

UFABC - Paradigmas de Programação (MCTA016-13)

Quadrimestre Suplementar de 2020

Turma NAMCTA016-13SA

Fabício Olivetti

folivetti@ufabc.edu.br

15 de setembro de 2020

Plano de ensino - Quadrimestre Suplementar (QS)

Diante da pandemia de COVID-19, que impõe a necessidade de adoção de medidas por parte do Poder Público (e da universidade) para a contenção da disseminação da doença, e considerando que o prazo de suspensão das atividades acadêmicas presenciais tende a ser longo, neste quadrimestre a disciplina será ministrada de maneira totalmente online. As regras que regulam esta modalidade são definidas pela [Resolução ConsEPE Nº 240/2020](#).

Todas as aulas, com e sem participação dos alunos, serão gravadas e disponibilizadas online segundo a [Licença Creative Commons Atribuição-NãoComercial 4.0 Internacional \(CC-BY-NC\)](#). Todos os participantes do curso dão sua tácita e irrevogável autorização para que suas imagens e falas sejam transmitidas, gravadas e editadas segundo a licença acima pelo docente responsável, sem nenhuma cobrança, para uso em distintos canais de comunicação e peças publicitárias sem fins comerciais.

1 Dinâmica de ensino

Serão disponibilizadas videoaulas com explicação do conteúdo teórico além do conteúdo prático com codificação em tempo real. Como material de apoio os alunos terão acesso aos códigos desenvolvidos durante as aulas, além de poderem contar com o uso do Discord (veja abaixo) para tirar dúvidas e discutir assuntos pertinentes à disciplina.

Semanalmente serão disponibilizadas listas de exercícios sobre o conteúdo apresentado. As entregas destas listas serão consideradas como uma avaliação e deverão ser feitas em até uma semana após a sua disponibilização. As notas dessas atividades serão utilizadas para a composição da média final. As aulas não serão dadas em tempo real devido aos inúmeros possíveis problemas técnicos, já que são dezenas de alunos matriculados na turma atualmente. Nos dias e horários em que haveria aula e atendimento, estarei online na ferramenta Discord (que além de chat, faz captura de voz e tela) para tirar dúvidas sobre o conteúdo previsto para aquela data.

2 Dias, horários e local das aulas

Todo material de aula, links para os vídeos, link para o servidor de Discord e o quadro de notas serão disponibilizados na página: <http://folivetti.github.io/teaching/2020-winter-teaching-1/>

O professor estará online no Discord (<https://discord.gg/JSgnfdE>):

- Segundas: 08h00 às 10h00
- Quintas: 10h00 às 11h00

Eventuais dúvidas e questionamentos poderão ser enviados em outros horários. Contudo, fora dos horários acima, o professor pode não atendê-los prontamente devido as suas outras atividades.

3 Datas Importantes

3.1 Listas semanais

Teremos listas semanais cujas datas de liberação do enunciado e prazo para entrega são listadas abaixo:

Lista	Enunciado	Prazo de entrega
1	24/09	02/10
2	01/10	09/10
3	08/10	16/10
4	15/10	23/10
5	22/10	30/10
6	29/10	06/11
7	05/11	13/11
8	12/11	20/11
9	19/11	27/11
10	26/11	04/12

Entrega toda sexta-feira até às 23:59. Listas entregues com atraso de até um dia serão aceitas, contudo com um desconto de 50% da nota.

3.2 Projeto de programação

Data para entrega: **04/12/2020**

Projetos entregues com atraso sofrerão descontos seguindo a seguinte tabela:

Dias em atraso	Nota máxima
1 dia	7
2 dias	6
3 dias	5
>3 dias	0

4 Aulas

As videoaulas serão disponibilizadas no youtube semanalmente. Além disto, em horários e locais a serem divulgados tanto no discord quanto no site da disciplina teremos sessões de live-coding. A gravação dessas seções será disponibilizada oportunamente.

