

PLANO DE DISCIPLINA

CURSO: ENGENHARIA DE COMPUTAÇÃO		
DISCIPLINA: ECONOMIA	CÓDIGO: EC B - 633	PERÍODO: 6º
CARGA HORÁRIA: 40 ha = 33,3 h		
REVISÃO: 01/2019		
I – COMPETÊNCIAS		
Entender as noções de macroeconomia e matemática financeira; juros simples e juros compostos; entender um fluxo de caixa; análise de viabilidade econômica de projetos e investimentos; riscos; oportunidades; taxa mínima de atratividade; custo anual uniforme, VPL, TIR; amortização de empréstimos e financiamentos; amortização.		
II – HABILIDADES		
Capacitar o aluno a fazer análise de viabilidade econômica de projetos e investimentos, possibilitando o levantamento dos riscos e oportunidades envolvidos e, através de indicadores de resultados, escolher a melhor alternativa quanto aos retornos esperados.		
III – CONTEÚDOS PROGRAMÁTICOS		
Fundamentos de engenharia econômica; definições básicas de matemática financeira; taxa nominal e taxa efetiva de juros; juros simples e juros compostos; análise do valor presente e valor futuro; fluxo de caixa; diagrama de fluxo de caixa; análise de alternativas de investimentos; métodos de comparação de alternativas de investimento; análise da taxa de retorno e custo anual equivalente; análise do ponto de equilíbrio; métodos de depreciação.		
IV – METODOLOGIA		
A metodologia se baseia em aulas expositivas em classe e/ou laboratório, exercícios, pesquisas, projetos, estudo de casos e seminário visando a fixação dos conceitos apresentados.		
V – AVALIAÇÃO		
Será atribuída ao aluno uma nota bimestral, decorrente de uma avaliação formal e das avaliações realizadas ao longo do bimestre. No final do período semestral, será atribuída nota final, decorrente da média aritmética das notas bimestrais desse período. Para aprovação, a nota final, também denominada média final (MF) deverá ser \geq a 5,0 (cinco inteiros).		
VI – BIBLIOGRAFIA		
<p>Básica ASSAF NETO, A. Matemática Financeira e Suas Aplicações. 12ª ed. São Paulo: Atlas, 2012. VIEIRA SOBRINHO, J. D. Matemática financeira. 7ª ed. São Paulo: Atlas, 2002. FERREIRA, R. G. Engenharia Econômica e Avaliação de Projetos de Investimento. São Paulo: Atlas, 2010.</p> <p>Complementar SAMANEZ, C. P. Engenharia Econômica. São Paulo: Prentice Hall, 2009. (Acesso Virtual) PUCCINI, A. de L. Matemática financeira objetiva e aplicada. 6ª ed. São Paulo: Saraiva, 2003. PENIDO, E. Matemática financeira essencial. São Paulo: Atlas, 2008. GIMENES, C. M. Matemática Financeira com HP12 c e Excel. 2ª ed. São Paulo: Atlas, 2010. FEIJÓ, R. Matemática financeira com conceitos econômicos e cálculo diferencial: utilização da HP - 12C e planilha de Excel. São Paulo: Atlas, 2009.</p>		

