UNIVERSIDADE FEDERAL DE UBERLÂNDIA
Faculdade de Computação
Av. João Naves de Ávila, nº 2121, Bloco 1A - Bairro Santa Mônica, Uberlândia-MG, CEP 38400-902
Telefone: (34) 3239-4144 - http://www.portal.facom.ufu.br/ facom@ufu.br

PLANO DE ENSINO

1. IDENTIFICAÇÃO

Componente Curricular:	**Programação Funcional**							
Unidade Ofertante:	FACOM							
Código:	GSI004		Período/Série:		3o	Turma:	C	
Carga Horária:						Natureza:		
Teórica:	30	Prática:	30	Total:	60	Obrigatória:	(X)	Optativa: ( )
Professor(A):	Lásaro Jonas Camargos					Ano/Semestre:	2021.2	
Observações:								

2. EMENTA

Tipos de objeto, expressões funcionais, operadores e funções, polimorfismo funcional, tuplas, listas, tipos algébricos de dados, classes de tipos, funções de ordem superior, formas de avaliação de programas, listas infinitas, entrada e saída, correção de programas e modularização.

3. JUSTIFICATIVA

O conhecimento dado na disciplina confere ao aluno a capacidade desenvolver programas em linguagens funcionais, conhecer os conceitos do paradigma de programação funcional e utilizar o Cálculo-Lambda, que é o modelo matemático deste paradigma de programação.

4. OBJETIVO

Objetivo Geral:

Ao final do curso, o aluno deverá ser conhecer os conceitos do paradigma de programação funcional, bem como ser capaz de desenvolver com fluência programas de porte médio escritos em linguagem funcional pura.

Objetivos Específicos:

-Conhecer e utilizar o Cálculo-Lambda, que é o modelo matemático deste paradigma de programação.

-Conhecer a sintaxe e operadores de uma linguagem funcional pura, como Haskell, sendo capaz de escrever com fluência programas de médio porte nessa linguagem.

-Diferenciar a abordagem de programação funcional de outras abordagens (ex. procedural e lógica) na implementação de programas.

5. PROGRAMA

Módulo 1. Introdução

1.1 Visão Geral da disciplina Programação Funcional

1.2 Principais paradigmas de Programação

1.3 Histórico das linguagens de programação

1.4 Introdução ao ambiente Haskell (GCC)

Módulo 2. Haskell

2.1 Introdução à Linguagem de Programação Haskell

2.2 Comandos úteis em Haskell

2.3 Tipos Básicos de Dados

2.4 Prototipação de tipos

2.3 Operadores matemáticos e lógicos

2.4 Funções

2.5 Expressões de seleção

2.6 Tuplas

2.7 Casamento de padrões

Módulo 3. Recursão

3.1 Função recursivas

3.2 Função recursivas matemáticas: fatorial, fibonacci, mdc, binomial

Módulo 4. Listas

4.1 Representação de listas

4.2 Lista por compreensão

4.3 Funções recursivas envolvendo listas

Módulo 5. Haskell: comandos

5.1 Sintaxe: where, let e case

5.2 Polimorfismo

5.3 Classes

Módulo 6. Cálculo Lambda

6.1 História: Alonzo Church

6.2 Sintaxe das expressões lambda e exemplos

6.3 Variáveis livres e ligadas

6.4 Redução de expressões lambda (Beta-redução)

6.5 Outras operações em expressões lambda: Alfa-conversão e η-redução

6.6 Expressões Lambda em Haskell

6.7 Funções anônimas em Haskell

Módulo 7. Funções de Alta Ordem

7.1 Mapeamento

7.2 Filtragem

7.3 Redução

Módulo 8. Algoritmos de Ordenação

8.1 Bubble

8.2 Insertion

8.3 Selection

8.4 Quicksort

Módulo 9. Tipos Algébricos

9.1 Sintaxe Geral

9.2 Tipos enumerados

9.3 Produto de Tipos

9.4 Tipos recursivos

9.5 Tipos polimórficos

Módulo 10. Árvores

10.1 Tipo abstrato de Dados

10.2 Árvores Binárias

10.3 Árvores Binárias de Busca

Módulo 11. Entrada e Saída

6. **METODOLOGIA**

A disciplina será desenvolvida através de atividades síncronas e assíncronas. As aulas síncronas serão realizadas presencialmente nos horários definidos pela coordenação: terça-feira 19:00-20:40 e quarta-feira 19:00-20:40/20:50-22:30.

A parte assíncrona será por meio de videoaulas disponibilizadas em um ambiente virtual de aprendizagem – AVA (preferencialmente, MS Teams) ou de domínio público, roteiros de estudos, listas de exercícios, trabalhos de implementação de códigos mais avançados e outras atividades letivas.

Essa disciplina possui 50% de conteúdo teórico e 50% de conteúdo prático. Tanto a parte teórica quanto prática da disciplina serão desenvolvidas através de atividades síncronas e assíncronas. A parte prática da disciplina envolve basicamente roteiros de estudo e lista de exercícios semanais, que visam a implementação em Haskell dos conceitos apresentados na parte teórica (videoaulas e aulas síncronas), além da implementação de dois trabalhos avaliativos com códigos mais avançados. Nesse processo, os alunos deverão utilizar o ambiente de implementação GHC para a linguagem Haskell.