O cronograma das aulas seguirá a programação seguinte:

Semana	Aula	Conteúdo
1	1	Paradigmas de Programação
	2	Ambiente de Programação; Conceitos básicos do Haskell
2	3	Introdução ao Haskell
	4	Cálculo Lambda; Programando com Lambdas; Combinador Y
3	5	QuickCheck
	6	Funções de Alta Ordem
4	7	Tipos de Dados Algébricos
	8	Type Class; Zippers
5	9	Monoids; Foldable
	10	Funtores
6	11	Funtores Aplicativos
	12	Traversable
7	13	Monads; SafeNum
	14	Monad Lista; Monad Either
8	15	Exemplos de monads: Writer e Reader
	16	Exemplos de monads: State e IO
9	17	Estruturas de dados funcionais: avaliação preguiçosa e persistência
	18	Pilhas e filas funcionais
10	19	Estruturas funcionais avançadas
	20	Heaps funcionais
11	21	Estratégias de paralelismo
	22	Programação concorrente
12	23	Type-safety, Phantom types
	24	Idris, Agda, Coq

5 Critério de Avaliação

5.1 Honestidade acadêmica

Entre outros, o código de ética da UFABC estabelece em seu artigo 25 que é **eticamente inaceitável** que os discentes:

I - fraudem avaliações;

II - fabriquem ou falsifiquem dados;

III - plagiem ou não creditem devidamente autoria;

IV - aceitem autoria de material acadêmico sem participação na produção;

V - vendam ou cedam autoria de material acadêmico próprio a pessoas que não participaram da produção.

Muitos ainda têm dúvidas sobre a interpretação das regras definidas pelo Código de Ética da UFABC. Por esta razão, diversos professores elaboraram um documento (**disponível aqui**) com vários exemplos e esclarecendo a interpretação das regras acima. Abaixo uma versão resumida. **Sempre consulte o documento completo ou converse com o seu professor em caso de dúvidas!**

- **Regra 1** - Você não pode enviar para avaliação um trabalho que não seja de sua própria autoria ou que seja derivado/baseado em soluções elaboradas por outros.
- **Regra 2** - Você não pode compartilhar a sua solução com outros alunos nem pedir aos seus colegas que compartilhem as soluções deles com você.
- **Regra 3** - Nos trabalhos enviados para avaliação você deve indicar eventuais assistências que você tenha recebido.

ATENÇÃO: todos os trabalhos enviados para avaliação poderão ser verificados por um sistema automatizado de detecção de plágio.

Qualquer violação às regras descritas acima implicará:

- Descarte dos conceitos atribuídos a TODAS as tarefas avaliativas regulares de TODOS os envolvidos, causando assim suas **reprovações automáticas com conceito F** .
- Possível **denúncia** à Comissão de Transgressões Disciplinares Discentes da Graduação, a qual decidirá sobre a punição adequada à violação que pode resultar em **advertência, suspensão ou desligamento** , de acordo com os artigos 78-82 do Regimento Geral da UFABC.
- Possível **denúncia** apresentada à Comissão de Ética da UFABC, de acordo com o artigo 25 do Código de Ética da UFABC.

5.2 Composição das notas

A avaliação da disciplina será composta pelas seguintes notas:

- N_P é a nota do projeto
- N_L é a nota das listas de exercícios semanais

A nota N_L será calculada como a média aritmética das listas semanais, cada lista valendo 10 pontos.

A nota final (N_F) será determinada pela média harmônica de N_P e N_L .

$$N_F = \frac{2}{\frac{1}{\max\{0,1,N_P\}} + \frac{1}{\max\{0,1,N_L\}}}$$

O conceito final (C_F) será obtido de acordo com a equação abaixo:

$$C_F = \begin{cases} \mathbf{F}, & \text{se } N_F \in [0, 0; 5, 0) \\ \mathbf{D}, & \text{se } N_F \in [5, 0; 6, 0) \\ \mathbf{C}, & \text{se } N_F \in [6, 0; 7, 0) \\ \mathbf{B}, & \text{se } N_F \in [7, 0; 8, 5) \\ \mathbf{A}, & \text{se } N_F \in [8, 5; 10, 0] \end{cases}$$

Caso seja verificado ocorrência de plágio no projeto final, o aluno será automaticamente reprovado com F

5.3 Recuperação

A resolução ConsEPE nº 182 assegura a todos os alunos de graduação com C_F igual a **D** ou **F** o direito a fazer uso de mecanismos de recuperação.