PLANO DE DISCIPLINA

CURSO: ENGENHARIA DE COMPUTAÇÃO		
DISCIPLINA: CONTABILIDADE E CUSTOS	CÓDIGO: EC E - 634	PERÍODO: 6º
CARGA HORÁRIA: 40 ha = 33,3 h		
REVISÃO: 01/2019		
I – COMPETÊNCIAS		
Simulação da criação de um produto, revenda de mercadoria ou oferta de um serviço; simulação das demandas de produção, de estocagem e venda; simulação da análise dos pontos de equilíbrio de um negócio (PEC, PEE, PEF); simulação do fluxo do produto entre deptos (departamentalização). Projeção de custos em gradiente.		
II – HABILIDADES		
Capacitar o aluno para: identificar os elementos de custos presentes à operação da empresa; quantificar a participação dos custos do produto (variáveis) e da estrutura (fixos); formar o preço de venda considerando os diversos gastos e a competitividade; interpretar os as informações de custos e relacioná-las aos objetivos do negócio da empresa.		
III – CONTEÚDOS PROGRAMÁTICOS		
Gastos (investimentos, custos e despesas); classificação do custo em relação ao volume (fixos e variáveis); classificação dos custos em relação à alocação (diretos e indiretos); critérios de rateio e departamentalização; formação do preço de venda; pontos de equilíbrio (contábil, financeiro e econômico); análise da margem de contribuição e mix de produtos; alavancagem operacional e financeira; margem de segurança; sistemas de custeio; curva ABC; cálculos em gradiente.		
IV – METODOLOGIA		
Aulas expositivas; listas de exercitação; uso de planilhas eletrônicas de cálculo.		
V – AVALIAÇÃO		
Será atribuída ao aluno uma nota bimestral, decorrente de uma avaliação formal e das avaliações realizadas ao longo do bimestre. No final do período semestral, será atribuída nota final, decorrente da média aritmética das notas bimestrais desse período. Para aprovação, a nota final, também denominada média final (MF) deverá ser \geq a 5,0 (cinco inteiros).		
VI – BIBLIOGRAFIA		
Básica MARTINS, E. Contabilidade de Custos . 9ª ed. São Paulo: Atlas, 2003. SILVA, E. J.; GARBRECHT, G. T. Custos empresariais. Uma visão sistêmica do processo de gestão de uma empresa . Curitiba: InterSaber, 2016. (Acesso Virtual e Físico) SOUZA, A. Gestão de custos . São Paulo: Atlas, 2007.		
Complementar MEGLIORINI, E. Custos . São Paulo: Makron Books, 2003. (Acesso Virtual) PEREZ JR, J. H. Gestão Estratégica de Custos . 5ª ed. São Paulo: Atlas, 2008. SANTOS, L. F. B. Gestão de custos . Curitiba: InterSaber, 2013. (Acesso Virtual) COELHO, F. S. Formação estratégica de precificação: como maximizar o resultado das empresas . 2ª ed. São Paulo: Atlas, 2009. BRUNI, A. L. Gestão de custos e formação de preços com aplicação na calculadora HP 12C e Excel . 3ª ed. São Paulo: Atlas, 2004.		

PLANO DE DISCIPLINA

CURSO: ENGENHARIA DE COMPUTAÇÃO		
DISCIPLINA: ENGENHARIA DE SOFTWARE I	CÓDIGO: EC E - 635	PERÍODO: 6º
CARGA HORÁRIA: 80 ha = 66,7 h		
REVISÃO: 01/2019		
I – COMPETÊNCIAS		
Introduzir a disciplina de Engenharia de Software apresentando: o contexto histórico, a visão macro de processo de software e sua decomposição, o processo de análise de requisitos e suas técnicas de elicitação e documentação, e a análise orientada a objetos. Introdução ao design de software, arquitetura de software, teste de software e ferramentas CASE.		
II – HABILIDADES		
Introduzir a disciplina de Engenharia de Software apresentando: o contexto histórico, a visão macro de processo de software e sua decomposição, o processo de análise de requisitos e suas técnicas de elicitação e documentação, e a análise orientada a objetos. Apresentar os conceitos fundamentais da Engenharia de Software que resultam na produção de um software de alta qualidade. Evidenciar a relação entre arquitetura de software e qualidade de software. Estudar os recursos oferecidos pelas atuais ferramentas CASE.		
III – CONTEÚDO PROGRAMÁTICO		
Introdução à Engenharia de Software; a crise de software; as características do software; aspectos sóciotécnicos; processo de software; modelos de processo de software; atividades do processo de software; análise de requisitos; classificação de requisitos; técnicas de elicitação de requisitos; documentação de requisitos; análise orientada a objetos; introdução ao design de software; arquitetura de software; teste de software; tipos de teste de software; ferramentas CASE; tipos de ferramentas CASE.		
IV – METODOLOGIA		
A metodologia aplicada utiliza aulas teóricas expositivas mescladas com parte prática (aplicação dos conceitos apresentados) através de exercícios e elaboração de projeto (equipes de alunos) e ainda seminários apresentados pelos alunos.		
V – AVALIAÇÃO		
Será atribuída ao aluno uma nota bimestral, decorrente de uma avaliação formal e das avaliações realizadas ao longo do bimestre. No final do período semestral, será atribuída nota final, decorrente da média aritmética das notas bimestrais desse período. Para aprovação, a nota final, também denominada média final (MF) deverá ser ≥ a 5,0 (cinco inteiros)		
VI – BIBLIOGRAFIA		
Básica		
SOMMERVILLE, I. Engenharia de Software . 9ª ed. São Paulo: Pearson, 2011. (Acesso Virtual e Físico)		
PFLEEGER, S. L. Engenharia de Software: Teoria e Prática . 2ª ed. São Paulo: Pearson, 2004. (Acesso Virtual e Físico)		
PRESSMAN, R. S.; SANTOS, J. C. B. dos (TRADUTOR). Engenharia de software . São Paulo: Person Education, 2006.		
Complementar		
HIRAMA, K. Engenharia de software: qualidade e produtividade com tecnologia . Rio de Janeiro: Elsevier, 2011.		
PRESSMAN, R. S.; LOWE, D. Engenharia Web . Rio de Janeiro: LTC, 2009.		
GAVA, V. L. Requisitos de software e cooperação . São Paulo: Blucher Acadêmico, 2011.		
HIRAMA, K. Engenharia de software: qualidade e produtividade com tecnologia . Rio de Janeiro: Elsevier, 2011.		
DENNIS, A.; WIXOM, B. H.; GEINHART, M. (TRADUTOR). Análise e projeto de sistemas . 2ª ed. Rio de Janeiro: LTC, 2005.		
MACHADO, F. N. R. Análise relacional de sistemas . 2ª ed. São Paulo: Érica, 2001.		

