



Universidade Federal do Ceará
Pró-Reitoria de Graduação
Coordenadoria de Projetos e Acompanhamento Curricular
Divisão de Pesquisa e Desenvolvimento Curricular

FORMULÁRIO PARA CRIAÇÃO DE DISCIPLINA

1. Unidade Acadêmica que oferta a Disciplina (Faculdade, Centro, Instituto, *Campus*):
Campus de Quixadá

2. Departamento que oferta a Disciplina (quando for o caso):

--

3. Curso(s) de Graduação que oferece(m) a disciplina

Código do Curso	Nome do Curso	Modalidade do Curso ¹	Currículo (Ano/Semestre)	Caráter da Disciplina ²	Semestr e de Oferta ³	Habilitação ⁴
402	Engenharia de Software	Bacharelado	2010.1	Optativa	06	--

4. Nome da Disciplina:

Leitura de Software

5. Código da Disciplina

 (preenchido pela PROGRAD):

QXD0067

6. Pré-Requisitos	Não (X)	Sim ()	
		Código	Nome da Disciplina

7. Turno da Disciplina

 (é possível marcar mais de um item):

Diurno Vespertino-Noturno Noturno

8. Regime da Disciplina:

Semestral Anual Modular

9. Justificativa para a criação desta disciplina – Máximo de 500 caracteres

(mostrar a importância da área / do conteúdo para a formação do aluno, a pertinência da disciplina na integralização curricular e outros aspectos):

¹ Preencher com *Bacharelado, Licenciatura* ou *Tecnólogo*.

² Preencher com *Obrigatória, Optativa* ou *Eletiva*.

³ Preencher quando obrigatória.

⁴ Quando eletiva, preencher com a habilitação a que se vincula a disciplina.

Por limitação de tempo, tamanho de equipe e maturidade da equipe, algumas vezes, é difícil exercitar dentro das disciplinas técnicas e práticas necessárias para projetos complexos ou de grande porte. Essa disciplina tem o objetivo de complementar a formação dos estudantes nesse aspecto através do estudo de projetos e produtos com reputação reconhecida (open source ou comerciais). Dessa forma é possível identificar e analisar as boas práticas utilizadas nesses projetos e com isso, fornecer experiência e conhecimento em um domínio que será importante para a vida profissional dos alunos.

10. Objetivo(s) da Disciplina:

Habilitar os alunos no reconhecimento as boas práticas empregadas em projetos de software por meio do estudo de projetos de desenvolvimento de software (ou seja, com o correspondente código, documentação, dados, ferramentas, processos utilizados e outros elementos empregados no desenvolvimento).

11. Ementa:

Estudar, investigar, analisar e discutir projetos de softwares existentes.

12. Descrição do Conteúdo e Carga Horária

Unidades e Assuntos das Aulas			Nº de Horas Teóricas	Nº de Horas Práticas	Nº de Horas EaD:
1. Técnicas de leitura de software			20	6	
2. Análise de código (avaliação do uso eficiente das estruturas de dados e de padrões de codificação, codificação e comentários)			4	6	
3. Análise de Requisitos e Testes (avaliação do uso eficiente de documentação de software e de testes)			4	6	
4. Análise de Projeto e Arquitetura (avaliação do uso eficiente de diferentes padrões arquiteturais)			4	4	
5. Estudo de casos incluindo avaliação de sistemas de complexos e de grande porte				10	
Número de Semanas:	Número de Créditos:	Carga Horária Total:	Carga Horária Teórica:	Carga Horária Prática:	Carga Horária EaD:
16	4	64	32	32	

13. Bibliografia (sugere-se a inclusão de até 10 títulos):

Bibliografia básica

MARTIN, Robert C. **Código limpo**: habilidades práticas do Agile Software . Rio de Janeiro: Alta Books, 2011. 413 p. ISBN 857608267-5 (broch.).

FEATHERS, Michael C. **Working effectively with legacy code**. Upper Saddle River, NJ: Prentice Hall, 2005. 434 p. (Robert C. Martin series) ISBN 0131177052.

...

SPINELLIS, Diomidis. **Code Quality**: the open source perspective. Boston: Addison-Wesley, 2006. ISBN 9780321166074.

Bibliografia complementar

ORAM, Andy; WILSON, Greg (Edit.). **Beautiful code**: leading programmers explain how they think. EUA: O'Reilly, 2007. 593 p. ISBN 9780596510046.

FOWLER, Martin. **Refatoração**: aperfeiçoando o projeto de código existente. Porto Alegre, RS:

Bookman, 2008. 365 p. ISBN 8536303956 (broch.).

MCCONNELL, Steve. **Code complete:** um guia prático para a construção de software . 2. ed. Porto Alegre, RS: Bookman, 2005. 928 p. ISBN 0735619670 (broch.).

HUNT, Andrew. **O programador pragmático:** de aprendiz a mestre. Porto Alegre, RS: Bookman, 2010. 343 p. ISBN 9788577807000 (broch.).

14. Avaliação de Aprendizagem:

O professor pode selecionar produtos open source, de reputação reconhecida, sobre os quais a análise do código disponível é realizada, por exemplo, por meio do acesso ao código fonte e de outros artefatos publicamente disponíveis. Nestes casos, a estrutura, a estratégia empregada, as ferramentas, o processo e outros deverão ser identificados e analisados. Esta análise deverá ser coletiva e de tal forma que o estudante possa se envolver com as opiniões dos colegas (todas moderadas pelo professor). O estudante deverá ser estimulado a confrontar o produto investigado com o que ele próprio produz, faz ou conhece.

Dessa forma, a avaliação na disciplina será realizada através da aplicação de 3 trabalhos práticos.

15. Aprovação do Colegiado do Departamento (quando for o caso)

Data de Aprovação:

Chefe(a) do Departamento
Assinatura e Carimbo

16. Aprovação do(s) Colegiado(s) de Curso(s)

Código do
Curso:

Data de
Aprovação:

Coordenador(a) do Curso
Assinatura e Carimbo

17. Aprovação do Conselho da Unidade Acadêmica

Data de Aprovação:

Diretor(a) da Unidade Acadêmica
Assinatura e Carimbo

18. Aprovação do Conselho de Ensino, Pesquisa e Extensão (Câmara de Graduação)

Data de Aprovação:	<hr/> Presidente(a) da Câmara de Graduação Assinatura e Carimbo
---------------------------	--

Orientação para tramitação do processo:

Deve ser aberto e encaminhado processo à Pró-Reitoria de Graduação / Câmara de Graduação, contendo: 1) Ofício de encaminhamento da Direção da Unidade Acadêmica; 2) Formulário para Criação de Disciplina integralmente preenchido, com assinaturas, datas e carimbos solicitados; e, 3) Parecer Técnico-Científico feito por profissional da área em questão.