

ESTRUTURA CURRICULAR

Coordenador: Prof. Dra. Sofia Larissa da Costa Paiva

Sub-Coordenador: Prof. Dr. Jacson Rodrigues Barbosa

Disciplinas	Carga Horária			Docente Responsável e Participantes	Titulação	IES onde está atuando
	T	P	Total			
Gerenciamento Ágil de Projeto	20	0	20	Juliano Lopes de Oliveira	Dr.	INF-UFG
Fundamentos de Orientação a Objetos com Java	10	30	40	Carlos Henrique Rorato Souza	Me.	INF-UFG
Experiência de Usuário - Fundamentos de IHC / Design e Usabilidade (UI/UX)	20	0	20	Jacson Rodrigues Barbosa	Dr.	INF-UFG
Fundamentos de Programação Back-End	10	30	40	Vinicius Sebba Patto	Dr.	INF-UFG
Projeto de Software	20	0	20	Edmundo Sérgio Spoto	Dr.	INF-UFG
Programação Back-End Avançada	10	30	40	Paulo Roberto Batista Júnior	Esp	INF-UFG
Fundamentos de Computação Móvel e Ubíqua	20	0	20	Iwens Gervásio Sene Júnior	Dr.	INF-UFG
Fundamentos de Programação Front-End	10	30	40	Jacson Rodrigues Barbosa	Dr.	INF-UFG
Programação Front-End Avançada	10	35	45	Daniel Ferreira Monteiro Alves	Me.	PUC-GO
Fundamentos de Programação Mobile com Android	8	22	30	Sandro Silva Moreira	Me.	UNIRV
Programação Mobile com Android Avançada	8	22	30	Sandro Silva Moreira	Me.	UNIRV
Programação Mobile com iOS	10	35	45	Silvano Maneck Malfatti	Me.	FACTO

DISCIPLINAS

1. Nome da Disciplina: Gerenciamento Ágil de Projeto

Carga horária: 20 horas

Docente responsável: Juliano Lopes de Oliveira

Ementa:

Manifesto ágil; métodos e frameworks ágeis; iniciação de projetos ágeis; planejamento de projetos ágeis; execução de projetos ágeis; monitoramento e controle de projetos ágeis; e melhoria contínua.

Metodologia:

Aulas teóricas expositivas para apresentação do conhecimento relevante e exercícios e dinâmicas para fixação do conhecimento. O conhecimento deve ser aplicado em disciplinas de cunho prático e no projeto final de curso. Um estudo de caso deverá ser definido no início do curso para se desenvolver trabalhos das demais disciplinas.

Bibliografia básica:

Rafael Priklandniki, Renato Willi, Fabiano Milani. *Métodos ágeis para desenvolvimento de Software*. Ed Bokman, 2014.
Project Management Institute, *A Guide to the Project Management Body of Knowledge*, Fifth Edition, 2015.
IEEE Computer Society, *Guide to the Software Engineering Body of Knowledge*, Version 3.0, 2014.

2. Nome da Disciplina: Fundamentos de Orientação a Objetos com Java

Carga horária: 40 horas

Docente responsável: Carlos Henrique Rorato Souza

Ementa:

Apresentação do paradigma de programação orientado a objetos (classes, objetos, herança, polimorfismo, encapsulamento, acoplamento, etc) e noções de UML (principais diagramas). Apresentação da linguagem de programação Java (sintaxe, atributos, métodos, construtores, sobrecarga, sobrescrita, exceções, responsabilidades e implementação dos conceitos de OO, etc) e ferramentas básicas (IDE, Controle de Versão, JUnit, etc.) empregadas no processo de desenvolvimento em Java. Teste de unidades, compilação, depuração, refatoração, e execução de programas em Java.