PLANO DE DISCIPLINA

CURSO: ENGENHARIA DE COMPUTAÇÃO		
DISCIPLINA: CONTROLE E AUTOMAÇÃO	CÓDIGO: EC P - 636	PERÍODO: 6º
CARGA HORÁRIA: 80 ha = 66,7 h		
REVISÃO: 01/2019		
I – COMPETÊNCIAS		
<p>Conhecer ferramentas matemáticas para modelagem de sistemas físicos; associar grandezas físicas com modelos matemáticos; elaborar projetos de sistemas de controle; compreender o comportamento dinâmico de sistemas; compreender a importância e utilidade da teoria de controle, da classificação e propriedades dos diversos tipos de sistemas. Compreender a utilização de Controladores Lógicos Programáveis na Automação Industrial. Aplicar ações de controle do tipo PID.</p>		
II – HABILIDADES		
<p>Capacidade de analisar diversos tipos de sistemas e projetar sistemas de controle para a resolução de problemas de engenharia. Analisar sistemas físicos a partir dos respectivos modelos matemáticos; construir modelos de sistemas. Implementar soluções de automação industrial utilizando Controladores Lógicos Programáveis. Aplicar os diferentes tipos de ações de controle.</p>		
III – CONTEÚDOS PROGRAMÁTICOS		
<p>Introdução e histórico sobre sistemas de controle; exemplos de sistemas de controle; controle em malha aberta e em malha fechada; modelagem matemática de sistemas de controle; transformada de Laplace; função de transferência e função de impulso-reposta; Análise de sistemas de primeira ordem; Análise de sistemas de segunda ordem. Conceitos de automação industrial, Controlador Lógico Programável – Principais componentes; hardware e software; Aplicações do CLP; Softwares de CLP; Linguagem Ladder; Funções Lógicas Básicas; Comandos principais: Contatos; bobinas; set; reset; temporizador; contador; entradas e saídas analógicas. Ações de controle: ON-OFF, proporcional (P), proporcional integral (PI), proporcional derivativa (PD) e proporcional, integral derivativa (PID).</p>		
IV – METODOLOGIA		
<p>Aulas expositivas com recursos audiovisuais, aulas práticas no laboratório utilizando softwares para simulação, programação e utilizando hardwares como Controladores Lógicos Programáveis e Sistemas Microprocessados.</p>		
V – AVALIAÇÃO		
<p>Será atribuída ao aluno uma nota bimestral, decorrente de uma avaliação formal e das avaliações realizadas ao longo do bimestre. No final do período semestral, será atribuída nota final, decorrente da média aritmética das notas bimestrais desse período. Para aprovação, a nota final, também denominada média final (MF) deverá ser \geq a 5,0 (cinco inteiros).</p>		
VI – BIBLIOGRAFIA		
<p>Básica OGATA, K. Engenharia de Controle Moderno. 5ª ed. São Paulo: Pearson, 2011. (Acesso Virtual e Físico) NISE, N. S. Engenharia de Sistemas de Controle. 6ª ed. Rio de Janeiro: LTC, 2012. MORAES, Cícero Couto de; CASTRUCCI, Plínio de Lauro. Engenharia de automação industrial. 2ª ed. Rio de Janeiro: LTC, 2015.</p>		
<p>Complementar GEROMEL, J. C.; PALHARES, A. G. B. Análise Linear de Sistemas Dinâmicos. 2ª ed. São Paulo: Edgard Blucher, 2011. SIGHIERI, L.; NISHINARI, A. Controle automático de processos industriais: instrumentação. 2ª ed. São Paulo: Edgard Blucher, 2003. MAYA, P. A.; LEONARDI, F. Controle essencial. São Paulo: Pearson, 2014. (Acesso Virtual) OPPENHEIM, A. V.; SCHAFER, R. Processamento em tempo discreto de sinais. 3ª ed. São Paulo: Pearson, 2013. (Acesso Virtual) OPPENHEIM, A. V.; WILLSKY, A. S; NAWAB, S. H. (COLAB.). Sinais e Sistemas. 2ª ed. São Paulo: Pearson, 2010. (Acesso Virtual)</p>		

