

PLANO DE ENSINO

1. IDENTIFICAÇÃO		
UNIDADE ACADÊMICA ESPECIAL		CURSO OFERTANTE
Biotecnologia		Ciências da Computação
COMPONENTE CURRICULAR: Programação Funcional e Lógica		
CÓDIGO: IBT 0277	MODALIDADE: (X) Presencial () EaD	
CURSO(S) ATENDIDO(S)	Ciências da Computação	
ANO/SEMESTRE: 2019/2	Horário(s) de aula do componente curricular: Quartas e Sextas, de 8:50-10:50	
DOCENTE(S): Ricardo Couto Antunes da Rocha		
HORÁRIO DE ATENDIMENTO: Sextas, 14:00-16:00		
2. EMENTA		
Introdução ao paradigma lógico. Unificação. Derivações e Resolução SLD. Técnicas de programação em lógica. Programação lógica com restrições. Introdução ao paradigma funcional. Cálculo lambda e combinadores. Mônadas. Técnicas de programação funcional.		
3. DISTRIBUIÇÃO DE CARGA HORÁRIA		
CH TOTAL 64h	CH TEÓRICA 64h	CH PRÁTICA 0h
4. OBJETIVOS		
4.1 OBJETIVO GERAL O objetivo geral da disciplina é permitir ao aluno compreender como problemas computacionais podem ser resolvidos utilizando paradigma de programação funcional e o paradigma lógico, por meio de experimentação prática em problemas de média complexidade.		
4.2 OBJETIVO ESPECÍFICO Ao final do curso, o aluno deverá: <ul style="list-style-type: none"> Diferenciar como o paradigma imperativo, funcional e programação em lógica promovem a solução de problemas computacionais. Experimentar o uso de uma linguagem de programação funcional e de programação lógica para resolução de problemas. Compreender os principais conceitos do paradigma de programação funcional e noções básicas de cálculo lambda. Compreender e saber aplicar os principais padrões de resolução de problemas em programação funcional, particularmente o conceito de mônadas. Entender como construir tipos abstratos de dados em uma linguagem de programação funcional. Compreender os principais conceitos do paradigma de programação em lógica. Compreender e saber aplicar os principais padrões de resolução de problemas em programação lógica. 		

- Compreender o conceito de gramática de cláusulas definidas, cortes e negação em programação lógica.

5. CONTEÚDO

1. Programação funcional
 1. Introdução ao Paradigma Funcional
 2. Conceitos Fundamentais
 3. Tipos de Dados Básicos
 4. Listas
 5. Tipos Abstratos de Dados
 6. Monadas
2. Programação lógica
 1. Introdução ao Paradigma de Programação em Lógica
 2. Restrições Lógicas
 3. Operadores
 4. Recursividade
 5. Listas e aritmética
 6. Gramática de cláusulas definidas
 7. Cortes e negação

6. METODOLOGIA

O professor utilizará aulas expositivas intermeadas com aulas práticas de resolução de problemas.

A cada período de 7 (sete) a 15 (quinze) dias será aplicada uma tarefa de resolução de problemas, individual, que será utilizada para avaliar o aluno (ver seção "Avaliação"). No total, serão aplicados 12 (doze) tarefas aos alunos. A resolução dos exercícios nas aulas práticas também será avaliada e computada como 2 (duas) tarefas adicionais, compreendendo os exercícios de cada um dos dois módulos da disciplina (Programação funcional e em lógica).

A disciplina adotará as linguagens de programação Haskell e Prolog como linguagens de referência para discutir os conceitos dos paradigmas de programação, assim como para desenvolver todos os exercícios e tarefas da disciplina.

7. PROCESSOS E CRITÉRIOS DE AVALIAÇÃO

A avaliação da disciplina será composta de 14 (quatorze) tarefas individuais que deverão ser entregues ao longo do curso. **Será considerado aprovado, o aluno que COMPLETAR um mínimo de 11 (onze) tarefas, ou 75% das tarefas.**

Para avaliação das duas tarefas referentes às aulas práticas feitas em sala de aula, será considerada completa a tarefa do aluno que resolver um mínimo de 70% das tarefas práticas.

A cada tarefa entregue, o professor elaborará um feedback indicando problemas na solução que devem ser resolvidos. O aluno deverá resolver os problemas indicados no feedback do professor e resubmeter a tarefa.

Uma tarefa será considerada completada quando não houver mais feedbacks indicados pelo professor. Haverá uma tabela online mantida na página web da disciplina, contendo o registro das tarefas submetidas, número de

feedbacks e tarefas completadas.

Cada tarefa terá um prazo de submissão que **deverá ser rigorosamente cumprido**. Após o prazo, o aluno poderá realizar até duas resubmissões ou o limite de uma semana após o prazo de submissão, valendo aquele que for menor. Para que não haja dúvida de que um problema foi corrigido, é recomendável que o aluno interaja com o professor, seja por e-mail, seja pessoalmente.

As tarefas referentes a aulas práticas serão compostas de diversas subtarefas, cada uma associada a uma aula ministrada. Neste caso, o prazo para entrega é definido por subtarefa e termina antes da aula seguinte ou três dias, sendo considerado aquele que for menor.