Metodologia:

Pequeno percentual de horas aulas teóricas expositivas para apresentação do conhecimento relevante e ênfase para realização de exercícios e atividades práticas em laboratório para aprendizagem e fixação dos conhecimentos sobre Orientação a Objetos com linguagem de programação Java. O conhecimento deve ser aplicado para modelar elementos e fundamentar o desenvolvimento de projeto de software ou de sistemas de informação pelo paradigma de orientação a objetos.

Bibliografia básica:

Grady Booch, James Rumbaugh & James Rumbaugh. *The Unified Modeling Language User Guide*, 2nd Edition, 2005.
Oracle, Java Tutorials, 2016. Disponível em <<http://docs.oracle.com/javase/tutorial/>>
Oracle, Java Platform Standard Edition 7 Documentation, 2016. Disponível em <<http://docs.oracle.com/javase/7/docs/index.html>>
Oracle, Java Platform, Enterprise Edition (Java EE) 7 Documentation, 2016. Disponível em <<http://docs.oracle.com/javaee/7/index.html>>
Pragmatic Unit Testing in Java with Junit Andrew Hunt & David Thomas, Pragmatic Bookshelf, 2003
Pragmatic Version Control Using Subversion Mike Mason Pragmatic Bookshelf, 2005

Git Reference Documentation, 2016. Disponível em
<<https://git-scm.com/documentation>>

3. Nome da Disciplina: Experiência de Usuário: Fundamentos de IHC, Design e Usabilidade
Carga horária: 20 horas

Docente responsável: Jacson Rodrigues Barbosa

Ementa:

Fundamentos de UX: experiência do usuário, design de interação, arquitetura de informação e usabilidade. Métodos, Técnicas e Ferramentas de UX Design: design centrado no usuário, técnicas para pesquisa com usuários. Validação de UX. Métodos ágeis focados em UX: Agile UX, Lean UX e Design Sprint.

Metodologia:

Aulas teóricas expositivas para apresentação do conhecimento relevante e exercícios e dinâmicas para fixação do conhecimento. O conhecimento deve ser aplicado em disciplinas de cunho prático e no projeto final de curso. O estudo de caso definido no início do curso deverá ser utilizado para desenvolver os trabalhos da disciplina.

Bibliografia básica:

Tidwell J., *Designing Interfaces: Patterns for Effective Interaction Design*. O'Reilly Media, 2005.
Weinschenk S., *100 Things Every Designer Needs to Know About People*. New Riders Publishing Thousand Oaks, CA, EUA, 2011.
Gothelf J., Seiden J., *Lean UX. Applying Lean Principles to Improve User Experience*, O'Reilly, 2012.

4. Nome da Disciplina: Fundamentos de Programação Back-End

Carga horária: 40 horas

Docente responsável: Vinícius Sebba Patto

Ementa:

Apresentação de protocolos HTTP e HTTPS, principais métodos e HTML. Apresentação de Servlets e Java Server Pages (JSP). Implementação de operações CRUD com JDBC e Framework JPA (Java Persistence API) com implementação Hibernate, utilizando mapeamento OR (Objeto Relacional) e controle de transações.

Metodologia:

Pequeno percentual de horas aulas teóricas expositivas para apresentação do conhecimento relevante e ênfase para realização de exercícios e atividades práticas em laboratório para aprendizagem e fixação dos conhecimentos importantes sobre programação back-end. O conhecimento deve ser aplicado para modelar e implementar elementos que executam e são persistidos em servidores de aplicações ou sistemas de informação, conforme estudo de caso definido no início do curso.

Bibliografia básica:

Oracle, Java Platform, Enterprise Edition (Java EE) 7 Documentation, 2016. Disponível em <<http://docs.oracle.com/javaee/7/index.html>>
Hibernate Reference Documentation, Versão 5.5.4, 2016. Disponível em <<https://docs.jboss.org/hibernate/stable/search/reference/en-US/html/>>

5. Nome da Disciplina: Projeto de Software

Carga horária: 20 horas

Docente responsável: Edmundo Sérgio Spoto

Ementa:

Fundamentos de design de software. Questões básicas de design de software: concorrência, controle e tratamento de eventos, persistência de dados, distribuição, tratamento de erro e exceção, tolerância a falhas, interação e apresentação, e segurança. Estrutura e arquitetura de software. Projeto de interface de usuário. Análise de qualidade e avaliação de design de software. Notação de design de software. Métodos e estratégias de design de software. Ferramentas de design de software.

