. et al. The Emerging Web of Social Machines. COMPSAC/IEEE, 2011. p. 26-27. Disponível em: <<http://arxiv.org/abs/1010.3045>>. Acesso em: 23 jan. 2013.

Compiladores

Ementa:

Introdução a Compiladores, Análise Léxica, Análise Sintática, Análise Semântica, Geração de Código, Tópicos Especiais em Compiladores.

Bibliografia Básica:

1. AHO, Alfred V.; LAM, Monica S.; SETHI, Ravi; ULLMAN, Jeffrey D., Compiladores: princípios, técnicas e ferramentas, Pearson, 2a Edição, 2007.
2. MAK, Ronald, Writing Compilers and Interpreters: A Software Engineering Approach, Wiley, 3rd Edition, 2009.

3. COOPER, Keith, TORCZON, Linda. Construindo Compiladores. Elsevier Brasil, 2013, ISBN 978-85-352-5564-5.

Bibliografia Complementar:

1. GRUNE, D., Jacobs, C., Parsing Techniques: A Practical Guide (Monographs in Computer Science), Springer, 2nd Edition, 2008.
2. LEVINE, J., Flex & Bison: Text Processing Tools. O'Reilly Media, 2009.
3. MOZGOVOY, Maxim, Algorithms, Languages, Automata, & Compilers: A Practical Approach, Jones and Bartlett Publishers, Inc., 2009.
4. JOSÉ NETO, João. Introdução à Compilação, Elsevier, 2016.
5. DES WATSON. A Practical Approach to Compiler Construction, Springer, 2017.

Teoria da Computação

Ementa:

Modelos computacionais universais. Computabilidade. Funções recursivas. Introdução à complexidade de problemas e tópicos avançados.

Bibliografia Básica:

1. CARNIELLI, W.; EPSTEIN, R. L. Computabilidade, Funções Computáveis, Lógica e os Fundamentos da Matemática. UNESP, 2009. ISBN: 9788571398979.
2. SIPSER, M. Introdução à teoria da computação. 2 ed. Thomson Learning, 2007. ISBN: 9788522104994.
3. DIVERIO, T.A. Teoria da computação: máquinas universais e computabilidade . 3 ed. Bookman, 2011. ISBN: 9788577808243.

Bibliografia Complementar:

1. HOPCROFT, John E. Introdução à teoria de autômatos, linguagens e computação. Rio de Janeiro: Elsevier, 2002. 560 p. ISBN 8535210725
2. ENDERTON, Herbert B. A mathematical introduction to logic. 2nd ed. San Diego, California: Harcourt/Academic Press, c2001. xii, 317 p. ; ISBN 0122384520 (enc.) – ISBN 0122384520/9780122384523.
3. PAPANITRIU, C. Computational complexity. Addison Wesley, 1994. ISBN: 9780201530827.
4. BARWISE, J. Language, proof and logic. Seven Bridges, 2002. ISBN: 9781575863740.
5. WATT, D.A. Programming language design concept. IE-Wiley, 2004. ISBN: 0470853204."

Reuso de Software

Ementa:

Conceitos básicos e importância de Reuso de Software. Principais abordagens de reuso: Engenharia de Domínio, Linha de Produtos de Software, Desenvolvimento Baseado em Componentes, Arquitetura Orientada a Serviços, Padrões e Frameworks.

Bibliografia Básica:

1. MILL, Hafedh. Reuse-based software engineering: techniques, organization and measurement. New York: Wiley, 2002. 636 p. ISBN 0471398195.
2. EZRAN, M.; MORISIO, M.; TULLY, C. Practical software reuse. Berlim: Springer, 2002
3. POHL, K.; BÖCKLE, G.; LINDEN, F. J. Software product line engineering: foundations, principles and techniques. Berlim: Springer, 2005.

Bibliografia Complementar:

1. KANG, K. C.; SUGUMARAN, V.; PARK, S. Applied software product line engineering. Boca Raton, Florida: CRC Press, 2010.
2. SOMMERVILLE, I. Engenharia de software. 7. ed. São Paulo: Pearson Addison-Wesley, 2007.
3. PRESSMAN, R. Engenharia de software. 6. ed. São Paulo: Mc Graw-Hill, 2007.
4. GAMMA, E.; HELM, JOHNSON, R.; R.; VLISSIDES, J. Padrões de projeto: soluções reutilizáveis de software orientado a objetos. Porto Alegre: Bookman, 2000. 364 p.
5. ALMEIDA, Eduardo Santana de et al. C.R.U.I.S.E: Component Reuse in Software Engineering. Recife: Gráfica Dom Bosco, 2007. Disponível em: <<http://cruise.cesar.org.br/index.html>> Acesso em: 14 set.

2008.

Estimativa de Custos em Projetos de Software

Ementa:

Estimativa do tamanho do software a ser desenvolvido; Estimativa dos riscos e incertezas do projeto; Estimativa de esforço necessário para construir o software; Estimativa de prazo necessário para construir o software e Estimativa do custo do projeto. Modelos de Estimativas como: Estimativa em pontos de Casos de Uso e Análise de Pontos de Função, COCOMO e COCOMO II.

Bibliografia Básica:

1. JONES, Capers. Estimating software costs: bringing realism to estimating. 2. ed. New York: Mc Graw-Hill, 2007. 644 p. ISBN 9780071483001.
2. HILL, P. Practical Software Project Estimation: a toolkit for estimating software development effort & duration. New York: McGraw-Hill Osborne Media, 2011. ISBN 0071717919, 9780071717915.
3. MCCONNELL, Steve. Software estimation: demystifying the black art. Redmond, Wa.: Microsoft Press, 2006. 308 p. ISBN 9780735605350.

Bibliografia Complementar:

1. COHN, Mike; HIGHSMITH, Jim. . Agile estimating and planning. Upper Saddle River, NJ: Prentice Hall PTR, 2006. 330 p. (Robert C. Martin series) ISBN 9780131479415 (broch.).
2. VAZQUEZ, Carlos Eduardo; SIMÕES, Guilherme Siqueira; ALBERT, Renato Machado. Análise de pontos de função: medição, estimativas e gerenciamento de projetos de software. São Paulo: Érica, 2011. 222 p. ISBN 8571948992.
3. GARMUS, David; HERRON, David. Function point analysis: measurement practices for successful software projects. Boston: Addison-Wesley, 2001. (Addison-Wesley information technology series) ISBN 9780201699443.
4. MENDES, Antonio. Custo de software. Elsevier Brasil, 2014. 184p. ISBN 9788535271997.
5. MARTINS, José Carlos Cordeiro. Técnicas Para Gerenciamento de Projetos de Software. Brasport, , 2007. ISBN 9788574523088.

Sistemas Distribuídos

Ementa:

Introdução: caracterização de sistemas de computação distribuída; aplicações distribuídas (caracterização e aspectos de projeto); objetivos básicos de sistemas distribuídos (transparência, abertura, escalabilidade, etc.). Modelos de sistemas distribuídos: sistemas cliente/servidor e sistemas multicamadas; sistemas peer-to-peer. Objetos distribuídos: interface versus implementação; objetos remotos; chamadas de métodos remotos (RMI). Processos em sistemas distribuídos: threads e seu uso em sistemas distribuídos; processos clientes e processos servidores; noções de código móvel e agentes de software. Sincronização e Coordenação: o conceito de tempo em sistemas distribuídos; consenso; exclusão mútua distribuída; eleição.