PLANO DE DISCIPLINA

CURSO: ENGENHARIA DA COMPUTAÇÃO		
DISCIPLINA: COMPILADORES	CÓDIGO: EC P - 637	PERÍODO: 6º
CARGA HORÁRIA: 80 ha = 66,7 h		
REVISÃO: 01/2019		
I – COMPETÊNCIAS		
Varredura (análise léxica); análise sintática descendente e ascendente; notação BNF; Yacc; análise semântica; projeto de um pequeno compilador; geração de código; otimização de código.		
II – HABILIDADES		
Executar os fundamentos na área de compilação de programas, através de abordagem teórica e prática. Apresentar conceitos relativos à Compilação. Conhecer e aplicar conceitos dos diversos tipos de análise de um processo de compilação. Estudar algumas linguagens de programação. Especificar uma linguagem de programação e seu processo de compilação.		
III – CONTEÚDO PROGRAMÁTICO		
Introdução à compilação de programas; conceito e funcionalidades; componentes de um compilador; fases de análise e síntese; análise léxica; tokens e lexemas; implementação manual e automatizada de analisadores léxicos; análise sintática; análise descendente (top down); análise sintática recursiva (determinística e não-determinística); análise sintática tabular; conceito de gramática LL(k); análise ascendente (bottom up); métodos LR(0), simple LR, LR(1) e LALR(1); conceito de gramática LR(k); geradores automáticos de analisadores sintáticos; análise semântica; introdução à semântica de linguagens de programação; verificação de tipos; tradução dirigida pela sintaxe; gramática de atributos; definição dirigida pela sintaxe; geração de código; formatos intermediários de código; mecanismos de tradução para código objeto; técnicas de otimização de código; ambientes de execução; carregadores, ligadores, bibliotecas de sistema; aspectos básicos sobre gerência de memória, concorrência, execução de procedimentos e métodos; interação com o subsistema: máquinas virtuais e sistema operacional; tópicos avançados e estudo de casos.		
IV – METODOLOGIA		
Aulas expositivas com auxílio de quadro branco, intercaladas com aulas de exercícios e laboratório, participação dos alunos de forma oral, escrita e seminário.		
V – AVALIAÇÃO		
Será atribuída ao aluno uma nota bimestral, decorrente de uma avaliação formal e das avaliações realizadas ao longo do bimestre. No final do período semestral, será atribuída nota final, decorrente da média aritmética das notas bimestrais desse período. Para aprovação, a nota final, também denominada média final (MF) deverá ser \geq a 5,0 (cinco inteiros)		
VI – BIBLIOGRAFIA		
Básica		
AHO, A. et al. Compiladores: princípios, técnicas e ferramentas . 2ª ed. São Paulo: Pearson, 2008. (Acesso Virtual)		
COOPER, K. D. Construindo Compiladores . São Paulo: Elsevier, 2014.		
HOPCROFT, J. E.; ULLMAN, J. D.; MOTWANI, R. Introdução à Teoria de Autômatos, linguagens e computação . 2ª ed. Rio de Janeiro: Elsevier, 2002.		
Complementar		
NETO, J. J. Introdução à compilação . São Paulo: Elsevier, 2016.		
MENEZES, P. B. Linguagens Formais e Autômatos . 6ª ed. Porto Alegre: Bookman, 2011.		
CHARLES, F. A Compiler . United States: Prentice Hall, 2009.		
LEUPERS, R. Retargetable Compiler Technology for Embedded Systems: Tools and Applications . United States: Springer, 2001.		
DIVERIO, T. A.; MENEZES, P. B. M. Teoria da Computação: Máquinas Universais e Computabilidade . Porto Alegre: Bookman, 2011.		

