

# **PROJETO PEDAGÓGICO DO CURSO DE CIÊNCIAS DA COMPUTAÇÃO DA UNIVERSIDADE FEDERAL DE SANTA CATARINA.**

Elaborado pelo Colegiado do Curso de Ciências da Computação do  
INE/CTC, composto pelos seguintes Professores:

## **Representantes Titulares**

Antônio Carlos Mariani  
Daniel Santana de Freitas  
Isaias Camilo Boratti  
Luis Fernando Friedrich  
Luiz Cláudio V. dos Santos  
Marcelo Menezes Reis  
Olinto José Varela Furtado  
Patrícia Vilain  
Ivan Pontual Costa e Silva  
Maria Álvaro Mendonça

## **Suplentes**

Ricardo Pereira e Silva  
Julio Felipe Szeremeta  
Maria Marta Leite  
Sérgio Peters  
Mauro Roisenberg  
Pedro Alberto Barbeta  
Antônio A. M. Fröhlich  
Ronaldo dos Santos Mello  
Roberto Corrêa da Silva  
Gloria Gil

## **Coordenador do CCO**

José Mazzucco Junior

## **Sub-Coordenador**

Leandro José Komosinski

## Sumário

1	Contextualização do Curso .....	03
1.1	Informações Gerais .....	03
1.2	Histórico .....	03
2	Diagnóstico da Situação Atual do Curso e Justificativa da Reforma .....	04
3	Concepção Filosófica, Teórico- Metodológica do Curso e o Perfil Profissional Almejado .....	12
3.1	Conjunto de Aptidões Esperadas dos Egressos .....	12
3.2	Classes de Problemas que os Egressos Estarão Capacitados a Resolver .....	12
3.3	Habilidades que os Alunos Poderão Desenvolver em Função do Conjunto de Disciplinas Optativas Cursadas ...	13
3.4	Funções que os Egressos Poderão Exercer no Mercado de Trabalho .....	14
3.5	Capacidade de Adaptação do Egresso à Evolução da Computação e de suas Tecnologias .....	14
4	Organização da Estrutura Curricular .....	17
4.1	Disciplinas Obrigatórias .....	18
4.2	Disciplinas Optativas .....	52
4.3	Distribuição das Disciplinas nas Fases Curriculares .....	65
4.3.1	Primeira Fase .....	65
4.3.2	Segunda Fase .....	66
4.3.3	Terceira Fase .....	66
4.3.4	Quarta Fase .....	67
4.3.5	Quinta Fase .....	67
4.3.6	Sexta Fase .....	68
4.3.7	Sétima Fase .....	68
4.3.8	Oitava Fase .....	69
4.3.9	Disciplinas Optativas .....	70
4.4	Metodologia do Curso em Função do Perfil dos Egressos e do Seu Papel na Sociedade .....	71
4.4.1	Área de Formação Básica .....	71
4.4.2	Área de Formação Tecnológica .....	77
4.4.3	Área de Formação Humanística .....	81
4.4.4	Área de Formação Complementar .....	83
4.4.5	Flexibilização Curricular .....	83
5	Formas e Instrumentos de Avaliação do Processo de Ensino e da Aprendizagem .....	85
6	Reajuste Estrutural Necessário para o Atendimento às Necessidades Colocadas no Projeto Pedagógico .....	85

## **1. Contextualização do Curso.**

### **1.1. Localização e Informações Gerais.**

Departamento de Informática e Estatística do Centro Tecnológico –  
INE/CTC; Campus Universitário – Trindade  
88040-900 – Florianópolis – SC – Brasil; Fone: 0XX 48 33319424  
Fax: 0XX 48 331; [www.inf.ufsc.br](http://www.inf.ufsc.br);

Duração Mínima e Máxima: 8 semestres e 17 semestres  
respectivamente;

Período: Diurno;

Número de Vagas: 100 por ano, 50 por semestre;

### **1.2. Histórico.**

A Universidade Federal de Santa Catarina (UFSC), autarquia de regime especial, vinculada ao Ministério da Educação e Cultura (Lei nº 3.849 de 18 de dezembro de 1960 - Decreto nº 64.824 de 15 de julho de 1969), localizada em Florianópolis, capital do estado de Santa Catarina, foi fundada em 1960. O processo de ingresso aos cursos da UFSC é feito através de vestibular, realizado anualmente.

O Departamento de Informática e de Estatística, criado em 20 de outubro de 1970 com o nome de Departamento de Ciências Estatísticas e da Computação - CEC, com a aprovação do Regimento da UFSC, começou a funcionar no primeiro semestre de 1971, tendo assumido a missão de produzir, sistematizar e transmitir conhecimentos nas áreas de Informática e Estatísticas, visando contribuir de forma integrada para o desenvolvimento da sociedade. Inicialmente seus principais objetivos foram oferecer apoio às crescentes pesquisas que vinham sendo realizadas na pós-graduação por diversos departamentos da UFSC e no ensino de graduação nas disciplinas de Informática, Estatística e de Cálculo Numérico a todos os cursos que delas necessitavam.

Em 1976, elaborou-se um projeto para a implantação do curso de Bacharelado em Ciências da Computação, que foi aprovado conforme Portaria nº 1153/GR/76 de 17 de dezembro de 1976, reconhecido pelo Decreto 233/81 de 25 de março de 1981.

O curso de Bacharelado em Ciências da Computação, com duração média de quatro anos, oferece à comunidade cem vagas anuais sendo cinquenta por semestre. O objetivo do curso é formar, com bases científicas e tecnológicas, profissionais para atuar na área de Informática como atividade fim, participando efetivamente do desenvolvimento tecnológico da Computação.

Em 1992, o Departamento de Informática e de Estatística, atendendo a um pleito da comunidade catarinense, implantou o curso de mestrado em Ciência da Computação. Essa criação trouxe grande benefício ao curso de Bacharelado em Ciências da Computação. Foram abertas várias frentes de pesquisa onde os alunos de graduação começaram a participar. Para o aluno do curso de Ciências da Computação, a convivência com diversos grupos e laboratórios de pesquisa é de extrema importância, pois eleva a qualidade da sua formação. Vários alunos de graduação engajam-se em algum projeto de pesquisa e acabam transformando seus projetos de iniciação científica e ou de conclusão de curso em projetos de mestrado.

## **2. Diagnóstico da Situação Atual do Curso e Justificativa da Reforma.**

A estrutura curricular atual do curso entrou em vigor em 1996. Para um curso de Ciência da Computação, onde a evolução tecnológica é impetuosa, o transcorrer de um período de dez anos exige uma atualização rigorosa tanto do projeto pedagógico como de sua estrutura curricular.

No ano de 2000, uma comissão de avaliação do MEC realizou uma análise apurada do curso dividida em três dimensões: corpo docente, plano pedagógico e infra-estrutura. O curso recebeu o conceito “muito bom” em todas as três dimensões e conseqüentemente, recebeu o conceito global “A”. Na dimensão “plano pedagógico”, resumidamente, a comissão elaborou o seguinte relato: “verifica-se que o perfil dos egressos apresenta, de forma adequada, as funções dos mesmos, bem como a metodologia a ser empregada para que os objetivos sejam atingidos. Além disso, a descrição da metodologia apresentada no relatório encontra-se bastante detalhada e clara, mostrando objetivamente como as disciplinas da estrutura curricular serão utilizadas para atingir o perfil do profissional a ser formado. Assim, atribuímos o conceito “A”. Como ponto favorável, pode-se

destacar: a grade curricular condiz com a formação pretendida e a metodologia apresentada se adapta aos objetivos a serem alcançados.

Quanto à estrutura curricular, de forma geral, é satisfatória para um curso de Bacharelado em Ciências da Computação. As disciplinas são semestrais. O currículo proposto possibilita que o perfil do egresso definido seja satisfatoriamente alcançado. É um curso que privilegia a parte conceitual da área com um forte embasamento matemático do contínuo. Observou-se, também, a existência de muitas disciplinas eletivas, com a presença de disciplinas orientadas para a formação empreendedora dos alunos. Existem disciplinas que cobrem razoavelmente a parte de hardware”.

Entretanto, a comissão apontou alguns problemas em indicadores de algumas dimensões. A estrutura curricular, por exemplo, recebeu o conceito “B”, onde foram destacados os seguintes pontos negativos: “Como fatores negativos, destacam-se: a carga horária de Matemática Discreta, relativa a Matemática em geral, é baixa deixando um pouco a desejar na formação dos alunos neste quesito; não há disciplinas contemplando o conteúdo de Análise e Síntese de Algoritmos; as subáreas de Computação Gráfica e Multimídia são contempladas apenas com disciplinas eletivas, sendo a primeira com apenas uma disciplina; banco de Dados possui somente uma disciplina obrigatória, com 72 h/a”.

Além desses pontos negativos explicitamente levantados, a comissão destaca outros problemas principalmente na atualização de ementas e bibliografias de disciplinas.

Dessa forma, um dos principais objetivos da reforma do projeto pedagógico do curso de Ciências de Computação da UFSC é a sua atualização. Nas providências tomadas para alcançá-lo, muitas sugestões oferecidas pela comissão foram consideradas.

O segundo grande objetivo definido pelo Colegiado do curso na reforma do projeto pedagógico foi o reforço do seu perfil de maneira que o mesmo melhor se caracterize como um curso de Ciência da Computação segundo orientações de entidades como MEC, SBC, ACM e IEEE.

Com a implantação do curso de Bacharelado em Sistemas de Informação no INE no ano de 2000, criou-se um espaço alternativo para atender uma demanda específica da sociedade catarinense. O departamento passou a oferecer a possibilidade de formação profissional em dois perfis bem distintos na área de Informática, ou seja, um curso de Bacharelado em Ciência da Computação e um curso de Bacharelado em Sistemas de Informação.

Em 1994, a Comissão de Especialistas em Ensino de Informática, criada pela portaria 161/94 MEC, organizou os cursos de graduação

dessa área em quatro modalidades, a saber: Bacharelado em Ciência da Computação, Bacharelado em Sistemas de Informação, Engenharia da Computação e Licenciatura em Computação. Entretanto, ainda hoje a caracterização exata de cada um desses cursos não é entendida completamente, principalmente pelo mercado de trabalho onde os egressos profissionais são absorvidos. Os cursos de Ciência da Computação e de Sistemas de Informação são os mais afetados por esse conflito reinante (O documento original que define tais categorias pode ser acessado em <http://www.inf.ufrgs.br/mec/ceeinf.perfis.html>). Levando-se em consideração as orientações emitidas pela referida Comissão de Especialista em Ensino de Informática, um dos itens incluídos na reforma trata-se da alteração do nome do curso. Assim sendo, o curso de Bacharelado em Ciências da Computação da UFSC passa a ser denominado de Bacharelado em Ciência da Computação.

O domínio de atuação da Ciência da Computação é extremamente vasto englobando desde os seus fundamentos teórico e algorítmico até os mais avançados desenvolvimentos em robótica e automação inteligente, computação gráfica, sistemas inteligentes, bio-informática etc. O curso de Ciência da Computação oferece ao seu graduado uma fundamentação teórica abrangente que lhe permite participar de maneira efetiva no desenvolvimento tecnológico da computação.

Os trabalhos desenvolvidos pelos cientistas da computação podem ser enquadrados em três grandes categorias:

Projetar e Implementar Software. Esta constitui a categoria clássica dos cientistas da computação e nela os mesmos desenvolvem o desafiante trabalho de análise e de programação, muitas vezes supervisionando outros programadores e sempre atentos às novas abordagens e tecnologias. As atividades de um cientista da computação que recaem sobre essa categoria podem apresentar uma certa intersecção com as atividades típicas de um profissional formado em Sistemas de Informação e provavelmente seja esse cruzamento a origem do conflito muitas vezes existente na caracterização dos dois cursos que, como será apresentado, são distintos.

Vislumbrar Novas Aplicações e Formas de se Utilizar Computadores. A rede mundial de computadores, a World Wide Web, por exemplo, constitui uma maneira revolucionária de se utilizar computadores e foi resultado do progresso marcante da Ciência da Computação, principalmente, nas áreas de redes de computadores, banco de dados e interface homem máquina. Atualmente os pesquisadores vêm trabalhando na construção de robôs inteligentes, na transformação de banco de dados em bancos de conhecimentos, no emprego de computadores para desvendar os

segredos do nosso DNA etc. Nessa categoria o profissional de ciência da computação deve atuar em áreas transdisciplinares envolvendo muitas vezes conhecimentos, técnicas e tecnologias de sub áreas não pertencentes à computação.

Propor e Desenvolver Maneiras Eficientes e Efetivas de se Tratar com Problemas Computacionais. Como exemplos de trabalhos que recaem sobre essa categoria, podemos citar os melhores métodos possíveis de se armazenar grandes quantidades de informações em um banco de dados, de se enviar dados utilizando redes de computadores de maneira extremamente rápida e segura, de gerar e trabalhar com imagens cada vez mais complexas etc. A bagagem teórica e a habilidade de análise e desenvolvimento de algoritmos, adquiridos no decorrer do curso podem auxiliam muito um cientista da computação no desenvolvimento das atividades que recaem sobre essa categoria.

Um curso de Ciência da Computação tem a computação como fim, diferentemente de um curso de Sistemas de Informação onde a computação é considerada como um meio de se tratar eficientemente a informação. Em Sistemas de Informação os problemas relacionados com o gerenciamento de informação se tornam cada vez mais complexos e os desafios consistem em se fazer uso adequado da informação e da tecnologia para se garantir a eficiência e a efetividade organizacional.

A seguir são apresentados dois gráficos que ilustram as diferenças entre os dois cursos considerando apenas os tópicos relacionados com a computação.

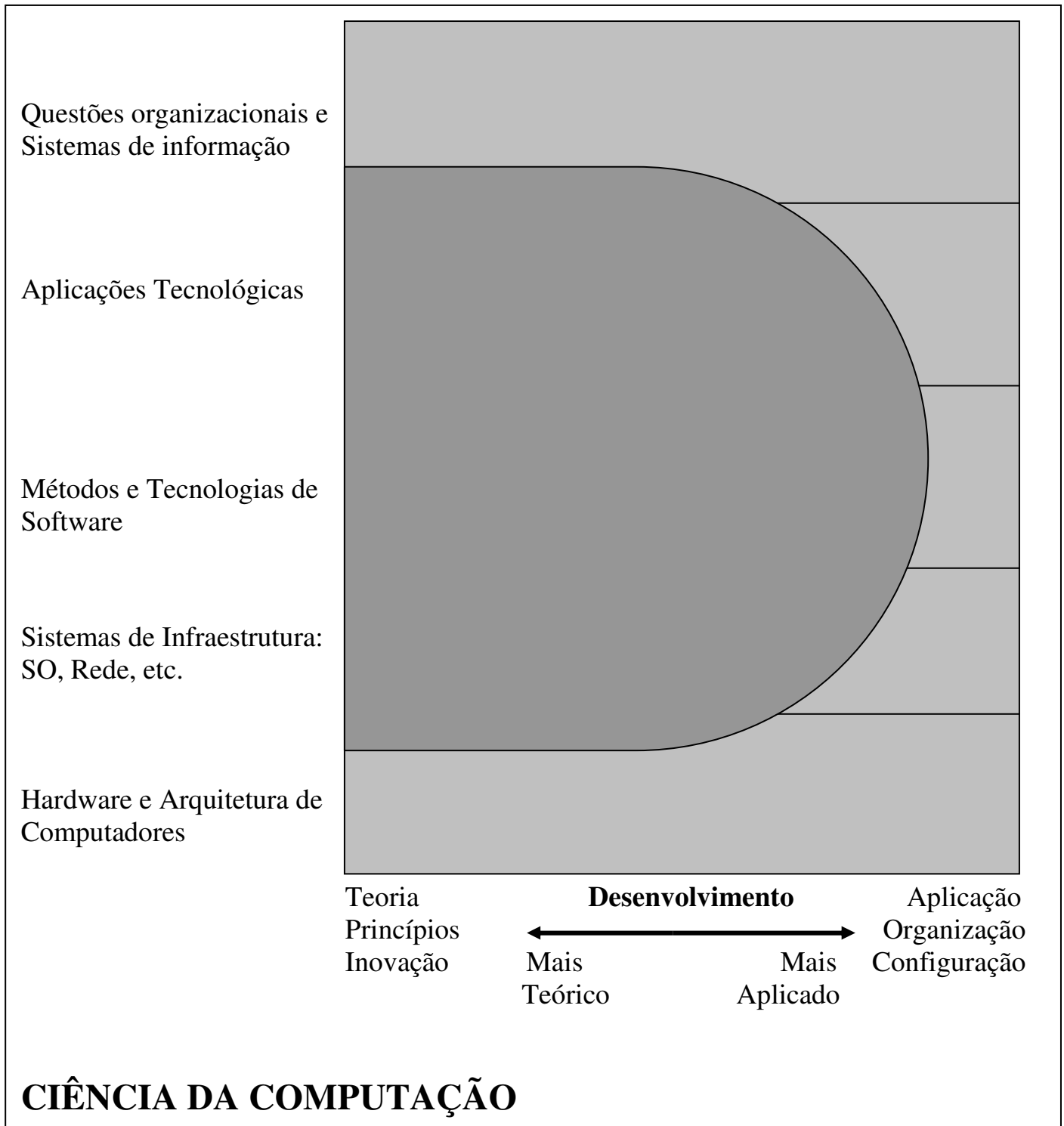


Figura 2.1 - Gráfico do Curso de Ciência da Computação  
Computing Curricula 2005- ACM/AIS/IEEE-CS

A porção escura do gráfico, apresentado na Figura 2.1, representa a abrangência do curso de Ciência da Computação que cobre a maioria do espaço horizontal central, porém, incluindo apenas uma pequena porção da faixa do topo que corresponde a assuntos



relacionados com questões organizacionais e de sistemas de informação e também uma pequena parte da faixa da base que corresponde ao hardware. A referida figura evidencia a competência do profissional de ciência da computação que é projetar e desenvolver todo tipo de software, desde sistemas de infraestruturas (sistemas operacionais, programas de comunicação etc) até aplicações tecnológicas (navegadores para a *web*, banco de dados, ambientes integrados de desenvolvimento etc).

O cientista da computação cria essas capacidades, porém, não gerencia a organização e a configuração das mesmas, conforme mostra a figura em questão através do estreitamento e interrupção da porção escura quando se move da esquerda para a direita. Isto porque não é função do cientista da computação auxiliar usuários na seleção de produtos computacionais, ajustar produtos às necessidades organizacionais ou mesmo aprender a utilizar produtos específicos.