O quadro a seguir apresenta o cronograma das atividades síncronas e assíncronas previstas e sua distribuição em quantidade de horas-aulas durante o semestre letivo. Para entendimento dos módulos e tópicos que serão oferecidos a cada semana, abaixo do quadro é apresentado um programa detalhado dos módulos e tópicos. Eventuais mudanças pontuais de tópicos podem ocorrer em função do andamento da aprendizagem da turma.

CRONOGRAMA DAS ATIVIDADES

SEMANA	**MÓDULOS**	**ATIVIDADES PRESENCIAIS PREVISTAS**	**CARGA-HORÁRIA PRESENCIAL**	**DATAS HORÁRIOS PRESENCIAIS**	**ATIVIDADES REMOTAS PREVISTAS**	**CARGA-HORÁRIA REMOTA**
INÍCIO DO SEMESTRE LETIVO						
1ª	Módulo 1: Introdução. Tópicos: 1.1, 1.2, 1.3, 1.4	Aulas	4ha	3/5;4/5		
2ª	Módulo 2: Haskell Tópicos: 2.1, 2.2, 2.3, 2.4,	Aulas	4ha	10/5;11/5	Lista de exercícios	0.5ha
3ª	Módulo 2: Haskell Tópicos: 2.5, 2.6 e 2.7	Aulas Especificação do primeiro trabalho	4ha	17/5;18/5	Lista de exercícios	0.5ha
4ª	Módulo 3: Recursão.	Aulas	4ha	24/5;25/5	Lista de exercícios	0.5ha

	Tópicos:3.1e 3.2					
5ª	Módulo 4: Listas Tópicos 4.1, 4.2 e 4.3	Aulas	4ha	31/5;1/6	Lista de exercícios	0.5ha
6ª	Módulo 4: Listas Tópicos: 4.4 Módulo 5: Haskell-comandos Tópicos 5.1, 5.2, 5.3	Aulas	4ha	7/6; 8/6	Lista de exercícios	0.5ha
7ª		Aula Entrega do primeiro trabalho	4ha	14/6;15/6		
8ª	Módulo 6: Cálculo Lambda 6.1, 6.2, 6.3 6.4, 6.5, 6.6 e 6.7	Aulas	4ha	21/6;22/6	Lista de exercícios	0.5ha
9ª	Módulo 7. Funções de Ordem Superior 7.1, 7.2, 7.3 Módulo 8. Algoritmos de Ordenação Tópicos 8.1, 8.2	Aulas Especificação do segundo trabalho	4ha	28/6;29/6	Lista de exercícios	0.5ha
10ª	Módulo 8. Algoritmos de Ordenação	Aulas	4ha	5/7 e 6/7	Lista de exercícios	0.5ha

	Tópicos 8.3, 8.4 Módulo 9. Tipos Algébricos Tópicos: 9.1, 9.2, 9.3, 9.4, 9.5					
11ª	Módulo 10: Árvores Tópicos: 10.1, 10.2, 10.3	Aulas	4ha	12/7; 13/7	Lista de exercícios	1ha
12ª	Módulo 11: Entrada e Saída	Aulas	4ha	19/7;20/7	Lista de exercícios	1ha
13ª		Entrega do segundo trabalho	4ha	26/7;27/7	Entrega do segundo trabalho	1ha
14ª	Tópicos adicionais		4ha	2/8;3/8	Lista de exercícios	1ha
15ª		2a Prova	4ha	9/8;10/8	Lista de exercícios	
16		Recuperação do trabalho	4ha	16/8;17/8	Lista de exercícios	
20/08/2022	TÉRMINO DO SEMESTRE LETIVO		Total carga-horária presencial: 64ha			Total Carga horária remota: 8ha

TÉCNICAS DE ENSINO E FERRAMENTAS TECNOLÓGICAS

O conteúdo programático da disciplina será desenvolvido presencialmente e por meio do ambiente virtual de aprendizagem MS Teams, mas podendo usar outras ferramentas em caso de necessidade.
Material de apoio como sítios e vídeos de livre acesso serão disponibilizados via MS teams.

As atividades assíncronas serão desenvolvidas por meio de videoaulas, roteiros de estudos, listas de exercícios, trabalhos de implementação, sendo esses materiais também disponibilizados via MS Teams.

ATENDIMENTO E COMUNICAÇÃO COM OS DISCENTES

Além das atividades síncronas prevista para o esclarecimento de dúvidas, o atendimento aos alunos também ocorrerá por meio da troca de mensagens via chat ou reuniões virtuais (individuais ou em grupo) previamente agendadas com o professor através do MS Teams.

A comunicação com a turma será feita preferencialmente pelo chat do MS Teams, onde todos os alunos devem ser previamente cadastrados no time "Programação Funcional" com código de auto inscrição a ser disponibilizado por email do professor para os alunos.