Bibliografia Básica:

1. COSTA, D. G. Java em rede: programação distribuída na internet. Brasport, 2008. ISBN: 9788574523361.
2. COULOURIS, G. F.; DOLLIMORE, J.; KINDBERG, T. Sistemas distribuídos: conceitos e projetos. 4 ed. Bookman, 2007. ISBN: 9788560031498.
3. TANENBAUM, A.; STEEN, V. M. Sistemas distribuídos: princípios e paradigmas. 2 ed. Prentice Hall, 2007. ISBN: 9788576051428.

Bibliografia Complementar:

1. WHITE, T. Hadoop: the definitive guide. O'Reilly Media, 2009.
2. TANENBAUM, A. S. Sistemas operacionais modernos. 2 ed. Prentice Hall, 2003.
3. ALONSO, G.; CASATI, F.; KUNO, K.; MACHIRAJU, V. Web Services: Concepts, Architectures and Applications. Springer, 2004. ISBN: 9783540440086
4. NAKAMURA, E. T.; GEUS, P.L. Segurança de redes em ambientes cooperativos. Novatec, 2007. ISBN: 9788575221365.

5. ERL, THOMAS. SOA. Princípios de design de serviço. Prentice Hall, 2009. ISBN: 9788576051893.

Sistemas Multiagentes

Ementa:

Agentes inteligentes: conceitos, modelos e arquiteturas; agentes reativos; agentes deliberativos; Fundamentos da Inteligência Artificial Distribuída e, em especial, dos Sistemas Multiagentes. Aspectos de comportamento emergente, comunicação, negociação e coordenação entre agentes. Metodologias de Desenvolvimento e Arquiteturas de Sistemas Multiagentes. Apresentação de aplicações existentes e Utilização de Plataformas para o desenvolvimento de Sistemas Multiagentes.

Bibliografia Básica:

1. WOOLDRIDGE, Michael J. An introduction to multiagent systems. 2. ed. New York: J. Wiley & Sons, 2009. 461 p. ISBN 9780470519462 (broch.).
2. RUSSELL, Stuart J.; NORVIG, Peter. Inteligência artificial. Rio de Janeiro: Elsevier, Campus, 2004. 1021 p. ISBN 8535211772 (broch.).
3. FOWLER, Martin. UML essencial: um breve guia para a linguagem-padrão de modelagem de objetos . 3. ed. Porto Alegre: Bookman, 2005. 160 p. ISBN 8536304545 (broch.).

Bibliografia Complementar:

1. DEITEL, Paul J.; DEITEL, Harvey M. Java: como programar. 8. ed. São Paulo: Pearson Prentice Hall, 2010. 1144 p. ISBN 9788576055631 (broch.).
2. RUMBAUGH, James. Modelagem e projetos baseados em objetos. Rio de Janeiro: Campus, 2006. 652p. ISBN 857001841X.
3. MEYER, Bertrand. Object - oriented software construction. 2nd. ed. New Jersey: Prentice Hall PTR, 1997. 1254 p. ISBN 0136291554.
4. FIPA. Especificações FIPA. Disponível em: <<http://www.fipa.org>>. Acesso em: 24 jan. 2013.
5. KENNETH BARCLAY, JOHN SAVAGE; Object-Oriented Design with UML and Java [Paperback]. Butterworth-Heinemann; 1 edition (February 10, 2004). ISBN-13: 978-0750660983.

Desenvolvimento de Software Concorrente

Ementa:

Introdução, Primitivas, Problema da Seção Crítica, Liveness e Safety, Problemas Clássicos, Programação em Threads, Compartilhamento e Composição de Objetos, Programação Concorrente em Java, Objetos Concorrentes de Alto Nível, Gerência do Ciclo de Vida de Software Concorrente, Tópicos Avançados de Programação Concorrente em Java.

Bibliografia Básica:

1. BEN-ARI. Principles of concurrent and distributed programming. 2. ed. São Paulo: Prentice-Hall, 2006.
2. DEITEL, Harvey M.; DEITEL, Paul J. Java como programar. 8. ed. São Paulo: Pearson Prentice Hall, 2010. 1144 p. ISBN 9788576055631.
3. GOETZ, B. Java concurrency in practice. Boston: Addison-Wesley, 2006. 403 p. ISBN 9780321349606.

Bibliografia Complementar:

1. MAGEE, J.; KRAMER, J. Concurrency: state models and Java programs. Michigan: Wiley, 2006. ISBN 9780470093559, 0470093552.
2. COULOURIS, George F.; DOLLIMORE, Jean; KINDBERG, Tim. Sistemas distribuídos: conceitos e projetos. 4. ed. Porto Alegre, RS: Bookman, 2007. 784 p. ISBN 9788560031498.
3. MANZANO, José Augusto N. G.; COSTA Jr., Roberto Affonso da. Java 7- Programação de Computadores - Guia Prático de Introdução, Orientação e Desenvolvimento. 1. ed. Editora Érica, 2011. ISBN: 9788536503745.
4. SILBERSCHATZ, A.; GALVIN, P.B.; GAGNE, G. Sistemas Operacionais com Java. 7 ed. Campus, 2008. ISBN:9788535224061.
5. TANENBAUM, A.; STEEN, V. M. Sistemas distribuídos: princípios e paradigmas. 2 ed. Prentice Hall, 2007. ISBN: 9788576051428.

Leitura de Software

Ementa:

Estudar, investigar, analisar e discutir projetos de softwares existentes.

Bibliografia Básica:

1. MARTIN, Robert C. Código limpo: habilidades práticas do Agile Software . Rio de Janeiro: Alta Books, 2011. 413 p. ISBN 857608267-5 (broch.).
2. FEATHERS, Michael C. Working effectively with legacy code. Upper Saddle River, NJ: Prentice Hall, 2005. 434 p. (Robert C. Martin series) ISBN 0131177052.
3. SPINELLIS, Diomidis. Code Quality: the open source perspective. Boston: Addison-Wesley, 2006. ISBN 9780321166074.

Bibliografia Complementar:

1. ORAM, Andy; WILSON, Greg (Edit.). Beautiful code: leading programmers explain how they think. EUA: O'Reilly, 2007. 593 p. ISBN 9780596510046.
2. FOWLER, Martin. Refatoração: aperfeiçoando o projeto de código existente. Porto Alegre, RS: Bookman, 2008. 365 p. ISBN 8536303956 (broch.).
3. MCCONNELL, Steve. Code complete: um guia prático para a construção de software . 2. ed. Porto Alegre, RS: Bookman, 2005. 928 p. ISBN 0735619670 (broch.).
4. HUNT, Andrew. O programador pragmático: de aprendiz a mestre. Porto Alegre, RS: Bookman, 2010. 343 p. ISBN 9788577807000 (broch.).
5. SOMMERVILLE, Ian, |d 1951-; OLIVEIRA, Kalinka; BOSNIC, Ivan. Engenharia de software. 9. ed. São Paulo, SP: Pearson/ Prentice Hall, 2011. 529 p. ISBN 9788579361081 (broch.).

