

PLANO DE ENSINO

1. IDENTIFICAÇÃO:

COMPONENTE CURRICULAR: Programação Funcional				
UNIDADE OFERTANTE: FACOM				
CÓDIGO: GBC033		PERÍODO: 3º período		TURMA:
CARGA HORÁRIA			NATUREZA	
TEÓRICA:	PRÁTICA:	TOTAL:	OBRIGATÓRIA:	OPTATIVA:
30	30	60	(X)	()
PROFESSOR(A): Gina Maira Barbosa de Oliveira				ANO/SEMESTRE: 2020/3
OBSERVAÇÕES: disciplina oferecida na modalidade AARE				

2. EMENTA DA DISCIPLINA:

Paradigma Funcional: Tipos de objeto, expressões funcionais, operadores e funções, polimorfismo funcional, tuplas, listas, tipos algébricos de dados, classes de tipos, funções de alta ordem, 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 DA DISCIPLINA:

Ao final do curso, o aluno deverá ser capaz de 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.

5. PROGRAMA DA DISCIPLINA:

1. Revisão sobre Linguagens em Geral em Computação (Linguagens de programação e modelagem)
2. Valores, Tipos, Expressões,
3. Definição e avaliação de funções
4. Funções e Recursão

5. Recursividade de Cauda
6. Módulos
7. Tipos Estruturados de Dados
8. Recursão, Listas e Tuplas
9. Compreensão de Listas
10. Funções de ordem superior
11. Entrada e Saída
12. Tipos de Dados Algébricos
13. Generalização - map/filter/fold
14. Sistemas de Tipos (Classes de Tipos)
15. Mecanismos de Avaliação (avaliação lazy)

6. METODOLOGIA:

A disciplina será desenvolvida através de atividades síncronas e assíncronas. Serão agendadas, em média, quatro aulas síncronas e quatro aulas assíncronas, por semana. A exceção será na sexta e nona semanas, sendo que a sexta semana terá uma carga maior assíncrona (6 horas) e a nona semana terá uma carga maior síncrona (6 horas). Esse desbalanceamento pontual visa permitir uma carga maior síncrona na última semana, para a aplicação de prova e apresentação de trabalhos. As aulas síncronas serão realizadas por meio de plataforma para videoconferências (inicialmente, será adotado o Mconf da RNP, sendo que ao longo da disciplina outras ferramentas poderão ser adotadas, como o MS-Teams) e alocadas preferencialmente nos horários disponibilizados pela coordenação: quarta-feira de 8:50 a 10:40 e sexta-feira de 8:50 a 10:40. Caso a professora necessite agendar atividades síncronas fora desses horários, será discutido previamente com a turma (ex: planejamento de 2 aulas síncronas extras na nona semana). A parte assíncrona será por meio de roteiros de estudos, listas de exercícios, trabalhos de implementação de códigos mais avançados, e, eventualmente, videoaulas disponibilizadas em um ambiente virtual de aprendizagem – AVA (ex: Moodle ou MS Teams) ou de domínio público 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. Serão agendadas, em média, três aulas síncronas por semana para a apresentação dos conceitos teóricos, exercícios de fixação e esclarecimento de dúvidas e uma aula será dedicada a apresentação do roteiro da parte prática (que no ensino presencial era realizado em laboratório de computadores). 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, além da implementação de dois trabalhos avaliativos com códigos mais avançados. Nesse processo, os alunos deverão utilizar o ambiente de implementação para a linguagem Haskell chamado GHC, que também era utilizado na modalidade presencial da disciplina.

