

Plano de Disciplina: Desenvolvimento de Aplicações de Banco de Dados

Carga Horária: 80 horas

Ementa:

Desenvolvimento de rotinas de programação para implementação de regras de segurança ou regras de negócio aplicadas à camada do banco de dados. Linguagem Procedural para manipulação das instruções SQL e controle do banco de dados (PL/SQL). Fundamentos de Bancos de Dados. Arquitetura de Banco de Dados. Instruções SQL. Blocos armazenados e anônimos de instruções. Cursor. Tratamento de exceção. Procedimentos. Funções. Triggers, Packages.

Objetivos:

- Conhecer as instruções SQL.
- Conhecer as estruturas de programação de seleção e repetição.
- Conhecer os conceitos de bancos de dados.
- Classificar corretamente tipos de dados;
- Utilizar estruturas de seleção e repetição;
- Implementar regras de segurança e regras de negócio apropriadas ao contexto.
- Criar procedimentos e funções para resolução de problemas ou automatização de rotinas de validação de dados.
- Criar gatilhos para implementação de regras de negócio e de segurança
- Utilizar corretamente as instruções de programação aplicadas à camada do banco de dados.

Conteúdos:

- Fundamentos de banco de dados.
- Revisão de arquitetura de banco de dados.
- Revisão de Instruções SQL.
- PL/SQL – Definição
- Estrutura do bloco PL/SQL
- Variáveis
- Conceito
- Declaração
- Tipo de dados
- Corpo do programa
- Manipulação de tabelas (inclusão de linhas, consultas, alterações, exclusões)
- Blocos Anônimos
- Blocos Identificados
- Cursor
- Exceções
- Procedimentos
- Funções
- Pacotes
- Gatilhos

Metodologia de Ensino:

- Apresentação de modelos de Aplicações
- Aulas expositivas e dialógicas
- Leituras Complementares
- Atividades de laboratórios virtuais
- Resolução de

exercícios e problemas • Pesquisas e seminários • Jogos e experiências práticas de aplicação • Debates • Utilização de aplicativos, filmes, publicações, divulgação científica da web, livros.

Bibliografia Básica:

PUGA, S.; FRANÇA, E.; GOYA, M. Banco de dados: implementação em SQL, PL/SQL e Oracle 11g. Pearson: 2013. ELMASRI, R.; NAVATHE, S. B. Sistemas de banco de dados. 7.ed. Pearson: 2018. TERADA, R. Segurança de dados: criptografia em rede de computador. 2.ed. Blucher: 2008.

Bibliografia Complementar:

STALLINGS, W. Criptografia e segurança de redes. 6.ed. Pearson: 2015.
MEDEIROS, L. F. de. Banco de dados: princípios e prática. Intersaber: 2013.
VICCI, C. (Org.). Banco de dados. Pearson: 2014. ASCENCIO, A. F. G.; ARAUJO, G. S. de. Estrutura de dados: algoritmos, análise da complexidade e implementações em java e C/C++. Pearson: 2010. PUGA, S.; RISSETTI, G. Lógica de programação e estruturas de dados, com aplicações em java. Pearson: 2016.

Plano de Disciplina: Desenvolvimento Dinâmico

Carga Horária: 80 horas

Ementa:

Compreender o uso do Javascript para desenvolvimento de sites dinâmicos com os requisitos de um projeto de sistema. Compreender as principais sintaxes, operadores e instruções da linguagem JavaScript. Fazer manipulação de arrays e utilizar recursos de iteração. Implementar funções. Compreender o uso de objetos e a sua construção. Manipulação de arquivos formato JSON. Manipular elementos do DOM. Usar interatividade com eventos.

Objetivos:

- Criar elementos dinâmicos em uma página web utilizando JavaScript
- Criar uma página web com JavaScript
- Manipulação do DOM
- Aplicar eventos nos elementos de uma página web com JavaScript
- Aplicar validações nos elementos de uma página web com JavaScript
- Consumo de APIREST de terceiros

Conteúdos:

