

2024 – PROGRAMAÇÃO FUNCIONAL (MCCC015-23)

Turmas: DA1MCCC015-23SA e NA1MCCC015-23SA

Professor: [Emilio Francesquini](#)

E-mail: e.francesquini@ufabc.edu.br

1 Avisos

- [2024-07-05 Fri] [Lista de estudo pré-quiz 01 disponível](#)
- [2024-07-05 Fri] Datas dos quizzes conforme o novo calendário divulgado pela ProGrad disponíveis.
- [2024-06-25 Tue] Página da disciplina no ar.

2 Informações Gerais

Esse oferecimento da disciplina será completamente **presencial**, incluindo aulas, plantões e avaliações. Contudo, o professor poderá responder a eventuais questionamentos e dúvidas no servidor do Discord no canal da disciplina cujo endereço está disponível abaixo.

Note

Todo o material das aulas é disponibilizado online segundo a [Licença Creative Commons Atribuição-NãoComercial 4.0 Internacional](#) (CC-BY-NC).

Tip

Inscreva-se o quanto antes no servidor do Discord da disciplina (<https://discord.gg/JSgnfdE>). Todos os anúncios e comunicações serão feitos por lá.



Figure 1: [Servidor do Discord](#) da disciplina.

Teremos Quizzes quinzenais, feitos em sala de aula com exercícios sobre o conteúdo apresentado. As entregas destes quizzes serão consideradas como uma avaliação e deverão ser feitas durante a aula. As notas dessas atividades serão utilizadas para a composição da média final. Teremos também dois projetos de programação que serão usados para compor a nota final. Veja mais sobre o [critério de avaliação aqui](#).

2.1 Horário e local das aulas

Turma	Quarta	Sala	Sexta	Sala
DA1MCCC015-23SA	10h-12h	A-108-0	08h-10h	A-108-0
NA1MCCC015-23SA	21h-23h	A-108-0	19h-21h	A-108-0

2.2 Atendimento

• Presencial

- Nos horários listados abaixo não é preciso confirmar ou marcar, apenas apareça! :-)
- **Quarta-feira, das 20:00 às 21:00, Sala 509-2.**
- **Sexta-feira, das 10:00 às 11:00, Sala 509-2.**
- **Agendado por e-mail**
 - Verifique [minha agenda](#) e sugira peelo menos dois possíveis horários!
- **Em sala de aula** - Após as aulas

• Online

- Por [e-mail](#).
- Pelo [Discord](#).

3 Sobre a Disciplina

MCCC015-23 Programação Funcional

- TPEI 4-0-0-4
- Recomendação: Algoritmos e Estrutura de Dados I; Programação Estruturada; Processamento da Informação

Objetivos

Apresentar noções básicas e intermediárias sobre paradigma funcional não-tipado (cálculo lambda) e tipado. Ensinar os conceitos fundamentais como funções, recursão, recursão de cauda, sistemas de tipos, álgebra de tipos, polimorfismo, funções de alta ordem, abstração, tratamento de efeitos colaterais, e conceitos elementares da teoria dos tipos e das categorias aplicados na programação funcional. Para tanto, será utilizada uma linguagem estritamente funcional moderna ou que estimule o uso dos conceitos desse paradigma.

Conteúdo Programático

Paradigma funcional e imutabilidade. Cálculo lambda. Funções, recursão, recursão de cauda. Funções de alta ordem e currying. Tipos de Dados Algébricos. Polimorfismo ad-hoc e paramétrico. Semigrupos e monoides. Funtores e monadas. Tratamento de efeitos colaterais com mônadas. Estruturas de dados funcionais.

Fonte: [Catálogo de Disciplinas 2023/2024](#)

4 Datas Importantes e Avaliações

ATENÇÃO - A prova de recuperação será feita durante as duas primeiras semanas do Q3, com a data a ser combinada com os interessados.