Metodologia:

Aulas teóricas expositivas para apresentação do conhecimento relevante e exercícios e dinâmicas para fixação do conhecimento. O conhecimento deve ser aplicado em disciplinas de cunho prático e no projeto final de curso. Um estudo de caso deverá ser definido no início do curso para se desenvolver trabalhos das demais disciplinas. O estudo de caso definido no início do curso deverá ser utilizado para desenvolver os trabalhos da disciplina.

Bibliografia básica:

Budgen, B. *Software Design*. 2. Ed., Addison-Wesley, 2003.
Bass, L. et al. *Software Architecture in Practice*. 3. Ed., Addison-Wesley, 2012.
Larman, C. *Utilizando UML e padrões: uma introdução à análise e ao projeto orientados a objetos e ao desenvolvimento iterativo*. Bookman, 2008.
Head First Object-Oriented Analysis and Design, Brett D. McLaughlin et all, O'Reilly, 2008.
Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and the Unified Process. Craig Larman, Prentice-Hall, 2002.

6. Nome da Disciplina: Programação Back-End Avançada

Carga horária: 40 horas

Docente responsável: Paulo Roberto Batista Júnior

Ementa:

Integrações entre back-end e frameworks full-stack: Node, organização de projetos, I/O Assíncrono, Promises, Módulos. Ferramentas para controle de pacotes, repositórios NPM. Servidores, webservices e criação de APIs. Implementação de operações CRUD com banco de dados NoSQL e implementação com MongoDB.

Metodologia:

Pequeno percentual de horas aulas teóricas expositivas para apresentação do conhecimento relevante e ênfase para realização de exercícios e atividades práticas em laboratório para aprendizagem e fixação dos conhecimentos importantes sobre programação back-end. O conhecimento deve ser aplicado para modelar e implementar elementos que executam e são persistidos em servidores de aplicações ou sistemas de informação. O estudo de caso definido no início do curso deverá ser utilizado para desenvolver os trabalhos da disciplina.

Bibliografia básica:

Node JS API Documentation, 2017. Disponível em <<https://nodejs.org/api/index.html>>
Node JS Application Developer's Guide, 2017. Disponível em <<https://docs.marklogic.com/guide/node-dev/>>
Angular JS Tutorials, 2017. Disponível em <<https://docs.angularjs.org/tutorial>>
Leonard Richardson, Sam Ruby. *RESTful Web Services - Web services for the real world*. O'Reilly Media, 2007.
Glenn Hostetler, Sandor Hasznos. *Web Service and SOA Technologies*. Practicing Safe Techs; First Ed., 2009.
MongoDB Database References, 2016. Disponível em <<https://docs.mongodb.com/manual/reference/database-references/>>

7. Nome da Disciplina: Fundamentos de Computação Móvel e Ubíqua

Carga horária: 20 horas

Docente responsável: Iwens Gervásio Sene Júnior

Ementa:

Computação móvel e ubíqua: conceitos, principais características, internet das coisas, smart spaces, sensibilidade a contexto, tecnologias de comunicação e desafios. Plataformas de desenvolvimento móvel e modelos de negócio para comercialização. Projeto de aplicações móveis: macro e micro arquitetura, padrões de interface com usuário, persistência de dados, segurança, privacidade e comunicação. Programação de aplicações móveis: middlewares, frameworks e sensores. Prática em desenvolvimento de aplicações móveis.