- 1) Introdução a JavaScript
 - a) Fundamentos
 - b) Notações
 - c) Variáveis
- 2) Função
 - a) Regular
 - b) Anônima
 - c) Arrow Function
- 3) Operadores
 - a) Operadores Aritméticos
 - b) Operadores de String
 - c) Operadores Lógicos
 - d) Operadores de Atribuição
- 4) Instruções
 - a) If
 - b) Switch
 - c) For
 - d) Do-while
 - e) While
 - f) Rótulos
- 5) String
 - a) slice
 - b) splice
 - c) substr
 - d) replace
 - e) concat
 - f) toLowerCase
 - g) toUpperCase
 - h) trim
 - i) split
- 6) JSON
 - a) Objetos
 - b) Manipulação JSON
- 7) Arrays
 - a) Criando e manipulando arrays
 - b) Praticando uso de Arrays
 - c) Métodos de Arrays
 - d) Filter
 - e) Map
 - f) reduce
 - g) forEach
- 8) Conceitos de Orientação a Objetos
 - a) Definindo classes
 - b) Propriedades de uma classe
 - c) Herança
 - d) Encapsulamento
- 9) Exception
 - a) Propriedades de uma classe
 - b) Prática com uso de exception
 - c) Encapsulamento
- 10) Promise
 - a) Conceito do uso
 - b) then(), catch(), finally()
 - c) Encadeamento
- 11) Elementos do DOM
 - a) Usando DOM
 - b) Encontrando elementos
 - c) Criando Elementos
 - d) Explorando o uso de Elementos
- 12) Eventos
 - a) eventos do mouse
 - b) eventos de teclado
 - c) eventos de botão
 - d) eventos gerais

Metodologia de Ensino:

Apresentação de modelos de Aplicações Aulas Ativas e dialógicas Leituras Complementares Atividades de laboratórios virtuais Resolução de exercícios e problemas

Bibliografia Básica:

FELIX, R. (Org.). Programação orientada a objetos. Pearson: 2016 DEITEL, P. J.; DEITEL, H. M. Ajax, rich internet applications e desenvolvimento web para programadores. Pearson: 2008 NEVES, M. C. B. de A. Sites de Alta Performance. Contentus: 2020

Bibliografia Complementar:

SEGURADO, V. S. (Org.). Projeto de interface com o usuário. Pearson: 2016 FLATSCHART, F.; BACHINI, C.; CUSIN, C. Open Web Platform. Brasport: 2013 PAGE-JONES, M. Fundamentos do desenho orientado a objeto com UML. Pearson: 2001 FERREIRA, R. D. Linguagem de programação. Contentus: 2020 PUGA, S.; RISSETTI, G. Lógica de programação e estruturas de dados, com aplicações em Java. Pearson: 2016

Plano de Disciplina: Desenvolvimento FrontEnd

Carga Horária: 80 horas

Ementa:

Compreender o uso do HTML para desenvolvimento de sites responsivo de acordo com os requisitos de um projeto de sistema. Compreender a semântica dos elementos HTM. Construção de site com HTML e CSS conforme os padrões W3C. Desenvolvimento de sites responsivos, dinâmicos e atraentes utilizando os componentes do BootStrap.

Objetivos:

Construção de sites responsivos Manipular elementos html Trabalhar com semântica Projetar sites no padrão W3C Conhecer motores de renderização Utilizar as principais tags do HTML Criar sites utilizando BootStrap Aplicar estilos CSS

Conteúdos:

1) Introdução a Internet a) O que é o W3C? b) Browser c) Motores 2) IDE de desenvolvimento a) Instalação b) Configuração 3) Elementos Básicos a) Estrutura de um código HTML b) Elementos e atributos c) Títulos, links, texto d) Cores e estilo 4) Lista e fontes a) Estilo de fontes b) Alinhamento de fontes c) Listas mais utilizadas d) Listas estruturadas 5) Tabelas a) Usando thead, tbody e tfoot b) Atributos de uma tabela c) Propriedades de uma tabela d) Aplicando o uso de tabelas 6) Formulários a) Campo input e textarea b) Campo Combobox, FieldSet c) Label, Hidden, E-mail e Data d) Tipos de Botões e) Metodos de envio de formulário f) Submissão formulário 7) Introdução ao BootStrap a) Entendo sobre GRID b) Criando um menu c) Criando textos d) Trabalhando com botões e) Trabalhando com imagens f) Usando Cards g) Trabalhando com Modal h) Explorando Tabelas i) Efeito Carousel j) Usando Forms k) Layout com base em outros 8) GitPage para montar portfolio

Metodologia de Ensino:

