



**Universidade Federal do Ceará**  
**Campus de Russas**

**PLANO DE ENSINO DE DISCIPLINA**

Ano/Semestre
<b>2019/1</b>

<b>1. Identificação</b>				
1.1. Unidade: Campus Russas				
1.2. Curso: Ciência da Computação				
1.3. Estrutura Curricular (ano-período): 2018.2				
1.4. Nome da Disciplina: Linguagens de Programação				
1.5. Código da Disciplina: RUS0059				
1.6. Caráter da Disciplina: ( X ) Obrigatória ( ) Optativa				
1.7. Regime de Oferta da Disciplina: ( X ) Semestral ( ) Anual ( ) Modular				
1.8. Carga Horária (CH) Total:	C.H. Teórica:	C.H. Prática:	C.H. EaD:	C.H. Extensão:
64H	48H	16H	-	-
1.9. Pré-requisitos (quando houver): RUS0001 - Fundamentos de Programação				
1.10. Co-requisitos (quando houver): -				
1.11. Equivalências (quando houver): -				
1.12. Professor(es): Bonfim Amaro Júnior.				
<b>2. Justificativa</b>				
Esta disciplina faz o fechamento teórico de diversos conceitos vistos nas disciplinas de Fundamentos de Programação, Laboratório de Programação e Estruturas de Dados. O estudo das linguagens de programação capacita o aluno a escolher a linguagem que mais se adéque à solução de um determinado problema.				
<b>3. Ementa</b>				
Conceitos básicos de LP: domínios de aplicação, influências no projeto, paradigmas, métodos de implementação, critérios de avaliação, evolução das linguagens. Análise léxica e sintática. Variáveis: identificadores, vinculações, verificação de tipos, escopo. Tipos de dados. Expressões e a declaração				

**ATENÇÃO!** As informações a serem preenchidas neste formulário devem ser exatamente iguais às aquelas constantes no formulário de criação/regulamentação da disciplina aprovado pela Câmara de Graduação.

de atribuição. Abstração de processos: subprogramas. Abstração de dados e orientação à objetos. Noções de programação funcional. Noções de programação lógica.

#### 4. Objetivos – Geral e Específicos

##### Objetivo Geral:

- O objetivo da disciplina visa capacitar o aluno a compreender os diferentes paradigmas de linguagens existentes assim como as principais características e peculiaridades das linguagens de programação.

##### Objetivos Específicos:

- Desenvolver no aluno o senso crítico na escolha da melhor linguagem de programação, levando em consideração os aspectos relevantes do problema a ser resolvido e a resolução ou estratégia de resolução adotada;
- Generalizar e abstrair as principais características de cada paradigma de linguagem, visando desenvolver no aluno a competência de abstração conceptual para o estudo e abordagem de novas linguagens de programação.

#### 5. Calendário de Atividades

Data	Descrição do Conteúdo	Carga Horária
18/02/2019	Apresentação da disciplina e aspectos preliminares das linguagens de programação.	2h
20/02/2019	Domínios de programação e linguagens.	2h
25/02/2019	Crêterios de avaliação para construção de uma linguagem e histórico da evolução das linguagens de programação.	2h
27/02/2019	Introdução aos diferentes tipos de Paradigmas de Programação: Imperativos, Declarativos e Concorrentes.	2h
04/03/2019	<i>Recesso Escolar – Carnaval.</i>	
06/03/2019	<i>Recesso Escolar – Quarta-feira de Cinzas (até 12:00).</i>	
11/03/2019	Processo de Compilação: Fases de um compilador, interpretação e compilação mista.	2h
13/03/2019	Conceito de Gramáticas/Linguagens para o processo de Compilação.	2h
18/03/2019	Análise Léxica: Conceito de Autômatos, Linguagens e Expressões Regulares. (Parte I).	2h
20/03/2019	Análise Léxica: Conceito de Autômatos, Linguagens e Expressões Regulares. (Parte II).	2h
25/03/2019	<i>Feriado Estadual – Data Magna do Ceará.</i>	
27/03/2019	Análise Sintática: Linguagens Livres de Contexto Forma de Backus-Naur (BNF).	2h
01/04/2019	Análise Sintática: Árvore de Derivação de Sentenças e	2h

ATENÇÃO! As informações a serem preenchidas neste formulário devem ser exatamente iguais àquelas constantes no formulário de criação/regulamentação da disciplina aprovado pela Câmara de Graduação.