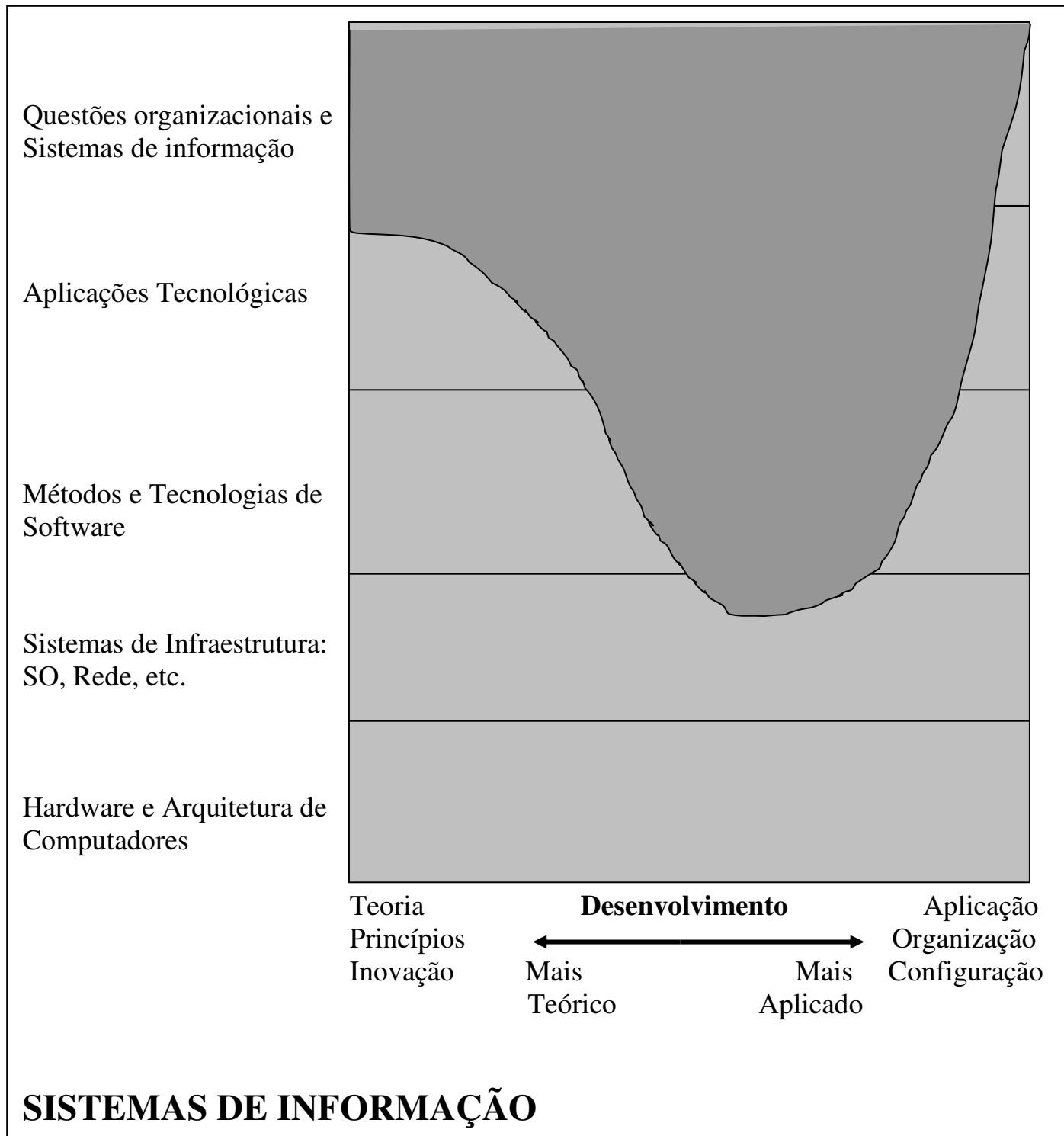


Figura 2.2 - Gráfico do Curso de Sistemas de Informação  
Computing Curricula 2005- ACM/AIS/IEEE-CS

Na Figura 2.2 a porção escura corresponde à abrangência do Curso de Sistemas de Informação considerando os mesmos tópicos relacionados na figura anterior. A área escura desta vez cobre praticamente de toda a faixa

do topo do gráfico uma vez que o profissional de Sistemas de Informação está preocupado com o relacionamento entre sistemas de informação e a sua organização, estendendo da teoria e princípios até aplicação e desenvolvimento. Muitos profissionais de sistemas de informação estão envolvidos com configuração, organização e gerência de sistemas, bem como treinamento de usuários.

Deve ser notado através da figura em questão que a área coberta pelo curso de Sistemas de Informação alcança as faixas correspondentes aos sistemas de infraestruturas, métodos e tecnologias de software. É importante salientar também que nessas faixas a área coberta é mais estreita e tende para a direita no eixo horizontal. Isto se justifica porque o profissional em sistemas de informação normalmente deve adaptar aplicações tecnológicas (por exemplo, banco de dados) às necessidades da organização e também desenvolver sistemas que utilizam outros produtos de software para ajustar às necessidades de informação de sua organização.

As providências tomadas para atingir o segundo objetivo da reforma que é o de melhor caracterizar o curso de Ciência da Computação, alterou significativamente o objetivo do curso, assim como, o perfil e as funções dos egressos. Conseqüentemente, a metodologia a ser empregada para que os objetivos sejam atingidos também o foi. A estrutura curricular sofreu alterações importantes: apenas quatro disciplinas da estrutura atual foram mantidas inalteradas; trinta e uma disciplinas foram atualizadas por meio de modificações de ementas, fusões e desmembramentos; quatorze novas disciplinas foram propostas.

É importante enfatizar então que o curso de Ciência da Computação da UFSC passará a apresentar um perfil fortalecido de um curso de Ciência da Computação conforme apregoado pelo Computing Curricula 2005- ACM/AIS/IEEE-CS. Ou seja, um curso que prima pela formação de um profissional de Ciência da Computação cuja principal competência seja a de projetar e desenvolver todo tipo de software, desde sistemas de infra-estrutura tais como sistemas operacionais, sistemas de comunicação etc, até aplicações tecnológicas como navegadores para a *web*, banco de dados, ambientes integrados de desenvolvimento etc.

O que também garante que o novo curso de Ciência da Computação da UFSC será de qualidade é o fato de que a construção da sua estrutura curricular foi inteiramente baseada nas Diretrizes Curriculares e que todas as matérias constantes do Currículo de Referência, proposto pela Sociedade Brasileira de Computação, foram contempladas da melhor forma possível.

### **3. Concepção Filosófica, Teórico-Metodológica do Curso e o Perfil do Profissional Desejado.**

Neste item são descritos o perfil desejado do egresso do curso incluindo o conjunto esperado de suas aptidões, as classes de problemas que os mesmos estarão aptos a resolver e também a capacidade de adaptação dos egressos à vigorosa evolução da computação e de suas tecnologias.

Também é abordada a metodologia do curso em função do perfil dos egressos e do seu papel na sociedade, onde se inclui a maneira como o curso forma profissionais capazes de atuar no desenvolvimento tecnológico da computação, bem como, de se tornarem agentes transformadores do mercado de trabalho, através da geração de novas tecnologias.

#### **3.1. Conjunto de Aptidões Esperadas dos Egressos.**

Promoção do desenvolvimento científico da Ciência da Computação por meio de pesquisas inovadoras.

Promoção do desenvolvimento tecnológico através da aplicação do embasamento teórico e dos conhecimentos científicos adquiridos.

Desenvolvimento de produtos em empresas de software e hardware.

Atuação como consultor e pesquisador em tecnologia.

Atuação na carreira acadêmica.

#### **3.2. Classes de Problemas que os Egressos Estarão Capacitados a Resolver.**

Problemas relacionados com a promoção e o desenvolvimento científico e tecnológico da computação, através de pesquisas inovadoras e de projetos e desenvolvimento de sistemas computacionais, em qualquer paradigma e metodologia, que possam envolver:

- Resolução de problemas relacionados à área de Análise Numérica;
- Desenvolvimento de programas em lógica;
- Utilização de probabilidade e estatística aplicada;
- Programação de componentes em baixo nível;
- Modelagem de sistemas utilizando técnicas de orientação a objetos, estruturas de dados e grafos;

Programação em linguagem funcional;  
Programação de componentes de sistemas operacionais;  
Construção de compiladores para linguagens dedicadas ou de uso geral;  
Análise de Sistemas utilizando técnicas modernas;  
Desenvolvimento de software para gerenciamento de bancos de dados;  
Construção de simuladores de propósito gerais;  
Aplicação de técnicas de inteligência artificial em problemas correlatos a computação e outras áreas;  
Sistemas multimídia;  
Sistemas em computação gráfica;  
Pesquisa e Desenvolvimento em segurança de sistemas;  
Identificação de problemas insolúveis ou NP-completos e proposta de soluções aproximadas (heurísticas);  
Coordenação do processo de Desenvolvimento de Software, utilizando metodologias modernas de engenharia de software;  
Projeto de software embarcado (embutido);  
Desenvolvimento de software para sistemas paralelos, concorrentes e distribuídos;  
Elaboração de projetos na área de ciência da computação;  
Especificação formal de sistemas;  
Identificação de questões éticas e legais na aplicação da informática no dia-a-dia;  
Desenvolvimento de sistemas críticos em relação à confiabilidade;  
Construção de um plano de negócios em informática;

### **3.3. Habilidades que os Alunos Poderão Desenvolver em Função do Conjunto de Disciplinas Optativas Cursadas.**

Projeto e implantação de redes locais de computadores;  
Desenvolvimento de soluções de gerenciamento para redes de computadores e de telecomunicações;  
Projetos de integração software/hardware;  
Projeto de protocolos de comunicação em redes;  
Projeto de microprocessadores simples;  
Construção de sistemas em realidade virtual;  
Construção de sistemas em tempo real;

### **3.4. Funções que os Egressos Poderão Exercer no Mercado de Trabalho.**

Pesquisador;  
Profissional da carreira acadêmica;  
Desenvolvedor nas diferentes áreas tecnológicas da computação (redes, bancos de dados, sistemas operacionais, realidade virtual, multimídia, sistemas distribuídos, etc.);  
Programador de sistemas de médio e grande porte;  
Engenheiro de software;  
Analista e projetista de software.

### **3.5. Capacidade de Adaptação do Egresso à Evolução da Computação e de suas Tecnologias.**

O curso de Ciências da Computação da UFSC prima pelo embasamento conceitual e teórico da área da computação, aliando as bases científicas da tecnologia à prática com a utilização dos principais produtos do mercado.

A visão do curso abrange um mercado de trabalho futuro, dessa forma, procura utilizar sempre tecnologias que recorrem as mais modernas técnicas computacionais, optando, por exemplo, pela perspectiva de programação e análise orientadas a objetos, em lugar da programação e análise estruturadas. O primeiro paradigma, por ser mais produtivo em aplicações referentes a sistemas de maior complexidade, vai proporcionar resultados de melhor qualidade, velocidade e confiabilidade do produto final. Considera-se assim que nossos alunos estarão, em pouco tempo, colaborando na transformação do mercado nacional, já que a grande maioria das empresas utiliza com técnicas mais antigas. Ao ingressar neste mercado, o aluno de Ciência da Computação da UFSC poderá produzir mais e melhor e, dentro do espírito da livre concorrência, acabará influenciando as empresas mais defasadas a atualizarem seus métodos, sob pena de perderem importantes fatias do mercado.

A ênfase em conceitos e não em tecnologias caracteriza-se, na área de programação, por exemplo, pelo fato de que o aluno tem contato com os diversos paradigmas de linguagens de programação ao longo do curso, sem contar com disciplinas específicas de programação. Nas duas primeiras fases, nas disciplinas de Programação Orientada a Objeto I e II, o aluno terá contato com linguagens orientadas a objetos. A ênfase do curso está nos conceitos trabalhados, que são os mesmos nas diferentes linguagens, e cuja sintaxe e forma de implementação pode variar. Na quarta fase, na disciplina de Paradigmas de Programação, o aluno tem contato com linguagens de outros paradigmas de programação, principalmente programação em lógica e funcional. Nessa disciplina, além de simplesmente ter contato com uma nova linguagem, de outro paradigma, o aluno é levado a conhecer os fundamentos de programação em lógica (sintática e semântica da LPO, notação clausal e os métodos de resolução genéricos, refutação por resolução, resolução linear, resolução para cláusulas definidas: LSD e LSDNF). Os alunos vêem a teoria e a prática (Prolog) em paralelo e ao final percebem que o Prolog é baseado em uma subclasse bem restrita destes métodos e pode ser estendido em linguagens mais poderosas (habilidade: Desenvolvimento de programas em lógica). Ainda nessa mesma disciplina o aluno terá contato com linguagens funcionais como Lisp e Haskell (habilidade: Desenvolvimento de programas em linguagem funcional). Além disso, em disciplinas de tópicos especiais, o aluno terá contato com outras tecnologias e linguagens de programação dedicadas.

Na terceira fase, a disciplina de estrutura de dados procura abrir as “caixas pretas” de algumas estruturas de dados que os alunos já aprenderam a utilizar na primeira e segunda fase. Nesta disciplina, linguagens que permitem uma manipulação mais direta dos elementos básicos do software, como C++, poderão ser utilizadas.

Nas áreas tecnológicas o curso também prima pela fundamentação teórica. Disciplinas das áreas de bancos de dados, redes, sistemas operacionais e compiladores apresentam aos alunos os principais conceitos fundamentais e resultados teóricos consolidados. Estes conceitos muito pouco tem mudado nos últimos vinte anos, tendo sido, na maioria dos casos, apenas acrescentados novos elementos tecnológicos. O aluno, de posse desse embasamento, tornará apto a trabalhar com qualquer ferramenta que o mercado venha a proporcionar, além de contribuir para a definição e criação de novas ferramentas e tecnologias do

futuro. Para tanto, o curso optou por cobrir a maioria das áreas de tecnologias através de duas disciplinas. Enquanto uma primeira disciplina aborda os aspectos fundamentais e as estruturas de uma determinada tecnológica, a outra trabalha essa mesma tecnologia com mais profundidade em pelo menos uma classe representante da mesma, envolvendo sempre que possível a realização de projetos relacionados com a mesma.

O projeto pedagógico do curso fornece realmente a direção do desenvolvimento de todos os conteúdos de forma que o profissional egresso seja capaz de atuar na sociedade, enfrentando os desafios específicos da área da computação que se deve transformar na sociedade do conhecimento, com o apoio tecnológico de forma consciente. Neste sentido e como já aludido anteriormente, a grande preocupação é a de formar profissionais conscientes, versáteis e criativos, os quais atenderão uma área de rápidas mudanças, como é a tecnológica. O curso está consciente da importância da definição e manutenção da metodologia pedagógica como um fator influente nos resultados a serem alcançados durante o seu ciclo de vida de, aproximadamente, quatro anos de duração. Dessa forma, o projeto pedagógico do Curso de Ciência da Computação da UFSC integra tanto uma visão da epistemologia (validade dos conteúdos abordados para a sociedade) através da perspectiva da complexidade (modelagem), quanto da gestão prática do conhecimento através do desenvolvimento de produtos (programação e desenvolvimento específicos). O caminho trilhado, da síntese (perspectiva global) para o analítico (perspectiva das partes, do específico) fica patente, na medida em que o aluno ingressante no curso entre em contato direto com a perspectiva de modelagem orientada a objetos, cuja visão sistêmica é do tipo complexa, superando a estruturada. Na gestão prática desta lida com esta perspectiva na organização dos conhecimentos, aparecem naturalmente as práticas de sala de aula estimulando trabalhos coletivos que podem levar à construção de modelos com graus de complexidade maiores. Os trabalhos em grupos também estimulam o aluno a desenvolver uma imagem pública, com discussões e apresentações de seminários, desde a sua primeira fase na universidade.



## **4. Organização da Estrutura Curricular.**

Os currículos dos cursos da área de computação e informática, segundo as diretrizes curriculares de cursos dessa área proposto pelo MEC/CEEInf, podem ser compostos por quatro grandes áreas de formação: formação básica, formação tecnológica, formação complementar e formação humanística.

A área de formação básica compreende os princípios básicos da área da computação, a ciência da computação, a matemática necessária para a defini-la formalmente, a física e eletricidade necessária para permitir o entendimento de projeto de computadores viáveis tecnicamente e a formação pedagógica que introduz os conhecimentos básicos da construção do conhecimento, necessário ao desenvolvimento da prática do ensino de computação. Essa área pode, então, ser subdividida em quatro sub áreas: área de formação básica em ciência da computação, em matemática, em física e eletricidade e em pedagogia. A sub área de ciência da computação, por sua vez, também pode ser subdividida em três outras sub áreas: programação, computação e algoritmos e arquitetura de computadores.

A área de formação tecnológica (também chamada de aplicada ou profissional) trata da aplicação dos conhecimentos básicos no desenvolvimento tecnológico da computação. Essa área pode ser subdividida em oito sub áreas: sistemas operacionais, redes de computadores e sistemas distribuídos; compiladores; bancos de dados; engenharia de software; sistemas multimídia, interface homem-máquina e realidade virtual; inteligência artificial; computação gráfica e processamento de imagens.

A área de formação complementar vem permitir uma maior interação dos egressos dos cursos de computação com outras profissões.

A área de formação humanística vem oferecer ao egresso uma dimensão social e humana.

Neste item serão identificadas todas as disciplinas que compõem a estrutura curricular, as suas distribuições nas referidas áreas de formação e a metodologia do plano pedagógico, ou seja, uma descrição de como as diversas matérias (programação, matemática, arquitetura de computadores etc) foram refinadas em disciplinas e encadeadas para formar o perfil do egresso desejado apontado no projeto pedagógico do curso.

#### **4.1. Disciplinas Obrigatórias.**

A seguir serão listadas todas as disciplinas obrigatórias que compõem a estrutura curricular com as suas respectivas identificações, ementas, cargas horárias, pré-requisitos e bibliografias.

Para a integralização curricular, o aluno deverá cursar os 172 créditos que corresponde ao total de disciplinas obrigatórias e 24 créditos de disciplinas optativas segundo critérios especificados no item 4.2.

##### **EELXX – Circuitos e Técnicas Digitais.**

**Carga Horária: 90 horas aulas.**

**Pré-requisito:**

**Ementa:**

Simulação e medidas elétricas de portas CMOS: tensões, correntes, atraso, minimização e mapeamento. Circuitos combinacionais (codificador, decodificador, somador, multiplicador, barrel shifter, unidade lógica e aritmética). Elementos de interconexão (multiplexador, barramento tri-state) e de armazenamento (latch, flip-flop, registrador, célula de memória). Relógio e temporização. Circuitos seqüenciais: registrador de deslocamento e contador. Memórias (DRAM, SRAM, ROM, FLASH). Arranjos lógicos (PAL, PLA, FPGA).

**Bibliografia:**

Livro-texto:

KATZ, Randy H., BORRIELLO, G., "Contemporary Logic Design", Prentice Hall, 2005, capítulos 1 a 5.

## **INE5401 – Introdução à Computação.**

**Carga Horária: 36 horas aulas.**

**Pré-requisito:**

**Ementa:**

Uso versus funcionamento interno de computadores. Organização de Computadores. Teoria da Computação. Linguagem de Computador. Sistemas Operacionais. Redes de Computadores. Sistemas Distribuídos. Engenharia de Software. Matemática na Computação. Tecnologias da informação: Bancos de Dados, Computação Gráfica, Segurança, Inteligência Artificial. Computação e Ética.

**Bibliografia:**

Castro, M. C. S., “Organização de Computadores I”, Universidade do Estado do Rio de Janeiro, 2005.

---

---

## **INE5402 – Programação Orientada a Objeto I.**

**Carga Horária: 108 horas aulas.**

**Pré-requisito:**

**Ementa:**

Algoritmo. Programação de computador. Resolução de problemas usando computador. Linguagem de Programação. Elementos de programação imperativa: variáveis simples, coleções uni e n dimensionais; tipos das variáveis; comandos; procedimentos e funções. Conceitos fundamentais do paradigma de Programação Orientada a Objetos: objeto, atributo, método, classe. Prática de programação usando alguma linguagem de programação orientada a objetos.

**Bibliografia:**

BORATTI, I. C., “Programação Orientada a Objetos Usando Delphi”, Visual Books, 2001.

BORATTI, I. C. e OLIVEIRA, A. B., “Introdução a Programação-Algoritmos”, Visual Books, 1999.

BOOCH, G., “Object Oriented Design”, Benjamin/Cummings Pub. 1994. MEYER, B., “Object-Oriented Software Construction”, Prentice Hall, 1988. TAKAHASHI, T. et alii., “Programação Orientada a Objetos”, VII Escola de Computação, 2000. RUMBAUCH, J. et alii., “Modelagem e Projetos Baseados em Objetos”, Ed. Campus, 1994.

---

---

COAD, P. e YOURDON, E., “Análise Baseada em Objetos”, Ed. Campus. 1991.

COAD, P. e YOURDON, E., “Projeto Baseado em Objetos”, Ed. Campus. 1993.

WEBER, C. E. et ali, “Developing with Delphi - Object Oriented Techniques”, Prentice Hall. 1996.

---

### **INE5403 – Fundamentos de Matemática Discreta para a Computação.**

**Carga Horária: 108 horas aulas.**

**Pré-requisito:**

**Ementa:**

Lógica Proposicional, Lógica de Primeira Ordem, Lógica Matemática (prova de teoremas), Teoria dos Conjuntos, Indução e Recursão. Divisão nos Inteiros. Seqüências e somas. Análise Combinatória. Probabilidade Discreta. Relações: Propriedades das Relações, Relações de Equivalência, Relações de Ordem, Reticulados, Álgebra Booleana. Funções Discretas: Funções Totais e Parciais, Composição de Funções. Noções de modelos Matemáticos de Computadores e de Computabilidade. Estruturas Algébricas: semigrupos e grupos, grupos e codificação.