Aulas expositivas e dialógicas Leituras Complementares Atividades de laboratórios virtuais Resolução de exercícios e problemas Pesquisas e seminários Jogos e experiências práticas de aplicação Debates Utilização de aplicativos, filmes, publicações, divulgação científica da web, livros

Bibliografia Básica:

NEVES, M. C. B. de A. Sites de Alta Performance. Contentus: 2020 JOÃO, B. do N. (Org.). Informática aplicada. 2.ed. Pearson: 2019 SOUSA, R. F. M. CANVAS HTML 5 - Composição gráfica e interatividade na web. Brasport: 2018

Bibliografia Complementar:

BONATTI, D. Desenvolvimento de Sites Dinâmicos com Dreamweaver CC. Brasport: 2013 MARINHO, A. L.; CRUZ, J. L. da. Desenvolvimento de aplicações para Internet. 2.ed. Pearson: 2020 TANENBAUM, A. S.; FEAMSTER, N.; WETHERALL, D. J. Redes de computadores. 6.ed. Pearson: 2021 FLATSCHART, F. HTML 5 - Embarque Imediato. Brasport: 2011 BONATTI, D. Desenvolvimento de Jogos em HTML5. Brasport: 2014

Plano de Disciplina: Design de Software

Carga Horária: 80 horas

Ementa:

- Arquitetura de software e modelagem de sistemas de informação. Atividades de modelagem em um ciclo de vida e no processo de software. Técnicas de levantamento de dados, especificação de processos, requisitos funcionais e não funcionais de software e regras de negócio. Declaração do escopo do projeto de software. Abordagens de modelagem Orientada a Objetos. Projetos de software Orientados a Objetos. UML aplicado à modelagem O.O. Elaboração de cronograma do projeto de software para apoio à gestão do tempo, monitoramento e controle de tarefas. Gerenciamento de configuração e mudanças. - Modelagem de sistemas assistida por computador (CASE).

Objetivos:

- Fazer levantamento de requisitos - Usar ferramentas CASE - Modelar aplicações orientadas a objetos em UML a partir de uma especificação de requisitos. - Representar corretamente funcionalidades do sistema utilizando UML. - Documentar sistemas utilizando a UML. - Utilizar ferramentas para Modelagem OO.

Conteúdos:

1) Conceitos básicos de orientação a objetos a) Objetos b) Classes c) Atributos d) Métodos e) Estado, comportamento e identidade de um objeto f) Encapsulamento g) Dependência entre classes h) Associações: multiplicidade e naveabilidade i) Herança j) Polimorfismo k) Troca de mensagens entre objetos
2) Introdução ao UML a) Diagramas Estruturais b) Diagrama de objetos c) Diagrama de classes d) Diagrama de componentes e) Diagrama de instalação f) Diagrama de pacotes g) Diagrama de estrutura h) Diagramas Comportamentais i) Diagrama de Caso de Uso j) Diagrama de Estados k) Diagrama de atividade l) Diagramas de Interação m) Diagrama de sequência n) Diagrama de Interatividade o) Diagrama de colaboração p) Diagrama de tempo 3) Mapeamento Objeto x Relacional. 4) Ferramentas para Modelagem OO.

Metodologia de Ensino:

- Apresentação de modelos de Aplicações - Aulas expositivas e dialógicas - Leituras Complementares - Atividades de laboratórios virtuais - Resolução de exercícios e problemas - Pesquisas e seminários - Jogos e experiências práticas de aplicação - Debates - Utilização de aplicativos, filmes, publicações, divulgação científica da web, livros

Bibliografia Básica:

GALLOTTI, G. M. A. (Org.). Arquitetura de software. Pearson: 2017. SOMMERVILLE, I. Engenharia de software. 10.ed. Pearson: 2019 PFLEGER, S. L. Engenharia de software: teoria e prática. 2.ed. Pearson: 2003

Bibliografia Complementar:

GALLOTTI, G. M. A. Qualidade de software. Pearson: 2015 BRAGA, P. H. Teste de software. Pearson: 2016 MEDEIROS, E. Desenvolvendo software com UML 2.0 definitivo. Pearson: 2004 ASCENCIO, A. F. G.; CAMPOS, E. A. V. de. Fundamentos da programação de computadores. 3.ed. Pearson: 2012

Plano de Disciplina: Empreendedorismo e Criação de Novos Negócios

Carga Horária: 80 horas

Ementa:

Empreendedorismo: conceitos e definições. O Perfil e as características do empreendedor. As habilidade e competências necessárias aos empreendedores. A Importância do Empreendedorismo para uma sociedade. A identificação das oportunidades de negócios. Conceitos e definições sobre crises e oportunidades. Técnicas de identificação de oportunidades. Os recursos da Tecnologia da Informação na criação de novos negócios. Ferramentas e Planilhas na elaboração do Plano de Negócios. Empreendedorismo na era do Comércio Eletrônico. Startups, Aceleradoras, Elaboração do Plano de Negócio. Conceitos e definições. A estrutura do Plano de Negócio. Plano de Marketing. O Plano Financeiro. O Plano de Produção. Plano Jurídico.

Objetivos:

- Conhecer as formas jurídicas de empreender. - Compreender os conceitos relativos ao empreendedorismo; - Identificar oportunidades de negócios;
- Desenvolver o potencial visionário de um projeto ou negócio; - Conceituar empreendedorismo; - Caracterizar os tipos de empreendedores e de negócios;
- Desenvolver sua criatividade para negócios - Realizar análises financeiras e de mercado para uma proposta de negócio. - Elaborar um plano de negócios.
- Conhecer aceleradoras e seus papéis. - Conhecer os modelos de startups.
- Conhecer as formas de apoio para startups.

Conteúdos:

I – Empreendedorismo: - Conceitos - Definições - Modelos de Negócios existentes II - O perfil e as características dos empreendedores: - Ser motivado pela auto realização - Desejo de assumir responsabilidades - Desejo de ser independente e autônomo III- As atitudes necessárias aos empreendedores: - Busca de oportunidades - Correr riscos calculados - Exigência de qualidade e eficiência - Persistência e comprometimento - Estabelecimento de metas - Planejamento e monitoramento - Persuasão e rede de contatos - Independência

e autoconfiança IV - A identificação das oportunidades de negócios: - Conceitos e definições sobre crise e oportunidades; - Técnicas de Identificar oportunidades. V – Os Recursos da Tecnologia da Informação na criação de novos negócios: - Os softwares disponíveis no mercado no auxílio à criação de novas empresas; - Ferramentas e Planilhas na Elaboração do Plano de Negócio. VI - Conceitos e definições do Plano de Negócios: - A importância do Plano de Negócio - A estrutura do Plano de Negócio - O Plano de Marketing - O Plano Financeiro - O Plano de Produção e Jurídico VII – Apoio ao Empreendedor - Incubadora - Aceleradora - Querer crescer, apresentar ideias inovadoras e ser capaz de manter os gastos abaixo do esperado.

Metodologia de Ensino:

- Aulas expositivas e dialógicas - Leituras Complementares - Resolução de exercícios - Pesquisas e seminários - Jogos e experiências práticas de aplicação - Debates - Utilização de aplicativos, filmes, publicações, divulgação científica da web, livros - Estudos de Casos, Cenários e Atividades Nucleadoras (parte de projeto integrador)

Bibliografia Básica:

Santo Agostinho. Empreendedorismo. Pearson SERTEK, Paulo.
Empreendedorismo. Intersaber: 2013 POSSOLLI, Gabriela Eymg. Gestão da inovação e do conhecimento. Intersaber: 2012

Bibliografia Complementar:

RAZZOLINI FILHO, Edelvino. Empreendedorismo: dicas e planos de negócios para o século XXI. Intersaber: 2012 STADLER, Adriano (Org.); HALICKI, Zélia; ARANTES, Elaine Cristina. Empreendedorismo e Responsabilidade Social. Intersaber: 2014 CHIAVENATO, Idalberto. Empreendedorismo: Dando asas ao espírito empreendedor. 4. ed. Manole BRUNING, Camila; RASO, Cristiane Cecchin Monte; PAULA, Alessandra de. Comportamento organizacional e intraempreendedorismo. Intersaber: 2015 BIAGIO, Luiz Arnaldo. Empreendedorismo: construindo seu projeto de vida. Manole TEIXEIRA, Tarcisio; LOPES, Alan Moreira. Startups e inovação direito no empreendedorismo. Manole: 2017 SOUZA NETO, Bezamat de. Contribuição e elementos para um metamodelo empreendedor brasileiro o empreendedorismo de necessidade do virador. 2. ed. Blucher: 2017

Plano de Disciplina: Estruturas de Dados para Sistemas Inteligentes

Carga Horária: 80 horas

Ementa:

Manipulação de Dados. Estrutura de Dados. Vetores. Recursividade. Pilhas. Filas. Listas. Árvores. Implementação em Java.

