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]	Semestre de Oferta[3]	Habilitação[4]
402	Engenharia de Software	Bacharelado	2010.1	Obrigatória	06	--

4. Nome da Disciplina:
Arquitetura de Software

5. Código da Disciplina (preenchido pela PROGRAD)**:**
QXD0064

6. Pré-Requisitos	Não ( )	Sim (X)	
		Código	Nome da Disciplina
		QXD0058	Projeto Detalhado de Software

7. Turno da Disciplina (é possível marcar mais de um item)**:**
(X) Diurno (X) Vespertino-Noturno ( ) Noturno

8. Regime da Disciplina:
(X) Semestral ( ) Anual ( ) Modular

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

A cada dia os sistemas de software se tornam parte integrante do cotidiano das pessoas. Esses sistemas são

[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.**

responsáveis por controlar desde simples ações, como abrir uma porta, até operações complexas como controlar aviões, gerenciar sistemas de telecomunicação e transações financeiras. Junto com o aumento do tamanho e da complexidade dos sistemas software, surge a necessidade de produzir sistemas cada vez mais confiáveis. Por outro lado, devido à competição de mercado, a necessidade das organizações por redução de tempo e custos no desenvolvimento e manutenção de software, aumenta. Nesse cenário, arquitetura de software tem assumido um papel importante no desenvolvimento de sistemas de software. Ela representa o núcleo de cada sistema de software bem projetado. A arquitetura de software lida com os "blocos de construção" de alto nível, que representam o sistema de software subjacente. Esses "blocos de construção" são os componentes (unidades de processamento em um sistema), os conectores (modelos das interações entre componentes de software), e as configurações (arranjos de componentes de software e conectores, e as regras que orientam sua composição). Essas abstrações proveem meios para que o engenheiro de software possa avaliar se o sistema em desenvolvimento está alinhado aos seus requisitos e aos interesses da organização. Desse modo, a adoção da arquitetura correta pode ajudar na gerência da complexidade e trazer diversos benefícios, tais como aumento da confiabilidade, manutenibilidade e redução de riscos. Dentro da matriz curricular do curso, a disciplina de Arquitetura de Software, além de apresentar novas técnicas e abordagens de abstração e modelagem de software sob uma perspectiva integrada e sistêmica, busca aprofundar conceitos básicos sobre modularidade, modelagem e analise de sistemas de software introduzidos nas disciplinas de "Programação Orientada a Objetos", "Projeto Detalhado de Software" e "Análise e Modelagem de Software", e introduzir conceitos de reuso de software alvo da disciplina de "Reuso de Software". Além do aprofundamento conceitual, a disciplina prioriza atividades teórico/prático associadas à modelagem, documentação e análise de arquiteturas, compreensão do formato padrão utilizado por engenheiros de software para documentar arquiteturas de famílias de aplicações para uma futura reutilização e como implantar, de forma sistemática, uma arquitetura.

10. Objetivo(s) da Disciplina:

Ao final do curso, os alunos deverão ser capazes de:

Objetivos Gerais:

- Compreender o conceito de arquitetura de software e a sua importância para o sucesso de um empreendimento de software;
- Compreender as dificuldades e a forma como projetar, analisar, documentar e implantar uma arquitetura software dentro de uma organização.

Objetivos específicos:

- Documentar arquiteturas utilizando padrões e estilos arquitetônicos;
- Identificar aspectos de qualidade de software que restringem uma arquitetura de software;
- Utilizar métodos sistemáticos para análise de arquiteturas de software;
- Compreender normas técnicas referentes à arquitetura de software;
- Identificar e gerenciar conflitos que podem afetar a arquitetura de software;
- Avaliar a qualidade de uma arquitetura de software;
- Mensurar os custos e os benefícios de uma arquitetura software para a organização.

11. Ementa:

Definição de arquitetura de software. A importância e o impacto em um empreendimento de software. Estilos arquiteturais (*pipes-and-filters*, *camadas*, *publish-subscribe*, baseado em eventos, cliente-servidor, dentre outros). Relação custo/benefício entre vários atributos arquitetônicos. Questões de hardware em projeto de software. Rastreabilidade de requisitos e arquitetura de software. Arquiteturas específicas de um domínio e linhas de produtos de software. Notações arquiteturais (ex., visões, representações e diagramas de componentes). Reutilização em nível arquitetural.