**Bibliografia:**

TREMBLAY, J. P. & MANOHAR, R. "Discrete Mathematical Structures with Applications to Computer Science", McGraw-Hill Computer Science Series, 1975, ISBN: 0-07-065142-6;

KOLMAN, BERNARD & BUSBY, ROBERT C., "Discrete Mathematical Structures for Computer Science", Prentice-Hall International Editions, 1987, ISBN: 0-13-215922-8;

MENEZES, PAULO BLAUTH. "Matemática Discreta para Computação e Informática", Serie Livros Didáticos - UFRGS n.16, Editora Sagra-Luzzatto, 2004, ISBN: 8524106913;

GERSTING, Judith L., "Fundamentos Matemáticos para a Ciência da Computação, 4a edição, LTC, 2001, ISBN: 85-2161422-5.

ROSEN, Kenneth H., "Discrete Mathematics and Its Applications", 5th edition, McGraw-Hill, 2003, ISBN: 0-07-242434-6.

---

---

**MTM5161 – Cálculo A.**

**Carga Horária: 72 horas aulas.**

**Pré-requisito:**

**Ementa:**

Funções reais de variável real; Funções elementares do cálculo; Noções sobre limite e continuidade; A derivada; Aplicações da derivada; Integral definida e indefinida.

**Bibliografia:**

FLEMMING, D. M., GONÇALVES, M. B., “Cálculo A”, Mkrn Books, 1992.

THOMAS, G. B., FINNEY, R. L., “Calculo Diferencial e Integral”, v.1, Addison Wesley, 2002.

KUELKAMP, N., “Cálculo I”, UFSC, 1999.

SIMMONS, G.F, “Cálculo com Geometria Analítica”, v. 1, Mc Graw-Hill, 1988.

STEWART, J., “Cálculo”, v.1, Pioneira Tomson Learning, 2002.

HOWARD, A., “Cálculo: Um Novo Horizonte”, v.1, Bookman, 1999.

LEITHOLD, L., “O Cálculo com Geometria Analítica”, v.1, Harbra, 1977.

IEZZI, G. e outros, “Fundamentos de Matemática Elementar”, v.1 e v.8, Atual Editora, 1985.

GUIDORIZZI, H.L., “Um Curso de Cálculo”, v.1, LTC, 1985.

---

---

**MTM5512 – Geometria Analítica.**

**Carga Horária: 72 horas aulas.**

**Pré-requisito:**

**Ementa:**

Matrizes. Determinantes. Sistemas lineares. Álgebra vetorial. Estudo da reta e do plano. Curvas planas. Superfícies.

**Bibliografia:**

STEINBRUCH, Alfredo e Winterle, Paulo - Geometria Analítica.

STEINBRUCH, Alfredo e Winterle, Paulo - Álgebra Linear.

BOULOS, Paulo - Geometria Analítica.

LEITE, Olímpio R. - Geometria Analítica Espacial.

KINDLE, Joseph H. - Geometria Analítica - Coleção Schaum.

FEITOSA - Cálculo Vetorial e Geometria Analítica.

BLASI, Francisco Lições de Geometria Analítica.

KOLMAN, Bernard - Álgebra Linear.

FRANK Ayres Júnior - Matrizes e vetores.

ROBERTO de Barros Lima Elementos de Álgebra Vetorial.

SYMOUR Lipschutz - Álgebra Linear.

BOLDRINI - Álgebra Linear.

---

---

## **INE5404 – Programação Orientada a Objeto II.**

**Carga Horária: 108 horas aulas.**

**Pré-requisito: INE5402.**

### **Ementa:**

Herança. Polimorfismo. Reusabilidade de software. Componentes. Criação e uso de bibliotecas de classes. Padrões de projeto. Interface Gráfica com o Usuário. Exceções. Relacionamentos entre classes: agregação, composição e especialização. Persistência de dados e de objetos. Prática de programação com alguma linguagem de programação orientada a objetos.

### **Bibliografia:**

Grady Booch, “Object-Oriented Analysis and Design”.Addison-Wesley, 2nd Edition, 1994.

Martin Fowler and Kendall Scott, “UML Essencial - 2a. Edição. Bookman, 2000.

“The Java Tutorial: Object-Oriented Programming for the Internet”, Sun Microsystems. [www.javasoft.com](http://www.javasoft.com)

John Lewis, William Loftus, “Java Software Solutions - Foundations of Program Design”. Addison-Wesley, 1999.

Paul J. Deitel & Harvey M. Deitel. “Java - Como Programar - 4a. Edição”, Bookman, 2001.

Bruce Eckel. “Thinking in Java”. Prentice-Hall, 2000.  
[www.bruceeckel.com](http://www.bruceeckel.com)

---

---

### **INE5405 – Probabilidade e Estatística.**

**Carga Horária: 90 horas aulas.**

**Pré-requisito: MAT5161.**

#### **Ementa:**

Análise combinatória. Planejamento de uma pesquisa. Análise exploratória de dados. Probabilidade. Variáveis aleatórias discretas e contínuas. Principais modelos teóricos. Estimação de parâmetros. Testes de hipóteses.

#### **Bibliografia:**

BARBETTA, P. A.; REIS, M. M., BORNIA, A. C. “Estatística para Cursos de Engenharia e Informática”. São Paulo: Editora Atlas, 2004;

BUSSAB, W. O., MORETTIN, P. A. – “Estatística básica”. 5 ed. São Paulo: Editora Saraiva, 2002. MAGALHÃES, A. N., LIMA, A. C. P. – Noções de probabilidade e estatística. 6 ed. São Paulo: EDUSP, 2005;

MONTGOMERY, D.C., RUNGER, G. C. – “Estatística aplicada e probabilidade para engenheiros”. Rio de Janeiro: LTC, 2003;

TRIOLA, M. F. “Introdução à Estatística”. 9. ed. Rio de Janeiro: LTC, 2005.

---

---

### **INE5406 – Sistemas Digitais.**

**Carga Horária: 90 horas aulas.**

**Pré-requisito: EELXX.**

#### **Ementa:**

Máquinas seqüenciais síncronas (Mealy e Moore) e sua representação (diagramas de transição e descrição em HDL). Síntese de circuitos seqüenciais (minimização e codificação de

estados). Mapeamento e alternativas de implementação de máquinas de estado ("*hardwired*", PLA, ROM e PLD). Estudos de casos: controladores de memória, de interrupção, de DMA. Simulação de sistemas digitais descritos em HDL no nível de transferência entre registradores. CPU vista como um sistema digital (*datapath* e unidade de controle). Unidade de controle de uma CPU simples ("*hardwired*" e microprogramada).

**Bibliografia:**

KATZ, Randy H., BORRIELLO, G., "Contemporary Logic Design", Prentice Hall, 2005, capítulos 6 a 10.

---

---

**MTM03 – Cálculo B para Computação.**

**Carga Horária: 72 horas aulas.**

**Pré-requisito: MTM5161.**

**Ementa:**

Aplicações da integral definida. Integrais impróprias. Funções de várias variáveis. Derivadas parciais. Séries de números reais. Séries de funções. Avaliação de funções: série de Taylor e Maclaurin.

**Bibliografia:**

STEWART, J., "Calculus", Brooks/Cole Publishing Company, 2000.

FLEMMINH, D. M., GONÇALVES, M. B., "Cálculo A", Makron Books, 1994.

GONÇALVES, M. B., FLEMMING, D. M., "Cálculo B", Makron Books, 1999.

AYRES, F.J., "Cálculo Diferencial e Integral", Makron Books, 2000.

SWOKOWSKI, W.G., "Cálculo com Geometria Analítica", v.1, v.2, Makron Books, 1994.

McCALLUM, W. G. e outros, "Cálculo de Várias Variáveis", Edgard Blucher Ltda, 1997.

NUNEM, M. A., FOULIS, D. J., "Calculo", v.1, v.2, Guanabara Dois S. A., 2000.

THOMAS, G.B., "Cálculo", v. 2, Addison Wesley, 2000.

LEITHOLD, L., "O Cálculo com Geometria Analítica", v. 1, v. 2, Harbra Ltda, 1986.

GUIDORIZZI, H. L., "Um Curso de Cálculo", v. 2, v. 3, L.T.C., 2000.



---

---

**INE5407 – Ciência, Tecnologia e Sociedade.**

**Carga Horária: 54 horas aulas.**

**Pré-requisito:**

**Ementa:**

Estudo das relações entre ciência, tecnologia e sociedade ao longo da história, com ênfase na atualidade; filosofia da ciência; análise de valores e ideologias envolvendo a produção e divulgação da ciência e da tecnologia; influências das diferenças culturais nas concepções de ciência e tecnologia e de suas relações com as sociedades; a participação da sociedade na definição de políticas relativas às questões científicas, tecnológicas, econômicas e ecológicas. O impacto da informática na sociedade.

**Bibliografia:**

ALVES, R., “Tecnologia e humanização”, In: Revista Paz e Terra, n. 18, Ed. Civilização Brasileira, s/d).

BRAVERMAN, H., “Trabalho e Capital Monopolista A Degradação do Trabalho no século XX”, Rio de Janeiro, Guanabara, 1981.

CORIAT, B., “Ciência Técnica e Capital”, H. Blume Edições, Madrid, 1976, p.51-52.

DAGNINO, R., “Sobre a neutralidade da tecnologia e da ciência: um guia de leitura”, 1977.

DICKSON, D., “Tecnología alternativa y políticas del cambio tecnológico”, Blume Ediciones, p. 11-15. 1980.

ELLIOTT, D. R., “El control popular de la tecnología, Colección Tecnología y Sociedad”, Editorial Nueva Sociedad.

FEENBERG, A., “ Critical Theory of Technology”, Oxford University Press, 1991.

GORZ, A., “Divisão do trabalho, hierarquia e luta de classes. In: Divisão social do trabalho, ciência, técnica e modo de produção capitalista”, Publicações Escorpião, Porto, 1974.

HESSON, B., “Las Raíces Socioeconómicas de la Mecánica de Newton”, Havana, Academia, 1985.

PINCH, T., BIJKER, W. E., “ The social construction of facts and artifacts: or how the sociology of Science and the Sociology of Technology might benefit each other. In Bijker et al, The Social construction of Technological systems”, Cambridge, MIT Press, 1990.

LANDER, E., “La ciencia y la tecnología como asuntos políticos: limites de la democracia en la sociedad tecnológica”, Venezuela: Editorial Nueva Sociedad, 1994.

MARGLIN, S.A., “Origem e funções do parcelamento das tarefas: para que servem os padrões”, 1971.

SALVATI, M., BECCALLI, B., “A divisão do trabalho - capitalismo, socialismo, utopia. In: a divisão capitalista do trabalho”, Córdoba, Argentina: Cuadernos Pasado e Presente, n. 32, 1972.

---

---

### **INE5408 – Estruturas de Dados.**

**Carga Horária: 108 horas aulas.**

**Pré-requisito: INE5404.**

**Ementa:**

Alocação dinâmica de memória. Variáveis estáticas e dinâmicas. Estruturas lineares. Tabelas de Espalhamento. Árvores. Árvores de Pesquisa. Métodos de ordenação. Métodos de acesso a arquivos. Técnicas de implementações iterativas e recursivas de estruturas de dados. Complexidade dos algoritmos em estruturas de dados.

**Bibliografia:**

BOOCH, G., RUMBAUGH, J. & JACOBSON, I., “UML - Guia do Usuário”, Ed. Campus, 2000.

COAD, P. e YOURDON, E. “Análise Baseada em Objetos”, Ed. Campus, 1991.

COAD, P. e YOURDON, E. “Projeto Baseado em Objetos”, Ed. Campus, 1993.

HOROWITZ, E., “Fundamentos de Estruturas de Dados”, Ed. Campus, 1986.

PREISS, B. R., “Data Structure and Algorithms With Object-Oriented Design Patterns”, Editora John Wiley, 1999.

VILLAS, M. V., “Estruturas de Dados: conceitos e técnicas de implementação”, Ed. Campus, 1993.

Links WWW relacionados às Estruturas de dados.

---

### **MTM5245 – Álgebra Linear.**

**Carga Horária: 72 horas aulas.**

**Pré-requisito: MTM5512.**

#### **Ementa:**

Espaço vetorial. Transformações lineares. Mudança de base. Produto interno. Transformações ortogonais. Autovalores e autovetores de um operador. Diagonalização. Aplicação da Álgebra linear às ciências.

#### **Bibliografia:**

LALLIOLI, Domingues Costa - Álgebra Linear e Aplicações.

BOLDRINI et alii - Álgebra Linear - Ed. Harper e Row do Brasil Ltda.

CARVALHO, João Pitombeira - Introdução à Álgebra Linear - Ed. Und.

CALLOLI, et alii - Álgebra Linear e Aplicações - Atual Editora.

HOWARD, Anton - Algebra Linear - Ed. Campus Ltda - RJ 1982.

LIPSCHUTZ - Álgebra Linear - Coleção Schaum - Ed. Mac-Graw-Hill.

STEINBRUCK, Alfredo - Álgebra Linear e Geometria Analítica - Ed. Mac-Graw-Hill.

HOFFMANN & KUNZE - Álgebra Linear.

STRANG, Gilbert - Linear Álgebra and its applications.

---

---

**INE5409 – Cálculo Numérico para Computação.**

**Carga Horária: 72 horas aulas.**

**Pré-requisito: MTM5512 e MTM03.**

**Ementa:**

Máquinas digitais: precisão, exatidão e erros. Aritmética de ponto flutuante. Sistemas de enumeração. Sistemas lineares. Resolução computacional de sistemas de equações lineares. Resolução de equações transcendentais. Aproximação de funções: interpolação *spline*, ajustamento de curvas, aproximação racional e por polinômios de Chebyshev. Integração numérica: Newton-Cotes e quadratura Gaussiana.

**Bibliografia:**

RUGGIERO, M. e LOPES, V., “Cálculo Numérico: Aspectos Teóricos e Computacionais”, McGraw-Hill, 1996.

CLÁUDIO, D. M. e MARINS, J. M., “Cálculo Numérico Computacional”, Teoria e Prática. São Paulo, Atlas, 1989.

CHENEY, W. and KINCAID, D., “Numerical Mathematics and Computing”, Brooks/Cole Publishing Company, 1994.

FAIRES, J.D. and BURDEN, R. L., “Numerical Methods”, PWS Publishing Company, 1993.

GERALD, C.F. and WEATLEY, P.O., “Applied Numerical Analysis”, 5th ed. New York: Addison Wesley, 1994.

RALSTON, A., “A First Course in Numerical Analysis”, McGraw-Hill, 1987.

CONTE, S. D., “Elementos de Análise Numérica”, São Paulo, Globo: 1977.

McCRACKEN, D. e DORN, W., “Cálculo Numérico com Estudos de Casos em FORTRAN IV”, Rio de Janeiro : Campus, 1978.

CHAPRA, S. and CANALE, R., “Numerical methods for Engineers: with personal computer applications”, McGraw-Hill, 1985.

PRESS, W.H., et al., “Numerical Recipes in C - The Art of Scientific Computing”, Cambridge Press, 2nd ed., 1992.

BARROSO, L. C. et al., “Cálculo Numérico (Com Aplicações) 2ª.ed.”, São Paulo, Harbra, 1987.

STEINBRUCH, A., “Matrizes, Determinantes e Sistemas de Equações Lineares”, McGraw Hill. São Paulo, 1989.

---

---

**INE5410 – Programação Concorrente.**

**Carga Horária: 72 horas aulas.**

**Pré-requisito: INE5404.**

**Ementa:**

Multiprogramação. Multitarefa. Execução concorrente. Recursos compartilhados e exclusão mútua. Regiões críticas. Coordenação de processos e *threads*. Semáforos. Monitores. Troca de mensagem. Programação concorrente orientada a objeto. *Deadlock*. Modelos de computação concorrente.

**Bibliografia:**

LEA, D., “Concurrent Programming in Java, Second Edition, Design Principles and Patterns”, Addison Wesley Longman, Inc, 2000.

HYDE, P., “Java Thread Programming”, Sams Publishing, 2000.

REILLY, D., REILLY, M., “Java Network Programming and Distributed Computing”, Addison-Wesley, 2002.

DAVIES, G. L., “Pascal FC Language Reference Manual – version 5”, University of Bradford, UK, 1992.

---

---

**INE5411 – Organização de Computadores.**

**Carga Horária: 108 horas aulas.**

**Pré-requisito: EELXX e INE5406.**

**Ementa:**

Tendências tecnológicas na fabricação de CPUs e memórias. CPU: instruções e modos de endereçamento. Formatos de instruções e linguagem de montagem. Simulador e montador. Aritmética. Avaliação de desempenho. *Datapath* e unidade de controle. Alternativas de implementação (monociclo, multiciclo, *pipeline*, superescalar). Exceções e interrupções. *Hazards* estruturais, de dados e de controle. Hierarquia de memória e associatividade (*cache* e TLB). Dispositivos de entrada e saída: tipos, características e sua conexão à CPU e à memória. Comunicação com a CPU (*polling*, interrupção, DMA).

**Bibliografia:**

Livro-texto:

PATTERSON, D., HENNESSY, J., "Computer Organization and Design: The Hardware Software Interface", 3rd edition, Morgan Kaufmann-Elsevier, 2005.

---

**INE5412 – Sistemas Operacionais I.**

**Carga Horária: 72 horas aulas.**

**Pré-requisito: INE5410 e INE5411.**

**Ementa:**

Programação de Sistemas. Histórico de Sistemas Operacionais. Arquitetura de Sistemas Operacionais. Funcionalidades de Sistemas Operacionais. Gerência de Processos e *Threads*: Controle e Escalonamento. Impasses: Modelagem e Tratamento. Memória: Alocação, Gerência e Memória Virtual. Entrada e Saída: princípios de hardware e software, dispositivos periféricos. Sistema de Arquivos: Arquivos, Diretórios e Implementação. Proteção e Segurança. Sistemas com Múltiplos Processadores.

**Bibliografia:**

Tanenbaum, A. S., "Sistemas Operacionais Modernos", 2ª. Edição, Prentice-Hall do Brasil, 2003.

Silberschatz, A., Galvin, P. B., "Sistemas Operacionais – 5a. Edição", Addison-Wesley, 2000.

Oliveira, R. S., Toscani, S. S., Carissimi, A. da S., "Sistemas Operacionais", Sagra Luzzatto, 2000.

Tanenbaum A. S. and Woodhull A. S., "Operating Systems Design and Implementation - Third Edition", Prentice Hall, 2006.

Silberschatz, A., Galvin, P. B., "Operating Systems Concepts – Fifth Edition", Addison-Wesley, 1998.

Stallings, W., "Operating Systems: Internals and Design Principles – Third Edition", Prentice Hall, 1998.

Bach, M. J., "The Design of the Unix Operating System", Prentice-Hall, 1990.

Vahalia, Uresh, "UNIX Internals the New Frontiers", Prentice-Hall, 1996.

---

---

**INE5413 – Grafos.****Carga Horária: 72 horas aulas.****Pré-requisito: INE5403 e INE5408.****Ementa:**

Grafos e grafos orientados. Representação de problemas com grafos. Caminhos, ciclos e caminho de custo mínimo. Conexidade e alcançabilidade. Árvores e árvore de custo mínimo. Coloração e planaridade de grafos. Grafos hamiltonianos e eulerianos. Fluxo máximo em redes. Estabilidade e emparelhamento em grafos. Problemas de cobertura e de travessia. Representações computacionais e complexidade de algoritmos em grafos.