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 ASSÍNCRONAS PREVISTAS ¹	CARGA-HORÁRIA	ATIVIDADES SÍNCRONAS PREVISTAS ²	CARGA-HORÁRIA ³
12/08/2020	INÍCIO DO SEMESTRE LETIVO	-	-	-	-
1ª	Módulo 1: Introdução. Tópicos: 1.1, 1.2, 1.3, 1.4	- Listas de exercícios - Video-aulas	4 horas	Aulas remotas, com a presença do professor e monitor, se disponível. Datas: 12/08/2020 e 14/08/2020	4 horas
2ª	Módulo 2: Haskell Tópicos: 2.1, 2.2, 2.3, 2.4, 2.5, 2.6 e 2.7	- Listas de exercícios - Video-aulas	4 horas	Aulas remotas com a presença do professor e monitor, se disponível. Datas: 19/08/2020 e 21/08/2020	4 horas
3ª	Módulo 3: Recursão. Tópicos: 3.1 e 3.2 Módulo 4: Listas Tópicos 4.1 e 4.2	- Listas de exercícios - Video-aulas - Desenvolvimento do 1º Trabalho	4 horas	Aulas remotas com a presença do professor e monitor, se disponível. Datas: 26/08/2020 e 28/08/2020	4 horas
4ª	Módulo 4: Listas Tópicos: 4.3, 4.4	- Listas de exercícios - Video-aulas - Desenvolvimento do 1º Trabalho	4 horas	Aulas remotas com a presença do professor e monitor, se disponível. Datas: 02/09/2020 e 04/09/2020	4 horas
5ª	Módulo 5: Haskell- comandos Tópicos 5.1, 5.2, 5.3 Aplicação da 1ª Prova	- Listas de exercícios - Video-aulas - Desenvolvimento do 1º Trabalho	4 horas	- Aula remota com a presença do professor e monitor, se disponível. Data: 09/09/2020 - Aplicação remota de prova : 11/09/2020	4 horas
6ª	Módulo 6:	- Listas de exercícios - Video-aulas	6 horas	- Apresentação remota individual e	2 horas

	<p>Cálculo Lambda</p> <p>6.1, 6.2, 6.3 6.4, 6.5, 6.6 e 6.7</p> <p>Avaliação do 1º Trabalho</p>	- Desenvolvimento do 2º Trabalho		<p>avaliação do 1º trabalho .</p> <p>Data: 18/09/2020</p>	
7ª	<p>Módulo 8. Algoritmos de Ordenação</p> <p>Tópicos 8.1, 8.2, 8.3, 8.4</p>	<p>- Listas de exercícios</p> <p>- Video-aulas</p> <p>- Desenvolvimento do 2º Trabalho</p>	4 horas	<p>Aulas remotas com a presença do professor e monitor, se disponível.</p> <p>Datas: 23/09/2020 e 25/09/2020</p>	4 horas
8ª	<p>Módulo 9. Tipos Algébricos</p> <p>Tópicos: 9.1, 9.2, 9.3 , 9.4, 9.5</p> <p>Módulo 10: Árvores</p> <p>Tópicos: 10.1, 10.2, 10.3</p>	<p>- Listas de exercícios</p> <p>- Video-aulas</p> <p>- Desenvolvimento do 2º Trabalho</p>	4 horas	<p>- Aulas remotas com a presença do professor e monitor, se disponível.</p> <p>Datas: 30/09/2020 e 02/10/2020</p>	4 horas
9ª	<p>Módulo 11: Entrada e Saída</p> <p>Aplicação da 2ª Prova</p> <p>Apresentação do 2º Trabalho</p>	<p>- Listas de exercícios</p> <p>- Video-aulas</p>	2 horas	<p>- Aplicação remota de prova: 07/10/2020</p> <p>- Apresentação remota individual e avaliação do 2º trabalho (09/10/2020) e 2 horas extras síncronas agendadas com os alunos, preferencialmente no mesmo dia.</p>	6 horas
XX/OX/2020	TÉRMINO DO SEMESTRE LETIVO	-	-	-	-

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

TÉCNICAS DE ENSINO E FERRAMENTAS TECNOLÓGICAS

O conteúdo programático da disciplina será desenvolvido por meio do(s) ambiente(s) virtual(is) de aprendizagem. Dentre os possíveis, Moodle, Mconf, YouTube e MS Teams.