Métodos e Ferramentas da Engenharia de Software

Ementa:

Métodos heurísticos, formais e de construção de protótipos. Ferramentas para auxiliar na produção de requisitos, projeto, construção, testes e manutenção. Ferramentas de gerência de configuração, gerência de projeto, processo de software, qualidade e outras.

Bibliografia Básica:

1. BECK, Kent. TDD desenvolvimento guiado por testes. Porto Alegre: Bookman, 2010. 240 p. ISBN 9788577807246.
2. Teste e análise de software: processos, princípios e técnicas. PEZZÉ, M.; YOUNG, M. Bookman, 2008.
3. HUNT, Andrew. O programador pragmático: de aprendiz a mestre. Porto Alegre, RS: Bookman, 2010. 343 p. ISBN 9788577807000 (broch.).

Bibliografia Complementar:

1. BASS, Len; CLEMENTS, Paul; KAZMAN, Rick. Software architecture in practice. 2. ed. Boston: Addison-Wesley, 2003. 528 p. ISBN 0321154959(enc.).
2. ABRAN, Alain (Ed.). Guide to the software engineering body of knowledge: trial version. Washington: Computer society, 2001. 205 p. Disponível em: <<http://www.computer.org/portal/web/swebok>>. Acesso em: 23 jan. 2013.
3. HEUSER, Carlos Alberto. Projeto de banco de dados. 6. ed. Porto Alegre: Bookman, 2009. 282 p. (Livros Didáticos Informática ; 4) ISBN 9788577803828.
4. MOLINARI, Leonardo. Gerência de configuração: técnicas e práticas no desenvolvimento do software. Florianópolis: Visual Books, 2007. 208 p. ISBN 8575022105 (broch.).
5. MARTINS, José Carlos Cordeiro. Técnicas para gerenciamento de projetos de software. Rio de Janeiro, RJ: Brasport, 2007. 432p. ISBN 9788574523088 (broch.).

Integração de Aplicações

Ementa:

Definição de integração de aplicações. Desafios de integração. Abordagens de integração (transferência de arquivos, bases de dados compartilhadas, chamada de procedimento remoto e troca de mensagens). Padrões para integração de aplicações.

Bibliografia Básica:

1. HOHPE, Gregor; WOOLF, Bobby. Enterprise integration patterns: designing, building, and deploying messaging solutions . Boston, Massachusetts: Addison-Wesley, 2004. 683 p. (The Addison-Wesley signature series) ISBN 0321200683 (broch.).
2. ERL, Thomas. SOA: princípios de design de serviços. São Paulo, SP: Pearson Prentice Hall, 2009. 320 p. ISBN 9788576051893 (broch.).
3. ALONSO, Gustavo. Web services: concepts, architectures and applications. Berlin: Springer, 2010. xx, 354 p. ISBN 9783642078880 (broch.).

Bibliografia Complementar:

1. JOSUTTIS, Nicolai M. SOA na prática: a arte da modelagem de sistemas distribuídos. Rio de Janeiro, RJ: Alta Books, 2008. 265 p. ISBN 9788576081845 (broch.).
2. SCHMIDT, Douglas C. Pattern-oriented software architecture: volume 2. Chichester [England]; New York: John Wiley & Sons, 2000. 633 p. (Wiley series in software design patterns) ISBN 0471606952(enc.).
3. BUSCHMANN, Frank et al. Pattern-oriented software architecture: volume 4: a system of patterns. New York: John Wiley & Sons, 1996. 457 p. ISBN 0471958697 .
4. VÖLTER, Markus; KIRCHER, Michael; ZDUN, Uwe. Remoting patterns foundations of enterprise, internet and realtime distributed object middleware. Chichester: John Wiley, 2005. 389 p. (Wiley series in software design patterns) ISBN 0470856629.
5. KERIEVSKY, Joshua. Refatoração para padrões. Porto Alegre: Bookman, 2008. 400 p. ISBN 9788577802449 (broch.)

Experimentação em Engenharia de Software**Ementa:**

Conceituação e esclarecimento acerca de experimento controlado, estudos de caso e surveys. Processo de desenvolvimento de um projeto de pesquisa (inclui atividades, formulação de questões, construção de teoria e análise qualitativa/quantitativa de dados). Investigação de experimentos científicos em engenharia de software. Prática acompanhada de pequeno experimento em engenharia de software.

Bibliografia Básica:

1. EXPERIMENTATION in software engineering: an introduction. Boston, MA: Kluwer Academic, 2000. 204 p. (The Kluwer international series in software engineering ; 6) ISBN 0792386825 (enc.).
2. SHULL, Forrest; SINGER, Janice; SJOBERG, Dag I. K SPRINGERLINK. Guide to advanced empirical software engineering. London: Springer-Verlag London Limited, 2008. ISBN 9781848000445.
3. JURISTO, Natalia; MORENO, Ana M. Basics of software engineering experimentation. Boston: Kluwer Academic Publishers, 2001. 395 p. ISBN 079237990X (enc.).

Bibliografia Complementar:

1. TRAVASSOS, G. et. al. Introdução a Engenharia de Software Experimental. Rio de Janeiro: COPPE/UF RJ, 2002. Relatório Técnico ES-590/02.
2. KITCHENHAM, B. Procedures for Performing Systematic Reviews. Australia: Joint Technical Report Keele University / NICTA Technical/ Keele University / NICTA, 2004.
3. Wazlawick, Raul Sidnei; Engenharia de Software - Conceitos e Práticas. Campus. Edição : 1 / 2013. I.S.B.N.: 9788535260847.
4. Pressman, Roger S.; Engenharia de Software - Uma Abordagem Profissional - 7ª Edição; Amgh Editora. I.S.B.N.: 9788563308337.
5. Junior, Hélio Engholm; Engenharia de Software na Prática / NOVATEC.

Computação em Nuvem**Ementa:**

Introdução a Computação em Nuvem, Princípios da Computação em Nuvem, Arquitetura da Computação em Nuvem, Modelos de Serviço: Infraestrutura como um Serviço, Plataforma como um Serviço e Software como um Serviço, Gerenciamento de Dados em Nuvem, Middlewares para a Computação em Nuvem, Gerenciamento e Monitoramento da Nuvem, Migração de Aplicações para

Nuvem.

Bibliografia Básica:

1. TAURION, Cezar. Cloud Computing: computação em nuvem, transformando o mundo da Tecnologia da Informação. Rio de Janeiro, RJ: Brasport, 2009.
2. COULOURIS, George F.; DOLLIMORE, Jean; KINDBERG, Tim. Sistemas distribuídos: conceitos e projeto. 4. ed. Porto Alegre: Bookman, 2007.
3. OZSU, M.Tamer. Principles of distributed database systems. 3rd. ed. New York: Springer, 2011.