**Bibliografia:**

- GERSTING, Judith L., “Fundamentos Matemáticos para a Ciência da Computação”. LTC - Livros Técnicos e Científicos, 1982.
- CAMPELLO, Ruy Educarado e MACULAN, Nelson, “Algoritmos e Heurísticas”, Universidade Federal Fluminense, 1994.
- CHARTRAND, Gary, “Graphs as Mathematical Models”, Prindle, Weber & Schmidt. Boston, 1977.
- NETTO, Paulo O.B., “Teoria e Modelos de Grafos”, Edgard blucher, São Paulo, 1979.
- FURTADO, A.L., “Teoria dos Grafos – Algoritmos”, PUC/RJ-LTC, 1973.
- SZWARCFILER, J. L., “Grafos e Algoritmos Computacionais”, Campus, 1984.
- WILSON, R.J., “Introduction to Graph Theory”, 1979.
- HARAY, F., “Graph Theory”, Addison-Wesley, 1969.
- CRISTOFIDES, N., “Graph Theory - an Algorithmic Approach”, Academic Press, 1975.
- 
- 

**INE5414 – Redes de Computadores I.****Carga Horária: 72 horas aulas.****Pré-requisito: INE5404.****Ementa:**

Capacidade de canal. Taxa de transmissão. Codificação analógica-digital, digital-digital e digital-analógica. Princípios dos modelos

das redes de computadores: OSI e TCP/IP. Meios de transmissão de dados. Serviços e tarefas ofertados na camada de enlace. Estudo de casos de protocolos e tecnologias de enlace de dados. Redes de comutação de circuito e de pacotes.

### **Bibliografia:**

Apostila de Redes de Computadores.

Anais dos Simpósios Brasileiros de Redes de Computadores.

Revistas especializadas em Comunicação de Dados e Redes de Computadores.

Sites com informações em Comunicações de Dados e Redes de Computadores.

JAKOBS, K.; PROCTER, R.; WILLIAMS, R. The Making of Standards: Looking Inside the Work Groups. IEEE Communications Magazine. Vol. 39, No. 4. Pg. 102-107, April. 2001.

HELD, G. Comunicação de Dados. Editora Campus. 1999.

MOURA, J. A. B. et al. Redes locais de computadores - Protocolo de alto nível e avaliação de desempenho. São Paulo. McGraw-Hill, 1986.

TANENBAUM, A.S. Redes de Computadores. Quarta edição. Editora Campus, 2003.

COMER, Douglas E. Internetworking with TCP/IP. Volume I, Principles, Protocols, and Architecture. Third Edition - Printice Hall, Inc. 1995.

COMER, D. E. Interligação em Redes com TCP/IP. Volume I. Princípios, Protocolos e Arquitetura. Segunda Edição. Editora Campus. 1998.

COMER, D. E.; STEVENS, D. L. Interligação em Redes com TCP/IP. Volume II. Projeto, implementação e detalhes internos. Editora Campus. 1999.

DERFLER, F.Jr. Guia de Conectividade, Editora Campus. Rio de Janeiro, 1993.

SOARES, L.F.G. et al. Redes de Computadores: das LANs, MANs e WANs às Redes ATM. Editora Campus, Rio de Janeiro, 1995.

ALVES, L. Protocolos para Redes de Comunicação de Dados. São Paulo, Atlas.1987.

CHURCHILL,B.,JORDAN,L.,“Comunicações e Redes com o PC , Axcel Books, 1994.

COMER, D. E., “Computer Networks and Internets”, Prentice Hall. NJ, USA, 1999.

COMER, D. E., Redes de Computadores e Internet. Bookman. Porto Alegre, 2001.



PERKIS, C. H. Mobile IP Design Principles and Practices. Addison Wesley Longman, 1998.

STALLINGS, W. SNMP, SNMPv2, SNMPv3, and RMON 1 and 2. Third Edition. Addison-Wesley. 1999.

AIDAROUS, S. & PLEVYAK, T., “Telecommunications Network Management Technologies and Implementations”, IEEE Press. 1988.

DANTAS, M., “Tecnologias de Redes de Comunicação e Computadores”, Axcel Books ISBN :85-7323-169-6. 2002.

KUROSE, J. F.; ROSS, K. W., “Redes de Computadores e a Internet (Uma Nova Abordagem)”, Addison Wesley. 2003.

PETERSON, L. L., DAVIE, B. S. , “Redes de Computadores (Uma Abordagem Sistêmica)”, Segunda Edição. Morgan Kaufmann Publishers. 2004.

Walrand, Jean, “Communication Networks, A First Course” - 2ª Edição, MCGraw-Hill. 1998,ISBN 0-256-17404-0.

Tanenbaum, A. S., “Computer Networks”, 3ª Edição, Prentice-Hall, ISBN 0-13-394248-1.

Soares, Lemos e Colcher, “Redes de Computadores, Das LANs MANs e WANs às Redes ATM” - 2ª Edição, Editora Campus, ISBN 85-7001-998-X.

Fred Halsall, “Data Communication, Computer Networks and Open Systems” - Editora Addison Welsey, 1995.

---

---

### **INE5415 – Teoria da Computação.**

**Carga Horária: 72 horas aulas.**

**Pré-requisito: INE5403 e INE5408.**

#### **Ementa:**

Programas, Máquinas e Computações. Máquinas de Turing. Funções Recursivas. Computabilidade. Decidibilidade. Análise e Complexidade de Algoritmos. Classes e complexidade de problemas computacionais.

#### **Bibliografia:**

HOPCROFT, E. J., ULMAN, D., MOTWANI, R., “Introdução à Teoria de Autômatos, Linguagens e Computação”, Editora Campus, 2002 (tradução da segunda edição americana).

HOPCROFT, J. E., ULLMAM, J. D., “Formal Languages and Their Relations to Automata”, Addison-Wesley, 1969.

- HOPCROFT, J. F., ULLMAN, J. D., “Introduction to Automata Theory, Languages and Computation”, Ed. Addison-Wesley, 1979.
- SUDKAMP, T. A., “Languages and Machines – An Introduction to the Theory of Computer Science, 2. edição”, Ed. Addison Wesley, 1997.
- MENESES, P. B., “Linguagens Formais e Autômatos”, Ed. Sagra Luzzato, 2. edição, 1998.
- LEWIS, H. R. e PAPADIMITRIOU, C. H. , “Elementos de Teoria da Computação”, Ed. Bookmam, 2. edição, 1998.
- AHO, A. V., ULLMAN, J. D., “The Theory of Parsing, Translation, and Compiling, Volume I: Parsing”, Ed Prentice-Hall, Inc. 1972, 542p.
- DIVERIO, T. A., MENEZES, P. B., “Teoria da Computação – Máquinas Universais e Computabilidade”, Ed. Sagra Luzzatto, Porto Alegre, 1999.
- 
- 

### **INE5416 – Paradigmas de Programação.**

**Carga Horária: 90 horas aulas.**

**Pré-requisito: INE5408.**

#### **Ementa:**

Caracterização e classificações dos paradigmas. Problemas tratáveis pelos paradigmas. Definição e caracterização dos principais paradigmas declarativos e imperativos. Programação em Lógica. Programação Funcional. Prática de programação com os principais paradigmas apresentados.

#### **Bibliografia:**

Revesz, G., “Lambda-Calculus Combinators and Functional Programming”, Cambridge University Press, 1988.

Bird, R. & Wadler, P., “Introduction to Functional Programming”, Prentice-Hall, 1988.

Hindley, J. R. & Seldin, J. P., “Introduction to Combinators and Calculus”, Cambridge University Press, 1986.

Steele Jr., Guy L., “Common Lisp – The Language”, disponível tanto sob forma de livro quanto em formato PostScript.

REVESZ, G., “Lambda-Calculus Combinators and Functional Programming”, Cambridge University Press, 1988.

BIRD, R. & WADLER, P., “Introduction to Functional Programming”, Prentice-Hall, 1988.

HINDLEY, J. R., SELDIN, J. P., “Introduction to Combinators and Calculus”, Cambridge University Press, 1986.

STEELE, J, GUY, L., “Common Lisp – The Language”, disponível tanto sob forma de livro quanto em formato PostScript.

NOLT, J., ROHATYN, D., “Lógica”, McGraw-Hill, São Paulo, 1991.

COPI, I. M., “Introdução à Lógica”, Editora Mestre Jou, 1981.

BELL, J. L., MACHOVER, M., “A Course in Mathematical Logic”, North-Holland, 1977.

ENDERTON, H.B., “A Mathematical Introduction to Logic”, Academic Press, 1972.

SMULLYAN, R.M., “First-Order Logic”, Dover, 1995.

LOVELAND, D. W., “Automated Theorem Proving: A Logical Basis”, Elsevier Science.

CHANG, Chin-Liang & LEE, R. C., “Symbolic Logic and Mechanical Theorem Proving”, Academic Pr.

BUCHSBAUN, A., PEQUENO, T., “O Método dos Tableaux Generalizado e sua Aplicação ao Raciocínio Automático em Lógicas Não Clássicas - O que nos faz pensar” - Cadernos do Departamento de Filosofia da PUC-Rio - nº 3 - setembro de 1990 - pp. 81-96.

---

---

## **INE5417 – Engenharia de Software I.**

**Carga Horária: 90 horas aulas.**

**Pré-requisito: INE5408.**

### **Ementa:**

Análise de requisitos: requisitos funcionais e requisitos não-funcionais; técnicas para levantamento e representação de requisitos, incluindo casos de uso. Modelagem OO: classe, atributo, associação, agregação e herança. Projeto OO: técnicas para projeto; padrões de projeto, componentes e *frameworks*; projeto de arquitetura; mapeamento objeto-relacional. Linguagem

de especificação orientada a objetos. Métodos de análise e projeto orientados a objetos. Desenvolvimento de um software OO.

### **Bibliografia:**

BOOCH, G., “Object Solutions - managing the object-oriented project”. Addison-Wesley: Menlo Park, 1996.

BOOCH, G., “Object-Oriented Analysis and Design with applications”. Ed. Benjamin/Cummings, 1994.

EMBLEY, D.W., et. Al., “Object-Oriented Systems Analysis - A model-driven approach”. Yourdon Press: Englewood Cliffs, 1992.

ERICKSON, H. E., PENKER, M., “UML Toolkit”, New York: John Wiley and Sons Inc, 1998.

FOWLER, M., SCOTT, K., “UML Essencial - Um breve guia para a linguagem padrão de modelagem de objetos”, Porto Alegre: Bookman, 2000.

JACOBSON, I., “The Object Advantage - Business process reengineering with object technology”, Addison-Wesley: Wokingham, 1994.

JACOBSON, I., “Object-Oriented Software Engineering - A Use Case Driven Approach”, Ed. Addison - Wesley, 1992.

RUMBAUGH, J., et. Al., “The Unified Modeling Language Reference Manual”. Object Technology Series, Addison Wesley Longman, Inc., December 1998.

LARMAN, C., “Applying UML and patterns: an introduction to object-oriented analysis and design”. Prentice Hall, 1997 (segunda edição revisada 2002).

LEE, M., et. Al. “Software Quality and Productivity - Theory, practice, education and training”. IFIP - Chapman & Hall: London, 1995.

MEYER, B., “Object-Oriented Software Construction”. Ed. Prentice Hall, 1988.

OMG. OMG, “Unified Modeling Language: Superstructure”, V.2.0. Aug. 2005.

PRESSMAN, R., “Engenharia de Software”, McGraw-Hill, 5 Ed. 2002.

RUMBAUGH, J., et al., “Object-Oriented modeling and Design”, Ed. Prentice Hall, 1991.

Sommerville, Ian Software Engineering. Fifth edition. Addison-Wesley: Harlow, 1995.

WAZLAWICK, R.S., “Análise e Projeto de Sistemas de Informação Orientados a Objetos”, Campus/Elsevier, 2004.

KOTONYA, G., SOMMERVILLE, I., “Requirements Engineering - Process and Techniques”, Ed. John Wiley & Sons, 1998.

JACOBSON, I., et. Al., “The Unified Software Development Process”, Ed. Addison-Wesley, 1999.

KRUCHTEN, P., “The Rational Unified Process - An Introduction”, Ed. Addison-Wesley, 2000.

GAMMA, E. et al., “Design Patterns - Elements of Reusable Object-Oriented Software”, Ed. Addison-Wesley, 1995.

FOWLER, M., “Patterns of Enterprise Application Architecture”, Ed. Addison-Wesley, 2003.

---

---

### **INE5418 – Computação Distribuída.**

**Carga Horária: 72 horas aulas.**

**Pré-requisito: INE5414 e INE5412.**

#### **Ementa:**

Arquitetura de Sistemas Distribuídos. Paradigmas de Computação Distribuída: Troca de Mensagens, Cliente/Servidor, Comunicação em Grupo, Objetos Distribuídos. Comunicação entre Processos. Suporte de SO para Computação Distribuída. Sincronização em Sistemas Distribuídos. Consistência e Replicação de Dados em Sistemas Distribuídos. Sistemas de Arquivo Distribuídos. Computação GRID.

#### **Bibliografia:**

Tanembaum, A. and Steen M., “Distributed Systems Principles and Paradigms”, Prentice-Hall, 2002.

Coulouris, G. et al., “Distributed Systems: Concepts and Design (Fourth Edition)”, Addison-Wesley, 2005.

Sinha, Pradeep K., “Distributed Operating Systems: Concepts and Design”, IEEE-Press, 1997.

M. L. Liu, “Distributed Computing: Principles and Applications”, Pearson Addison Wesley, 2004.

---

---

### **INE5419 – Engenharia de Software II.**

**Carga Horária: 72 horas aulas.**

**Pré-requisito: INE5417.**

**Ementa:**

Evolução da prática de desenvolvimento de software; qualidade de artefatos de software; modularidade e reusabilidade; modelagem estrutural e dinâmica em orientação a objetos, diferentes visões de um sistema; metodologias de análise e projeto orientadas a objetos; teste de software; manutenção de software; modelos de ciclo de vida; engenharia reversa; modelagem formal de sistemas; abordagens voltadas ao reuso de software; gerenciamento do processo de produção de software e técnicas de apoio ao gerenciamento do processo de produção de software; apoio automatizado ao desenvolvimento de software.

**Bibliografia:**

BOOCH, G. "Object oriented design - with applications", Redwood City, California: Benjamin Cummings, 1991.

COAD, P., YOURDON, E., "Análise baseada em objetos", Rio de Janeiro: Campus, 1992.

COAD, P., YOURDON, E., "Projeto baseado em objetos", Rio de Janeiro: Campus, 1993.

COLEMAN, D. *et al.*, "Object-oriented development: the Fusion method", Englewood Cliffs: Prentice Hall, 1994.

COMMER, "Principles of program design induced from experience with small public programs", IEEE Transactions on Software Engineering, New York, v.SE-7, n.2, 1981.

DEREMER, F., KRON, H.H., "Programming-in-the-large versus programming-in-the-small", IEEE Transactions on Software Engineering, New York, v.SE-2, n.2, 1976.

DEUTSCH, P., "Frameworks and reuse in the smalltalk 80 system", In: BIGGERSTAFF, T. Software reusability, New York: ACM Press, 1989. v.1, p. 57-71.

D'SOUZA, D. F. *et al.*, "Objects, Components, and Frameworks With Uml: The Catalysis Approach", [S.l.]: Addison-Wesley, 1997.

FAIRLEY, R., "Software Engineering concepts", [S.l.]: McGraw-Hill, 1986.

GAMMA, E., "Design patterns: elements of reusable object-oriented software", Reading: Addison-Wesley, 1994.

HAREL, D. *et al.* "Statecharts: a visual formalism for complex systems", Science of Computer Programming, [S.l.], n.8, p.231-274, 1987.

JACOBSON, I. *et al.* "Object-oriented software engineering - a use case driven approach", Reading: Addison-Wesley, 1992.

JOHNSON, R. E., "Components, frameworks, patterns", Feb. 1997. Disponível por FTP anônimo em st.cs.uiuc.edu (dez. 98).

LEWIS, T. *et al.* "Object-oriented application frameworks", Greenwich: Manning, 1995.

MEYER, B., "Object-oriented software construction. 2.ed", [S.l.]: Prentice Hall PTR, 1997.

OLIVEIRA, G., "As ferramentas CASE orientadas pelos objetos", Lisboa: Universidade Nova de Lisboa, 1996. Dissertação de mestrado.

PARNAS, D.; CLEMENTS, P., "A rational design process: how and why to fake it", IEEE Transactions on Software Engineering, New York, v.SE-12, n.2, 1986.

PAULK, M. C. *et al.*, "The Capability Maturity Model: Guidelines for Improving the Software Process", Addison-Wesley, Reading, 1995.

PREE, W., "Design patterns for object oriented software development", Reading: Addison-Wesley, 1994.

PRIETO-DÍAZ, R.; ARANGO, G. Domain analysis and software modeling. Los Alamitos: IEEE Computer Society Press, 1991.

PRESSMAN, R., "Software engineering. a practioner's approach", [S.l.]: McGraw-Hill, 1982.

ROSS, D. T., "Structured analysis (SA): a language for

communicating ideas”, IEEE Transactions on Software Engineering, New York, v.SE-3, n.1, 1977.

RUMBAUGH, J., et al., “Modelagem e projetos baseados em objetos”, Rio de Janeiro: Campus, 1994.

Software Engineering Institute. CMMI for Development. v.1.2. Aug. 2006.

WEST, R., “Reverse Engineering - an overview”, London: HSMO, 1993.

---

---

### **INE5420 – Computação Gráfica.**

**Carga Horária: 72 horas aulas.**

**Pré-requisito: MTM5512, MTM03, MTM5245 e INE5408.**

#### **Ementa:**

Computação Gráfica Básica. Sistema Gráfico Interativo. Transformações geométricas 2D e coordenadas homogêneas. *Clipping*. Curvas paramétricas em 2D e 3D. Estruturas de dados gráficas 3D. Navegação 3D. Projeções, perspectiva e *clipping* 3D. Superfícies paramétricas bicúbicas. Visualização em 3D contendo, *Rayshading*, *Raycasting* e *Raytracing*. Conversão por varredura e buffer de profundidade. Iluminação de objetos. Implementação de um *rayshader*. APIs Gráficas e OpenGL. Animação e utilização de modelos hierárquicos. Simulação de movimentação de animais e humanos. Realidade virtual e VRML.

#### **Bibliografia:**

Foley, J.D. van Dam, A. Feiner K.S., Jughes, J.F., “Computer Graphics: Principles And Practice”, Addison Wesley, 1993.

Neider, J. Davis, T. Woo, M., “OpenGL Programming Guide: The Official Guide to Learning OpenGL, Release 1”, Addison Wesley, 1993.

Rankin J.R., “Computer Graphics Software Construction”, Advances in Computer Science Series, Prentice Hall, 1989.

Persiano, R.C.M. de Oliveira, A.A.F., “Introdução à Computação Gráfica”, Livros Técnicos e Científicos Editora Ltda. 1989.

---

---



## **INE5421 – Linguagens Formais e Compiladores.**

**Carga Horária: 72 horas aulas.**

**Pré-requisito: INE5408 e INE5415.**

### **Ementa:**

O processo de compilação. Linguagens e suas representações. Gramáticas: definição formal, classificação (Hierarquia de Chomsky), propriedades, problemas de decisão e aplicações. Gramáticas regulares, autômatos finitos, conjuntos regulares e expressões regulares. Gramáticas livres de contexto. Autômatos de pilha. Teoria de *Parsing*. Análise léxica e sintática.

### **Bibliografia:**

FURTADO, O. J. V., “Apostila de Linguagens Formais e Compiladores – versão 2”, UFSC, 2002.

WOOD, D. , “Theory of Computation”, Ed. John Wiley & Sons, 1987.

AHO, A. V., SETHI, R., ULLMAN, J. D., “Compiler-Principles, Techniques and tools”, Ed. Addison-Wesley, 1986.

HOPCROFT, J. E., ULLMAN, J. D., “Formal Languages and Their Relations to Automata”, Addison-Wesley, 1969.

HOPCROFT, J. F., ULLMAN, J. D., “Introduction to Automata Theory, Languages and Computation”, Ed. Addison-Wesley, 1979.

SUDKAMP, T. A., “Languages and Machines – An Introduction to the Theory of Computer Science, 2. edição”, Ed. Addison Wesley, 1997.

MENESES, P. B., “Linguagens Formais e Autômatos”, Ed. Sagra Luzzato, 2. edição, 1998.

LEWIS, H. R. e PAPADIMITRIOU, C. H. , “Elementos de Teoria da Computação”, Ed. Bookman, 2. edição, 1998.

