

# Apresentação do Curso

MCTA016-13 - Paradigmas de Programação

---

Emilio Francesquini

[e.francesquini@ufabc.edu.br](mailto:e.francesquini@ufabc.edu.br)

2019.Q2

Centro de Matemática, Computação e Cognição  
Universidade Federal do ABC



- Estes slides foram preparados para o curso de **Paradigmas de Programação na UFABC**.
- Este material pode ser usado livremente desde que sejam mantidos, além deste aviso, os créditos aos autores e instituições.



# Apresentação

---

- Prof. Dr. Emilio Francesquini
- [e.francesquini@ufabc.edu.br](mailto:e.francesquini@ufabc.edu.br)
- <http://professor.ufabc.edu.br/~e.francesquini>
- Santo André, Bloco A, Sala 531-2

Todas as informações relativas à disciplina tais como:

- Datas importantes
- Critérios de avaliação
- Bibliografia
- Avisos
- ...

Estarão disponíveis em:

**http:**

[//professor.ufabc.edu.br/~e.francesquini/2019.q2.paradigmas/](http://professor.ufabc.edu.br/~e.francesquini/2019.q2.paradigmas/)

Ou simplesmente busque pelo meu nome e ache o link na minha página.

## Multitasking

*Attention, multitaskers (if you can pay attention, that is): Your brain may be in trouble.*

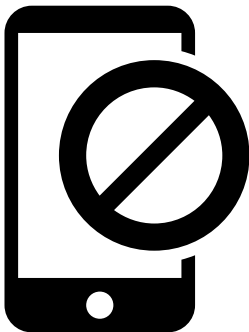
*People who are regularly bombarded with several streams of electronic information do not pay attention, control their memory or switch from one job to another as well as those who prefer to complete one task at a time, a group of Stanford researchers has found.*

*(...)*

*So maybe it's time to stop e-mailing if you're following the game on TV, and rethink singing along with the radio if you're reading the latest news online. By doing less, you might accomplish more.*

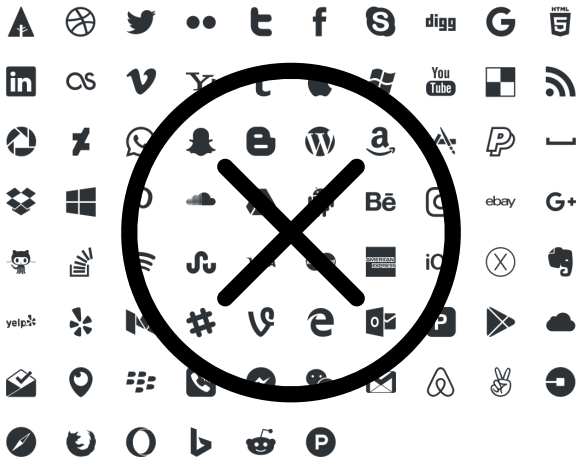
<http://news.stanford.edu/2009/08/24/multitask-research-study-082409/>

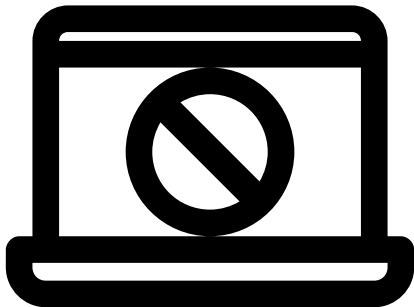
- Veja o vídeo de Clifford Nass (Stanford) em <https://youtu.be/PriSFBu5CLs>
- Se render às distrações do mundo digital (e-mail, mensagens instantâneas, Facebook, etc.) faz o cérebro lançar pequenas doses de dopamina
- Com o tempo, ficamos viciados nisso
- Resultado: *multitaskers* gastam muito mais poder de processamento cerebral do que *monotaskers* quando são distraídos
- Efeitos a longo prazo são difíceis de reverter



No Phone by Rflor from the Noun Project







blocked laptop by unlimicon from the Noun Project

Contudo, em algumas aulas haverá demonstrações de uso de tecnologias de programação. Nestes momentos aqueles que quiserem seguir em seus notebooks estão liberados.

**Avise seu professor o quanto antes** sobre a necessidade de cuidados extras para acessibilidade nos casos de deficiência:

- visual,
- física,
- auditiva,
- dislexia,
- etc.

<http://proap.ufabc.edu.br/>

As aulas de teoria pela manhã são iguais às da noite do mesmo dia. Caso deseje você pode assistir a aula no horário mais conveniente (**contanto que haja espaço na sala**).

### Turma Diurno (DA2MCTA016-13SA)

- Quartas das 10:00 às 12:00, Sala 409-2 (P)
- Sextas das 08:00 às 10:00, Sala S-008-0 (T)

### Turma Noturno (NA2MCTA016-13SA)

- Quartas das 21:00 às 23:00, Sala 407-2 (P)
- Sextas das 19:00 às 21:00, Sala S-006-0 (T)

## ■ Presencial

### ▶ Horários de atendimento

- Nos horários listados abaixo não é preciso confirmar ou marcar, apenas apareça! :-)
- Quinta-feira, das 17:00 às 20:00, Sala 531-2.
- Sexta-feira, das 10:00 às 12:00, Sala 531-2.

### ▶ Agendado por e-mail

- Verifique minha agenda e sugira pelo menos dois possíveis horários!