As atividades síncronas serão realizadas em salas virtuais por meio de plataformas para videoconferências (preferencialmente, o Mconf da CAPES/CAFE mas outras plataformas como Zoom e Google Meet poderão ser eventualmente utilizadas). A parte assíncrona será por meio de roteiros de estudos, listas de exercícios, trabalhos de implementação e, eventualmente, videoaulas, sendo esses materiais disponibilizados em um ambiente virtual de aprendizagem – AVA (por exemplo, Moodle da disciplina).

7. 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 de e-mail.

A comunicação com a turma será feita preferencialmente pelo quadro de avisos e fórum do ambiente AVA. Além disso, também pode ser usado o aplicativo UFU Mobile, o qual envia mensagens para os e-mails cadastrados dos alunos matriculados na disciplina e o fórum da plataforma Moodle, onde todos os alunos devem ser previamente cadastrados, através de auto-inscrição (a professora irá disponibilizar chave e senha na primeira semana de aula).

8. AVALIAÇÃO

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

- **Listas de Exercícios (entrega individual):** consiste na implementação do conteúdo aprendido na semana. A lista com os exercícios será disponibilizada pelo AVA durante a aula prática da semana (síncrona). Nessa aula, o professor irá explicar o roteiro de atividades com os alunos e iniciar o desenvolvimento da lista, com auxílio do monitor (se disponível). 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. Essa entrega será pelo moodle, preferencialmente, embora o aplicativo run.codes, ou ferramentas similares, seja uma opção a ser avaliada pela docente ao longo do semestre (<https://run.codes/>). A pontuação máxima que pode ser obtida nessa atividade avaliativa, computando todas as listas semanais entregues, é de **10 pontos**, pois são exercícios de baixa dificuldade apenas para fixação de conceitos.

- Data-limite para envio das listas semanais : 6 dias (corridos) após a aula síncrona, onde o roteiro de atividades será 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 Moodle. As respectivas entregas serão definidas para o dia imediatamente anterior à 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 AVA. A correção dos códigos consiste na verificação de seu funcionamento através de casos de testes e do correto uso da estrutura de dados solicitadas. Além disso, cada aluno da dupla passará 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 $Nc \cdot Na$, 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.

- Data-limite para envio do primeiro trabalho : 08/09/2020 11H55

- Data de apresentação (INDIVIDUAL) para o primeiro trabalho : 18/09/2020 8H50

- Data-limite para envio do segundo trabalho : 05/10/2020 11H55
- Data de apresentação (INDIVIDUAL) para o primeiro trabalho : 09/10/2020 8H50

- **Provas:** o plano de avaliação prevê a realização de duas provas individuais de 25 pontos (**50 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 será aplicada através de alguma plataforma AVA, sendo que a professora ainda está avaliando as ferramentas disponíveis para tal e deverá confirmar o procedimento até 1 semana antes da realização da primeira prova, ou seja, na quarta semana. O aluno enviará suas respostas pelo sistema antes do final do tempo previsto. O sistema não permitirá múltiplas tentativas, nem a alteração de uma resposta já registrada. A avaliação será realizada pelo professor a partir das respostas devidamente registradas. A prova é 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. Todos os alunos devem assinar um termo de compromisso, se comprometendo a ter um comportamento ético, não praticando plágio ou comunicação não-autorizada.

- Aplicação remota da primeira prova : 11/09/2020 às 8:50
- Aplicação remota da segunda prova: 07/10/2020 às 8:50

A assiduidade será avaliada com as entregas atividades assíncronas dentro dos prazos estipulados e através de chamadas durante as atividades síncronas.

9. BIBLIOGRAFIA E MATERIAL DE APOIO:

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.
- OLIVEIRA, G.M.B. Programação Funcional: módulos de aulas. Slides de aula disponibilizados pela profa. no AVA.

10. 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.

11. APROVAÇÃO

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

Coordenação do Curso de Graduação em: _____