AHO, A. V., ULLMAN, J. D., “The Theory of Parsing, Translation, and Compiling, Volume I: Parsing”, Ed Prentice-Hall, Inc. 1972, 542p.

DIVERIO, T. A., MENEZES, P. B., “Teoria da Computação – Máquinas Universais e Computabilidade”, Ed. Sagra Luzzatto, Porto Alegre, 1999.

---

---

## **INE5422 – Redes de Computadores II.**

**Carga Horária: 72 horas aulas.**

**Pré-requisito: INE5414.**

**Ementa:**

Camada de Rede TCP/IP: endereçamento, roteamento, protocolos. Camada de Aplicação: conceitos e protocolos. Aspectos de segurança em redes de computadores. Dispositivos de interconexão, conceito de processadores de redes. Protocolos leves. Métodos formais para especificação e verificação de protocolos. Administração de redes de computadores.

**Bibliografia:**

Walrand, Jean, “Communication Networks, A First Course” - 2ª Edição, MCGraw-Hill. 1998, ISBN 0-256-17404-0.  
Tanenbaum, A. S., “Computer Networks”, 3ª Edição, Prentice-Hall, ISBN 0-13-394248-1.  
Soares, Lemos e Colcher, “Redes de Computadores, Das LANs MANs e WANs às Redes ATM” - 2ª Edição, Editora Campus, ISBN 85-7001-998-X.  
Fred Halsall, “Data Communication, Computer Networks and Open Systems” - Editora Addison Welsey, 1995.  
Dantas, Mario, “Tecnologias de Redes de Comunicação e Computadores”, Axcel Books, 2002.

---

---

**INE5423 – Banco de Dados I.**

**Carga Horária: 72 horas aulas.**

**Pré-requisito: INE5408.**

**Ementa:**

Banco de Dados (BD). Sistema de Gerência de BD: funcionalidades, módulos principais, categorias de usuários, dicionário de dados. Modelo relacional: conceitos, restrições de integridade, álgebra relacional, cálculo relacional. Linguagens SQL: DDL, DML, restrições de integridade, visões, autorização de acesso. Modelagem de dados: etapas do projeto de um BD relacional, modelo Entidade-Relacionamento (ER), mapeamento ER-relacional. Teoria da Normalização: objetivo, dependências funcionais, formas normais.

**Bibliografia:**

Korth, H. F., Sudarshan, S, Silberschatz, A., “ Sistema de Banco de Dados”, 3ª edição. Editora Makron Books, 1999.

Imasri, R., Navathe S. B., “ Sistema de Banco de Dados, 3<sup>a</sup> edição”, Editora LTC. 2002.

Elmasri, R., Navathe S. B., “Fundamentals of Database Systems. 4<sup>th</sup> ed.”, Addison-Wesley. 2003.

Date, C. J., “Introdução a Sistemas de Bancos de Dados. 8<sup>a</sup> edição”, Editora Campus, 2004.

Atzeni, P., Ceri, S., Paraboschi, S. and Torlone, R., “Database Systems\_Concepts, Languages and Architectures”, McGraw Hill, 2000.

Heuser, C.A., “Projeto de Banco de Dados. 5<sup>a</sup> edição”, Série Livros Didáticos – Instituto de Informática da UFRGS, número 4. Editora Sagra-Luzzatto, 2004.

Ramakrisnan, R. and Gehrke, J., “Database Management Systems”, McGraw Hill, second edition, 2000.

---

---

## **INE5424 – Sistemas Operacionais II.**

**Carga Horária: 72 horas aulas.**

**Pré-requisito: INE5412.**

### **Ementa:**

Desenvolvimento de Sistemas Operacionais: projeto lógico, arquitetura, inicialização de Sistemas Computacionais, componentes (processos, *threads*, escalonador, mecanismos de sincronização, gerenciadores de memória, sistemas de arquivos, *drivers* de dispositivos de entrada e saída), teste, depuração, estudos de caso.

### **Bibliografia:**

ANDERSON, t., “The Case for Application-Specific Operating Systems”, in Proceedings of the Third Workshop on Workstation Operating Systems, pages 92-94, 1992.

BACH, M.J., “The Design of the UNIX Operating System”, Prentice-Hall, 1987.

BAR, M., “Linux Internals”, Osborne McGraw-Hill, 2000.

BEUCHE, D., at al., “The PURE Family of Object-Oriented Operating Systems for Deeply Embedded Systems”, in Proceedings of the 2nd IEEE International Symposium on Object-Oriented Real-Time Distributed Computing, 1999.

CAMPBELL, R. H., JOHNSTON, G.M. and RUSSO, F., “hoices (Class Hierarchical Open Interface for Custom Embedded systems), Operating Systems Review”, 21(3):9-17, 1987.

CZARNECKI, K., EISENECKER, U., “Generative programming: Methods, Tools, and Applications”, Addison-Wesley, 2000.

ENGLER, D.R., KAASHOEK, M. F., O'TOOLE, J., “Exokernel: An Operating System Architecture for Application-level Resource Management”, in Proceedings of the Fifteenth ACM Symposium on Operating Systems Principles, pages 251-266, 1995.

FORD, B. at al., “The Flux OS Toolkit: Reusable Components for OS Implementation”, in Proceedings of the Sixth Workshop on Hot Topics in Operating Systems, pages 14-19, 1997.

FROHLICH, A. A., “Application-Oriented Operating Systems”, GMD - Forschungszentrum Informationstechnik, 2001.

LEFFLER, S., at al., “The Design and Implementation of The 4.3 BSD UNIX Operating System”, Addison-Wesley, 1989.

PARNAS, L., “On the Design and Development of Program Families”, IEEE Transactions on Software Engineering, SE-2(1):1-9, 1976.

SILBERSCHATZ, A., GALVIN, P., PETERSON, J., “Operating Systems Concepts”, 5th ed., John Wiley and Sons, 1998.

SCHRODER, W.P., “The Logical Design of Parallel Operating Systems”, Prentice-Hall, 1994.

STROUDTRUP, B., “The C++ Programming Language”, Addison-Wesley, 1997.

TANENBAUM, A.S., “Modern Operating Systems”, Prentice-Hall, 1992.

---

---

## **INE5425 – Modelagem e Simulação.**

**Carga Horária: 72 horas aulas.**

**Pré-requisito: INE5405.**

### **Ementa:**

Modelagem de sistemas. Modelos analíticos e Modelos de simulação. Processos estocásticos. Introdução à teoria das filas. Redes de filas. Geração de números pseudo-aleatórios. Geração de variáveis aleatórias. Metodologia de projetos de modelagem e simulação de sistemas. Modelos voltados à simulação: definição, taxonomia e desenvolvimento. Validação de modelos de simulação. Práticas de modelagem e simulação. Linguagens de simulação. Estudo de casos.

### **Bibliografia:**

Freitas Filho , P. J., “Introdução à Modelagem e Simulação Discreta de Sistemas”.

Kelton e outros, “Simulation with Arena”, McGraw-Hill, 1998.

Banks, J., Carson, J., “Discrete-Event System Simulation”, Prendice-Hall, 1996.

Law, A. M., Kelton, W. D., “Simulation Modeling and Analisis”, McGraw-Hill, 1991.

Profozich, David, “Managing Changing with BPS”, Prendice-Hall, 1997.

---

---

### **INE5426 – Construção de Compiladores.**

**Carga Horária: 72 horas aulas.**

**Pré-requisito: INE5421.**

#### **Ementa:**

Projeto de especificação de linguagens de programação. Implementação das etapas que compreendem o processo de compilação: Análise Léxica, Análise Sintática, Análise Semântica, Geração e Otimização de Código. Evolução e tendências da área de compiladores e linguagens de programação.

#### **Bibliografia:**

FURTADO, O. J. V., “Apostila de Linguagens Formais e Compiladores – versão 2”, UFSC, 2002.

WOOD, D. , “Theory of Computation”, Ed. John Wiley & Sons, 1987.

AHO, A. V., SETHI, R.,ULLMAN, J. D., “Compiler-Principles, Techniques and tools”, Ed. Addison-Wesley, 1986.

HOPCROFT, J. E., ULLMAN, J. D., “Formal Languages and Their Relations to Automata”, Addison-Wesley, 1969.

HOPCROFT, J. F., ULLMAN, J. D., “Introduction to Automata Theory, Languages and Computation”, Ed. Addison-Wesley, 1979.

SUDKAMP, T. A., “Languages and Machines – An Introduction to the Theory of Computer Science, 2. edição”, Ed. Addison Wesley, 1997.

MENESES, P. B., “ Linguagens Formais e Autômatos”, Ed. Sagra Luzzato, 2. edição, 1998.

LEWIS, H. R. e PAPANIMITRIOU, C. H. , “Elementos de Teoria da Computação”, Ed. Bookmam, 2. edição, 1998.

AHO, A. V., ULLMAN, J. D., “The Theory of Parsing, Translation, and Compiling, Volume I: Parsing”, Ed Prentice-Hall, Inc. 1972, 542p.

DIVERIO, T. A., MENEZES, P. B., “Teoria da Computação – Máquinas Universais e Computabilidade”, Ed. Sagra Luzzatto, Porto Alegre, 1999.

---

---

### **INE5427 – Planejamento e Gestão de Projetos.**

**Carga Horária: 72 horas aulas.**

**Pré-requisito: INE5419.**

**Ementa:**

Projetos. Metodologias de planejamento e gestão de projetos. Áreas de conhecimento da gerência de projetos: Escopo, Tempo, Risco, Integração, Comunicação, Custo, Recursos Humanos, Aquisição, Qualidade. Grupos de processos: Iniciação, Planejamento, Execução, Controle, Encerramento. Técnicas de acompanhamento de projetos. Ferramentas computacionais de apoio ao planejamento e gerência de projetos. Estudo de casos.

**Bibliografia:**

BRUZZI, Demerval G., “Gerência de Projetos: Uma Visão Prática”, São Paulo: Érica, 2002.

CARVALHO, M. M., RABECHINI JR., “Construindo competências para gerenciar projetos”, São Paulo: Atlas, 2005.

CLELAND, D., IRELAND, L, R., “Gerência de Projetos”, Rio de Janeiro: Reichmann & Affonso Editores, 2002.

DINSMORE, P.C., “Gerenciamento de Projetos: como gerenciar seu projeto com qualidade, dentro do prazo e custos previstos”, Rio de Janeiro: Qualitymark, 2004.

DINSMORE, P.C., SILVEIRA, N.F.H., “Gerenciamento de Projetos e o Fator Humano”, Rio de Janeiro: Qualitymark, 2005.

- GASNIER, D.G., “Guia prático para gerenciamento de projetos: manual de sobrevivência para os profissionais de projetos. 3. ed”, São Paulo: IMAM, 2003.
- HELDMAN, K., “Gerência de Projetos”, Rio de Janeiro: Campus, 2003.
- KEELLING, R , “Gestão de Projetos: Uma Abordagem Global”, São Paulo: Saraiva, 2002.
- KERZNER, H., “Gestão de Projetos: As Melhores Práticas. 2. ed”, Porto Alegre: Bookman, 2006.
- MENEZES, L.C.M., “Gestão de Projetos”, São Paulo: Atlas, 2003.
- MEREDITH, J.R., MANTEL, S.J. J., “Administração de Projetos: Uma Abordagem Gerencial. 4. Ed”, Rio de Janeiro: LTC, 2003.
- MOLINARI, L., “Gestão de Projetos - Técnicas e Práticas com Ênfase em Web”, Rio de Janeiro: Érica, 2004.
- PHILLIPS, J., “Gerência de projetos de Tecnologia da Informação”, Rio de Janeiro: Campus, 2003.
- PRADO, D.S., “Gerenciamento de Projetos nas Organizações, 2. ed”, Belo Horizonte: Desenvolvimento Gerencial, 2003.
- PROJECT MANAGEMENT INSTITUTE (Editor), “Um Guia do Conjunto de Conhecimentos do Gerenciamento de Projetos” – PMBOK (*Project Management Body of Knowledge*) Guide. PMI, Edição em português – 2004.
- PROJECT MANAGEMENT INSTITUTE, “A guide to the project management body of knowledge”, (PMBOK Guide) – última edição disponível.
- RABECHINI J. R., “O Gerente de Projetos na Empresa”, São Paulo: Atlas, 2005.
- VALERIANO, D. L., “Gerenciamento Estratégico e Administração por Projetos”, São Paulo: Makron Books, 2001.
- VALERIANO, D. L., “Moderno Gerenciamento de Projetos”, São Paulo: Prentice Hall, 2005.
- VARGAS, R.V., “Gerenciamento de Projetos: Estabelecendo Diferenciais Competitivos, 5. ed.”, Rio de Janeiro: Brasport, 2003.
- VIEIRA, M.F., “Gerenciamento de Projetos de Tecnologia da Informação”, Rio de Janeiro: Campus, 2004.

WYSOCKI, R.K., MCGARY, R., “Effective Project Management: Traditional, Adaptive, Extreme, 3. ed.”, USA: John Wiley & Sons, 2003.

WYSOCKI, R.K., et al., “Effective Project Management. 2. ed.”, USA: John Wiley & Sons, Inc, 2000.

XAVIER, C.M.S., “Gerenciamento de Projetos: Como Definir e Controlar o Escopo do Projeto”, Rio de Janeiro: Saraiva, 2005.

---

---

### **INE5428 – Informática e Sociedade.**

**Carga Horária: 72 horas aulas.**

**Pré-requisito: INE5407.**

**Ementa:**

Ética pessoal, profissional e pública na área da informática. Dilemas éticos do profissional da informática; privacidade, vírus, *hacking*, uso da internet, direitos autorais, etc. Desemprego e informatização. Responsabilidade social. O profissional e o mercado de trabalho. Trabalho e relações humanas. O empreendedorismo como opção do profissional da informática. Legislação: Política nacional e tendências atuais referentes à regulamentação da profissão.

**Bibliografia:**

ROSZAK, T., “O Culto da Informação”, Ed. Brasiliense, 1989.

BRETON, P., “História da Informática”, Ed. São Paulo UNESP, 1989.

MASIERO, P.C., “Ética em Computação”, Editora da Universidade de São Paulo, 2000.

FORRESTER, T., MORRISON, P., “Computer Ethics”, The MIT Press, London, 1994.

DEGEN, R., “O Empreendedor - Fundamentos da Iniciativa Empresarial”, McGraw-Hill, S.P. 1989.

DORNELAS, J.C. A., “Empreendedorismo – Transformando Idéias em Negócios”, Rio de Janeiro. Ed. Campus, 2001.



FILION, L.J., “Vision et Relacion”, Clefs du Succès de L’Entrepreneur; Les éditions de L’entrepreneur, 1991.  
OECH, R., “Toc na Cuca”, Livraria Cultura Editora, Rio de Janeiro, 1988.  
CIAGE/FGV, CADERNOS – 12, “ Projeto Básico de Negócios”, Fundação Getúlio Vargas, 1992.  
PAVANI, C., DEUTSCHER, J.A., LOPS, S.M., “Planos de Negócios”, Ed. Lexicon Informática; R.J.; 1997.  
GOUVÊA, S., "O Direito na Era Digital", Ed. Mauad, R.J., 1997.

---

---

**INE5429 – Segurança em Computação.**

**Carga Horária: 72 horas aulas.**

**Pré-requisito: INE5403, INE5414, INE5415.**

**Ementa:**

Segurança em aplicações: programação segura, detecção de falhas, códigos maliciosos (*malware*). Segurança em sistemas operacionais: princípios de controle de acesso, sistemas confiáveis. Segurança em redes de computadores: ataques e defesas. Princípios de criptografia: criptografia simétrica e assimétrica, integridade de dados. Protocolos de autenticação: princípios, infra-estrutura de chaves públicas e aplicações (X.509, OpenPGP, SPKI, IBE), protocolos criptográficos (S/Mime, IPSec, SSL, OpenSSH, Kerberos, VPNs).

**Bibliografia:**

Stallings, W., “Cryptography and Network Security: Principles and Practice”, 3rd ed, Prent-Hall, 2003.  
Chandra, P., Messier, M., Viega, J., “Network Security with OpenSSL”, O’Reilly, 2002.  
Mao, W., “Modern Cryptography”, Prentice-Hall, 2004.  
Pfleeger, C.P., Pfleeger, L.P., “Security in Computing, 3rd ed.”, Prentice-Hall, 2003.

---

---

**INE5430 – Inteligência Artificial.****Carga Horária: 72 horas aulas.****Pré-requisito: INE5405, INE5413 e INE5416.****Ementa:**

Histórico, Teoria de Problemas, Paradigma Simbólico da IA, Modelagem de Agentes Inteligentes, Métodos de Busca, Representação de Conhecimento, Métodos de Inferência, Tratamento de Incertezas, Sistemas Especialistas, Lógica Nebulosa, Paradigma Conexionista da IA, Redes Neurais Diretas e Algoritmos de Aprendizado, Paradigma Evolucionário da IA, Algoritmos Genéticos.

**Bibliografia:**

Russell and Norvig. Artificial Intelligence: A modern approach  
J.M. Barreto. Inteligência Artificial, uma abordagem híbrida  
E. Rich, K. Knight. Artificial Intelligence  
G. Bittencourt. Inteligência Artificial, ferramentas e teoria.  
R.A. Rabuske. Inteligência Artificial

---

---

**INE5431 – Sistemas Multimídia.****Carga Horária: 72 horas aulas.****Pré-requisito: INE5414.****Ementa:**

Definição de sistemas multimídia e problemática envolvida. Representação digital de áudio, imagens e vídeos. Técnicas e padrões de compressão. Aplicações multimídia: caracterização e requisitos. Protocolos de transporte de mídia e de controle-sinalização. Qualidade de Serviço.

**Bibliografia:**

FLUCKIGER, F., “Understanding Networked Multimedia: Applications and Technology”, Prentice Hall, 1995.  
KUO, F., EFFELSBERG, W., GARCIA-LUNA-ACEVES, J.J., “Multimedia Communications: Protocols and Applications”, Prentice Hall PTR, 1998.  
STEINMETZ, R., NAHRSTEDT, K., “Multimedia: Computing, Communications and Applications”, Prentice Hall Series in Innovative Technology, 1995.

HERSENT, O., GUIDE, D., PETIT, J.P, “Telefonia IP: Comunicação Multimídia Baseada em Pacotes”, São Paulo: Addison Wesley, 2002.

JAMES, F., KUROSE, F., KEITH, W., “Redes de Computadores e a Internet, São Paulo: Addison Wesley, 2003.

---

---

## **INE5432 – Banco de Dados II.**

**Carga Horária: 72 horas aulas.**

**Pré-requisito: INE5423.**

### **Ementa:**

SQL embutida: instruções estáticas e dinâmicas, cursores. Processamento de consultas: otimização algébrica; plano de execução de uma consulta considerando estimativas sobre os dados, índices, *buffers* e *pipelines*. Transações: definição, propriedades, estados. Recuperação de falhas: categorias de falhas, gerência de buffer, técnicas de recuperação. Controle de concorrência: teoria da serializabilidade, escalonadores otimistas e pessimistas, tratamento de *deadlock*. Noções básicas de bancos de dados distribuídos: arquiteturas, projeto, processamento de consultas, gerência de transações.

### **Bibliografia:**

Korth, H. F., Sudarshan, S, Silberschatz, A., “ Sistema de Banco de Dados”, 3ª edição. Editora Makron Books, 1999.

Imasri, R., Navathe S. B., “ Sistema de Banco de Dados, 3ª edição”, Editora LTC. 2002.

Elmasri, R., Navathe S. B., “Fundamentals of Database Systems. 4<sup>th</sup> ed.”, Addison-Wesley. 2003.

Date, C. J., “Introdução a Sistemas de Bancos de Dados. 8ª edição”, Editora Campus, 2004.

Atzeni, P., Ceri, S., Paraboschi, S. and Torlone, R., “Database Systems\_Concepts, Languages and Architectures”, McGraw Hill, 2000.

Heuser, C.A., “Projeto de Banco de Dados. 5ª edição”, Série Livros Didáticos – Instituto de Informática da UFRGS, número 4. Editora Sagra-Luzzatto, 2004.

Ramakrisnan, R. and Gehrke, J., “Database Management Systems”, McGraw Hill, second edition, 2000.