Objetivos:

- Classificar as diferentes implementações de estruturas de dados de acordo com a implementação em disco ou em memória - Classificar os diferentes tipos de dados
- Compreender a implementação das estruturas de dados.
- Compreender as estruturas para manipulação de dados
- Identificar as diferentes estruturas de dados
- Propor soluções adequadas para a resolução de problemas
- Implementar as estruturas de dados utilizando a linguagem de programação Java
- Desenvolver soluções que utilizem corretamente as estruturas de dados
- Propor soluções para melhorar o desempenho de aplicações que manipulem dados.
- Escolher o melhor meio para manipulação de dados
- Manipular pilhas, filas, listas e árvores

Conteúdos:

- 1) Vetores e Matrizes
- 2) Métodos para Ordenação
- 3) Recursividade.
- 4) Pilhas
- 5) Conceito
- 6) Implementação de pilhas com o uso de vetores
- 7) Manipulação a)
Inserção de elementos
- b) Remoção de elementos
- c) Atualização do valor de um elemento
- d) Recuperação de um elemento
- 8) Filas
- a) Conceito
- b) Implementação de filas com o uso de vetores
- c) Manipulação
- i) Inserção de elementos
- ii) Remoção de elementos
- iii) Atualização do valor de um elemento
- iv) Recuperação de um elemento
- 9) Listas Ligadas
- a) Implementação de listas Simples com o uso de estruturas dinâmicas
- i) Manipulação
- (1) Inserção de elementos
- (2) Remoção de elementos
- (3) Atualização do valor de um elemento
- (4) Recuperação de um elemento
- b) Implementação de listas Duplas com o uso de estruturas dinâmicas
- i) Manipulação
- (1) Inserção de elementos
- (2) Remoção de elementos
- (3) Atualização do valor de um elemento
- (4) Recuperação de um elemento
- 10) Árvores
- a) Implementação de Árvores com o uso de estruturas dinâmicas
- b)

Manipulação i) Inserção de elementos ii) Remoção de elementos iii) Atualização do valor de um elemento iv) Recuperação de um elemento

Metodologia de Ensino:

- Apresentação de modelos de Aplicações - Aulas expositivas e dialógicas - Leituras Complementares - Atividades de laboratórios virtuais - Resolução de exercícios e problemas - Pesquisas e seminários - Jogos e experiências práticas de aplicação - Debates - Utilização de aplicativos, filmes, publicações, divulgação científica da web, livros

Bibliografia Básica:

PUGA, S.; RISSETTI, G. Logica de programação e estruturas de dados, com aplicações em Java. Pearson: 2016. ASCENCIO, A. F. G.; ARAUJO, G. S. de. Estrutura de dados: algoritmos, análise da complexidade e implementações em Java e C/C++. Pearson: 2010 FORBELLONE, A. L. V.; EBERSPACHER, H. F. Lógica de programação: a construção de algoritmos e estruturas de dados. 3.ed. Pearson: 2005

Bibliografia Complementar:

DEITEL, P.; DEITEL, H. Java: como programar. Pearson: 2016. BARNES, D. J.; KOLLING, M. Programação orientada a objetos com java: uma introdução prática usando o BlueJ. 4.ed. Pearson: 2009. ELMASRI, R.; NAVATHE, S. B. Sistemas de banco de dados. 4.ed. Pearson: 2005 TANENBAUM, A. S. Organização estruturada de computadores. 5.ed. Pearson: 2006

Plano de Disciplina: Experiência do Usuário

Carga Horária: 80 horas

Ementa:

Essa disciplina aborda a importância da Experiência do Usuário e Interface do Usuário na criação de interfaces digitais eficazes. Começando com conceitos fundamentais de UX, explora sua relação com UI e os diferentes tipos de UX no mercado. A ênfase na identidade visual, arquitetura de informação, usabilidade, heurísticas e acessibilidade enriquece a compreensão. Metodologias como Design Thinking e Design Sprint são exploradas, abrangendo fases de empatia, ideação, prototipagem e testes. A adaptação responsiva a dispositivos e a importância dos design systems também são destacadas. A disciplina faz uma introdução sobre as metodologias ágeis, como Scrum e Lean UX, e enfatiza a colaboração, métricas e acompanhamento contínuo no papel do profissional de UX/UI.