DIREITOS AUTORAIS

Todo o material produzido e divulgado pelo docente, como vídeos, textos, arquivos de voz, etc., está protegido pela Lei de Direitos Autorais, a saber, a lei nº 9.610, de 19 de fevereiro de 1998, pela qual fica vetado o uso indevido e a reprodução não autorizada de material autoral por terceiros. Parágrafo Único: responsáveis pela reprodução ou uso indevido do material de autoria dos docentes ficam sujeitos às sanções administrativas e as dispostas na Lei de Direitos Autorais.

7. AVALIAÇÃO

O método de avaliação pretendido prevê a realização das seguintes atividades avaliativas:

- Listas de Exercícios (entrega individual): consiste na implementação do conteúdo estudado na semana. Os exercícios são individuais e devem ser iniciados pelo aluno na aula prática semanal prevista no cronograma, com apoio do professor e do monitor. O restante da lista deverá ser implementado pelo aluno de forma assíncrona, sendo possível agendar atendimento para esclarecer dúvidas. Além disso, será dado o prazo até um dia antes da aula prática síncrona da semana seguinte para o seu envio, de modo que os alunos ausentes ou que apresentem maior dificuldade no aprendizado (ou dúvidas) tenham tempo de assimilar o conteúdo e desenvolver as atividades. A ferramenta a ser usada para entrega será especificada no início das aulas e possibilitará a auto correção. A solução das listas também será disponibilizada. A pontuação máxima que pode ser obtida nessa atividade avaliativa, computando todas as listas semanais entregues, é de 30 **pontos**.

- Data-limite para envio das listas semanais : 6 dias (corridos) após a aula síncrona onde o roteiro de atividades for apresentado.

- Trabalhos em dupla: ao longo da disciplina serão realizados dois trabalhos que envolvem a implementação de funções mais complexas da linguagem Haskell, bem como o seu uso em aplicações da área de computação. A descrição de cada trabalho será disponibilizada pelo MS Teams. As respectivas entregas estão definidas nesse plano. O trabalho também ajudará na preparação do aluno para a prova de conteúdo equivalente, de forma que o aluno utilize o seu desenvolvimento como autoavaliação e treinamento para a realização das provas. Essas entregas serão feitas pela dupla também pelo MS Teams, apontando para repositórios github e para um vídeo com a entrega. A correção dos códigos consiste na verificação de seu funcionamento através de casos de testes e do correto uso das estruturas de dados solicitadas. Além disso, cada aluno da dupla poderá passar por uma arguição individual, em um horário previamente agendado com o professor. Essa arguição será realizada em uma sala virtual particular. Cada trabalho vale 20 pontos, sendo 20 pontos atribuídos à correção dos códigos (Nc) e será atribuída uma nota de 0 a 1 para o desempenho de cada aluno durante a sua arguição (Na). A nota final do aluno em cada trabalho será dada pela multiplicação Nc*Na, sendo que embora a nota da dupla seja a mesma em NC, as notas de arguição (Na) serão diferenciadas, podendo resultar em diferentes notas de trabalho para integrantes de uma mesma dupla. Assim, as notas dos dois trabalhos somam 40 **pontos** no máximo. As datas-limite para envio dos trabalhos estão especificadas no cronograma de atividades. As datas de apresentação dos trabalhos estão especificadas no cronograma de atividades. Este trabalho terá a possibilidade de reapresentação, com atraso, com decréscimo da nota máxima para 30 pontos.

- Provas: serão realizadas duas provas individuais de 15 pontos (30 **pontos** no total), cujos horários e datas estão previstos no cronograma de atividades, mas serão confirmados com os alunos na primeira semana de

aulas para eventuais ajustes nas datas. A prova terá avaliação escrita e INDIVIDUAL. O plágio e as tentativas de comunicação lateral durante a prova não serão aceitas, sendo zeradas as provas com conteúdos idênticos ou muito similares, além de situações identificadas de tentativa de comunicação não-autorizada.

A assiduidade será avaliada por chamada nas atividades síncronas. Todos os alunos são considerados presentes nas atividades assíncronas.

8. BIBLIOGRAFIA

Básica

- Cláudio César de Sá, Márcio F. Silva. Haskell – uma abordagem prática. Novatec, 2006.
- Simon Thompson. Haskell – The Craft of Functional Programming (Second Edition). Addison-Wesley, 1999.

Complementar

- Paul Hudak. The Haskell School of Expression. Cambridge University Press. 2000.
- R. BIRD. Introduction to Functional Programming using Haskell, Prentice-Hall, 1998.

Material de apoio com acesso remoto:

- Gabriel, Paulo Henrique Ribeiro. Vídeo-Aulas de Programação Funcional. Portal do PET-SI da FACOM/UFU. Em: https://www.youtube.com/user/petsiufu, acessado em 23/07/2020.
- Camargos, Lásaro. Notas de aula de Programação Funcional. Em https://lasarojc.github.io/teaching/gbc033/index.html.
- The Haskell Programming Language: Em https://wiki.haskell.org/Haskell

9. APROVAÇÃO

Aprovado em reunião do Colegiado realizada em: ____/____/_____

Coordenação do Curso de Graduação: _____