Bibliografia Complementar:

1. JENNINGS, Roger. Cloud computing with the Windows Azure Platform. Indianapolis, Indiana: Wiley Pub., 2009.
2. WHITE, Tom. Hadoop: the definitive guide. California: O'Reilly, 2009.
3. TANENBAUM, Andrew S.; STEEN, Van Maarten; MARQUES, Arlete Simille. Sistemas distribuídos: princípios e paradigmas. 2. ed. São Paulo, SP: Prentice Hall, 2007.
4. JOSUTTIS, Nicolai M. SOA na prática: a arte da modelagem de sistemas distribuídos. Rio de Janeiro, RJ: Alta Books, 2008.
5. HAY, Chris; PRINCE, Brian H. Azure in action. Stamford, Ct: Manning, 2011.

Inglês Instrumental I

Ementa:

Vocabulário ligado à Informática. Técnicas de Leitura e Compreensão. Tradutores. Verbos e tempos verbais. Nomes e Pronomes.

Bibliografia Básica:

1. MURPHY, R. English Grammar in use.3 ed. Cambridge University, 2004.
2. GALLO, L. R. Inglês instrumental para informática. Ícone Ed.2008.
3. PHILIPS, D. Longman Preparation Course for the TOEFL Test Ibt. Longman do Brasil, 2007.
4. MARINOTTO, D. Reading on info tech: inglês para informática. 2 ed. Novatec, 2007.

Bibliografia Complementar:

1. COLLINS. COLLINS: dicionário escolar. Martins Fontes, 2009.
2. EFRAIM, T. Decision support and business. 9 ed. Prentice Hall, 2010.
3. MACMILLAN English dictionary for advanced learners. 2 ed. Macmillan Education, 2007. ISBN: 9780230025455
4. MACMILLAN EDUCATION (EDITORA). MacMillan English Dictionary for advanced learners of American English. Macmillan Education
5. Longman gramática escolar da língua inglesa: gramática de referências com exercícios e respostas LONGMAN, 2004, Longman do Brasil

Inglês Instrumental II

Ementa:

Conversação, leitura de artigos e jornais da área. Escrita de trabalhos técnicos. Apresentação de seminários. Noções avançadas de gramática e compreensão de texto.

Bibliografia Básica:

1. MURPHY, R. English Grammar in use.3 ed. Cambridge University, 2004.
2. GALLO, L. R. Inglês instrumental para informática. Ícone Ed.2008.
3. PHILIPS, D. Longman Preparation Course for the TOEFL Test Ibt. Longman do Brasil, 2007.
4. MARINOTTO, D. Reading on info tech: inglês para informática. 2 ed. Novatec, 2007.

Bibliografia Complementar:

1. COLLINS. COLLINS: dicionário escolar. Martins Fontes, 2009.
2. EFRAIM, T. Decision support and business. 9 ed. Prentice Hall, 2010.
3. MACMILLAN English dictionary for advanced learners. 2 ed. Macmillan Education, 2007. ISBN: 9780230025455
4. MACMILLAN EDUCATION (EDITORA). MacMillan English Dictionary for advanced learners of

American English. Macmillan Education

5. Longman gramática escolar da língua inglesa: gramática de referências com exercícios e respostas LONGMAN, 2004, Longman do Brasil

Educação em Direitos Humanos

Ementa:

Direitos Humanos, democratização da sociedade, cultura e paz e cidadanias. O nascituro, a criança e o adolescente como sujeitos de direito: perspectiva histórica e legal. O ECA e a rede de proteção integral. Educação em direitos humanos na escola: princípios orientadores e metodologias. O direito à educação como direito humano potencializador de outros direitos. Movimentos, instituições e redes em defesa do direito à educação. Igualdade e diversidade: direitos sexuais, diversidade religiosa e diversidade étnica. Os direitos humanos de crianças e de adolescentes nos meios de comunicação e nas mídias digitais.

Bibliografia Básica:

1. BRASIL. Congresso Nacional. Constituição da República Federativa do Brasil. Brasília, 1988.
2. BRASIL. Secretaria Especial de Direitos Humanos. Estatuto da Criança e do Adolescente (Lei no 8069/90). Brasília, 2008.
3. BRASIL. MEC. Lei de Diretrizes e Bases da Educação Nacional (Lei no 9394/96). Brasília, 1996.
4. BRASIL. Comitê Nacional de Educação em Direitos Humanos/Secretaria Especial dos Direitos Humanos.
5. FERREIRA, Lúcia de Fátima G.; ZENAIDE, Maria de N. T. E DIAS, Adelaide Alves (Orgs). Direitos humanos na educação superior: subsídios para a educação em direitos humanos na pedagogia. João Pessoa: editora Universitária da UFPB, 2010.
6. JARES, Xesus R. Educação para a paz: sua teoria e sua prática. 2. ed. Porto Alegre: Artmed, 2002.

Bibliografia Complementar:

1. MOTA, Maria Dolores de Brito et al. A Escola diz não à violência. Fortaleza: Expressão Gráfica e Editora, 2007.
2. OLINDA, Ercília Maria Braga de. Educação em Direitos Humanos. Material Instrucional do Curso de Pedagogia Semipresencial da UFC. Fortaleza, 2012.
3. PEREIRA, Lúcia. Ludicidade: algumas reflexões. IN Porto, B. Ludicidade: o que é mesmo isso? Salvador, Universidade Federal da Bahia, Faculdade de Educação, PPGE, GEPEL, 2002.
4. Plano Nacional de Educação em Direitos Humanos. Brasília: MEC/MJ/UNESCO, 2009.
5. Projeto de Lei no 478/2007. Dispõe sobre o Estatuto do Nascituro e dá outras providências. Comissão de Seguridade Social e Família. Brasília, 2010.
6. RAYO, José Tuvilla. Educação em direitos humanos: rumo a uma perspectiva global. 2. ed. Porto Alegre: Artmed, 2004.
7. SILVEIRA, Rosa Maria Godoy et al. Educação em direitos humanos: fundamentos teórico-metodológicos. João Pessoa: Editora Universitária, 2007.

Educação Ambiental

Ementa:

Educação Ambiental, conceitos e metodologias na pesquisa e no ensino. Princípios da Educação Ambiental. Fundamentos filosóficos e sociológicos da Educação Ambiental. Tratado de Educação Ambiental para Sociedades Sustentáveis; A Agenda XXI; A Carta da Terra e outros marcos legais da EA. Educação Ambiental e sua Contextualização (Urbana e Rural). Paradigmas Epistemo-educativos Emergentes e a Dimensão Ambiental. Educação Ambiental: uma abordagem crítica. Educação Ambiental Dialógica e a Práxis em Educação Ambiental.

Bibliografia Básica:

1. CARVALHO, I. C. M. A invenção ecológica: sentidos e trajetórias da educação ambiental no Brasil. 2. Ed. Porto Alegre, RS: Editora da UFRGS, 2002.
2. FIGUEIREDO, João B. A. Educação Ambiental Dialógica e Representações Sociais da Água em Cultura Sertaneja Nordestina: uma contribuição à consciência ambiental em Irauçuba-Ce (Brasil). 2003. Tese (Doutorado em Ciências Biológicas/Ecologia/Educação Ambiental) – Universidade Federal de São

Carlos – UFSCar, São Carlos, SP, 2003.