	Ambiguidade.	
03/04/2019	Apresentação de Trabalhos sobre linguagens – I – 4 grupos.	2h
08/04/2019	Apresentação de Trabalhos sobre linguagens – II – 4 grupos.	2h
10/04/2019	Apresentação de Trabalhos sobre linguagens – III – 4 grupos.	2h
15/04/2019	Apresentação de Trabalhos sobre linguagens – IV – 4 grupos.	2h
17/04/2019	Aula de exercícios sobre linguagens formais.	2h
22/04/2019	Avaliação de Conhecimentos I (AV1).	2h
24/04/2019	Avaliação de Conhecimentos I (AV1) 2ª Chamada.	2h
29/04/2019	Introdução à programação funcional usando Haskell.	2h
01/05/2019	<i>Feriado Nacional – Dia do Trabalho.</i>	
06/05/2019	Tipos e Classes de funções em Haskell.	2h
08/05/2019	Desvios Condicionais em funções com if / case of / “guardas”.	2h
13/05/2019	Listas em Haskell.	2h
15/05/2019	Funções de Ordem Superior.	2h
20/05/2019	Resolução de Exercícios na linguagem Haskell - I (LAB).	2h
22/05/2019	Programas interativos usando Haskell.	2h
27/05/2019	Resolução de Exercícios na linguagem Haskell - II (LAB).	2h
29/05/2019	Avaliação de Conhecimentos II (AV2) 1ª parte – Programação funcional.	2h
03/06/2019	Introdução à programação Lógica usando Prolog.	2h
05/06/2019	Sintaxe do Prolog.	2h
10/06/2019	Componentes básicos: Relações x Fatos x Regras.	2h
12/06/2019	Regras Recursivas.	2h
17/06/2019	Resolução de Exercícios na linguagem Prolog - I (LAB).	2h
19/06/2019	Avaliação de Conhecimentos II (AV2) 2ª parte – Programação Lógica.	2h
24/06/2019	Avaliação de Conhecimentos II (AV2) 2ª Chamada ( Programação Lógica + Funcional )	2h
26/06/2019	Correção das avaliações e considerações finais para última avaliação.	2h
<b>01/07/2019</b>	<b>Avaliação Final (AF)</b>	<b>2h</b>

#### 6. Metodologia de Ensino

- Aulas expositivas utilizando os recursos didáticos (Quadro Branco, Pincel e Projetor);
- Aulas práticas ou de exercícios em sala de aula e no laboratório de informática;
- Resolução de problemas computacionais;
- Trabalhos individuais ou em grupo.

**ATENÇÃO!** As informações a serem preenchidas neste formulário devem ser exatamente iguais às aquelas constantes no formulário de criação/regulamentação da disciplina aprovado pela Câmara de Graduação.

7. Atividades Discentes
<p>Os discentes deverão resolver os exercícios propostos durante as aulas teóricas, sempre apresentando ao professor suas soluções e dificuldades. Além disso, os mesmos terão que desenvolver uma apresentação em grupo com 4 participantes, no máximo, sobre linguagens de programação de diversos paradigmas, uma linguagem por equipe.</p> <p>Os alunos ficarão responsáveis por definir suas próprias equipes. O Professor realizará um sorteio para estabelecer a relação entre as equipes e linguagens propostas, bem como as datas de apresentação ( Reservadas no calendário de atividades (item 5) ).</p> <p>A avaliação dessa atividade representará a nota do Trabalho Prático (TP1). Os critérios de avaliação ficarão de responsabilidade do docente e serão validados para cada equipe.</p> <p>As apresentações terão duração mínima de 20 minutos e máxima de 25 minutos. Reservado para um máximo de 4 perguntas ao final. A presença dos alunos em todas as apresentações também contará como critério de pontuação (individual) consolidada.</p> <p>Todos os detalhes mais específicos serão apresentados pelo professor durante as aulas.</p>
8. Sistema de Avaliação
<p>Conforme o Regimento Geral da UFC, a avaliação de rendimento do aluno far-se-á segundo os critérios de assiduidade e eficiência. Na verificação da assiduidade será aprovado o aluno que frequentar 75% (setenta e cinco por cento) ou mais da carga horária da disciplina, vedado o abono de faltas. A verificação da eficiência compreenderá, no mínimo, duas avaliações progressivas e uma avaliação final. Será aprovado por média o aluno que apresentar média aritmética das notas resultantes das avaliações progressivas igual ou superior a 07 (sete). O aluno que apresentar a média igual ou superior a 04 (quatro) e inferior a 07 (sete), será submetido à avaliação final. Nesse caso, o aluno será aprovado quando obtiver nota igual ou superior a 04 (quatro) na avaliação final e média final igual ou superior a 05 (cinco).</p> <p>Diante desse contexto, o sistema de avaliação dessa disciplina será composto, exatamente, por duas avaliações progressivas (AP1 e AP2). A nota da primeira será computada por meio da média entre a soma da Avaliação de Conhecimentos I (AV1) ou da Avaliação de Conhecimentos I (AV1) (2ª Chamada) e do trabalho prático (TP1). Logo,</p> $AP1 = \frac{(AV1 + TP1)}{2}$ <p>Assim, a nota de AP2 será calculada através média entre soma da Avaliação de Conhecimentos II - 1ª parte (AV2.1) e Avaliação de Conhecimentos II 2ª parte (AV2.2). Assim,</p> $AP2 = \frac{(AV2.1 + AV2.2)}{2}$ <p>A Avaliação de Conhecimentos II (AV2) 2ª Chamada ( Programação Lógica + Funcional ) deverá</p>