---

---

**INE5433 – Trabalho de Conclusão de Curso I.****Carga Horária: 108 horas aulas.****Pré-requisito: INE5427.****Ementa:**

Conceituação da pesquisa em cursos de graduação em ciência da computação. Conceitos e técnicas para preparação de projetos de pesquisa: introdução, objetivos, hipóteses, metodologia, justificativa, resultados esperados, estado da arte, desenvolvimento, experimentos, conclusões. Conceitos e técnicas para proceder à revisão bibliográfica e a escrita de artigos científicos. Orientação para o desenvolvimento do projeto em Ciência da Computação. Relatórios sobre o andamento dos projetos.

**Bibliografia:**

Não definida.

---

---

**INE5434 – Trabalho de Conclusão de Curso II.****Carga Horária: 108 horas aulas.****Pré-requisito: INE5433.****Ementa:**

Orientação para o desenvolvimento da fase final do projeto em Ciência da Computação. Defesa final do trabalho perante banca examinador.

**Bibliografia:**

Não definida.

---

---

**4.2. Disciplinas Optativas.**

Para a integralização curricular, o aluno deverá cursar 172 créditos correspondentes ao total de disciplinas obrigatórias e 24 créditos de disciplinas optativas. Com relação às disciplinas optativas o aluno poderá cursar até oito créditos de disciplinas optativas de caráter geral oferecidas pela UFSC e pelo menos dezesseis créditos de disciplinas optativas eletivas escolhidas na lista que segue:

## **INE5435 – Integração Software/Hardware.**

**Carga Horária: 72 horas aulas.**

**Pré-requisito: INE5411.**

### **Ementa:**

Conceitos, técnicas e metodologias pertinentes ao desenvolvimento de projetos integrados de *software* e *hardware*, projeto baseado em plataforma, arquiteturas de *software* e *hardware*, componentes de *software* e *hardware*, estratégias de implementação, microcontroladores, sistemas em chip único (*system-on-a-chip*), aplicações de tempo-real, aplicações distribuídas. Estudo de caso: inicialização de sistemas computacionais, *firmware*, tratadores de dispositivos, projeto integrado de *software* e *hardware*.

### **Bibliografia:**

Maurice J. Bach, “The Design of the UNIX Operating System”, Prentice-Hall, 1987.

Moshe Bar, “Linux Internals”, Osborne McGraw-Hill, 2000.

Amit Dhir, “The Digital Consumer Technology Handbook”, Newnes, 2004.

Antônio Augusto Fröhlich, “Application-Oriented Operating Systems”, GMD - Forschungszentrum Informationstechnik, 2001.

Samuel J. Leffler, Marshall Kirk McKusick, and Michael J. Karels, “The Design and Implementation of The 4.3 BSD UNIX Operating System”, Addison-Wesley, 1989.

Alessandro Rubini and Jonathan Corbet, “Linux Device Drivers, 2nd ed.”, O'Reilly, 2001.

Abraham Silberschatz, Peter Galvin and James Peterson, “Operating Systems Concepts, 5th ed.”, John Wiley and Sons, 1998.

Bjarne Stroustrup, “The C++ Programming Language”, Addison-Wesley, 1997.

Andrew S. Tanenbaum, “Modern Operating Systems”, Prentice-Hall, 1992.

Wayne Wolf, “Computers as Components: Principles of Embedded Computing System Design”, Morgan Kaufman, 2000.

---

---

## **INE5436 – Arquitetura de Computadores I.**

**Carga Horária: 72 horas aulas.**

**Pré-requisito: INE5411.**

### **Ementa:**

Fundamentos do projeto de computadores (mercados, custo, preço e desempenho). Arquiteturas de conjuntos de instruções (máquinas baseadas em acumulador, pilha e registradores, máquinas *load-store*). Arquiteturas RISC, CISC e DSP. *Pipelining* e emissão múltipla (máquinas superescalares e VLIW). Exploração de paralelismo entre instruções (escalonamento estático e dinâmico, previsão estática e dinâmica de desvios, execução especulativa, *software pipelining*, *trace scheduling*). Projeto de hierarquia de memória.

**Bibliografia:**

Livro-texto:

HENNESSY, J., PATTERSON, D., "Computer Architecture: A Quantitative Approach", 4th edition, Morgan Kaufmann-Elsevier, 2006.

---

---

**INE5437 – Arquitetura de Computadores II.**

**Carga Horária: 72 horas aulas.**

**Pré-requisito: INE5411.**

**Ementa:**

Arquitetura de computadores modernos. Contextualização da taxonomia de Flynn (SISD, SIMD, MISD, MIMD).

Redes de Interconexão. Multicomputadores. Multiprocessadores. Máquinas com Acesso Uniforme à Memória (UMA). Multiprocessadores simétricos (SMP). Máquinas com Acesso Não Uniforme à Memória (NUMA). Máquinas com Coerência de Cache e Acesso Não Uniforme à Memória (ccNUMA).

Processadores Massivamente Paralelos (MPP).

Sistemas Distribuídos. Clusters. Grids.

**Bibliografia:**

Buyya, Rajkumar (Ed.). "High Performance Cluster Computing: Programming and Applications". Prentice-Hall, 2v. ISBN 0-13-013785-5, 1999.

Casavant, Thomas L.; Kuhl, Jon G. "A Taxonomy of Scheduling in General-Purpose Distributed Computing Systems". IEEE Transactions on Software Engineering, v. 14, n. 2, p. 141-154, Feb. 1988.

Charlotte, Project, Disponível em Setembro de 2004:

<http://www.cs.nyu.edu/milan/charlotte/frmain.html>.

Coulouris, G., Dollimore, J., Kindberg, T., “Distributed Systems: Concepts and Design (3rd Edition)”, Addison Wesley; ISBN: 0201619180, 2000.

Culler, D.E., Singh, J.P., “Parallel Computer Architecture: A Hardware/Software Approach”, Morgan Kaufmann, ISBN 1-55860-343-3, 1999.

Tanenbaum, A S., van Steen, M., “Distributed Systems: Principles and Paradigms”, Prentice Hall; 1st edition, ISBN: 0130888931, 2002.

---

---

## **INE5438 – Laboratório de Microprocessadores e Lógica Programável.**

**Carga Horária: 72 horas aulas.**

**Pré-requisito: INE5411.**

### **Ementa:**

Perspectiva histórica, revisão de organização de computadores, microprocessadores e microcontroladores. Princípios de projeto de sistemas embarcados, restrições de tempo-real, técnicas de implementação, teste, emulação e depuração. Entrada e Saída (portas de E/S, tratamento de interrupções, E/S programada, DMA). Temporizadores, contadores e relógios. Interfaceamento analógico (conversores A/D e D/A, sensores e atuadores). Estudo de casos: sistemas embarcados baseados em microcontroladores, prototipação de sistemas embarcados em lógica programável, sistemas embarcados baseados em sistemas operacionais de tempo-real (RTOS).

### **Bibliografia:**

Bruce Powel Douglass, “Doing Hard Time: Developing Real-Time Systems with UML, Objects, Frameworks and Patterns”, Addison-Wesley, 1999.

David E. Simon, “An Embedded Software Primer”, Addison-Wesley, 1999.

Peter J. Ashenden, “The Designer's Guide to VHDL”, Morgan Kaufmann, 2nd ed., 2002.

Qing Li and Caroline Yao, “Real-time Concepts for Embedded Systems”, CMP, 2003.

Steve Heath, “Embedded Systems Design”, Newnes, 2003.

Microcontroller's Manuals, User Guides and Data sheets.

---

---

**INE5439 – Sistemas Embarcados.**

**Carga Horária: 72 horas aulas.**

**Pré-requisito: INE5411.**

**Ementa:**

Especificação (requisitos, linguagens, níveis e estilos de descrição). *Hardware*: entrada e saída (*sample-hold*, conversores A/D e D/A, sensores e atuadores), unidades de processamento (microprocessadores, DSPs, ASIPs e lógica reconfigurável) e memórias (*flash, cache e scratch pad*). Eficiência energética: compiladores energeticamente conscientes e gerenciamento de potência (DVS e DPM). Compactação de código. Ferramentas de projeto de hardware e de software (simulador, síntese comportamental e lógica, gerador de código e depurador). *Systems-on-Chip* e co-projeto de *hardware e software*.

**Bibliografia:**

Livro-texto:

MARWEDEL, P., "Embedded System Design", Springer, 2003.

---

---

**INE5440 – Tópicos Especiais em Arquitetura de Computadores.**

**Carga Horária: 72 horas aulas.**

**Pré-requisito: INE5411.**

**Ementa:**

Ementa livre sobre novas técnicas ou tendências em Arquitetura de Computadores.

**Bibliografia:**

Não definida.

---

---

**INE5441 – Sistemas de Tempo Real.**

**Carga Horária: 72 horas aulas.**

**Pré-requisito: INE5412.**

**Ementa:**

Conceituação básica e Caracterização de um Sistema de Tempo Real. Escalonamento de Tempo Real: modelo de tarefas;



abordagens de escalonamento de tempo real. Controle de acesso a recursos: uso de protocolos de acesso a recursos. Comunicação em Tempo Real: requisitos, protocolos. Sistemas Operacionais de Tempo Real: aspectos funcionais, aspectos temporais, estudo de casos. Linguagens de Programação de Tempo Real: características, requisitos, estudo de casos.

**Bibliografia:**

WOLF, w., “Computers as Components Principles of Embedded Computing System Design”, Morgan Kaufmann Publishers, 2001.  
KOPETZ, H., “Real-Time Systems”, Kluwer Academic Publishers, 1997.  
BUTTAZO, G., “Hard Real-Time Computing Systems”, Kluwer Academic Publishers, 1998.  
SON, S. H., “Advances in Real-Time Systems”, Prentice Hall, 1995.  
BURNS, A., WLLING, A., “Real-Time Systems and Programming Languages”, Addison-Wesley, 1997.

---

---

**INE5442 – Circuitos e Sistemas Integrados.**

**Carga Horária: 72 horas aulas.**

**Pré-requisito: INE5411.**

**Ementa:**

Modelagem simplificada de transistores NMOS e PMOS. O inversor CMOS: característica de transferência, atraso, consumo estático e dinâmico. Lógica combinacional, *flip-flops* e registradores em CMOS. Estruturas regulares: ULA, PLA, ROM. Princípios básicos de processos de fabricação. Regras geométricas e elétricas de projeto. *Scaling*. Variabilidade em tecnologias submicrométricas. Estilos *full* e *semi-custom*, lógica programável pelo usuário. Leiaute. Níveis de abstração, metodologias de projeto e ferramentas de EDA (simuladores, analisadores, extratores, ferramentas de verificação e síntese).

**Bibliografia:**

Livro-texto:

RABAEY, J., "Digital Integrated Circuits: A Design Perspective", Prentice-Hall, 2005.

Leitura-complementar:

Weste, N., Harris, D., "CMOS VLSI Design: A Circuits and Systems Perspective", 3rd edition, 2005.

---

---

**INE5443 – Reconhecimento de Padrões.**

**Carga Horária: 72 horas aulas.**

**Pré-requisito: INE5420.**

**Ementa:**

Considerações iniciais sobre padrões. Técnicas simbólicas. Técnicas sub simbólicas: redes neurais. Raciocínio baseado em caso – estendendo RP com um *framework* de inteligência artificial. Técnicas estatísticas. Geração de padrões: análise de sinais e imagens. Criação de aplicação de reconhecimento de padrões.

**Bibliografia:**

A definida.

---

---

**INE5444 – Estágio Supervisionado I.**

**Carga Horária: 72 horas aulas.**

**Pré-requisito: INE5417.**

**Ementa:**

Estágio supervisionado na área de computação.

**Bibliografia:**

Não definida.

---

---

**INE5445 – Estágio Supervisionado II.**

**Carga Horária: 72 horas aulas.**

**Pré-requisito: INE5417.**

**Ementa:**

Estágio supervisionado na área de computação.

**Bibliografia:**

Não definida.

---

---

**INE5446 – Tópicos Especiais em Sistemas de Infra-estrutura I.**

**Carga Horária: 72 horas aulas.**

**Pré-requisito: INE5412.**

**Ementa:**

Ementa livre para assuntos relevantes na área de Sistemas de Infra-Estrutura.

**Bibliografia:**

Não definida.

---

---

**INE5447 – Tópicos Especiais em Sistemas de Infra-estrutura II.**

**Carga Horária: 72 horas aulas.**

**Pré-requisito: INE5412.**

**Ementa:**

Ementa livre para assuntos relevantes na área de Sistemas de Infra-Estrutura.

**Bibliografia:**

Não definida.

---

---

**INE5448 – Tópicos Especiais em Aplicações Tecnológicas I.**

**Carga Horária: 72 horas aulas.**

**Pré-requisito: INE5412.**

**Ementa:**

Ementa livre para assuntos relevantes na área de Aplicações Tecnológicas.

**Bibliografia:**

Não definida.

---

---

**INE5449 – Tópicos Especiais em Aplicações Tecnológicas II.**

**Carga Horária: 72 horas aulas.**

**Pré-requisito: INE5412.**

**Ementa:**

Ementa livre para assuntos relevantes na área de Aplicações Tecnológicas.

**Bibliografia:**

Não definida.

---

---

**INE5450 – Tópicos Especiais em Aplicações Tecnológicas III.**

**Carga Horária: 72 horas aulas.**

**Pré-requisito: INE5412.**

**Ementa:**

Ementa livre para assuntos relevantes na área de Aplicações Tecnológicas.

**Bibliografia:**

Não definida.

---

---

**INE5619 – Administração e Gerência de Redes de Computadores.**

**Carga Horária: 72 horas aulas.**

**Pré-requisito: INE5422.**

**Ementa:**

Sistemas de administração de redes, fontes, e fluxos de informação para administração e objetivos estratégicos; Administração Inovadora de Redes; Tecnologias e Suporte à Administração de Redes (TMN do ITU-T, OSI/NM da ISO, SNMP da Internet ou assemelhadas); Forças do Mercado de Administração de Redes (gerência proprietária, redes locais, e redes heterogêneas); Tendências em Administração e Gerência de Redes. Avaliação de plataformas de gerência, segurança da gerência de redes.

**Bibliografia:**

Apostila de Redes de Computadores.

Anais dos Simpósios Brasileiros de Redes de Computadores.

Revistas especializadas em Comunicação de Dados e Redes de Computadores.

Sites com informações em Comunicações de Dados e Redes de Computadores.

JAKOBS, K.; PROCTER, R.; WILLIAMS, R. The Making of Standards: Looking Inside the Work Groups. IEEE Communications Magazine. Vol. 39, No. 4. Pg. 102-107, April. 2001.

HELD, G. Comunicação de Dados. Editora Campus. 1999.

MOURA, J. A. B. et al. Redes locais de computadores - Protocolo de alto nível e avaliação de desempenho. São Paulo. McGraw-Hill, 1986.

TANENBAUM, A.S. Redes de Computadores. Quarta edição. Editora Campus, 2003.

COMER, Douglas E. Internetworking with TCP/IP. Volume I, Principles, Protocols, and Architecture. Third Edition - Printice Hall, Inc. 1995.

COMER, D. E. Interligação em Redes com TCP/IP. Volume I. Princípios, Protocolos e Arquitetura. Segunda Edição. Editora Campus. 1998.

COMER, D. E.; STEVENS, D. L. Interligação em Redes com TCP/IP. Volume II. Projeto, implementação e detalhes internos. Editora Campus. 1999.

DERFLER, F.Jr. Guia de Conectividade, Editora Campus. Rio de Janeiro, 1993.

SOARES, L.F.G. et al. Redes de Computadores: das LANs, MANs e WANs às Redes ATM. Editora Campus, Rio de Janeiro, 1995.

ALVES, L. Protocolos para Redes de Comunicação de Dados. São Paulo, Atlas.1987.

CHURCHILL,B.,JORDAN,L.,“Comunicações e Redes com o PC , Axcel Books, 1994.

COMER, D. E., “Computer Networks and Internets”, Prentice Hall. NJ, USA, 1999.

COMER, D. E., Redes de Computadores e Internet. Bookman. Porto Alegre, 2001.

PERKIS, C. H. Mobile IP Design Principles and Practices. Addison Wesley Longman, 1998.

STALLINGS, W. SNMP, SNMPv2, SNMPv3, and RMON 1 and 2. Third Edition. Addison-Wesley. 1999.

AIDAROUS, S. & PLEVYAK, T., “Telecommunications Network Management Technologies and Implementations”, IEEE Press. 1988.

DANTAS, M., “Tecnologias de Redes de Comunicação e Computadores”, Axcel Books ISBN :85-7323-169-6. 2002.

KUROSE, J. F.; ROSS, K. W., “Redes de Computadores e a Internet (Uma Nova Abordagem)”, Addison Wesley. 2003.

PETERSON, L. L., DAVIE, B. S. , “Redes de Computadores (Uma Abordagem Sistêmica)”, Segunda Edição. Morgan Kaufmann Publishers. 2004.

Walrand, Jean, “Communication Networks, A First Course” - 2ª Edição, MCGraw-Hill. 1998,ISBN 0-256-17404-0.

Tanenbaum, A. S., “Computer Networks”, 3ª Edição, Prentice-Hall, ISBN 0-13-394248-1.

Soares, Lemos e Colcher, “Redes de Computadores, Das LANs MANs e WANs às Redes ATM” - 2ª Edição, Editora Campus, ISBN 85-7001-998-X.

Fred Halsall, “Data Communication, Computer Networks and Open Systems” - Editora Addison Welsey, 1995.

---

---

### **INE5624 – Engenharia da Usabilidade.**

**Carga Horária: 72 horas aulas.**

**Pré-requisito: INE5419.**

**Ementa:**

Cognição humana, Semiótica Computacional; Critérios e recomendações ergonômicas; Engenharia de requisitos visando a usabilidade, Técnicas de Análise Ergonômica do Trabalho; Técnicas de Concepção e de Especificação funcional da tarefa interativa; Técnicas de Projeto e Especificação da interface com o usuário; Técnicas de Prototipagem *Off-line* e *On-line*; Técnicas de avaliações heurísticas; Técnicas de inspeções ergonômicas, Norma ISO 9241, Técnica de ensaios de interação.

**Bibliografia:**

A definir.

---

---

### **INE5640 – Computação Móvel.**

**Carga Horária: 72 horas aulas.**

**Pré-requisito: INE5422.**

**Ementa:**

Computação Ubíqua; Sistemas de Comunicação sem Fios; Serviços de Localização e Posicionamento; Transações e Gestão da Informação em Operação Desligada; Interfaces para dispositivos de dimensão muito reduzida; Engenharia de Sistemas de Informação para suporte à mobilidade.

**Bibliografia:**

A definir.

---

---

**INE5646 – Programação Web.**

**Carga Horária: 72 horas aulas.**

**Pré-requisito: INE5417.**

**Ementa:**

Sistemas distribuídos na *web*: protocolos, segurança, clusters, arquiteturas, concorrência. Tecnologias para camada *web*: modelos, linguagens e *frameworks*. Tecnologias para camada lógica: serviços, linguagens e *frameworks*. Tecnologias para integração com a camada de persistência: fontes de dados, *frameworks* para acesso direto e para mapeamento objeto-relacional.

**Bibliografia:**

A definir.

---

---

**INE5644 – Data Mining.**

**Carga Horária: 72 horas aulas.**

**Pré-requisito: INE5423.**

**Ementa:**

Considerações iniciais da área e aplicações. Processos de descoberta do conhecimento (KDD) em base de dados. Análise exploratória de dados. Tipos de variáveis, Técnicas de *Data Mining* para classificação, estimação, predição, análise de agrupamentos, análise de associação: redes neurais, árvores de decisão, regras de decisão, análise discriminante, regressão linear, regressão logística, análise de cluster, análise de componentes principais. Uso de softwares de *Data Mining*. Exercícios. *Data Mining* de caso.

**Bibliografia:**

A definir.

---

---

**INE5628 – Sistemas Multiagentes.**

**Carga Horária: 72 horas aulas.**

**Pré-requisito: INE5430.**

**Ementa:**

Propriedades de agentes. Arquitetura de construção de agentes. Linguagens para programação de agentes. Estudo de casos de agentes implementados.