4.1 Avaliações escritas (Quizzes quinzenais)

- Quiz 1 - 10/07/2024 - [Lista Pré Quiz](#)
- Quiz 2 - 24/07/2024
- Quiz 3 - 07/08/2024
- Quiz 4 - 21/08/2024
- Quiz 5 - 04/09/2024
- Quiz 6 - 13/09/2024

- Quiz Substitutivo - 13/09/2024

Exemplos dos exercícios que podem cair nos quizzes podem ser vistos nas playlists com a resolução de todas as listas dadas nos oferecimentos anteriores da disciplina podem ser encontradas aqui:

- [2021](#)
- [2020](#)
- [2022](#) (apenas exercícios, sem resolução)

4.2 Projeto de programação

- Projeto 1 - 04/08/2024
- Projeto 2 - 08/09/2024

Durante o quadrimestre teremos dois projetos de programação as serem entregues via GitHub Classroom. O link será disponibilizado nesta página em breve.

- O projeto pode ser feito em grupos de no máximo 3 pessoas
 - Apenas um dos integrantes do grupo deve fazer o upload do projeto para o GitHub e deve estar bem claro quais são os integrantes da equipe no relatório.
- Todos os *commits* e o *push final* devem ter sido feitos até no máximo às **23h59 do respectivo dia de entrega.**
 - Atenção, caso ocorram/commits/pushes após o prazo, será considerada apenas a última versão para avaliação. Caso o último esteja com atraso > 3 dias, então será considerado o último commit até 3 dias (valendo no máximo 5).
 - Entregas em atraso sofrerão descontos conforme a tabela abaixo.

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

- No mesmo repositório do seu código, deverá haver um relatório de no máximo 1 páginas (em PDF! nada de `.md`, `.docx`, `.odt`, ou qualquer outro formato da moda...) descrevendo:
 - Qual é o seu projeto
 - Como utilizar (se for qualquer coisa diferente de `stack run`) o seu código
 - Dificuldades, surpresas e destaques do seu código
- O seu relatório também deverá conter **obrigatoriamente** um link para um vídeo que:
 - Esteja abrigado no Youtube (ou qualquer serviço semelhante)
 - Seja privado/não listado e disponível apenas àqueles com o link
 - Tenha no máximo 3 minutos (ou seja, <= 180 segundos). **Sem exceções.**
 - Mostre a compilação, o uso e uma apresentação do código do projeto
- Durante a correção o professor pode pedir por esclarecimentos adicionais.

****Que tipos de projetos serão aceitos? ****

Lembre-se que você terá o quadrimestre todo para entregar o seu projeto final, em duas etapas. Logo é esperado que o seu projeto reflita um esforço de desenvolvimento compatível com o prazo disponível.

Como exemplo, em oferecimentos anteriores da disciplina tivemos alunos que implementaram jogos (inclusive dois deles que implementaram um clone completo do Pac-Man), algoritmos de IA e aprendizado de máquina e avaliações de desempenho, paralelizações de algoritmos clássicos, ferramentas completas para sincronização de arquivos, clientes de e-mail, servidores de banco de dados com suporte para SQL,...

Caso esteja com dúvidas se a sua ideia é o suficiente para um projeto final que será bem avaliado converse com o professor o quanto antes!

E, caso esteja sem ideias, veja [a seguinte proposta de Projeto](#) e os projetos de exemplo feitos pelos alunos nos oferecimentos anteriores.

5 Projetos de exemplo de oferecimentos anteriores

- **Haskeroid** - Por **Artur Henrique Allen Santos**

- Código: [haskeroid.zip](#)



- **Pac-Man** - Por **Edson Gomes Martinelli e Rafael Akio Shishito Matos**

- Código: [pacman.zip](#)



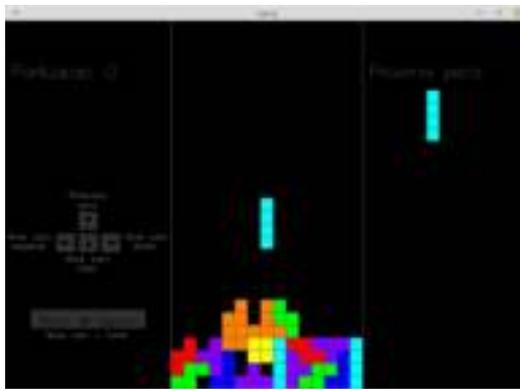
- **Pac-Man: Get the Curry** - Por **Jair Edipo Jerônimo e Michelle Kaori Hamada**

- Código: [pacman_curry.zip](#)



- **Tetris I** - Por **Bárbara Dias de Sena**

- Código: [tetris_i.zip](#)



- **Tetris II** - Por **Fabio Luis Arruda Fernandes**

- Código: [tetris_ii.zip](#)



- **Xadrez** - Por **Eduardo Castilho Ferreira**

- Código: [xadrez.zip](#)



6 Listas de Exercícios Complementares

Listas de exercícios extras (não valem nota) para aqueles que quiserem ir um pouco além...