**ATENÇÃO!** As informações a serem preenchidas neste formulário devem ser exatamente iguais àquelas constantes no formulário de criação/regulamentação da disciplina aprovado pela Câmara de Graduação.

ser realizada pelo aluno que faltar uma das avaliações (AV2.1 OU AV2.2). Dessa forma, a nota obtida será substituída na equação supracitada.

Obs: Se o aluno faltar as duas (AV2.1 E AV2.2) terá nota máxima igual a 5.0, assumindo que realizou a Avaliação de Conhecimentos II (AV2) 2ª Chamada e obteve 100% de aproveitamento.

Finalmente, a Média Parcial ( MP ) e a Média Final ( MF ) serão efetivadas mediante as equações abaixo:

$$MP = \frac{(AP1 + AP2)}{2} \text{ e } MF = \frac{(MP + AF)}{2}$$

## 9. Bibliografia Básica e Complementar

### Bibliografia Básica:

1. SEBESTA, R.W. Conceitos de linguagens de programação. 9 ed. Bookman, 2011. ISBN: 9788577807918;
2. TUCKER, A. B.; NOONAN, R. Linguagens de programação: princípios e paradigmas. 2 ed. McGrawHill, 2008. ISBN: 9788577260447;
3. WATT, D.A. Programming language: concepts and paradigms. Prentice Hall, 1990.

### Bibliografia Complementar:

1. DEITEL, H. M. C++ como programar. 5 ed. Prentice Hall, 2006. ISBN: 8576050560;
2. AHO, A. V.; SETHI, R.; ULLMAN, J. D. Compiladores: princípios, técnicas e ferramentas. 2 ed. Pearson/Addison-Wesley, 2008. ISBN: 9788588639249;
3. HOPCROFT, J. E.; ULLMAN, J. D.; MOTWANI, R. Introdução à teoria dos autômatos: linguagens e computação. Campus, 2002. ISBN: 9788535210729;
4. BARWISE, J. Language, proof and logic. Seven Bridges, 2002. ISBN: 9781575863740;
5. MENEZES, P.B. Linguagens formais e autômatos. 5 ed. Sagra Luzzato, 2008. ISBN: 9788577807659;
6. BARNES, D.J.; KOLLING, M. Programação orientada a objetos com Java: uma introdução prática usando BLUEJ 4 ed. Prentice Hall Brasil, 2009. ISBN: 9788576051879;
7. URUBATAN, R. Ruby on rails: desenvolvimento fácil e rápido. Novatec, 2009.

## 10. Parecer

Assinatura do Professor

\_\_\_\_/\_\_\_\_/\_\_\_\_

\_\_\_\_\_  
Professor Responsável

**ATENÇÃO!** As informações a serem preenchidas neste formulário devem ser exatamente iguais às aquelas constantes no formulário de criação/regulamentação da disciplina aprovado pela Câmara de Graduação.

Aprovação da Coordenação do Curso

\_\_\_\_/\_\_\_\_/\_\_\_\_

\_\_\_\_\_  
Coordenador do Curso

Aprovação da Coordenação Acadêmica

\_\_\_\_/\_\_\_\_/\_\_\_\_

\_\_\_\_\_  
Coordenadora Acadêmica

ATENÇÃO! As informações a serem preenchidas neste formulário devem ser exatamente iguais às constantes no formulário de criação/regulamentação da disciplina aprovado pela Câmara de Graduação.