



UNIVERSIDADE
FEDERAL DO CEARÁ

UNIVERSIDADE FEDERAL DO CEARÁ
PRÓ-REITORIA DE GRADUAÇÃO
COORDENADORIA DE PROJETOS E ACOMPANHAMENTO CURRICULAR
DIVISÃO DE DESENVOLVIMENTO CURRICULAR

FORMULÁRIO PARA CRIAÇÃO E/OU REGULAMENTAÇÃO DE DISCIPLINA

() **Regulamentação** (se a disciplina está prevista no Projeto Pedagógico)

() **Criação/Regulamentação** (se a disciplina não está prevista no Projeto Pedagógico)

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 oferta(m) a disciplina

Código do Curso	Nome do Curso	Grau do Curso ¹	Currículo (Ano/Semestre)	Caráter da Disciplina ²	Semestre de Oferta ³	Habilitação ⁴
406	Design Digital	Bacharelado	2015.1	Obrigatória	2	

4. Nome da Disciplina:

Programação para Design

5. Código da Disciplina (preenchido pela PROGRAD):

QXD0129

6. Pré-Requisitos

Não ()

Sim ()

Código

Nome da Disciplina/Atividade

QXD0125

Introdução à Programação para Design

7. Correquisitos

Não ()

Sim ()

Código

Nome da Disciplina/Atividade

8. Equivalências

Não ()

Sim ()

Código

Nome da Disciplina/Atividade

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* ou *ênfase* a que se vincula a disciplina.

9. Turno da Disciplina (é possível marcar mais de um item):

Matutino Vespertino Noturno

10. Regime da Disciplina:

Semestral Anual Modular

11. Justificativa para a criação/regulamentação 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):

O estudo das técnicas de programação constitui uma componente fundamental no aprendizado do desenvolvimento de interfaces interativas e responsivas. É através da programação que a interface dos sistemas de software tornam-se interativas, usáveis e ricas. A Orientação a Objetos é um paradigma de programação maduro e largamente utilizado tanto na indústria quanto na academia. Um programa orientado a objetos é uma abstração do mundo real implementado através de um conjunto de objetos que interagem entre si para resolver, no universo computacional, um problema ou conjunto de problemas de um domínio de aplicação. Por esse motivo, o aprendizado das técnicas de programação orientada a objetos e, concomitantemente, a aplicação desse aprendizado através de uma linguagem de programação orientada a objetos, observando as boas práticas de desenvolvimento, torna-se uma necessidade e um fator diferencial.

12. Objetivo(s) da Disciplina:

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

Objetivos Gerais:

- Compreender os princípios da Orientação a Objetos.

Objetivos específicos:

- Desenvolver programas orientados a objetos em uma linguagem de programação;
- Entender o conceito de separação de interesses e modularidade através do encapsulamento;
- Utilizar e desenvolver bibliotecas de classes;
- Noções básicas de teste de software e tratamento de erros.

13. Ementa:

Estudo e utilização das estruturas dinâmicas que permitem o desenvolvimento de aplicações mais eficientes, confiáveis e estruturadas. Introduzir o paradigma de programação orientada a objetos (OO), juntamente com seus conceitos de classes, objetos, herança, encapsulamento e polimorfismo, além dos conceitos de Interfaces e exceções que são inerentes às linguagens de programação orientadas a objetos.

14. Descrição da Carga Horária

Número de Semanas:	Número de Créditos:	Carga Horária Total:	Carga Horária Teórica:	Carga Horária Prática:
16	4	64 horas	32 horas	32 horas

15. Bibliografia Básica (sugere-se a inclusão de, pelo menos, 03 títulos):

SILVEIRA, Paulo; ALMEIDA, Adriano. **Lógica de programação: crie seus primeiros programas usando JavaScript e HTML**. São Paulo, SP: Casa do Código, [2012]. vii , 163 p. (Caelum). ISBN 9788566250220

ZAKAS, Nicholas C. **Princípios de orientação a objetos em JavaScript**. São Paulo: Novatec, 2014. 126 p. ISBN 9788575223895 (broch.).

RESIG, John; BIBLEAULT, Bear. **Segredos do ninja JavaScript**. São Paulo: Novatec, 2013. 488 p. ISBN 9788575223284

16. Bibliografia Complementar (sugere-se a inclusão de, pelo menos, 05 títulos – de acordo com instrumento de avaliação de Curso de Graduação, INEP/maio-2012 ou legislação posterior):

MCLAUGHLIN, Brett; POLLICE, Gary; WEST, David. **Use a cabeça: análise e projeto orientado ao objeto**. Rio de Janeiro, RJ: Alta Books, 2007. xxviii, 441 p. ISBN 9788576081456.

BLAHA, Michael; RUMBAUGH, James. **Modelagem e projetos baseados em objetos com UML 2**. 2.ed. rev. e atual. Rio de Janeiro, RJ: Campus; Elsevier, 2006. xvii, 496 p. ISBN 9788535217537

FLANAGAN, David. **JavaScript: o guia definitivo**. 6. ed. Porto Alegre: Bookman, 2013. 1062 p. ISBN 9788565837194.

FREEMAN, Eric; ROBSON, Elisabeth. **Use a cabeça! programação em HTML 5: desenvolvendo aplicativos para web com JavaScrip**. Rio de Janeiro: Alta Books, 2014. xxxii, 573 p. (Use a cabeça!). ISBN 9788576088455.

FOWLER, Martin. **UML essencial: Um breve guia para a linguagem-padrão de modelagem de objetos**. Editora Bookman. 3a edição. 2005. ISBN: 8536304545.

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

Data de Aprovação:

Chefe(a) do Departamento
Assinatura e Carimbo

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

Código do Curso:

Data de Aprovação:

Coordenador(a) do Curso
Assinatura e Carimbo

19. Aprovação do Conselho da Unidade Acadêmica

Data de Aprovação:

Diretor(a) da Unidade Acadêmica
Assinatura e Carimbo

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

Data de Aprovação:

Presidente(a) da Câmara de Graduação
Assinatura e Carimbo

Orientações para tramitação do processo:

Deve ser aberto e encaminhado processo à Pró-Reitoria de Graduação / Câmara de Graduação, contendo: 1) Ofício(s) informando a data de aprovação da criação e/ou regulamentação da(s) disciplina(s) pela Coordenação do Curso, pelo(s) Departamento(s) envolvido(s) – se for o caso – e pela Direção da Unidade Acadêmica; 2)

Formulário para Criação e/ou Regulamentação de Disciplina integralmente preenchido, com assinaturas, datas e carimbos solicitados.

ANEXO - Descrição do Conteúdo e Carga Horária

Descrição do Conteúdo e Carga Horária				
Unidades e Assuntos das Aulas		Nº de Horas Teóricas	Nº de Horas Práticas	
1 – Revisão de Fundamentos de Programação com JavaScript		2	2	
2 – Introdução a linguagem JavaScript		2	4	
3 – Classes e Objetos		4	2	
4 – Atributos e métodos		4	2	
5 – Relacionamento entre objetos		2	4	
6 – UML		4		
7 – Herança e cadeia de protótipos		8	8	
8 – Polimorfismo		6	6	
9 – Padrões de Projeto			2	
10 – Testes Unitários			2	
14. Descrição da Carga Horária				
Número de Semanas: 16	Número de Créditos: 4	Carga Horária Total: 64h	Carga Horária Teórica: 32h	Carga Horária Prática: 32h