PLANO DE DISCIPLINA

CURSO: ENGENHARIA DA COMPUTAÇÃO		
DISCIPLINA: LINGUAGEM DE PROGRAMAÇÃO II	CÓDIGO: EC P - 638	PERÍODO: 6º
CARGA HORÁRIA: 80 ha = 66,7 h		
REVISÃO: 01/2019		
I – COMPETÊNCIAS		
Praticar e desenvolver programas com recursos mais avançados da linguagem orientada a objetos; oferecer o conhecimento e habilitar os alunos a serem capazes de desenvolver programas utilizando tipos e classes parametrizadas da linguagem OO; Utilizar os principais Design Patterns para programação, classes e métodos anônimos, conectividade remota, sockets e programação concorrente.		
II – HABILIDADES		
A disciplina tem o objetivo de demonstrar as práticas para desenvolver aplicativos com recursos mais avançados no paradigma orientado a objetos, como métodos e classes anônimas; Desenvolvimento de sistemas para utilizar os recursos de conectividades entre equipamentos e também a execução concorrente e paralela de processos. Aplicar os principais Design Patterns no desenvolvimento de aplicações. Manipulação de arquivos. E aplicações gráficas cliente servidor. Gerenciamento de código fonte com Git e GitHub.		
III – CONTEÚDO PROGRAMÁTICO		
Principais elementos e características da linguagem Java; Orientação a Objetos com Java; Java Collections; Genéricos em Java; AutoBoxing e Unboxing; Enumerações type-safe; Importação estática; Metadados (programação declarativa); Interfaces gráficas com o usuário; Canvas; Componentes JavaFX; Gerenciadores de layout; Manipulação de Eventos em Java; Eventos de Ação; Eventos do Mouse; Eventos do Teclado e Joystick; Classes Adaptadoras de Evento e Classes Internas; Desenvolvimento de Programas Concorrentes; Conceitos de Java Threads; Deadlocks e Starvation; Conectividade entre equipamentos; Conceitos de Sockets; Serviços REST/JSON e XML; Microserviços; Criação e consumo de API em nuvem; Conceitos de RMI/IIOP; Principais Design Patterns; Conectividade com JDBC, SQL e DAO; Classes e métodos anônimos. Eclipse IDE, banco de dados MySQL e Linux como ferramentas de desenvolvimento.		
IV – METODOLOGIA		
A metodologia é baseada em aulas teóricas expositivas mescladas com exemplos práticos em laboratório projetos e desafios em cada unidade do conteúdo programático, além de trabalhos práticos. Também será realizado um projeto interdisciplinar envolvendo a disciplina de Engenharia de Software, com o propósito de contextualizar o conteúdo programático favorecendo assim a efetiva aquisição de habilidades por parte dos alunos, mais próximo a uma situação real corporativa.		
V – AVALIAÇÃO		
Será atribuída ao aluno uma nota bimestral, decorrente de uma avaliação formal e das avaliações realizadas ao longo do bimestre. No final do período semestral, será atribuída nota final, decorrente da média aritmética das notas bimestrais desse período. Para aprovação, a nota final, também denominada média final (MF) deverá ser \geq a 5,0 (cinco inteiros)		
VI – BIBLIOGRAFIA		
Básica		
SIERRA, K. e BATES, B. Use a Cabeça! JAVA . 2ª ed. Rio de Janeiro: Alta Books, 2007.		
DEITEL, H. M.; DEITEL, P. J. Java como programar . 8ª ed. Porto Alegre: Bookman, 2010. (Acesso Virtual e Físico)		
FREEMAN, E.; FREEMAN, E. Use a Cabeça! Padrões de Projetos (design Patterns) . 2ª ed. Rio de Janeiro, Alta Books, 2007.		
Complementar		
FURGERI, S. Ensino didático da linguagem XML . São Paulo: Érica, 2001.		
TAMASSIA, R.; GOODRICH M. T. Estruturas de Dados e Algoritmos em Java . 2ª ed. Porto Alegre: Bookman, 2002.		
THOMPSON, M. A. Java2 & banco de dados . São Paulo: Érica, 2002.		
HORSTMANN, C. S.; CORNELL, G. Core Java . 8º ed. São Paulo: Pearson. 2009. Vol. 1. (Acesso Virtual)		
ASCENCIO, A. F. G.; ARAÚJO, G. S. de. Estrutura de Dados: algoritmos, análise da complexidade e implementações em Java e C/C++ . São Paulo: Pearson. 2011. (Acesso Virtual)		
JORGE, M. Java: passo a passo lite . São Paulo: Pearson, 2004. (Acesso Virtual)		
BARNES, DAVID J.; KÖLLING, MICHAEL. Programação orientada a objetos com Java . São Paulo: Pearson, 2004. (Acesso Virtual)		