3. GUIMARÃES, Mauro. A dimensão ambiental na educação. Campinas, SP: Papirus, 1995.

4. GUIMARÃES, Mauro. Educação ambiental: No consenso um embate? Campinas, SP: Papirus, 2000.

5. LOUREIRO, Carlos Frederico B. Trajetória e fundamentos da educação ambiental. São Paulo, SP: Cortez, 2004.

6. REIGOTA, M. O que é educação ambiental. 1a reimp. São Paulo, SP: ed. Brasiliense, 1996. (Coleção Primeiros Passos).

7. TRISTÃO, Martha. Espaços/Tempos de formação em educação ambiental. In: GUERRA, Antônio F. S. & TAGLIEBER, José E. (orgs.). Educação Ambiental: fundamentos, práticas e desafios. Itajaí: Universidade do Vale do Itajaí, 2007.

Bibliografia Complementar:

1. DUSSEL, Enrique. Europa, modernidade e eurocentrismo. In: LANDER, Edgardo. (Org.). A colonialidade do saber: eurocentrismo e ciências sociais. 1a. Ed. Buenos Aires: Consejo Latinoamericano de Ciencias Sociales – CLACSO, 2005.

2. FREIRE, Paulo. Pedagogia do oprimido. Rio de Janeiro, RJ: Paz e Terra, 1974/13 ed., 1983.

3. FREIRE, Paulo. Pedagogia da Esperança: em reencontro com a pedagogia do oprimido. Rio de Janeiro, RJ: Paz e Terra, 1992.

4. FREIRE, Paulo. Pedagogia da autonomia: saberes necessários à prática educativa. São Paulo: Paz e Terra, 1997.

5. LANDER, Edgardo. (Org.). Ciências Sociais: saberes coloniais e eurocêntricos. In: LANDER, Edgardo. (Org.). A colonialidade do saber: eurocentrismo e ciências sociais. 1a ed. Buenos Aires: Consejo Latinoamericano de Ciencias Sociales – CLACSO, 2005.

6. QUIJANO, Aníbal. Colonialidad y Modernidad/Racionalidad. Revista Perú Indígena. Vol. 13, No. 29, 1991, pp.11-20, Lima, Perú, 1991.

7. QUIJANO, Aníbal. Colonialidade do poder, eurocentrismo e ciências sociais. In: LANDER, Edgardo. (Org.). A colonialidade do saber: eurocentrismo e ciências sociais. 1a. Ed. Buenos Aires: Consejo Latinoamericano de Ciencias Sociales – CLACSO, 2005.

Relações Étnico-Raciais e Africanidades

Ementa:

Negritude e pertencimento étnico. Conceitos de africanidades e afrodescendência. Cosmovisão africana: valores civilizatórios africanos presentes na cultura brasileira. Ancestralidade e ensinamentos das religiosidades tradicionais africanas nas diversas dimensões do conhecimento no Brasil. Introdução à geografia e história da África. As origens africanas e as nações africanas representadas no Brasil. O sistema escravista no Brasil e no Ceará. Aportes dos africanos à formação social e cultural do Brasil e do Ceará. Personalidades africanas, afrodescendentes e da diáspora negra que se destacaram em diferentes áreas do conhecimento. Contexto das Ações Afirmativas hoje. Atualização do legado africano no Brasil. Desconstrução de preconceitos e desdobramentos teórico-práticos para a atuação do profissional na sua área de inserção no mercado de trabalho.

Bibliografia Básica:

1. ARCO-VERDE, Yvelise Freitas de Souza. Prefácio. In Cadernos Temáticos – História e cultura afro-brasileira e africana: educando as relações étnico-raciais. Curitiba: SEED-PR, 2006.

2. BRASIL. CNE. Parecer no. 03 de 10 de março de 2004. Dispõe sobre as diretrizes curriculares nacionais para a educação das relações étnico-raciais e para o ensino de história e cultura afro-brasileira e africana. Relatora: Petronilha Beatriz Gonçalves e Silva. Ministério da Educação. Brasília, julho de 2004.

3. BRASIL. Constituição da República Federativa do Brasil. São Paulo: Ed. Revistas dos Tribunais, 1988.

4. BRASIL. Ministério do Planejamento, Orçamento e Gestão. IBGE. Síntese de indicadores Sociais: Uma análise das condições de vida da população brasileira 2007. Rio de Janeiro, 2007. Disponível em http://200.130.7.5/spmu/docs/indic_sociais2007_mulher.pdf.

5. BRASIL. Lei 10639 de 09 de janeiro de 2003. Inclui a obrigatoriedade da temática “História e Cultura Afro-Brasileira e Africana” no currículo oficial da rede de ensino. Diário Oficial da União. Brasília, 2003.

6. BRASIL. Lei 11645 de 10 de março. Altera a Lei no 9.349 de 20 de dezembro de 1996, modificada pela Lei no. 10639, de 9 de janeiro de 2003, que estabelece as diretrizes e bases da educação nacional,

para incluir no currículo oficial da rede de ensino a obrigatoriedade da temática “História e Cultura Afro-Brasileira e Indígena”. Diário Oficial da União. Brasília, 2008.

7. BITTENCOURT, Circe. Identidade nacional e ensino de História do Brasil. In: KARNAL, Leandro (org.). História na sala de aula: conceitos, práticas e propostas. São Paulo: Contexto, 2005.

8. CAVALLEIRO, Eliane. Educação anti-racista: compromisso indispensável para um mundo melhor. In: 9. CAVALLEIRO, Eliane (org.). Racismo e anti-racismo na educação: repensando nossa escola. São Paulo. SUMMUS, 2001.

10. CRUZ, Mariléia dos Santos. Uma abordagem sobre a história da educação dos negros. In: ROMÃO, Jeruse (org.). História do negro e outras histórias. Secretaria de Educação Continuada, Alfabetização e Diversidade: - Brasília: Ministério da Educação, Secretaria de Educação Continuada, Alfabetização e Diversidade, 2005.

Bibliografia Complementar:

1. CUNHA JUNIOR, Henrique. A história africana e os elementos básicos para o seu ensino. In. COSTA LIMA, Ivan e ROMÃO, Jeruse (org.). Negros e currículo. Série Pensamento Negro em Educação no. 2. Florianópolis: Núcleo de Estudos Negros/NEN, 1997.

2. CUNHA JUNIOR, Henrique. Abolição inacabada e a educação dos afrodescendentes. In Revista Espaço Acadêmico, no. 89, Outubro de 2008. Disponível em <http://www.espacoacademico.com.br/089/89cunhajr.pdf>.

3. DIAS, Lucimar Rosa. Quantos passos já foram dados? A questão da raça nas leis educacionais. Da LDB de 1961 à Lei 10639 de 2003. In: ROMÃO, Jeruse (org.). História do negro e outras histórias. Secretaria de Educação Continuada; Alfabetização e Diversidade: - Brasília: Ministério da Educação, Secretaria de Educação Continuada, Alfabetização e Diversidade, 2005.