12. Descrição do Conteúdo e Carga Horária

	Nº de Horas	Nº de Horas	Nº de Horas EaD

Unidades e Assuntos das Aulas	Teóricas	Práticas	(quando for o caso):
1. Conceitos Básicos 1.1. Contextualização 1.2. Arquiteturas, Componentes e Conectores 1.3. Configurações 1.4. Estilos e Padrões Arquiteturais 1.5. Modelos e Processos para Arquiteturas	4h		
2. Projetando Arquiteturas 2.1. Concepção Arquitetural 2.2. Padrões e Estilos Arquiteturais 2.2.1. Arquiteturas para Domínios Específicos 2.2.2. Padrões versus Estilos Arquiteturais 2.3. Principais Estilos Arquiteturais 2.4. O Processo de Projeto Arquitetural	4h	2h	
3. Conectores 3.1. Fundamentos sobre Conectores 3.2. Papéis dos Conectores 3.3. Tipos de Conectores e suas Dimensões de Variação 3.4. Exemplos de Conectores	4h	2h	
4. Modelagem de Arquiteturas 4.1. Conceitos de Modelagem 4.2. Ambiguidade e Precisão 4.3. Trabalhando com Múltiplas Visões 4.4. Técnicas de Modelagem	6h	4h	
5. Análise de Arquiteturas 5.1. Metas da Análise 5.2. Escopo da Análise 5.3. Nível de Formalidade e Automação de Arquiteturas 5.4. Técnicas de Análise	6h	4h	
6. Implementação de Arquiteturas 6.1. Conceitos 6.2. Frameworks e Ferramentas Existentes 6.3. Exemplos	6h	4h	
7. Implantação e Mobilidade 7.1. Visão Geral 7.2. Arquitetura e o Processo de Implantação 7.3. Arquitetura de Software e a Mobilidade	4h	2h	
8. Estilos Arquiteturais Aplicados 8.1. Arquiteturas para Sistemas Distribuídos 8.2. Arquiteturas Descentralizadas 8.3. Arquiteturas Orientadas a Serviços 8.4. Arquiteturas para Domínios Específicos	4h	2h	
9. Projetando para Atender Requisitos Nãofuncionais 9.1. Eficiência 9.2. Complexidade 9.3. Escalabilidade e Heterogeneidade 9.4. Adaptação 9.5. Dependabilidade	4h	2h	

Número de Semanas:	Número de Créditos:	Carga Horária Total:	Carga Horária Teórica:	Carga Horária Prática:	Carga Horária EaD:
16	4	64	**42**	**22**	

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

Básica

TAYLOR, R. N.; MEDVIDOVIC, N.; DASHOFTY, E. M. **Software architecture**: Foundations, Theory, and Practice. Wiley, 2009. 750 p.

CLEMENTS, Paul et al. **Documenting software architectures**: views and beyond. 2. ed. Massachusetts: Addison-Wesley Professional. 2010. 592 p.

BASS, Len; CLEMENTS, Paul; KAZMAN, Rick. **Software architecture in practice**. 2. ed. Boston, Massachusetts: Addison-Wesley Professional, 2003. 560 p.

Complementar

SHAW, Mary; GARLAN, David. **Software architecture**: perspectives on an emerging discipline. São Paulo: Prentice Hall. 1996. 242 p.

SOMMERVILLE, I. **Engenharia de software**. 7. ed. São Paulo: Pearson Addison-Wesley, 2007.

GORTON, Ian. **Essential software architecture**. Berlin: Springer, 2006. 283 p. ISBN 3540287132 (enc.).

REEKIE, John. **A software architecture primer**. Sydney, Australia: Angophora Press, 2006. 179 p. ISBN 0646458418 (broch.).

BEZERRA, E. **Princípios de análise e projeto de sistemas com UML**. 2. ed. Rio de Janeiro: Elsevier, 2007.

14. Avaliação de Aprendizagem:

Avaliação individuais: AP1 e AP2
Trabalho em equipe: T1

Média = (AP1 + AP2 + T1) / 3

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:	_____ Presidente(a) da Câmara de Graduação **Assinatura e Carimbo**

<u>Orientação para tramitação do processo:</u>
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.