Os seguintes requisitos se aplicam a todas as tarefas para que elas sejam consideradas completadas:

1. Todos os requisitos indicados no problema devem ser atendidos.
2. A solução deve incorporar código de testes de verificação da sua corretude.
3. A solução deve ser submetida em arquivo de texto (código fonte).
4. A solução deve ser submetida no sistema online indicado pelo professor (não serão aceitas soluções submetidas por email, por exemplo).
5. A solução não pode ser copiada de nenhum aluno (de qualquer ano), livro ou Internet, e deve ser desenvolvida individualmente pelo próprio aluno.

Caso o requisito (5) não seja atendido, a tarefa poderá ser ou considerada rejeitada, sem direito a resubmissão, ou substituída por uma nova tarefa.

Para efeito de cumprimento do RGCG, que exige que o resultado do aluno seja expresso em uma nota de 0 a 10, será adotada a seguinte tabela de mapeamento entre tarefas completadas e nota final (tarefas→nota):

14 → 10,0
13 → 8,6
12 → 7,3
11 → 6,0
10 → 5,5
9 → 4,9
8 → 4,4
7 → 3,8
6 → 3,3
5 → 2,7
4 → 2,2
3 → 1,6
2 → 1,1
1 → 0,5
0 → 0,0

Caso o número de tarefas da disciplina seja menor do que 14 (quatorze), será aplicado um critério de mapeamento proporcional, mantendo a exigência de 75% de tarefas completadas para aprovação.

8. LOCAL DE DIVULGAÇÃO DOS RESULTADOS DAS AVALIAÇÕES

No sítio web da disciplina (<http://www.inf.ufg.br/~ricardo/pfl>), no SIGAA e em sala de aula.

9. BIBLIOGRAFIA

9.1 BIBLIOGRAFIA BÁSICA

1. APT, K. R. From logic programming to Prolog. New York: Prentice Hall, 1997. <http://learnyouahaskell.com/>
2. BIRD, R. Introduction to Functional Programming using Haskell. 2 nd Edition. Editora Prentice Hall, 1998.
3. DOETS, K. From Logic to Logic Programming. Editora MIT Press, 1994.

9.2 BIBLIOGRAFIA COMPLEMENTAR

1. LIPOVACA, MIRAN. Learn you a haskell for great good!: a beginner's guide. no starch press, 2011.
2. APT, K. e WALLACE, M. Constraint Logic Programming using Eclipse. Editora Cambridge University Press, 2007.
3. HINDLEY, J. R. e SELDIN, J. P. Lambda--Calculus and Combinators: an introduction. 2 nd edition. Editora Cambridge University Press, 2008.
4. DOETS, K. e van EIJCK, J. The Haskell Road to Logic, Maths and Programming. Editora College Publications, 2004.
5. BRAMER, M. Logic Programming with Prolog. Editora Springer, 2005.
6. BRATKO, I. PROLOG Programming for Artificial Intelligence. 4th edition. Editora Addison-Wesley, 2011.

10. CRONOGRAMA

Data	Aula
14-08-2019 Qua	0.1 Apresentação do curso
16-08-2019 Sex	1.1 Introdução ao Paradigma Funcional
21-08-2019 Qua	1.2 Introdução à Programação em Haskell e Conceitos Fundamentais
23-08-2019 Sex	1.2 Introdução à Programação em Haskell e Conceitos Fundamentais
28-08-2019 Qua	1.3 Tipos de Dados Básicos
30-08-2019 Sex	1.3 Tipos de Dados Básicos
04-09-2019 Qua	Não haverá aula
06-09-2019 Sex	1.3 Tipos de Dados Básicos
11-09-2019 Qua	1.4 Listas
13-09-2019 Sex	1.4 Listas
18-09-2019 Qua	1.5 Tipos Abstratos de Dados
20-09-2019 Sex	1.5 Tipos Abstratos de Dados
25-09-2019 Qua	1.6 Entrada e Saída
27-09-2019 Sex	1.7 Funções de Alta Ordem
02-10-2019 Qua	1.7 Funções de Alta Ordem
04-10-2019 Sex	1.8 Mônadas
09-10-2019 Qua	1.8 Mônadas
11-10-2019 Sex	1.8 Mônadas
16-10-2019 Qua	1.8 Mônadas
18-10-2019 Sex	2.1 Introdução ao Paradigma de Programação em Lógica

23-10-2019 Qua	2.1 Introdução ao Paradigma de Programação em Lógica
25-10-2019 Sex	2.2 Restrições Lógicas
30-10-2019 Qua	CONPEEX
01-11-2019 Sex	2.2 Restrições Lógicas
06-11-2019 Qua	2.3 Operadores
08-11-2019 Sex	2.3 Operadores
13-11-2019 Qua	Não haverá aula
15-11-2019 Sex	FERIADO: Proclamação da República
20-11-2019 Qua	2.4 Recursividade
22-11-2019 Sex	2.4 Recursividade
27-11-2019 Qua	2.5 Cortes e negação
29-11-2019 Sex	2.5 Cortes e negação
04-12-2019 Qua	2.6 Listas e aritmética
06-12-2019 Sex	2.6 Listas e aritmética
11-12-2019 Qua	2.7 Gramática de cláusulas definidas
13-12-2019 Sex	2.7 Gramática de cláusulas definidas

11. APROVAÇÃO

APROVADO EM REUNIÃO DO COLEGIADO DA UNIDADE ACADÊMICA ESPECIAL
DE BIOTECNOLOGIA

EM ____ / ____ / ____.

Assinatura e Carimbo do Docente

Assinatura e Carimbo do Chefe da Unidade Acadêmica Especial IBIOTEC