4. FOGAÇA, Azuete. Educação e identidade negra. Série-Estudos – Periódico do Mestrado em Educação da UCDB. Campo Grande – MS, n.22, p. 31-46, jul./dez. 2006.

5. LOPES, Marta Teixeira e GALVÃO, Ana Maria de Oliveira. História da Educação. Coleção [o que você precisa saber sobre...]. Rio de Janeiro: DP&A, 2001.

6. MAESTRI, Mário. A pedagogia do medo: disciplina, aprendizado e trabalho na escravidão brasileira. In: STEPHANOU, Maria e BASTOS, Maria Helena Câmara (org.). História e memórias da educação no Brasil, vol. I: séculos XVI – XVIII. Petrópolis, RJ; Vozes, 2004.

7. PARANÁ, CEE. Deliberação no. 04 de 02 de agosto de 2006. Institui normas complementares às Diretrizes Curriculares Nacionais para a Educação das Relações Étnico-Raciais e para o Ensino de História e Cultura Afro-Brasileira e Africana. Relator: Romeu Gomes de Miranda, Marília Pinheiro Machado de Souza, Lygia Lumina Pupatto, Domenico Costella e Maria Tarcisa Silva Bega. Secretaria de Estado da Educação. Curitiba, 2006.

8. PARANÁ, SEED. Diretrizes Curriculares de Artes para os anos finais do Ensino Fundamental e para o Ensino Médio. Curitiba: SEED, 2008. Disponível em http://www.diaadiaeducacao.pr.gov.br/diaadia/arquivos/File/livro_e_diretrizes/diretrizes.

Língua Brasileira de Sinais

Ementa:

Fundamentos histórico culturais da Libras e suas relações com a educação dos surdos. Parâmetros e traços linguísticos da Libras. Cultura e identidades surdas. Alfabeto datilológico. Expressões não-manuais. Uso do espaço. Classificadores. Vocabulário da Libras em contextos diversos. Diálogos em língua de sinais.

Bibliografia Básica:

1. CAPOVILLA, Fernando. C; RAPHAEL, Walkyria. D. Dicionário Enciclopédico Ilustrado Trilíngue da Língua de Sinais. 3ª Ed. São Paulo: EDUSP, 2008

2. FELIPE, Tânia Amara. Libras em Contexto: curso básico. Brasília: MEC/SEESP, 2007

3. LABORIT, Emmanuelle. O Vão da Gaivota. Best Seller, 1994.

4. QUADROS, Ronice Muller; KARNOPP, Lodenir B. Língua de Sinais Brasileira: estudos lingüísticos. Porto Alegre: ArtMed, 2004.

5. SACKS, Oliver. Vendo Vozes: uma viagem ao mundo dos surdos. São Paulo: Cia. Das Letras, 1998.

Bibliografia Complementar:

1. CHAVES, Ernando P. Sinaliza, surdo!: caracterização da construção de um modelo de escola de surdos.

- Dissertação (Mestrado em Educação Brasileira). Faculdade de Educação, UFC. 2003. 110 p.
2. FERNANDES, Eulália. Linguagem e surdez. Porto Alegre. Editora Artmed, 2003
 3. FERREIRA-BRITO, Lucinda. Integração Social & Educação de Surdos. Rio de Janeiro: Babel Editora, 1993.
 4. FERREIRA-BRITO, Lucinda. Por uma Gramática da Língua de Sinais. Rio de Janeiro: Tempo Brasileiro, 1995.
 5. GOES, Maria Cecília Rafael; SMOLKA, Ana Luiza B. A linguagem e o outro no espaço escolar: Vygotsky e a construção do conhecimento. Campinas: Papyrus, 1993.
 6. GOLDFELD, Marcia. A Criança Surda: linguagem e cognição numa perspectiva sócio-interacionista. São Paulo: Plexus, 1997.
 7. LACERDA, Cristina Broglia. GOES, Cecília Rafael de. Surdez: processos educativos e subjetividade. São Paulo: LOVISE, 2000
 8. LANE, Harlan. A máscara da benevolência: comunidade surda amordaçada. Lisboa: Instituto PIAGET, 1997.
 9. LIMA-SALLES, Heloisa Maria Moreira (org). Bilinguismo dos surdos: questões lingüísticas e educacionais. Goiânia: Cãnone Editorial, 2007
 10. SOUZA, Margarida M. P. Voando com Gaivotas: um estudo das interações na educação de surdos. Dissertação (Mestrado em Educação Brasileira). Faculdade de Educação, UFC. 2008. 152 p.
 11. QUADROS, Ronice Muller de. Educação de surdos: a aquisição da linguagem. Porto Alegre: Artes Médicas, 1997
 12. SÁ, Nídia Regina Limeira de. Cultura, Poder e Educação de Surdos. Manaus: INEP, 2002.
 13. SKLIAR, Carlos. (org). Educação e Exclusão: abordagens sócio-antropológicas em educação especial. Porto Alegre: Mediação, 1997.
 14. SKLIAR, Carlos. A Surdez: um olhar sobre as diferenças. Porto Alegre: Mediação, 1998.
 15. THOMA, Adriana da Silva e LOPES, Maura Corcini (orgs), A Invenção da Surdez: cultura, alteridade, identidade e diferença no campo da educação, Santa Cruz do Sul, EDUNISC, 2004
 16. LEITÃO, Vanda M. Narrativas silenciosas de caminhos cruzados: história social de surdos no Ceará. Tese (Doutorado em Educação Brasileira). Faculdade de Educação, UFC. 2003. 225 p.
 17. WILCOX, Sherman; WILCOX, Phyllis. Aprender a ver. Trad.: Tarcísio Leite. Rio de Janeiro: Arara Azul, 2005.

4. GESTÃO ACADÊMICA DO CURSO

4.1 Coordenação

O coordenador do curso de Engenharia de Software do Campus de Russas é o Prof. Dr. Markos Oliveira Freitas. Este assumiu a coordenação em 2017.1, com a realocação da antiga coordenadora, Profa. Dra. Maria Viviane de Menezes, para o Campus de Quixadá da Universidade Federal do Ceará. O coordenador foi responsável por coordenar as atividades de revisão deste Projeto Pedagógico.

4.2 Colegiado

O Colegiado do curso de Engenharia de Software é composto pelo Prof. Dr. Markos Oliveira Freitas (coordenador), pelo Prof. Ms. Marcos Vinicius de Andrade Lima (vice-coordenador e representante da unidade curricular de Engenharia de Software Aplicada), pela Profa. Ms. Tatiane Fernandes Figueiredo (representante da unidade curricular de Ciência da Computação), pelo Prof. Ms. Filipe Maciel Roberto (representante da unidade curricular de Matemática), pelo Prof. Ms. José Osvaldo Mesquita Chaves (representante da unidade curricular de Engenharia de Software), pela

Profa. Dra. Marília Soares Mendes (representante da unidade curricular de Tecnologia da Informação), pelo Prof. Ms. Daniel Márcio Batista de Siqueira (representante da unidade curricular de Formação Complementar e Humanística), pela Profa. Dra. Valéria Lelli Leitão Dantas (representante da unidade curricular de Formação Suplementar) e por Jeferson da Silva Juliani (representante discente).