- [Lista 1](#)
 - [Lista 2](#)
 - [Lista 3](#)
 - [Lista 4](#)
 - [Lista 5](#)
 - [Lista 6](#)
-

7 Aulas

A disciplina será completamente presencial. Contudo, o material da disciplina mantém uma boa interseção com os oferecimentos anteriores da disciplina que foram online. Os vídeos e material escrito destas aulas está disponível online no seguinte endereço:

<http://haskell.pesquisa.ufabc.edu.br/haskell>

A playlist com todos os vídeos das aulas teóricas do curso pode ser encontrada aqui:

<https://www.youtube.com/playlist?list=PLYItvall0TqJ25sVTLcMhxsE0Hci58mpQ>

Para facilitar, na tabela abaixo você encontra os assuntos juntamente com links diretos para o conteúdo e vídeos.

Semana	Tópico	Material de Estudo	Playlist
1	Preparando o ambiente	https://haskell.pesquisa.ufabc.edu.br/haskell/00.bootstraping/	https://www.youtube.com/playlist?list=PLYItvall0TqLedbLNsncIUfk3cHv_FS7O
1	Paradigmas de programação	https://haskell.pesquisa.ufabc.edu.br/haskell/01.paradigmas/	https://www.youtube.com/playlist?list=PLYItvall0TqLZ_ih1mSoc1roCZGifj8-t
2	Cálculo Lambda	https://haskell.pesquisa.ufabc.edu.br/haskell/02.lambda/	https://www.youtube.com/playlist?list=PLYItvall0TqKPbnSblJ_fxNIFRgEol-7_
2	Conceitos Básicos - Parte 1	https://haskell.pesquisa.ufabc.edu.br/haskell/03.haskell.basico.1/	https://www.youtube.com/playlist?list=PLYItvall0TqLLCPN9vbDlc8FAKhG-RfbM
2	Conceitos Básicos - Parte 2	https://haskell.pesquisa.ufabc.edu.br/haskell/04.haskell.basico.2/	https://www.youtube.com/playlist?list=PLYItvall0TqLLCPN9vbDlc8FAKhG-RfbM
3	QuickCheck	https://haskell.pesquisa.ufabc.edu.br/haskell/05.quickcheck/	https://www.youtube.com/playlist?list=PLYItvall0TqJ25sVTLcMhxsE0Hci58mpQ
3	Funções de Alta Ordem	https://haskell.pesquisa.ufabc.edu.br/haskell/06.higher.order/	https://www.youtube.com/playlist?list=PLYItvall0TqLBlT6oXFVBaloU7-xZsV-v
4	Tipos de Dados Algébricos	https://haskell.pesquisa.ufabc.edu.br/haskell/07.adt/	https://www.youtube.com/playlist?list=PLYItvall0TqJWdfiLuMslm_4ag3fJjHTo
5	Monoid e Foldable	https://haskell.pesquisa.ufabc.edu.br/haskell/08.monoids.foldable/	https://www.youtube.com/playlist?list=PLYItvall0TqLzShhLgcVvti0d0kAaJMWC
5	Functor	https://haskell.pesquisa.ufabc.edu.br/haskell/09.functors/	https://www.youtube.com/playlist?list=PLYItvall0TqKtSdeBAjz0GOM_3oq3pP8-
6	Applicative Functor e Traversable	https://haskell.pesquisa.ufabc.edu.br/haskell/10.applicatives.traverse/	https://www.youtube.com/playlist?list=PLYItvall0TqLlKtL_Cnx0g_H4Bk686e4Y
7	Monads - Parte 1	https://haskell.pesquisa.ufabc.edu.br/haskell/11.monads/	https://www.youtube.com/playlist?list=PLYItvall0TqLW_mPtIpVA8qHpuQ3YbnVO
8	Monads - Parte 2	https://haskell.pesquisa.ufabc.edu.br/haskell/11.monads/	https://www.youtube.com/playlist?list=PLYItvall0Tqll6y3FydOuEjFMH6jp7ROm
9	Avaliação Preguiçosa	https://haskell.pesquisa.ufabc.edu.br/haskell/12.laziness/	https://www.youtube.com/playlist?list=PLYItvall0TqJwLa9rY-bT_B9-ErmtaiPTO
9	Haskell Paralelo	https://haskell.pesquisa.ufabc.edu.br/haskell/13.paralelismo/	https://www.youtube.com/playlist?list=PLYItvall0TqJ1jteUbGOHfBycg5NM9thq
10	Programação Concorrente	https://haskell.pesquisa.ufabc.edu.br/haskell/14.concorrencia/	https://www.youtube.com/playlist?list=PLYItvall0TqKz0Jw8RTA2epq8VimzSqGp
10	Estruturas de dados	https://haskell.pesquisa.ufabc.edu.br/estruturas-de-dados/01.persistencia/	https://www.youtube.com/playlist?list=PLYItvall0TqI0xQzCMsK3zciIXxxgzlcG