**Bibliografia:**

A definir.

---

---

**INE5643 – *Datawarehouse*.**

**Carga Horária: 72 horas aulas.**

**Pré-requisito: INE5417.**

**Ementa:**

Os ecossistemas de informação. O ciclo de vida do DW: Planejamento e Administração, Levantamento de Requisitos, Modelagem Dimensional, Projeto Físico, o *Back-Room* e o *Front-Room*, Metadados, Arquiteturas, Implementações, Segurança e Internet. Implementação, Suporte e Treinamento.

**Bibliografia:**

A definir.



### 4.3. Distribuição das Disciplinas nas Fases Curriculares.

Neste item é apresentada a distribuição da estrutura curricular, nas oito fases que compõem o tempo mínimo necessário para a integralização curricular do curso de Ciência da Computação da UFSC.

Como já foi dito anteriormente, para a integralização curricular, o aluno deverá cursar os 172 créditos que corresponde ao total de disciplinas obrigatórias e 24 créditos de disciplinas optativas. Com relação às disciplinas optativas o aluno poderá cursar até oito créditos de disciplinas optativas de caráter geral oferecidas pela UFSC e pelo menos dezesseis créditos de disciplinas optativas eletivas escolhidas na tabela 4.3.9.

#### 4.3.1 – Primeira Fase – Total de Créditos: 23

<b>COD</b>	<b>NOME</b>	<b>CHS</b>	<b>CHT</b>	<b>PR</b>
EELXX	Circuitos e Técnicas Digitais	5	90	
INE5401	Introdução à Computação	2	36	
INE5402	Programação Orientada a Objetos I	6	108	
INE5403	Fundamentos de Matemática Discreta para a Computação	6	108	
MTM5161	Cálculo A	4	72	

### 4.3.2 – Segunda Fase – Total de Créditos: 27

<b>COD</b>	<b>NOME</b>	<b>CHS</b>	<b>CHT</b>	<b>PR</b>
INE5404	Programação Orientada a Objetos II	6	108	INE5402
INE5405	Probabilidade e Estatística	5	90	MTM5161
INE5406	Sistemas Digitais	5	90	EELXX
MTMXX	Cálculo B para Computação	4	72	MTM5161
MTM5512	Geometria Analítica	4	72	
INE5407	Ciência, Tecnologia e Sociedade	3	54	

### 4.3.3 – Terceira Fase – Total de Créditos: 24

<b>COD</b>	<b>NOME</b>	<b>CHS</b>	<b>CHT</b>	<b>PR</b>
INE5408	Estruturas de Dados	6	108	INE5404
INE5409	Cálculo Numérico para Computação	4	72	MTM5512 MTMXX
INE5410	Programação Concorrente	4	72	INE5404
INE5411	Organização de Computadores I	6	108	EELXX INE5406
MTM5245	Álgebra Linear	4	72	MTM5512

#### 4.3.4 – Quarta Fase – Total de Créditos: 26

<b>COD</b>	<b>NOME</b>	<b>CHS</b>	<b>CHT</b>	<b>PR</b>
INE5412	Sistemas Operacionais I	4	72	INE5410 INE5411
INE5413	Grafos	4	72	INE5403 INE5408
INE5414	Redes de Computadores I	4	72	INE5404
INE5415	Teoria da Computação	4	72	INE5403 INE5408
INE5416	Paradigmas de Programação	5	90	INE5408
INE5417	Engenharia de Software I	5	90	INE5408

#### 4.3.5 – Quinta Fase – Total de Créditos: 24

<b>COD</b>	<b>NOME</b>	<b>CHS</b>	<b>CHT</b>	<b>PR</b>
INE5418	Computação Distribuída	4	72	INE5414 INE5412
INE5419	Engenharia de Software II	4	72	INE5417
INE5420	Computação Gráfica	4	72	INE5408 MTM5512 MTMXX MTM5245
INE5421	Linguagens Formais e Compiladores	4	72	INE5408 INE5415
INE5422	Redes de Computadores II	4	72	INE5414
INE5423	Banco de Dados I	4	72	INE5408

#### 4.3.6 – Sexta Fase – Total de Créditos: 20

<b>COD</b>	<b>NOME</b>	<b>CHS</b>	<b>CHT</b>	<b>PR</b>
INE5424	Sistemas Operacionais II	4	72	INE5412
INE5425	Modelagem e Simulação	4	72	INE5405
INE5426	Construção de Compiladores	4	72	INE5421
INE5427	Planejamento e Gestão de Projetos	4	72	INE5419
INE5428	Informática e Sociedade	4	72	INE5407

#### 4.3.7 – Sétima Fase – Total de Créditos: 22

<b>COD</b>	<b>NOME</b>	<b>CHS</b>	<b>CHT</b>	<b>PR</b>
INE5429	Segurança em Computação	4	72	INE5403 INE5414 INE5415
INE5430	Inteligência Artificial	4	72	INE5405 INE5413 INE5416
INE5431	Sistemas Multimídia	4	72	INE5414
INE5432	Banco de Dados II	4	72	INE5423
INE5433	Trabalho de Conclusão de Curso I	6	108	INE5427
	Disciplina Optativa I			
	Disciplina Optativa II			

#### 4.3.8 – Oitava Fase – Total de Créditos: 6

<b>COD</b>	<b>NOME</b>	<b>CHS</b>	<b>CHT</b>	<b>PR</b>
INE5434	Trabalho de Conclusão de Curso II	06	108	INE5433
	Disciplina Optativa III			
	Disciplina Optativa IV			
	Disciplina Optativa V			
	Disciplina Optativa VI			

As disciplinas optativas, apresentadas anteriormente, estão resumidas na tabela a seguir com suas respectivas identidades, nomes, cargas horárias e pré-requisitos.

### 4.3.9 – Disciplinas Optativas.

<b>COD</b>	<b>NOME</b>	<b>CHS</b>	<b>CHT</b>	<b>PR</b>
INE5435	Integração Software/Hardware	4	72	INE5411
INE5436	Arquitetura de Computadores I	4	72	INE5411
INE5437	Arquitetura de Computadores II	4	72	INE5411
INE5438	Laboratório de Microprocessadores e Lógica Programável	4	72	INE5411
INE5439	Sistemas Embarcados	4	72	INE5411
INE5440	Tópicos Especiais em Arquitetura de Computadores	4	72	INE5411
INE5441	Sistemas de Tempo Real	4	72	INE5412
INE5442	Circuitos e Sistemas Integrados	4	72	INE5411
INE5443	Reconhecimento de Padrões	4	72	INE5420
INE5444	Estágio Supervisionado I	4	72	INE5417
INE5445	Estágio Supervisionado II	4	72	INE5417
INE5446	Tópicos Especiais em Sistemas de Infra-estrutura I	4	72	INE5412
INE5447	Tópicos Especiais em Sistemas de Infra-estrutura II	4	72	INE5412
INE5448	Tópicos Especiais em Aplicações Tecnológicas I	4	72	INE5412
INE5449	Tópicos Especiais em Aplicações Tecnológicas II	4	72	INE5412
INE5450	Tópicos Especiais em Aplicações Tecnológicas III	4	72	INE5412
INE5619	Administração e Gerência de Redes de Computadores	4	72	INE5422
INE5624	Engenharia da Usabilidade	4	72	INE5419
INE5640	Computação Móvel	4	72	INE5422
INE5646	Programação <i>Web</i>	4	72	INE5417
INE5644	<i>Data Mining</i>	4	72	INE5423
INE5628	Sistemas Multiagentes	4	72	INE5430
INE5643	<i>Datawarehouse</i>	4	72	INE5417

#### **4.4. Metodologia do Curso em Função do Perfil dos Egressos e do seu Papel na Sociedade.**

Neste item é descrito detalhadamente como o perfil dos egressos e seu papel na sociedade, propostos no projeto pedagógico, podem ser alcançados no encadeamento das disciplinas que compõem a estrutura curricular. Antes de abordar as disciplinas que compõem cada área de formação e os seus inter-relacionamentos, é importante tecer algumas considerações a respeito da disciplina Introdução à Computação que é oferecida no primeiro semestre. Um dos objetivos dessa disciplina é fornecer ao aluno ingressante uma visão ampla e detalhada do curso que o mesmo se propõe a realizar. A visão referida deve contemplar: o perfil do egresso desejado e o seu papel na sociedade e na sua carreira profissional como um cientista da computação; todas as áreas de conhecimento que compõem a estrutura curricular, juntamente com as justificativas da existência de cada uma.

Um segundo objetivo também muito importante e que releva a existência dessa disciplina é uma questão humanista. O professor dessa disciplina, preferivelmente o coordenado do curso, terá a incumbência de orientar e estimular os alunos, promovendo assim, uma adaptação mais suave dos mesmos ao curso que se inicia.

##### **4.4.1. Área de Formação Básica.**

Na área de formação básica em programação, o aluno é apresentado aos principais paradigmas de programação: orientado a objetos, funcional e lógico. Os paradigmas funcional e lógico são abordados em uma disciplina específica e retomados na disciplina de inteligência artificial, onde os alunos devem elaborar sistemas inteligentes usando essas tecnologias.

O curso de Ciências da Computação da UFSC foi um dos primeiros do Brasil a adotar o paradigma de orientação a objetos desde a sua primeira fase. Isso já vem acontecendo desde 1996, embora esta matéria seja vista no curso desde 1987. Metodologicamente, procura-se apresentar na disciplina de Programação Orientada a Objetos I os conceitos gerais de objeto, classe, mensagem, variáveis, instruções e controle, concentrando maior ênfase nos métodos dos objetos para que os alunos, ao final da disciplina, já saibam produzir programas tendo como base esse paradigma.

Na disciplina de Programação Orientada a Objetos II o aluno é levado a ampliar os conceitos, além de entrar em contato com a teoria relacionada a sistemas de tipos, acoplamento dinâmico, tratamento de exceções, programação por contrato etc (habilidade: Programação de Sistemas de forma geral, especialmente com o a utilização da perspectiva de modelagem orientada a objetos).

Na terceira fase, a disciplina de Estruturas de Dados apresenta em detalhes os algoritmos relacionados às estruturas que, na sua maioria os alunos já sabem manipular, como filas, árvores e dicionários, mostrando como implementar tabelas de *hash*, listas invertidas e outras estruturas, indicando também a complexidade dos algoritmos associados aos problemas de busca, inserção e ordenação nestas estruturas.

Na quarta fase, a disciplina de Grafos tem, além da característica teórica, forte ênfase na modelagem e implementação de sistemas com o uso da estrutura de grafo (habilidade: Modelagem de sistemas utilizando técnicas de orientação a objetos, estruturas de dados e grafos).

A área de arquitetura de sistemas de computação compreende um conjunto de seis disciplinas básicas obrigatórias (Circuitos e Técnicas Digitais, Sistemas Digitais, Organização de Computadores, Sistemas Operacionais I e II e Construção de Compiladores), de seis disciplinas complementares optativas (Arquitetura de Computadores I, Arquitetura de Computadores II, Sistemas Embarcados, Circuitos e Sistemas Integrados, Laboratório de Microprocessadores e Lógica Programável e Integração Software-hardware), além de uma disciplina de Tópicos Especiais para acomodar análise de tendências, temas emergentes ou avançados, técnicas inovadoras e promissoras na área de arquitetura de sistemas de computação.

As disciplinas Circuitos e Técnicas Digitais e Sistemas Digitais foram o resultado da adaptação e síntese do conteúdo clássico correlato na área de Engenharia (Eletrônica Geral, Eletrônica Digital), orientando-o às necessidades específicas da Computação e atualizando-o particularmente em relação aos requisitos da crescente importância da Computação Embarcada (por exemplo, baixo consumo de potência, uso de memórias FLASH, síntese lógica do circuito a partir de linguagens de descrição de hardware).

A disciplina Organização de Computadores aborda o conteúdo clássico de implementação contemporânea de processador, hierarquia de memória e interface com dispositivos de entrada e saída. Os fundamentos da técnica de *pipelining* e *cache* são



apresentados em profundidade, enquanto que conceitos como emissão múltipla e paralelismo entre instruções são apresentados de forma introdutória, uma vez que há uma disciplina optativa específica para abordá-los (Arquitetura de Computadores I).

A disciplina de Construção de Compiladores está fortemente relacionada à disciplina de Organização de Computadores, uma vez que a geração de código binário é dependente da arquitetura do conjunto de instruções e a otimização de código leva em conta a estrutura do *pipeline* (*code scheduling*), o mapeamento de instruções (*instruction selection*) e a estrutura da memória *cache* (*cache awareness*).

A disciplina de Sistemas Operacionais I representa o conteúdo clássico que aborda as rotinas que controlam e gerenciam a utilização de um computador de uso geral, tal como visto na disciplina Organização de Computadores. Especificamente, aborda a gerência e a coordenação de processos, a gerência de memória, a gerência de dispositivos de entrada e saída, e a gerência de arquivos. Por outro lado, a disciplina de Sistemas Operacionais II aborda metodologias de projeto orientadas a componentes, assim como teste e depuração de software básico, estudos de caso e prática em laboratório.

As disciplinas Arquitetura de Computadores I e II abordam conteúdos clássicos. Essencialmente, a primeira aborda máquinas com emissão múltipla (superescalares e VLIW), além de técnicas de exploração de paralelismo entre instruções. A segunda aborda multiprocessadores e multicomputadores, bem como técnicas de exploração de paralelismo entre *threads* e computação de alto desempenho.

A disciplina de Laboratório de Microprocessadores e Lógica Programável aplica conceitos obtidos em Organização de Computadores e Sistemas Operacionais I através de estudos de caso de sistemas embarcados prototipados com lógica programável.

A disciplina de Integração *Software-Hardware* estuda a interface entre o *hardware* e o *software* dependente do *hardware*, aplicando conceitos de Organização de Computadores e Sistemas Operacionais II para a elaboração de *drivers* e rotinas de inicialização e rotinas de serviço de interrupção. Esta disciplina ampara-se no paradigma tradicional de integração do software após o desenvolvimento do hardware na forma de lógica programável. Este paradigma é bastante difundido na indústria automotiva e de controle e automação.

O grupo de disciplinas formado por Sistemas Operacionais II, Laboratório de Microprocessadores e Lógica Programável, e Integração Software/Hardware tem como objetivo capacitar os alunos ao pleno desenvolvimento de software básico (bibliotecas de suporte à execução, *drivers* de dispositivos para sistemas operacionais e *firmware*) tanto para sistemas computacionais de uso geral como para o desenvolvimento de sistemas embarcados baseados em microcontroladores.

A disciplina de Sistemas Embarcados aborda os requisitos, desafios, plataforma de hardware, software dependente de hardware e metodologias para o suporte à Computação Embarcada, especialmente na forma de “*Systems-on-Chip*”. O foco da disciplina ampara-se em paradigmas contemporâneos: projeto orientado a plataforma, abstração em nível de sistema eletrônico (ESL) e projeto concorrente de hardware e software. Tais paradigmas são cruciais em aplicações na área de multimedia e sistemas *wireless*.

A disciplina de Circuitos e Sistemas Integrados apresenta conceitos e técnicas de Microeletrônica e Nanotecnologia. Esta disciplina representa importante insumo para a implementação de componentes de sistemas de computação de uso geral (CPUs e memórias) ou para suporte à Computação Embarcada, através da implementação de sistemas em um único circuito integrado (“*Systems-on-Chip*”).

O grupo de disciplinas formado por Sistemas Embarcados e Circuitos e Sistemas integrados tem como objetivo capacitar os alunos em co-projeto de hardware e software de “*Systems-on-Chip*”.

A área de matemática visa oferecer forte embasamento teórico e maturidade ao aluno, promovendo o desenvolvimento do seu raciocínio abstrato e fornecendo o ferramental necessário ao desenvolvimento de outras disciplinas do curso.

Cálculo A apresenta os fundamentos do cálculo: limite, derivada e integral, necessários em várias atividades do dia-a-dia, especialmente aquelas ligadas a estudos que envolvam métodos numéricos e computação científica.

As disciplinas de Álgebra Linear e Geometria Analítica oferecem as bases necessárias para aplicações tecnológicas como redes neurais, computação gráfica etc, além de valorizar o raciocínio abstrato do aluno.

A disciplina de Fundamentos Matemáticos da Informática visa dar os conhecimentos da lógica matemática essenciais para a

compreensão de vários conceitos da computação como estruturas de dados, grafos, programação em lógica, circuitos lógicos etc. Essa disciplina aborda tópicos que vão desde a teoria de conjuntos, números, recursão e indução, até as estruturas algébricas fundamentais para a compreensão dos sistemas de tipo e enumeração. O Cálculo B visa aprofundar as habilidades do aluno com métodos de integração e suas aplicações nos problemas do dia a dia.

A disciplina de Cálculo Numérico para a Computação trabalha com métodos para resolução computacional, utilizando conceitos dos cálculos, geometria analítica e álgebra linear, problemas que não permitem tratamento analítico ou simbólico. O aluno inicialmente é introduzido à idéia do sistema de ponto flutuante, a fim de compreender suas vantagens e limitações. Disto decorrem as noções de erros numéricos (de truncamento e arredondamento) e aproximações numéricas, que permeiam estes cursos durante a exposição de métodos que objetivam resolver eficientemente vários problemas como: raízes de equações, sistemas lineares e não-lineares, interpolação e ajuste de curvas, aproximação de funções, integração numérica, etc. A ênfase principal está em apresentar aos alunos métodos numéricos eficientes, com estimativa de erros (a fim de conhecer a qualidade da solução obtida), detalhamento dos algoritmos, número de FLOPS envolvidos (a fim de aferir o custo computacional), vantagens e desvantagens de cada método (quando devem ser usados e quando outros métodos seriam mais adequados) e implementação (programação em alguma linguagem). O que os alunos de Ciência da Computação precisam (pré-requisito) para as disciplinas de análise numérica é um bom conhecimento sobre os conceitos de limites, derivadas (ordinárias e parciais) e integrais (simples e múltiplas), e principalmente a idéia da expansão em série de Taylor (que aparece a todo o momento durante o curso - por exemplo: a série de Taylor multi-dimensional é a base para o entendimento dos métodos de Newton para sistemas não-lineares). Este material é visto nos Cálculos A e B.

A álgebra linear é importante no momento em que se fala sobre a solução numérica de sistemas lineares e não-lineares (na álgebra linear os alunos já estudaram o problema da existência e unicidade de soluções, normas matriciais e vetoriais - por exemplo, é através das normas que se mede a qualidade da solução obtida).

Outros conceitos fundamentais ao curso são: linearidade, independência linear, bases, etc. (Por exemplo, a interpolação de Lagrange usa uma base particularmente importante de polinômios, e o ajuste de curvas emprega funções linearmente independentes).

Em suma, ao cursarem a disciplina de cálculo numérico, os alunos aprendem vários métodos numéricos que visam resolver computacionalmente (com grande eficiência e exatidão) de forma aproximada uma variedade de problemas (já citados) que não podem ser resolvidos normalmente de forma simbólica (habilidade: Resolução de problemas relacionados à área de Análise Numérica).

As disciplinas de Probabilidade e Estatística bem como a de Modelagem e Simulação lançam as bases para a compreensão do método de amostragem e da combinatória. Tais conhecimentos são fundamentais tanto na experimentação científica (teste de hipóteses), quanto na realização de simulações de sistemas discretos (habilidades: Utilização de probabilidade e estatística aplicada e Construção de simuladores de propósito gerais).

A disciplina de Grafos, situando-se no limiar entre a matemática e a área de programação e estruturas de dados é um bom laboratório para trabalhar a questão de modelagem de problemas. A partir de problemas do mundo real, o aluno deve produzir modelos baseados em grafos, usando-os como estrutura de dados, podendo assim reutilizar algoritmos clássicos sobre grafos para resolver novos problemas, sem a necessidade de criar novos algoritmos, além de ser capaz de compreender os principais resultados teóricos desta área. Assim o aluno sai com formação teórica, mas também compreendendo a imensa utilidade destas estruturas para a modelagem e resolução de problemas.