Objetivos:

- Compreender os Fundamentos de UX e UI: • Dominar Metodologias e Processos • Aplicar Conhecimentos de Design • Proporcionar aos alunos habilidades práticas em prototipagem, design responsivo e utilização de ferramentas de prototipagem • Incentivar a colaboração entre os alunos, bem como capacitar de entender a importância das métricas para avaliar o sucesso das interfaces desenvolvidas

Conteúdos:

1. EXPERIÊNCIA DO USUÁRIO (UX) • O que é UX? Por que é importante? • A relação entre UX e UI • Os tipos de UX no mercado de trabalho • As entregas de UX
2. INTERFACE DO USUÁRIO (UI) • O designer e as entregas de UI • As partes de uma interface digital • Dispositivos e responsividade • O design system
3. CONHECIMENTOS IMPORTANTES • Identidade visual • Arquitetura de informação • Usabilidade • Heurísticas • Acessibilidade
4. METODOLOGIAS DE DESIGN • Tipos de metodologias de design • O design thinking • Por que aplicar o design thinking? • Design Sprint
5. FASES DE EMPATIA E DE DEFINIÇÃO •

Pesquisas qualitativas e quantitativas • Entrevistas com usuários • Personas e Jornadas • Nosso exemplo: a pesquisa e a definição do e-commerce 6. FASE DE IDEAÇÃO • Benchmark • Moodboard • Uma técnica de criatividade • Nosso exemplo: a ideação para o site 7. FASE DE PROTOTIPAÇÃO • Alta e baixa fidelidade • Ferramentas de prototipação • A entrega do arquivo (handoff) • Nosso exemplo: o protótipo do site 8. FASE DE TESTES • Por que fazer testes com usuários • Teste de usabilidade • Nosso exemplo: testando com usuários • E depois dos testes? 9. METODOLOGIAS ÁGEIS • A metodologia ágil Scrum no mercado de trabalho • Lean UX e UX Ágil • Nosso exemplo: o projeto do site na metodologia ágil 10. O TRABALHO DO PROFISSIONAL DE UX E DESIGN • Vamos falar de métricas • A importância da colaboração • Acompanhamento contínuo • Nosso exemplo: conclusão e próximos passos

Metodologia de Ensino:

Apresentação de modelos de Aplicações Aulas Ativas e dialógicas Leituras Complementares Atividades de laboratórios virtuais Resolução de exercícios e problemas

Bibliografia Básica:

JOÃO, B. do N. (Org.). Usabilidade e interface homem-máquina. Pearson: 2017
SEGURADO, V. S. (Org.). Projeto de interface com o usuário. Pearson: 2017
BENYON, D. Interação humano-computador. 2. ed. Pearson: 2011

Bibliografia Complementar:

CHAK, A. Como criar sites persuasivos. Pearson: 2003 OLIVEIRA, I. de L.; SOARES, A. T. N. (Orgs.). Interfaces e tendências da comunicação no contexto das organizações. 2. ed. Difusão: 2019 SCHLITTLER, J. P. A. TV digital interativa: convergência das mídias e interfaces do usuário. Blucher: 2012 SOMMERVILLE, I. Engenharia de software. 8. ed. Pearson: 2007 LAWSON, B. Como arquitetos e designers pensam. Oficina de textos: 2011

Plano de Disciplina: Lógica de Programação

Carga Horária: 80 horas

Ementa:

Construção de algoritmos. Estruturas de controle de fluxo de execução: sequência, seleção e repetição. Tipos de dados básicos e estruturados (vetores e matrizes). Recursão, busca e ordenação. Pseudocódigo. Implementação de algoritmos usando uma linguagem de programação. Desenvolver programas básicos.