A recuperação será feita através de uma nova entrega do projeto final levando em consideração a correção do projeto original e seguido de uma entrevista com o docente a ser marcada entre os dias 18/12/2020 e 19/12/2020. A sua nota será utilizada para compor a o conceito pós-recuperação C_R conforme as equações abaixo:

$$N_R = \frac{P_R + N_F}{2}$$

Caso 1 $C_F = D$:

$$C_R = \begin{cases} \mathbf{C}, & \text{se } N_R \geq 6,0 \\ \mathbf{D}, & \text{caso contrário} \end{cases}$$

Caso 2 $C_F = F$:

$$C_R = \begin{cases} \mathbf{D}, & \text{se } N_R \geq 5,0 \\ \mathbf{F}, & \text{caso contrário} \end{cases}$$

5.4 Regulamentações Relevantes

- [Resolução ConsEPE Nº 240](#) - Estabelece a autorização para a oferta excepcional de componentes curriculares e de outras atividades acadêmicas remotas durante o(s) chamado(s) "Quadrimestre(s) Suplementar(es)"
- [Resolução ConsEPE nº 182](#) - Regulamenta a aplicação de mecanismos de recuperação
- [Código de Ética da UFABC](#)
- [Resolução ConsUni nº 63](#) - Regimento Geral da UFABC]]

6 Recursos Online

6.1 Grupos, listas, páginas, ...

- [Grupo de Estudos em Haskell da UFABC](#)
 - Não esqueça de participar do Discord: <https://discord.gg/JSgnfdE>
- [Haskell Home Page](#)
- [#haskell IRC channel](#)
- [StackOverflow](#)
- A lista [Haskell-beginners](#) é um bom lugar para buscar respostas para perguntas básicas
- A lista [Haskell-cafe](#) é boa para buscar respostas a perguntas mais elaboradas
- O [Haskell Wiki](#) tem uma lista de respostas às perguntas mais comuns

6.2 Disciplinas em Haskell

- MCTA016-13: Paradigmas de Programação (em Haskell), UFABC. [2019](#), [2018](#).
- CR062-Programação Funcional em Haskell, UFABC. [2019](#), [2018](#).
- Estruturas de Dados Puramente Funcionais. [2019](#).
- G51PGP: Programming Paradigms, University of Nottingham. [2019](#).
- CS653: Functional Programming, Indian Institute of Technology Kanpur. [2018](#).
- CIS 194: Introduction to Haskell, University of Pennsylvania. [2016](#), [2015](#), [2014](#), [2013](#).
- ... mais exemplos [aqui](#)

6.3 Leituras

- [Haskell Wikibook](#)
- [A Gentle Introduction to Haskell](#)
- [Haskell Cheat Sheet](#)
- [What I Wish I Knew When Learning Haskell](#)

6.4 Documentação

- [Standard library documentation](#)
- [Hackage](#) - Repositório de pacotes
- [Hoogle](#) - Procurando funções para as quais você não sabe o nome?
- [Hayoo](#) - Parecido com o Hoogle, porém mais completo (procura todo o Hackage, mas está frequentemente fora do ar...)
- Se você realmente precisar, a linguagem e a biblioteca padrão do Haskell são definidos pelo [Haskell 2010 - Language Report](#)

7 Bibliografia

Os principal texto utilizado neste curso será o **GH Segunda Edição**.

- *Programming in Haskell*. 2nd Edition.

– Por *Graham Hutton*.

A primeira edição, que tem boa parte do conteúdo da segunda edição, está disponível na biblioteca:

- Link Biblioteca: http://biblioteca.ufabc.edu.br/index.php?codigo_sophia=15287

- *Real World Haskell*.

– Por *Bryan O’Sullivan, John Goerzen e Don Stewart*.

– Disponível **gratuitamente** em: <http://book.realworldhaskell.org/>

- *Learn You a Haskell for Great Good!: A Beginner’s Guide*.

– Por *Miran Lipovača*.

– Disponível **gratuitamente** em: <http://learnyouahaskell.com/>

- *Parallel and Concurrent Programming in Haskell: Techniques for Multicore and Multithreaded Programming*.

– Por *Simon Marlow*.

– Disponível **gratuitamente** em: <https://www.oreilly.com/library/view/parallel-and-concurrent-programming-in-haskell/9781449335939/>