A área de teoria da computação começa a ser vista com profundidade na quarta fase, na disciplina Teoria da Computação onde são introduzidos seus conceitos fundamentais através da apresentação dos modelos abstratos de computação como máquina de Turing, sistemas de Post ou cálculo lambda, e usando estes modelos para apresentar os conceitos relacionados aos limites da computação, ou seja, aos conceitos de computabilidade efetiva e também de complexidade (habilidade: identificação de problemas insolúveis ou NP-completos).

Na disciplina Linguagens Formais e Compiladores, quinta fase, são abordados os conceitos de linguagens, representações e sistemas de reescrita. Em especial, são aprofundados os conceitos de gramáticas do ponto de vista da hierarquia de Chomsky e seus impactos na decidibilidade e complexidade de análise das linguagens, além dos conceitos de autômato finito e autômato de pilha, importantes não só como modelos de computação, mas

também como analisadores de linguagem (habilidade: Identificação de problemas insolúveis e NP-Completo e proposta de soluções envolvendo aproximações (heurísticas)).

As disciplinas de Fundamentos Matemáticos da Informática e Grafos também contemplam aspectos da teoria da computação ao apresentar a fundamentação matemática para a compreensão dos modelos de computabilidade e complexidade. Assim, apesar desta área só começar a ser vista em profundidade a partir da quarta fase, os alunos entram em contato com conceitos básicos desde a primeira fase. Ao deparar com os conceitos de indução e recursão na disciplina de fundamentos matemáticos, os alunos são levados a ter uma primeira visão sobre funções computáveis, computabilidade e Máquina de Turing. Além disso, nas disciplinas de programação em lógica e funcional, vários conceitos fundamentais da teoria da computação são apresentados aos alunos. Já os conceitos relativos à matéria de análise de algoritmos são tratados especialmente nas disciplinas de Estruturas de Dados, Grafos e Teoria da Computação.

#### **4.4.2. Área de Formação Tecnológica.**

O curso de Ciência da Computação da UFSC busca uma autonomia profissional e intelectual do aluno de forma progressiva, habilitando o mesmo na superação dos desafios das renovações das condições do exercício profissional, da produção de conhecimento e do domínio das tecnologias computacionais. Assim sendo, as disciplinas que compõem essa área de formação foram agrupadas em duplas de maneira que, enquanto uma disciplina aborda os fundamentos e as estruturas de uma determinada tecnologia, a outra trabalha essa mesma tecnologia com mais profundidade envolvendo sempre que possível a realização de projetos relacionados com a mesma. Um outro aspecto positivo fruto dessa idéia de se criar duas disciplinas para uma mesma tecnologia é que, de certa forma, exige-se a existência de pelo menos dois docentes atuando na área correspondente. Esse fato é muito importante, pois obriga que os professores envolvidos estejam sempre muito bem articulados para que as duas disciplinas se ajustem e se complementem perfeitamente.

O ciclo tecnológico inicia na quarta fase, com as disciplinas de Sistemas Operacionais I e Redes de Computadores I. A disciplina de Sistemas Operacionais I, que como visto anteriormente, possui Organização de Computadores como pré-requisito, apresenta ao aluno os conceitos relacionados ao software que interfaceia a arquitetura com os aplicativos. Gerência de memória, de arquivos e de processos são conceitos fundamentais nesta disciplina (habilidade: Programação de componentes de sistemas operacionais). A disciplina, também obrigatória, de Sistemas Operacionais II tem caráter prático e leva o aluno a projetar e implementar um sistema operacional (ou partes dele) para uma máquina real.

O assunto referente à computação concorrente é preconizado na disciplina de Programação Concorrente onde são vistos e fixados os conceitos fundamentais desse assunto por meio de programação de sistemas dessa natureza. Assim sendo, essa disciplina trabalha o assunto sob a ótica de programação, possuindo para tanto uma considerável carga de aulas práticas onde são abordados os principais modelos de programação concorrente e também as técnicas de coordenação de processos e “*threads*” existentes. Fazem parte também do conteúdo dessa disciplina os assuntos relacionados com “*deadlock*” e suas formas de tratamento. A Rede de Petri é vista como uma ferramenta para verificação de sistemas concorrentes, com a qual, são especificados alguns sistemas simples. Em uma forma de “currículo invertido”, nas disciplinas posteriores, de Sistemas Operacionais I e Computação Distribuída os conceitos da computação concorrente são retomados, desta feita, sob o enfoque mais teórico e de suporte a esse paradigma.

Não existe na estrutura curricular do curso uma disciplina específica que aborde os assuntos relacionados com a especificação formal de sistemas computacionais. Entretanto, essa matéria encontra-se distribuída em três disciplinas curriculares que de alguma forma podem fazer uso desse formalismo. São elas: Programação Concorrente, Engenharia de Software II e Redes de Computadores II. Esse fato também ocorre com o assunto relacionado com Complexidade dos algoritmos que não é tratado em uma disciplina específica, mas integra as ementas das disciplinas de Estrutura de Dados e Grafos.

Na sétima fase é oferecida uma disciplina com o nome de Desenvolvimento de Sistemas para *Web*, cujo objetivo é integrar a

maioria das tecnologias vistas em disciplinas isoladas. Essa disciplina trabalha com sistemas envolvendo conjuntamente computação distribuída, redes e protocolos de comunicação, *frameworks*, banco de dados e suas interconexões etc. Essa disciplina tem como pré-requisito a disciplina de Computação Distribuída criada para suprir a lacuna que existia nessa área que se tornou tão importante na computação.

Ainda com relação às áreas de tecnologias, o aluno terá disciplinas optativas que poderá aprofundá-lo na área de sua preferência.

A matéria de compiladores, além de fundamentar o conhecimento sobre o funcionamento dos compiladores comerciais, é importante para que o aluno seja capaz de conceber e implementar novas linguagens de programação, tanto de propósito geral, quanto de propósito específico, integrando, por exemplo, a interface de determinados sistemas. Os conceitos de analisador léxico e sintático começam a ser tratados na disciplina de Linguagens Formais e Compiladores.

A disciplina de Construção de Compiladores (obrigatória) leva o aluno a realizar um trabalho prático de construção de um compilador completo, da análise léxica à geração e otimização de código. Além disso, características especiais das linguagens de programação e tópicos avançados são tratados nesta disciplina (habilidade: Desenvolvimento de compiladores para linguagens dedicadas ou de propósito geral).

A matéria de engenharia de software, por sua importância na formação do cientista da computação com o perfil descrito, é vista em profundidade por todos os alunos através de três disciplinas. A disciplina de Engenharia de Software I apresenta os conceitos e técnicas relacionadas com a engenharia de requisitos e análise e projeto orientados a objetos. Um método de análise e projeto é estudado em profundidade, bem como uma linguagem de especificação orientada a objetos (UML) é utilizada para representar as decisões de análise e projeto. Durante a disciplina também é desenvolvido um software orientado a objetos. Uma ferramenta CASE é utilizada para dar suporte à modelagem e projeto dos problemas modelados durante a disciplina. (habilidade: Análise de Sistemas utilizando técnicas modernas). A disciplina de Engenharia de Software II apresenta os conceitos relacionados ao processo de desenvolvimento de software de forma mais aprofundada e são conhecidas as diferentes visões sobre um

sistema, as metodologias de análise e projeto orientadas a objetos, teste de software, manutenção de software, modelos de ciclo de vida, engenharia Reversa, abordagens voltadas ao reuso de software, ambientes e ferramentas de apoio ao desenvolvimento de software, gerenciamento do processo de produção de software, técnicas de apoio ao gerenciamento do processo de produção de software. Ainda fazendo parte do conjunto de disciplina que cobrem essa matéria de engenharia de software, foi criada a disciplina de Gerência de Projetos onde são tratados assuntos relacionados com áreas de conhecimento da gerência de projetos: Escopo, Tempo, Risco, Integração, Comunicação, Custo, Recursos Humanos, Aquisição, Qualidade. Grupos de processos: Iniciação, Planejamento, Execução, Controle, Encerramento. Técnicas de acompanhamento de projetos. Ferramentas computacionais de apoio ao planejamento e gerência de projetos. Estudo de casos. (habilidades Elaboração de Projetos na área de Ciência da Computação). Como optativa da área é oferecida a disciplina de Tópicos avançados em Análise e Projeto de Sistemas Computadorizados, aprofundando as técnicas de análise e projeto orientadas a objetos.

A área de banco de dados conta com duas disciplinas obrigatórias, onde os conceitos fundamentais de sistemas relacionais e suas aplicações a bancos de dados são apresentados. O aluno deve ser capaz de identificar as formas normais, compreender os conceitos que fundamentam a linguagem SQL, como o cálculo relacional e as simplificações permitidas por este no projeto de bancos de dados. Aspectos de modelagem de sistemas também são abordados (habilidades: Produção de Software para gerenciamento de bancos de dados, Modelagem de bancos de dados). Com essas duas disciplinas o aluno cobre praticamente todo o assunto dessa área chegando a tratar de bancos de dados distribuídos e orientados a objetos.

A área de inteligência artificial é coberta em abrangência por uma disciplina que aborda os principais métodos dos paradigmas simbólico, evolucionário e conexionista (habilidade: Aplicação de técnicas de inteligência artificial em problemas correlatos a computação e/ou outras áreas).

A área de redes de computadores é tratada com profundidade a partir da quarta fase, através de duas disciplinas: Redes de Computadores I e II. Essas duas disciplinas de caráter teórico e



prático procuram cobrir praticamente todo o assunto da área de redes, mas que ainda pode ser complementado através de uma disciplina optativa nessa área.

A área de multimídia é coberta de forma abrangente pela disciplina obrigatória de Sistemas de Multimídia, de caráter teórico e prático (habilidade: Pesquisa e Desenvolvimento em sistemas multimídia).

A área de computação gráfica é coberta com profundidade em seus fundamentos pela disciplina obrigatória de Computação Gráfica, com forte ênfase nos algoritmos básicos de manipulação de objetos gráficos em três dimensões. Essa área ainda pode ser complementada através de uma disciplina optativa que aborda assuntos relacionados com reconhecimento de padrões (habilidade: Pesquisa e Desenvolvimento em computação gráfica).

Foi criada uma disciplina de caráter obrigatório que trata de um assunto que passou a ser de extrema importância para todas as áreas tecnológicas da computação, trata-se da disciplina de Segurança em Computação que aborda temas como: Segurança em redes de computadores: ataques e defesas. Princípios de criptografia: criptografia simétrica e assimétrica, integridade de dados. Protocolos de autenticação: princípios, padrões seguros (SSL, IPSec, S/MIME), estrutura para criptografia com chaves públicas (ICP, PGP). Segurança em sistemas operacionais: princípios de controle de acesso, sistemas confiáveis. Segurança em aplicações: detecção de falhas, códigos maliciosos (vírus e vermes) (habilidade: Pesquisa e Desenvolvimento em segurança de sistemas).

#### **4.4.3. Área de Formação Humanística.**

A formação humanística dos egressos do Curso de Ciência da Computação é tratada com seriedade. A inserção da informática em todos os setores da sociedade e na vida privada passou a exigir do profissional da área, uma qualificação que sai do tecnicismo e adentra nos valores pessoais agregados a este profissional. Valores considerados éticos, criatividade e habilidade nos relacionamentos humanos, são quesitos que não podem ser relegados à sorte ou ao

acaso. Os estudantes, jovens por excelência, estão neste período de formação universitária, consolidando e adquirindo valores sendo, portanto, responsabilidade da Instituição zelar para que estes valores sejam éticos, preparando-os para uma vida pessoal e profissional mais plena, tornando-os cidadãos mais construtivos capazes de agregar valores à sociedade.

Foram criadas duas disciplinas diretamente vinculadas à formação humanística dos egressos. Na segunda fase do curso, a disciplina de Ciência, Tecnologia e Sociedade aborda temas multidisciplinares. O objetivo é fornecer uma conexão entre o mundo acadêmico e futuro mundo profissional do estudante com o mundo real. Os temas abordados visam identificar problemas potenciais, considerar soluções alternativas e levar em consideração as conseqüências de soluções particulares. A principal preocupação é entender como a ciência e a tecnologia interagem mutuamente, modificando as sociedades e alterando aspectos culturais. Para isto os temas freqüentemente abordados em CTS são diferenças culturais, valores éticos e morais, sistemas econômicos e meios de produção e utilização dos recursos ambientais.

Também a formação de uma visão sócio-histórica é objetivada buscando uma análise não-ingênua da realidade. Procura destacar que a ciência e a tecnologia estão sempre a serviço de um sistema ideológico, que podem legitimar decisões eticamente questionáveis, dar poder de dominação a empresas e nações, subjugar pessoas, ou mesmo destruir nações ou culturas.

No contexto da computação, questões como propriedade intelectual, automatização, privacidade e liberdade civil, dependência tecnológica, dentre outras são temas importantes.

O profissional com uma visão de CTS tem um conhecimento potencial para ver a realidade com outros olhos, tomar decisões mais conscientes, analisar melhor as conseqüências de soluções ou produtos, bem como propor soluções preventivas e corretivas de problemas sociais e até mesmo econômicos.

A disciplina de Informática e Sociedade, oferecida posteriormente no sexto semestre, vai ampliar o desenvolvimento histórico da computação e analisar as interações do profissional com a sociedade do ponto de vista ético e legal, além de procurar incutir nos discentes a possibilidade de atuação profissional como empreendedores em qualquer posição e setor empresarial. Ela trata especificamente da ética pessoal, profissional e pública na área da informática. Dilemas éticos do profissional da informática.

Privacidade, vírus, "hacking", uso da Internet, direitos autorais etc. Desemprego e informatização. Responsabilidade social. O profissional e o mercado de trabalho. Trabalho e relações humanas. O empreendedorismo como opção do profissional da informática. Legislação: Política nacional e tendências atuais referentes a regulamentação da profissão.

#### **4.4.4. Área de Formação Complementar.**

A formação complementar do egresso é obtida através de disciplinas optativas das seguintes áreas: matemática, física, sistemas de informação e engenharias. Além disso, para formar o perfil do profissional técnico, mais capaz de administrar negócios na sua própria área a disciplina obrigatória de Informática e Sociedade procura dar a formação do empreendedor necessária para que o aluno tenha os conhecimentos básicos para criar e gerir seu próprio negócio. Para o aluno que pretenda criar uma empresa de informática o curso conta com um centro Geness, para incubar pequenas empresas, além da Empresa Júnior de projetos em informática.

#### **4.4.5. Flexibilização Curricular: Disciplinas de Estágio e Trabalho de Conclusão de Curso.**

Além de o curso objetivar a formação de profissionais altamente capacitados a geradores de inovação tecnológica nas diversas áreas da computação, ele se preocupa com o princípio da flexibilização curricular. Esse princípio procura inculcar no currículo uma maior fluidez e dinamização através do oferecimento de disciplinas optativas, de livre escolha do aluno, da existência de atividades complementares de caráter técnico-científico-culturais e de atividades de extensão.

Para tanto, como atividades complementares, o curso conta com disciplinas específicas para a elaboração de um projeto de grande porte, com orientação de docentes do curso. Este grupo de disciplinas inicia na sexta fase com as disciplinas de Gerência de Projetos, Metodologia Científica para a Computação e Introdução

ao Projeto em Ciência da Computação, onde as técnicas científicas são aprendidas e a proposta inicial de projeto é idealizada, respectivamente. Na sétima fase, a disciplina de Projeto em Ciência da Computação I visa à produção de uma monografia com um projeto propriamente dito, considerando a realização de uma revisão bibliográfica que garanta que o aluno está a par do estado da arte na área escolhida. Na oitava fase a disciplina de Projeto em Ciência da Computação II, trata da implantação efetiva do projeto proposto. O aluno deve defender o projeto realizado em público, no final da disciplina e ser aprovado por uma banca de três avaliadores, entre eles o orientador. Estes avaliadores acompanham o desenvolvimento do projeto desde a sua proposição inicial no final da sexta fase (habilidade: Elaboração de Projetos na área de Ciência da Computação).

Como atividades de extensão, o aluno pode contar com duas disciplinas optativas de estágio, oferecidas a partir do sexto semestre. Os estágios são supervisionados por um professor do departamento com duas incumbências importantes. A primeira de procurar selecionar estágios que possam agregar valores na formação do aluno. A segunda de acompanhar o desenvolvimento do aluno no trabalho do seu estágio, analisando e julgando os relatórios gerados.

Com relação às disciplinas optativas, como já foi visto, o curso procura oferecê-las sempre que em uma área se faça necessário para cobrir um assunto com mais profundidade ou mesmo para abordar um assunto novo. Dessa forma, o curso conta com dois tipos de disciplinas optativas: as de caracteres mais flexíveis, conhecidas como disciplinas de tópicos avançados, onde um assunto novo em uma determinada área possa ser abordado e as disciplinas optativas com ementas e nomes definidos.

À medida que uma disciplina de tópicos especiais se consagra em audiência e interesse, ela é incorporada ao currículo como optativa com ementa definida, recebendo código próprio. Além disso, o colegiado está sempre atento às questões curriculares. Grandes ou pequenas reformas curriculares serão freqüentemente realizadas para manter a atualidade e a qualidade do currículo.

## **5. Formas e Instrumentos de Avaliação do Processo de Ensino e da Aprendizagem.**

A avaliação do processo de ensino e da aprendizagem de cada disciplina, normalmente, inclui aspectos de caráter teórico e prático onde o aluno deverá desenvolver programas envolvendo a teoria e ou as tecnologias abordadas naquela disciplina. Os critérios de avaliação constam do programa de cada disciplina e podem ser restabelecidos, a cada novo semestre, pelo professor da disciplina, desde que aprovado pelo colegiado do curso. Os critérios devem ser apresentados aos alunos, juntamente com o conteúdo programático da disciplina, a cada início de semestre letivo, devendo ser criteriosamente respeitado durante aquele semestre.

## **6. Reajuste Estrutural Necessário para o Atendimento às Necessidades Colocadas no Projeto Pedagógico.**

Com relação à infra-estrutura o curso de Ciência da Computação da UFSC apresenta dois aspectos que devem ser prioritariamente reparados: o acervo bibliográfico e os laboratórios de ensino.

O acervo bibliográfico é reduzido e está desatualizado. Com a reforma curricular em curso, a medida em que os professores vão construindo os planos de ensino de suas disciplinas está sendo realizado um levantamento rigoroso das necessidades reais das bibliografias que será enviado aos órgãos institucionais competentes.

No tocante aos laboratórios de ensino, atualmente o curso vem utilizando laboratórios de forma compartilhada com todos os cursos de graduação do Centro Tecnológico. Com a criação de disciplinas da área de arquitetura e sistemas de infra-estrutura, torna-se imprescindível à criação de novos laboratórios específicos. Para o atendimento das disciplinas de Sistemas digitais, Organização de Computadores I e II, Laboratórios de Microprocessadores, Sistemas Operacionais I e II, seriam suficientes dois laboratórios com treze máquinas (dois alunos por máquinas) em espaços que permitam distanciamentos entre as

máquinas suficientes para a instalação de bancadas para kits de prototipação. O Departamento já conta com os espaços necessários.

Para as disciplinas de Sistemas digitais e Organização de Computadores I e II, os equipamentos necessários para montar as referidas bancadas são: 30 kits de prototipação ALTERA (NIOS-DEVKIT-PROMO-Hardware ou similar), 30 licenças de software para NIOS-DEVKIT-PROMO-Hardware ou similar.

Para a disciplina de Laboratórios de Microprocessadores três classes de kits são desejáveis: microcontroladores para sistemas profundamente embutidos (8-bits, KBs de memória); microcontroladores para sistemas embutidos complexos (32-bits, MBs de memória); FPGAs para sistemas sintetizáveis. Supondo uma demanda de 30 alunos por semestre, teríamos 5 kits de cada um dos três tipos. Estes kits seriam atualizados anualmente, introduzindo novas tendências, como por exemplo, redes sem fios, sensores inteligentes, atuadores de potência, entre outros.

As disciplinas de Redes de Computadores I e II, Banco de Dados I e II, Desenvolvimento de Sistemas para *Web* e Sistemas Distribuídos, também necessitam de um laboratório específico. Um laboratório contendo vinte máquinas seria suficiente para o pleno atendimento das disciplinas referidas. O Departamento também já conta com o espaço necessário para este laboratório.