Objetivos:

- Criar algoritmos • Desenvolver pequenas aplicações • Fazer controle de fluxo de aplicações • Entender e utilizar recursão • Conhecer os principais algoritmos de busca • Implementar algoritmos • Executar algoritmos • Fazer identificação e correção de erros em algoritmos

Conteúdos:

1. Introdução a computação; a. Introdução a lógica de programação; i. Noções de lógica; ii. Algoritmos; b. Pseudocódigos e fluxogramas; c. Teste de mesa. 2. Elementos básicos de algoritmos a. Constantes, b. Variáveis simples e compostas; c. Tipos enumerados; 3. Comandos de entrada e saída; a. Expressões, b. Estruturas sequenciais e condicionais; 4. Estruturas de repetição; 5. Funções. 6. Linguagem de programação: a. Sintaxe da linguagem; b. Modularização: procedimentos e funções c. Passagem de parâmetros por valor e referência; d. Funções recursivas. e. Vetores, f. Matrizes, g. Registros e uniões; h. Busca sequencial e binária em vetores;

Metodologia de Ensino:

Apresentação de modelos de Aplicações Aulas expositivas e dialógicas Leituras Complementares Atividades de laboratórios virtuais Resolução de exercícios e problemas Pesquisas e seminários Jogos e experiências práticas de aplicação Debates Utilização de aplicativos, filmes, publicações, divulgação científica da

web, livros

Bibliografia Básica:

ASCENCIO, A. F. G.; ARAUJO, G. S. de. Estruturas de dados: algoritmos, análise da complexidade e implementações em Java e C/C++. Pearson: 2010 ARAUJO, S. de. Lógica de programação e algoritmos. Contentus: 2020 PUGA, S.; RISSETTI, G. Lógica de programação e estruturas de dados, com aplicações em Java. 3.ed. Pearson: 2016

Bibliografia Complementar:

ADAMI, A. G. Introdução à construção de algoritmos. Educs: 2009 FORBELLONE, A. L. V.; EBERSPÄCHER, H. F. Lógica de programação: a construção de algoritmos e estruturas de dados com aplicações em Python. 4.ed. Pearson: 2022 ASCENCIO, A. F. G.; CAMPOS, E. A. V. de. Fundamentos da programação de computadores: algoritmos, PASCAL, C/C++ e JAVA. 2.ed. Pearson: 2007 SOUZA, S. G. de. Lógica de programação algorítmica. Pearson: 2014 BORIN, V. P. Estrutura de dados. Contentus: 2020

Plano de Disciplina: Prática Integradora Desenvolvimento de Apps

Carga Horária: 72 horas

Ementa:

- Disciplina que integra os conteúdos vistos no módulo de Smart Data Structures e Mobile Developer para desenvolver as competências e habilidades necessárias para a construção de Apps e utilização de metodologias ágeis para desenvolvimento.

Objetivos:

- Desenvolver apps • Fazer identificação e correção de erros em apps • Fazer levantamento de requisitos • Fazer integração com Banco de Dados • Utilizar metodologias ágeis de desenvolvimento de sistemas

Conteúdos:

Cada prática deve ser desenvolvida em duas aulas. 1) Prática de Design Thinking para a criação de um App 2) Prática de Levantamento de Requisitos para Apps 3) Prática de uso de kanban 4) Prática de Usabilidade e Navegabilidade de Interface Gráfica 5) Prática de Integração com Banco de Dados 6) Prática de Programação do App 7) Prática de Tratamento de Exceção

Metodologia de Ensino:

- Apresentação de modelos de Aplicações • Aulas expositivas e dialógicas • Leituras Complementares • Atividades de laboratórios virtuais • Resolução de exercícios e problemas • Pesquisas e seminários • Jogos e experiências práticas de aplicação • Debates • Utilização de aplicativos, filmes, publicações, divulgação científica da web, livros

Bibliografia Básica:

BOYLESTAD, R. L.; NASHESKY, L. Dispositivos eletrônicos e teoria de circuitos. 11.ed. Pearson: 2013. LEE, V.; SCHNEIDER, H.; SCHELL, R. Aplicações móveis:

arquitetura, projeto e desenvolvimento. Pearson: 2005. SILVA, D. (Org.) Desenvolvimento para dispositivos móveis. Pearson: 2017.

Bibliografia Complementar:

DEITEL, P. J.; DEITEL, H. M. Ajax, rich internet Applications e desenvolvimento web para programadores. Pearson: 2008. DUARTE, W. Delphi para Android e iOS: desenvolvendo aplicativos móveis. Brasport: 2015. FELIX, R.; SILVA, E. L. da. Arquitetura para computação móvel. 2.ed. Pearson: 2019 MARINHO, A. L.; CRUZ, J. L. da. Desenvolvimento de aplicações para internet. 2.ed. Pearson: 2019. MOLETTA, A. Você na tela: criação audiovisual para a internet. Summus: 2019.