O colegiado vem atuando desde que foi formado, em 2016, principalmente no estímulo motivacional dos alunos. O colegiado tem incentivado a participação dos alunos em ações de extensão, cursos extra-curriculares e palestras, bem como em seleções de monitorias e iniciação a docência e em grupos de pesquisa. O colegiado tem intensificado a divulgação da Avaliação Institucional, para que cada vez mais alunos participem e exponham suas perspectivas com relação ao curso.

O colegiado ainda organizará palestras sobre o ENADE (Exame Nacional de Desempenho de Estudantes), bem como encorajará os professores do curso a incluírem discussões sobre o exame nas disciplinas, juntamente com a utilização de questões de provas passadas. O colegiado também planejará palestras para os professores sobre como criar questões no estilo das presentes no ENADE.

4.3 Núcleo Docente Estruturante

O Núcleo Docente Estruturante do curso de Engenharia de Software é composto pelo Prof. Dr. Markos Oliveira Freitas (presidente), pela Profa. Dra. Marília Soares Mendes, pela profa. Dra. Valéria Lelli Leitão Dantas, pelo Prof. Ms. Marcos Vinicius de Andrade Lima, pelo Prof. Ms. Daniel Márcio Batista de Siqueira e pelo prof. Ms. José Osvaldo Mesquita Chaves.

A principal ação do NDE implementada foi a revisão da matriz curricular do curso, especialmente, para se adequar às Diretrizes Curriculares Nacionais, as quais não estavam completamente presentes no curso. Em meio a isso, também foram feitas recomendações sobre mudanças em ementas, cargas horárias, pré-requisitos e bibliografias de disciplinas, para atualizar as disciplinas e adequá-las à realidade do curso, do corpo docente e da região.

4.4 Integração com as redes públicas de ensino

Devido à implantação recente do Campus de Russas, apenas alguns projetos têm sido feitos com os alunos de escolas públicas. Algumas das principais ações de extensão do curso são o projeto “Tecnologia para Todos”, que visa contribuir para a inclusão digital de jovens e adolescentes, o projeto “Incentivo a Projetos de Engenharia com a Plataforma Arduino”, que proporciona aos alunos de ensino médio contato imediato com conhecimentos relacionados a projetos com sensores e sistemas embarcados, o projeto “Desenvolvimento de Sistemas Acessíveis para o Município de Russas”, em que alunos de escolas públicas de ensino médio de Russas trabalham no desenvolvimento de

softwares acessíveis sob a supervisão de alunos do curso, e a “SESCOMP” que, em 2016, contou com um dia voltado a alunos do ensino médio, divulgando o curso e visando atrair novos alunos.

4.5 Apoio ao discente

O curso demanda conhecimentos prévios de matemática do ensino médio. Alguns alunos nos primeiros semestres do curso necessitam de reforço escolar em matérias ligadas à matemática. Pretende-se aqui fornecer atividades extracurriculares para o ensino da matemática do ensino médio.

Para combater a reprovação e a evasão no curso de Engenharia de Software, o Campus de Russas mantém programas/projetos de monitoria, de iniciação acadêmica, de iniciação à docência, de apoio ao curso e de aprendizagem colaborativa que são mantidas com bolsas providas pela Universidade e conseguidas através de editais internos para permitir que alunos que se destacaram nas referidas disciplinas possam ajudar outros alunos a estudar, tirando dúvidas e ajudando no estudo de materiais complementares atribuídos pelos professores. A formação de grupos de estudo com a participação dos alunos também é incentivada pela coordenação e professores.

Um projeto que merece destaque é o de “Apoio seu curso”, no qual estudantes veteranos acompanham os novatos no seu primeiro semestre, através de diálogos sobre experiências prévias nos semestres iniciais, grupos de estudo, realização de atividades extra-curriculares motivacionais e divulgação das atividades do campus. Cada aluno veterano é responsável por um grupo de alunos novatos, orientando-os no início da vida acadêmica e ajudando-os em suas dificuldades escolares e sociais. O projeto conta ainda com um aluno do primeiro semestre, com a função de observar os colegas e descobrir as principais causas de reprovação em disciplinas e de abandono do curso.

Além dos programas mencionados, o Campus de Russas mantém programas e projetos que tem o objetivo de permitir a aplicação mais prática dos conhecimentos das diversas disciplinas como forma de aprofundamento e engajamento dos alunos no ambiente educacional, como a maratona de programação e projetos de inclusão digital, de desenvolvimento de software e de desenvolvimento de sistemas embarcados. Esses programas permitem que os alunos trabalhem em conjunto fora do ambiente das disciplinas, mas recebam retorno sobre a efetividade do aprendizado adquirido nas disciplinas, funcionando como avaliação informal oriunda dos pares (outros alunos que também participam das iniciativas), do professor responsável pela atividade e auto-avaliação feita pelo próprio aluno com base no desempenho pessoal.

O Campus de Russas possui ainda um setor de Assuntos Estudantis, que busca disponibilizar uma ação psicopedagógica, onde existe orientação e acompanhamento aos alunos que se encontram em dificuldades emocionais, vocacionais e outras dificuldades que possam comprometer o aprendizado do aluno. Além disso, esse setor

de Assuntos Estudantis promove o programa Auxílio Moradia, que tem como objetivo viabilizar a permanência dos estudantes matriculados do Campus de Russas, em comprovada situação de vulnerabilidade econômica, assegurando-lhes auxílio institucional para complementação de despesas com moradia e alimentação durante todo o período do curso ou enquanto persistir a mesma situação. Para tanto, o aluno precisa estar matriculado e frequentando o curso, não ter concluído nenhum curso de graduação e não ter família nuclear residindo na sede do curso.

4.6 Acompanhamento e avaliação do PPC

A avaliação e a atualização curricular devem constituir um processo contínuo, dinâmico, aberto e flexível, com o intuito de manter o curso de Bacharelado em Engenharia de Software sintonizado com as necessidades do ambiente externo e propiciar o aperfeiçoamento constante das condições de ensino do curso. Assim, a avaliação deve ser uma concepção incorporada ao desenvolvimento das atividades do curso no âmbito da sala de aula, no âmbito da unidade acadêmica que é responsável pelo curso e no âmbito da própria instituição de ensino superior.

Para efetivação dessa avaliação, poderão ser utilizados instrumentos e técnicas diversos, tais como questionários, entrevistas, grupos focais, entre outras metodologias que permitam o levantamento de dados acerca da implementação do curso de Engenharia de Software do Campus de Russas.