Semana	Tópico	Material de Estudo	Playlist
	funcionais e persistência		
11	Árvores: Roseiras e Rubro-Negras	https://haskell.pesquisa.ufabc.edu.br/estruturas-de-dados//02.arvores/	https://youtube.com/playlist?list=PLYitvall0TqLE8OELzykHZOn7LaKu6-Ze
12	Heaps funcionais	https://haskell.pesquisa.ufabc.edu.br/estruturas-de-dados//03.heaps/	

7.1 Materiais das aulas práticas

Em oferecimentos anteriores da disciplina, ainda haviam aulas práticas. O material dessas podem ser vistos abaixo. Vasculhe o site em busca dos oferecimentos anteriores da disciplina em busca dos vídeos sobre sua resolução! ;)

Slides
Slides Prática 01 - Preparando o Ambiente
Slides Prática 02 - Tipos e Classes
Aula de Resolução de Exercícios
Atividade - Cifra de César
Atividade - Exercícios de recursão
Atividade - Tautologia
Atividade - Transformando Coelhos 🐰🐰 em Tomates 🍅🍅
Atividade - Monóides (Arquivo base)

8 Notas

As notas serão disponibilizadas aqui.

9 Critério de avaliação



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.

Regulamentações relevantes:

- [Código de Ética da UFABC](#)
- [Resolução ConsUni nº 63](#) - Regimento Geral da UFABC

9.1 Presença

A [resolução CONSEPE nº 139](#) estabelece no seu Artigo 2º, § 4 que nas disciplinas presenciais, a **frequência mínima obrigatória para aprovação é de 75%** das aulas ministradas e/ou atividades realizadas. Alunos que não atingirem a frequência mínima receberão conceito **O**.

- **Abonos de faltas:** Conforme [descrito no portal do MEC](#), na educação superior **não há abono de faltas**. Há, contudo, casos especiais [para alunos reservistas](#), alunos com representação na CONAES, [gestantes](#) e [em dias de guarda religiosa](#).
- **Substituição de faltas por exercícios domiciliares:** As situações em que a falta às aulas podem ser preenchidas por exercícios domiciliares são regulamentadas pelo [Decreto-Lei nº 1.044, de 21 de outubro de 1969](#), e estendidos pela [Lei nº 6.202, de 17 de abril de 1975](#). Nestes casos os alunos devem protocolar requerimento [diretamente junto à Central de Atendimento ao Estudante da Pró-reitoria de Graduação da UFABC](#). A [Resolução nº 25/2020 da CG](#) dispõe o procedimento a ser seguido para o Regime de Exercícios Domiciliares na UFABC.
- **Mecanismos de avaliação substitutivos:**
 - Atividades não presenciais: Não existe adiamento ou reposição de mecanismos de avaliação não presenciais (como, por exemplo, listas de exercícios) devido a faltas justificadas, afastamentos médicos, etc., uma vez que tais atividades tem um prazo de entrega extenso. Em particular, o [Artigo 1º, da Resolução 227 do CONSEPE](#) deixa claro que mecanismos de avaliação substitutivos só se aplicam a "avaliações presenciais".
 - Atividades presenciais: No caso de ausência justificada em uma avaliação presencial (e.g., prova escrita) através de um dos comprovantes previstos na [Resolução 227 do CONSEPE](#), o aluno terá direito a requerer uma avaliação substitutiva em uma data a ser combinada.