Plano de Disciplina: Programação I

Carga Horária: 80 horas

Ementa:

Programação Orientada a Objetos. Paradigmas de linguagens de programação. O paradigma de orientação a objetos. Estudo de uma linguagem de programação orientada a objetos. Estrutura da linguagem. Tipos de dados. Manipulação de strings, vetores e coleções. Tipos parametrizados. Tratamento de exceções. Manipulação de arquivos. Acesso a banco de dados. Padrões de projeto para persistência de dados.

Objetivos:

- Criar algoritmos e programas orientadas à objeto
- Conhecer os principais paradigmas de linguagens de programação
- Manipular arquivos
- Implementar corretamente Classes.
- Desenvolver rotinas para tratamento de erros no programa.
- Integrar aplicativos a bancos de dados.
- Conhecer os conceitos de Programação Orientada a Objetos.
- Identificar e declarar corretamente variáveis.
- Classificar variáveis de acordo com os tipos de dados.
- Identificar objetos, atributos e métodos.
- Projetar soluções adequadas à resolução de problemas computacionais

Conteúdos:

1. Introdução à Linguagem Orientada a objeto a. Ambientes de desenvolvimento
- b. Arquivo-fonte, arquivo compilado e empacotamento
2. Tipos Primitivos de Dados
3. Literais
4. Classe String
5. Entrada e Saída de Dados
6. Documentação do código
7. Objetos, Classes, Métodos e Atributos
- a. Objetos e Classes
- b. Características e Comportamento de objetos
- c. Encapsulamento
- d. Visibilidade
- e. Chamadas de métodos (mensagens)
8. Métodos
- a. Assinatura
- b. Nomenclatura
- c. Parâmetros
- d. Passagem de parâmetros por valor e por referência
- e. Retorno de métodos
- f. Chamada de métodos
- g. Visibilidade
- h. Recursividade
- i. Sobrecarga de métodos
9. Exceções
- a. Capturando uma exceção
- b. Capturando múltiplas exceções
- c. Ordem da captura
- d. Criando novas exceções
10. Tipos agregados homogêneos
- a. Tipos agregados homogêneos simples
- b. Tipos agregados

homogêneos bidimensionais c. Tipos agregados homogêneos multidimensionais d. Tipos agregados homogêneos multidimensionais irregulares e. Tamanho de agregações f. Varredura de tipos agregados simples g. Varredura de tipos agregados multidimensionais (inclusive irregulares) 11. Herança a. Herança Simples b. Classes Abstratas c. Interfaces d. Classes Anônimas e. Heranças Múltiplas f. Sobreposição de métodos g. Chamada a construtores da super-classe 12. Composições a. Conceito de composição b. Definindo composições c. Tipos agregados heterogêneos d. Acessando partes do objeto 13. Interfaces Gráficas 14. Modelo de Eventos 15. Manipulação de arquivos 16. Acesso a banco de dados 17. Serialização de objetos a. Armazenando objetos em disco

Metodologia de Ensino:

- Apresentação de modelos de Aplicações • Aulas expositivas e dialógicas • Leituras Complementares • Atividades de laboratórios virtuais • Resolução de exercícios e problemas • Pesquisas e seminários • Jogos e experiências práticas de aplicação • Debates • Utilização de aplicativos, filmes, publicações, divulgação científica da web, livros

Bibliografia Básica:

FELIX, R. (Org.). Programação orientada a objetos. Pearson: 2017 BARNES, D. J.; KOLLING, M. Programação orientada a objetos com java: uma introdução prática usando o bluej. 4.ed. Pearson: 2009 FERREIRA, R. D. Linguagem de programação. Contentus: 2020

Bibliografia Complementar:

SINTES, A. Aprenda programação orientada a objetos em 21 dias. Pearson: 2002. FORBELLONE, A. L. V; EBERSPACHER, H. F. Lógica de programação: a construção de algoritmos e estruturas de dados. 3.ed. Pearson: 2005 ASCENCIO, A. F. G.; ARAUJO, G. S. de. Estrutura de dados: algoritmos, análise da complexidade e implementações em java e c/c++. Pearson: 2010. SINTES, A. Aprenda programação orientada a objetos em 21 dias. Pearson: 2002 MEDEIROS, L. F. de. Banco de dados: princípios e prática. Intersaber: 2013.