Metodologia:

Aulas teóricas expositivas para apresentação do conhecimento relevante e exercícios e dinâmicas para fixação do conhecimento. O conhecimento deve ser aplicado em disciplinas de cunho prático e no projeto final de curso. Um estudo de caso deverá ser definido no início do curso para se desenvolver trabalhos das demais disciplinas. O estudo de caso definido no início do curso deverá ser utilizado para desenvolver os trabalhos da disciplina.

Bibliografia básica:

Greenfield, A. *Everyware: The Dawning Age of Ubiquitous Computing*. 1. ed., New Riders Publishing, 2006.
Krumm, J. *Ubiquitous Computing Fundamentals*. 1. ed., Chapman & Hall/CRC, 2009.
Neil, T. *Padrões de Design para Aplicativos Móveis*. O'Reilly, 2012. ISBN: 978-85- 7522-319-2.

8. Nome da Disciplina: Fundamentos de Programação Front-End**Carga horária:** 40 horas**Docente responsável:** Jacson Rodrigues Barbosa**Ementa:**

Os fundamentos de HTML, recomendações da W3C, semântica e acessibilidade. Aplicação de estilos com CSS: formatação de tags, utilização com HTML, seletores, herança, cores e gradiente. Padrões de nomenclaturas para organizar o CSS: BEM (Block, Element, Modifier), herança, etc. Técnicas de design responsivo. JavaScript (tipos de Dados – variáveis, strings, arrays, objetos, verificação de valores, JSON; controle de fluxo – estruturas de decisão, estruturas de repetição, funções; encapsulamento, controle de evento, processamento de arrays e manipulação de DOM).

Metodologia:

Pequeno percentual de horas aulas teóricas expositivas para apresentação do conhecimento relevante e ênfase para realização de exercícios e atividades práticas em laboratório para aprendizagem e fixação dos conhecimentos importantes sobre programação back-end. O conhecimento deve ser aplicado para modelar e implementar elementos que executam e são persistidos em servidores de aplicações ou sistemas de informação. O estudo de caso definido no início do curso deverá ser utilizado para desenvolver os trabalhos da disciplina.

Bibliografia básica:

W3C. *HTML: The Markup Language (an HTML language reference)*. 2013.

Disponível em <<https://www.w3.org/TR/html-markup/>>

W3C. *Cascading Style Sheets - Articles and Tutorials*. Disponível em

<<https://www.w3.org/Style/CSS/learning>>

MDN. *JavaScript Reference*. Disponível em

<<https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference>>

W3School. *JavaScript and HTML DOM Reference*. Disponível em

<<http://www.w3schools.com/jsref/>>

9. Nome da Disciplina: Programação Front-End Avançada**Carga horária:** 45 horas**Docente responsável:** Daniel Ferreira Monteiro Alves**Ementa:**

HTML5: elementos e atributos. APIs na especificação do HTML5 (canvas, audio e video, geolocation e storage). Especificação do Google Material Design e sua aplicação. Introdução ao SVG para sintetização de imagens vetoriais. Resolução de problemas de Acessibilidade.

Metodologia:

Pequeno percentual de horas aulas teóricas expositivas para apresentação do conhecimento relevante e ênfase para realização de exercícios e atividades práticas em laboratório para aprendizagem e fixação dos conhecimentos importantes sobre programação back-end. O conhecimento deve ser aplicado para modelar e implementar elementos que executam e são

persistidos em servidores de aplicações ou sistemas de informação. O estudo de caso definido no início do curso deverá ser utilizado para desenvolver os trabalhos da disciplina.