Regulamentações Relevantes:

- [Resolução CONSEPE nº 139](#) - Estabelece frequência mínima de 75%
- [Perguntas frequentes sobre educação superior - Secretaria de Educação Superior - MEC](#)
- [Decreto-Lei Nº 715, de 30 de julho de 1969](#) - Serviço militar
- [Lei nº 6.202, de 17 de abril de 1975](#) - Atribui à estudante em estado de gestação o regime de exercícios domiciliares
- [Lei nº 13.796, de 3 de janeiro De 2019](#) - Define prestações alternativas à aplicação de provas e à frequência a aulas realizadas em dia de guarda religiosa
- [Parecer CNE/CES 336/2000](#) e [Parecer CNE/CES 224/2006](#) do Conselho Nacional De Educação sobre abono de faltas
- [Decreto-Lei nº 1.044, de 21 de outubro de 1969](#) - Dispõe sobre tratamento excepcional para os alunos portadores das afecções que indica.
- [Resolução nº 25/2020 da CG](#) dispõe sobre procedimento a ser seguido para o Regime de Exercícios Domiciliares na UFABC.
- [Resolução CONSEPE nº 227](#) - Estabelece os critérios para mecanismos de avaliação substitutivos

9.2 Mecanismos de avaliação

A avaliação da disciplina será composta pelas seguintes notas:

- M_Q é média dos quizzes
- M_{EP} é a média dos exercícios programas

A M_Q será calculada como a média aritmética entre as notas dos quizzes 1 a 6 ($Q_1, Q_2, Q_3, Q_4, Q_5, Q_6$) conforme a fórmula abaixo:

$$M_Q = \frac{Q_1 + Q_2 + Q_3 + Q_4 + Q_5 + Q_6}{6}$$

Os quizzes substitutivos substituem algum quiz que tenha sido perdido (com justificativa).

A M_{EP} é a média ponderada dos dos exercícios programas (EP_1 e EP_2) com pesos 2 e 3 respectivamente, conforme a fórmula abaixo:

$$M_{EP} = \frac{2EP_1 + 3EP_2}{5}$$

A nota final (N_F) será determinada pela **média harmônica ponderada** de M_Q e M_{EP} com pesos 2 e 3 respectivamente, conforme a fórmula abaixo:

$$N_F = \frac{5}{\frac{2}{\max\{0,1,M_Q\}} + \frac{3}{\max\{0,1,M_{EP}\}}}$$

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

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

Warning

Caso seja verificado ocorrência de plágio nos EPs ou nos quizzes, o aluno será automaticamente reprovado com F.

1. 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 avaliação escrita a ser marcada durante o Q3. 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 $C_F = F$:

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

2. Regulamentações Relevantes

- [Resolução ConsEPE nº 182](#) - Regulamenta a aplicação de mecanismos de recuperação
- [Resolução ConsUni nº 63](#) - Regimento Geral da UFABC

10 Recursos Online

10.1 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

10.2 Disciplinas em Haskell

- MCTA016-13: Paradigmas de Programação (em Haskell), UFABC. [2023](#), [2022](#), [2021](#), [2020](#), [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](#)

10.3 Leituras

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

10.4 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](#)

11 Bibliografia

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



[GH]

- ***Programming in Haskell***. 2nd Edition.
 - Por *Graham Hutton*.

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



1st Edition (antiga)

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



[SGS]

- **Real World Haskell**
 - Por *Bryan O'Sullivan, John Goerzen e Don Stewart*.
 - Disponível **gratuitamente** em: <http://book.realworldhaskell.org/>
-



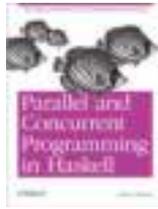
[ML]

- **Learn You a Haskell for Great Good! A Beginner's Guide.**
 - Por *Miran Lipovača*.
 - Disponível **gratuitamente** em: <http://learnyouahaskell.com/>
-



[HIPE]

- **Haskell: Uma introdução à programação funcional**
 - Por *Alexandre Garcia de Oliveira*
 - Disponível em: https://www.casadocodigo.com.br/products/livro-haskell?_pos=2&_sid=8ab00a083&_ss=r
-



[SM]

- ***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/9781449335939/>
-