Como estratégias de ação, planeja-se realizar de regularmente:

- A discussão ampla do projeto pedagógico com o corpo docente do curso para avaliação da proposta formativa, buscando averiguar sua adequação aos parâmetros curriculares da área das TIC, bem como sua relação com o contexto local e regional em que o curso está inserido e com a qualificação e experiência acadêmica e profissional de seus professores. Entende-se que o Colegiado do Curso e o Núcleo Docente Estruturante serão proponentes e executores dessa conjectura, desenvolvendo adequadamente os seus instrumentos e metodologias. Contudo, pode-se adiantar, sabe-se que a necessidade de avaliação do projeto pedagógico é permanente, então se subentende que frequentemente o ensino, o currículo e o PPC deverão ser objeto de discussão, refletindo sobre o prescrito nesta projeção e o exequível.
- A escuta dos alunos, para averiguar se suas expectativas em relação à formação estão sendo atendidas, para levantar as possíveis dificuldades existentes nas disciplinas, nos processos de ensino e de aprendizagem, como também se as condições de infraestrutura (salas de aula, laboratório, acervo da biblioteca) atendem às suas necessidades.

O curso de Engenharia de Software possui um Núcleo Docente Estruturante do Curso, formado por professores da área do curso, que contém atribuições consultivas,

propositivas e de assessoria sobre matéria de natureza acadêmica, corresponsável pela elaboração, implementação, acompanhamento, atualização e consolidação do Projeto Pedagógico do Curso. A resolução No 10/2012/CEPE rege sobre as atribuições desse núcleo.

Tecnologias mais modernas que venham a ser empregadas no mercado serão incorporadas nas práticas de laboratório e nos trabalhos passados aos alunos, de forma que a grade curricular proposta se mostre concisa, abrangente e adaptável, o que permite um acompanhamento e avaliação bastante adequados dos resultados obtidos na implementação deste projeto pedagógico.

Sobre o acompanhamento e avaliação do Projeto Pedagógico deste curso, expõe-se o entendimento da necessidade de que o acompanhamento e a avaliação desta projeção sejam feitos por todos os membros da comunidade acadêmica. Assim, dizemos que professores e estudantes farão avaliação da proposta, analisando sua concepção, sua execução e o atendimento aos objetivos expostos neste documento. Seguir-se-á as recomendações da Comissão Própria de Avaliação (CPA), analisando: o planejamento docente, a atuação do professor na execução do planejado, as formas de acompanhamento da aprendizagem discente. Do mesmo modo, entende-se a importância de que professores e estudantes se autoavaliem e avaliem o curso, em especial, através da Avaliação Institucional da UFC, identificando-se, possivelmente, procedimentos, hábitos e métodos capazes de progressão do ensino e obtenção de qualidade na formação profissional dos estudantes.

Por fim, a revisão geral deste PPC acontece a cada 3 (três) anos, ou quando se fizer necessário, sem prejuízo de ajustes pontuais que podem ser realizados a qualquer momento pelo Colegiado para correção de imperfeições detectadas, utilizando como insumos os resultados das autoavaliações institucionais, resultados de avaliações ENADE, Conceito Preliminar de Curso (CPC) obtidos no triênio e outros indicadores utilizados pelo INEP.

5. INFRAESTRUTURA DO CURSO

A primeira unidade didática do Campus de Russas possui 12 salas de aula com capacidade para 60 alunos cada, três salas de aula com capacidade para 30 alunos cada, e três laboratórios de informática com capacidade para 30 alunos cada, equipados com computadores de boa capacidade, que atendem aos requisitos de programação do curso. A unidade didática conta ainda com biblioteca e auditório, e salas de videoconferência, de estudos, de atendimento a alunos, de professores, de coordenações, de administração, de atendimento psicológico, de assistência social, de secretarias de cursos, de coordenação acadêmica, de reunião, e de direção. Por possuir um piso superior, dois elevadores estão instalados na unidade didática.

O Campus de Russas possui ainda um prédio localizado no centro da cidade, a cerca de um quilômetro de distância da unidade didática. Este prédio possui salas administrativas onde, temporariamente, funcionam os centros acadêmicos e os escritórios das empresas juniores dos cursos oferecidos pelo campus. Além disso, nesse prédio, funciona, também temporariamente, o Restaurante Universitário do campus.

No momento, em 2017.1, encontra-se em construção mais uma unidade didática, semelhante à unidade existente, e de frente a esta. No mesmo terreno, estão também em construção o Restaurante Universitário e um bloco de laboratórios a serem utilizados pelos cursos de Engenharia Civil, Engenharia Mecânica e Engenharia de Produção do campus.

Entre os equipamentos disponibilizados aos professores, o campus possui computadores, projetores, impressoras e multifuncionais.

A diretoria do Campus de Russas é composta pelo Prof. Dr. Lindberg Lima Gonçalves e pela Profa. Dra. Aliny Abreu de Sousa Monteiro. O corpo docente, além de contar com os diretores, conta com cerca de 30 professores mestres e doutores nas áreas de Ciência da Computação e Engenharia de Software, Engenharias Civil, Mecânica e de Produção, e Matemática, Física e Química.

O corpo de servidores técnico-administrativos é composto pelas bibliotecárias, pela assistente social, pela psicóloga, pelos técnicos de laboratórios de informática, de física, de química e das áreas de engenharia, pelos assistentes administrativos e secretários do campus e do curso, pelo analista de tecnologia de informação, pelo engenheiro civil prefeito do campus e pela técnica em edificações.

6. REFERÊNCIAS

[IEEE-CS/ACM 2014]: IEEE/ACM Joint Task Force on Computing Curricula, **Software Engineering 2014, Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering**, IEEE Computer Society and ACM, 2014. Disponível em: <http://www.acm.org/education/curricula-recommendations>.

[IEEE-CS 2004]: IEEE, **SWEBOK 2004 - Guide to the Software Engineering Body of Knowledge**, IEEE Computer Society, 2004. Disponível em: <http://swebok.org>.

[SIT 2009]: Stevens Institute of Technology, **Curriculum Guidelines for Graduate Degree Programs in Software Engineering**, Integrated Software & Systems Engineering Curriculum (iSSEc) Project, 2009. Disponível em: <http://www.acm.org/education/curricula-recommendations>.

[RUF 2016]: Folha de São Paulo, **Ranking Universitário Folha**, 2016, Disponível em: <http://ruf.folha.uol.com.br/2016/ranking-de-universidades/>

[QS 2016]: Quacquarelli Symonds, **QS University Rankings - Latin America**, 2016, Disponível em: <https://www.topuniversities.com/university-rankings/latin-american-university-rankings/2016>

[UFC 2016]: Universidade Federal do Ceará, **Anuário Estatístico da UFC 2016 - Base 2016**, 2016, Disponível em: <http://www.ufc.br/a-universidade/documentos-oficiais/322-anuario-estatistico>

[UFC 2013]: Universidade Federal do Ceará, **Plano de Desenvolvimento Institucional 2013-2017**, 2013, Disponível em: <http://www.ufc.br/a-universidade/documentos-oficiais/313-plano-de-desenvolvimento-institucional-pdi>

[CNE 2016]: Conselho Nacional de Educação/Câmara de Educação Superior, **Diretrizes Curriculares Nacionais para os cursos de graduação em Computação**, Ministério da Educação, 2016. Disponível em: <http://portal.mec.gov.br/component/content/article?id=12991>.

7. ANEXOS