Bibliografia básica:

W3C. *W3C Recommendation: HTML5 A vocabulary and associated APIs for HTML and XHTML*. 2014. Disponível em <<https://www.w3.org/TR/html5/>>
Bootstrap. *Getting Started*. Disponível em <<http://getbootstrap.com/getting-started/>>
Bootstrap. *CSS*. Disponível em <<http://getbootstrap.com/css/>>
Bootstrap. *Components*. Disponível em <<http://getbootstrap.com/components/>>
Bootstrap. *JavaScript*. Disponível em <<http://getbootstrap.com/javascript/>>
Google. *Google Material Design*. Disponível em <<https://material.google.com/>>
Angular 2. *Bootstrap components written in pure AngularJS by the AngularUI Team*. Disponível em <<https://angular-ui.github.io/bootstrap/>>
W3School. *AngularJS Tutorial*. Disponível em <<http://www.w3schools.com/angular/default.asp>>

10. Nome da Disciplina: Fundamentos de Programação Mobile com Android

Carga horária: 30 horas

Docente responsável: Sandro Silva Moreira

Ementa:

Introdução ao sistema Android: arquitetura do sistema e SDK. Desenvolvimento de interfaces gráficas: Activity, Fragment e bibliotecas. Criação de listas. Arquiteturas para a aplicação. Conexão à internet através de protocolo HTTP. Publicação de aplicativos no loja Google Play.

Metodologia:

Aulas expositivas serão utilizadas para demonstrar o conhecimento que cobre a ementa seguida de prática em laboratório onde cada estudante irá exercitar e adquirir habilidade de desenvolvimento de aplicativos por meio da construção de programas em Android. Esta prática deverá exercitar todas as ferramentas básicas e boas práticas definidas na ementa.

Bibliografia básica:

Nelson Glauber. *Dominando o Android - 2ª edição: Do básico ao avançado*. NOVATEC, 2015.
Ricardo R. Lecheta. *Google Android*. NOVATEC, 2015.
João Bosco Monteiro. *Google Android: crie aplicações para celulares e tablets*. Casa do Código, 2015.

11. Nome da Disciplina: Programação Mobile com Android Avançado

Carga horária: 30 horas

Docente responsável: Sandro Silva Moreira

Ementa:

Tópicos em otimização de aplicação para contexto nacional. Persistência de dados. Leitura de sensores: câmera, geolocalização, beacons e NFC.

Metodologia:

Aulas expositivas serão utilizadas para demonstrar o conhecimento que cobre a ementa seguida de prática em laboratório onde cada estudante irá exercitar e adquirir habilidade de desenvolvimento de aplicativos por meio da construção de programas em Android. Esta prática deverá exercitar todas as ferramentas básicas e boas práticas definidas na ementa.

Bibliografia básica:

Nelson Glauber. *Dominando o Android - 2ª edição: Do básico ao avançado*. NOVATEC, 2015.

Ricardo R. Lecheta. *Google Android*. NOVATEC, 2015.

João Bosco Monteiro. *Google Android: crie aplicações para celulares e tablets*. Casa do Código, 2015.

12. Nome da Disciplina: Programação Mobile com iOS

Carga horária: 45 horas

Docente responsável: Silvano Maneck Malfatti

Ementa:

Introdução ao sistema iOS: arquitetura do sistema e ferramentas. Desenvolvimento de interfaces gráfica no XCode: Ações, delegates, listas e componentes de terceiros (bibliotecas). Arquiteturas para a aplicação. Conexão à internet através de protocolo HTTP. Publicação de aplicativos na Apple Store.

Metodologia:

Aulas expositivas serão utilizadas para demonstrar o conhecimento que cobre a ementa seguida de prática em laboratório onde cada estudante irá exercitar e adquirir habilidade de desenvolvimento de aplicativos por meio da construção de programas em iOS. Esta prática deverá exercitar todas as ferramentas básicas e boas práticas definidas na ementa.

Bibliografia básica:

Chris Eidhof Florian Kugler, Wouter Swierstra, *Swift Programming: The Big Nerd Ranch Guide (Big Nerd Ranch Guides)*, Big nerd Ranch, 2016

Chris Eidhof Florian Kugler, Wouter Swierstra, *Functional Swift*, Big nerd Ranch, 2015.