


PERÍODO 2022.2		
CENTRO UNIVERSITÁRIO CTC DEPARTAMENTO DE INFORMÁTICA	2022.2	
INF2125	PROJETO DE SISTEMA DE SOFTWARE	
	CARGA HORÁRIA TOTAL: 45h	CRÉDITOS: 3
	Título do Tópico: Projeto de Sistema de Software Pré-requisitos: (Não há) Professor: Carlos José Pereira de Lucena	

OBJETIVOS	<p>O objetivo central da disciplina é permitir que alunos aprendam técnicas e modelos que ajudem a desenvolver projetos de sistema de software seguindo padrões recomendados pela engenharia de software. Esse aprendizado acontece a partir do uso de uma metodologia de aprendizado moderna chamada <i>Challenge Based Learning</i> (CBL). CBL estimula a colaboração para identificar ideias, investigar e resolver desafios. Além disso, permite com que alunos obtenham um conhecimento profundo na área considerada na disciplina e desenvolvam habilidades necessárias para prosperar em um mundo em constante mudança. A seguir há alguns exemplos de tópicos considerados na disciplina de PSS: modelagem UML, paradigma orientado objetos, padrões de projeto, framework, linha de produto de software, SOLID etc.</p>
EMENTA	<p>Na edição 2022.2, a disciplina será composta por aulas e desafios práticos que ajudem os alunos a desenvolver projetos de sistemas de software. A disciplina adota a metodologia de aprendizado Challenge Based Learning (CBL), que é uma abordagem moderna, envolvente, colaborativa, aplicada ao problema de ensino-aprendizado, e tem sido empregada com sucesso no mundo todo. Esta metodologia explora recursos oferecidos pelas tecnologias e procura praticar habilidades importantes para o profissional do século XXI.</p> <p>Os alunos participarão de desafios que poderão contar com convidados externos voltados a contribuir com as soluções a serem desenvolvidas. A multidisciplinaridade é um ponto alto em que a metodologia usada prega.</p>

	<p>Dessa forma, a disciplina procura oferecer tal experiência durante a jornada de aprendizado dos alunos.</p> <p>Visando oferecer essa importante base de conhecimento sobre engenharia de software, um conjunto de tópicos importantes serão apresentados. Abaixo segue lista desses tópicos.</p> <ol style="list-style-type: none"> 1. O que é engenharia de software e o que esperar da disciplina. 2. Conhecendo a metodologia Challenge Based Learning 3. Prototipação de Sistemas na Engenharia de Software 4. Conhecendo Orientação Objetos 5. Diagrama de Caso de Uso 6. Diagrama de Classe 7. Diagrama de Sequência 8. Padrões de Projeto de Criação 9. Padrões de Projeto Estruturais 10. Padrões de Projeto Comportamentais 11. SOLID 12. Framework 13. Linha de Produto de Software 14. Boas práticas adicionais de desenvolvimento (p.ex. Nomenclatura de artefatos)
<p>PROGRAMA</p>	<p>A disciplina é dividida em duas fases, como explicadas a seguir.</p> <p>Fase 1: Apresentação de conceitos importantes relacionados a engenharia de software e primeira parte do desafio</p> <p>Nessa primeira parte da disciplina, um conjunto de aulas e a primeira parte de um desafio são oferecidos aos alunos. Sobre as aulas, os tópicos abordados são os seguintes:</p> <ol style="list-style-type: none"> 1. Introdução da disciplina. 2. Conhecendo a metodologia Challenge Based Learning 3. Prototipação de Sistemas na Engenharia de Software 4. Conhecendo Orientação Objetos – Aula parte 1 5. Conhecendo Orientação Objetos – Aula parte 2 6. Diagrama de Caso de Uso 7. Diagrama de Classe 8. Diagrama de Sequência 9. Padrões de Projeto de Criação 10. Padrões de Projeto Estruturais 11. Padrões de Projeto Comportamentais <p>Já em relação a primeira parte do desafio, ele permite colocar em prática boa parte dos conceitos mencionados nas aulas, focando principalmente na concepção e preparação do que é necessário para iniciar o desenvolvimento de um projeto de sistema de software. O tema do desafio será oferecido pelo</p>

	<p>professor, e geralmente conta com convidados que visam contribuir com a concepção das ideias de solução dos alunos. Esse desafio aplica a metodologia CBL, permitindo que haja uma experiência de aprendizado moderna e dinâmica.</p> <p>Fase 2: Segunda parte do desafio e aulas adicionais sobre engenharia de software</p> <p>Nesta fase os alunos são apresentados a segunda parte do desafio, que é continuidade da primeira parte mencionada acima. O foco do desafio é permitir que alunos evoluam o trabalho, permitindo o desenvolvimento de um framework que aplique boas práticas recomendadas da engenharia de software. A fim de consolidar esse aprendizado, diversos encontros voltados a orientar a evolução do desafio, além de aulas adicionais dos seguintes temas serão oferecidas.</p> <ol style="list-style-type: none"> 1. SOLID 2. Framework 3. Linha de Produto de Software 4. Boas práticas adicionais de desenvolvimento (p.ex. Nomenclatura de artefatos)
AVALIAÇÃO	Entregas a partir dos desafios: parte 1 e 2.
BIBLIOGRAFIA PRINCIPAL	<p>Grady Booch, James Rumbaugh and Ivar Jacobson. Unified Modeling Language User Guide, 2nd Edition, Addison-Wesley Object Technology Series, 2005.</p> <p>Craig Larman. Applying UML and Patterns: An Introduction to Object-Oriented, 3rd Edition, Prentice Hall PTR, 2004.</p> <p>Erich Gamma, Richard Helm, Ralph Johnson and John Vlissides. Design Patterns: Elements of Reusable Object-Oriented Software, 1st edition, Addison-Wesley Professional Computing Series, 1994.</p> <p>Elisabeth Freeman, Eric Freeman, Bert Bates and Kathy Sierra. Head First Design Patterns, O'Reilly, 2004.</p> <p>Challenge Based Learning. Available at: http://challengebasedlearning.org/.</p>

Mohamed E. Fayad, Douglas C. Schmidt and Ralph E. Johnson. Building Application Frameworks: Object-Oriented Foundations of Framework Design. John Wiley & Sons, 1999

Marcus Eduardo Markiewicz and Carlos J.P. Lucena. Object Oriented Framework Development. ACM Crossroads, Volume 7, Issue 4, 2001, pp.3-9.

Jennings, N. and Wooldridge, M. Agent-Oriented Software Engineering. Proceedings of the 9th European Workshop on Modelling Autonomous Agents in a Multi-Agent World : Multi-Agent System Engineering (MAAMAW'99), p. 1-7, 1999.

Joseph D. Gradecki, Nicholas Lesiecki, Mastering AspectJ: Aspect-Oriented Programming in Java, 1nd Edition, Wiley (2003)

Object Management Group – Agent Platform Special Interest Group. Agent Technology - Green Paper. Version 1.0, September 2000.

Klaus Pohl, Günter Böckle, Frank J. van der Linden, Software Product Line Engineering: Foundations, Principles and Techniques, Springer-Verlag New York, Inc., Secaucus, NJ, 2005

Ramnivas Laddad. Aspectj in Action: Practical Aspect-Oriented Programming, 1nd Edition, Manning Publications (2003)

**BIBLIOGRAFIA
COMPLEMENTAR**