### ▶ Em sala de aula - Após as aulas

## ■ Online

### ▶ Por e-mail.

# A Disciplina

---

- TPI: 2-2-4
- Recomendação: Processamento da Informação, Programação Orientada a Objetos

## Objetivos

Esta disciplina traz à atenção do aluno as diversas diferenças fundamentais entre grandes famílias de linguagens de programação, tanto em teoria como de forma prática. Esta visão tem efeitos importantes, nem sempre perceptíveis: ao apreciar diversas técnicas de programação e mecanismos peculiares de linguagens de programação diferentes, o estudante expande seu leque de técnicas e rompe sua rigidez de concepção a respeito do que vem a ser programar. Além disso, há situações onde um paradigma se aplicará com mais sucesso do que outro. Pode-se sem dúvida afirmar que a programação em diferentes paradigmas auxilia na melhoria da qualidade da programação de forma geral.

- TPI: 2-2-4
- Recomendação: Processamento da Informação, Programação Orientada a Objetos

## Conteúdo Programático

Visão comparativa entre os paradigmas de programação. Paradigma funcional. Paradigma concorrente.



- "Não fiz a disciplina **XPTO** dá pra fazer sem?"
  - ▶ Processamento da Informação → **Não**
  - ▶ Programação Orientada a Objetos → **Com esforço**
  - ▶ Programação estruturada/AED... → Ajudaria muito!
  - ▶ ...
- A disciplina **exige** conhecimentos de desenvolvimento de software
  - ▶ Se você acha que vai conseguir escrever na gambiarra o projeto ou as atividades semanais, pode tirar o cavalinho da chuva
  - ▶ Utilizaremos exclusivamente a **linguagem de programação Haskell**.

- Nunca programou em C, Java, Python, ...
- Não fez Processamento da Informação ou Programação Orientada a Objetos
- Não consegue se concentrar em um só problema e quebrar a cabeça por horas para resolvê-lo
- Não se sente orgulhoso por finalmente resolver um problema (que você nem consegue explicar qual é pra sua família) depois de horas de trabalho

Então você terá **MUITA** dificuldade neste curso!

- **Comece a estudar agora!**

# Avaliação

---

Qualquer tentativa de fraude nas provas, listas de exercícios ou projetos implicará:

- **Conceito final CF = F (reprovado)** para **TODOS** os envolvidos.
- **Possível denúncia apresentada 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.



- Diversos professores se reuniram e escreveram um **Código de Honra**
  - ▶ <http://professor.ufabc.edu.br/~e.francesquini/codigodehonra/>
- Nesta disciplina seguiremos este código
- Leiam o texto completo e, em caso de dúvidas, perguntem ao professor

## 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 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 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.



A avaliação da disciplina será composta por três notas principais: provas, atividades e projeto. Considere:

- $N_F$  é a nota final;
- $N_{\text{Atividades}}$  é a nota das atividades semanais;
- $N_{\text{Provas}}$  é a nota da prova;
- $N_{\text{Projeto}}$  é a nota do projeto.

A nota da teoria  $N_{\text{Provas}}$  será formada por duas provas  $P_1$  e  $P_2$ . Todas as provas serão efetuadas em sala de aula, sem qualquer tipo de consulta.

Haverá também uma prova substitutiva  $P_5$  que será aberta a todos os interessados, ainda que eles tenham feito tanto a  $P_1$  quanto a  $P_2$

## Atenção

A nota da  $P_5$  será utilizada obrigatoriamente em substituição à menor nota entre  $P_1$  e  $P_2$  ainda que isto diminua a nota final do aluno!

Assim, a nota de provas ( $N_{\text{Provas}}$ ) será calculada conforme a seguinte fórmula:

$$N_{\text{Provas}} = \begin{cases} \frac{2 \cdot P_S + 3 \cdot P_2}{5}, & \text{caso tenha feito a } P_S \text{ e } P_1 < P_2 \\ \frac{2 \cdot P_1 + 3 \cdot P_S}{5}, & \text{caso tenha feito a } P_S \text{ e } P_2 \leq P_1 \\ \frac{2 \cdot P_1 + 3 \cdot P_2}{5}, & \text{caso contrário} \end{cases}$$

- Haverá uma atividade de laboratório por semana, totalizando 10 atividades ao longo do quadrimestre.
- O enunciado de cada atividade será divulgado toda quarta antes da aula prática no repositório da disciplina no GitHub.
- A solução dessas atividades deverá ser entregue **apenas pelo Github**, até o domingo da semana seguinte à liberação do enunciado (11 dias de prazo).

- A nota de cada laboratório poderá chegar a 10 caso o programa execute corretamente em todos os casos de teste e será 0 caso contrário.
- Exemplos de problemas que causam redução da nota: código mal escrito, nomes de variáveis/funções pouco claros, desorganização do código.
- Cada atividade terá um peso específico, divulgado em seu enunciado, que irá variar de acordo com sua complexidade.
- Não serão aceitas soluções fora do prazo.
- Não haverá laboratórios substitutivos.
- Essas atividades devem ser feitas individualmente.

O enunciado do projeto será anunciado em breve. Ele poderá ser feito em grupos de até 4 pessoas. A sua nota será dada pela qualidade do código (organização, documentação, ...), complexidade do projeto, documentação, etc.

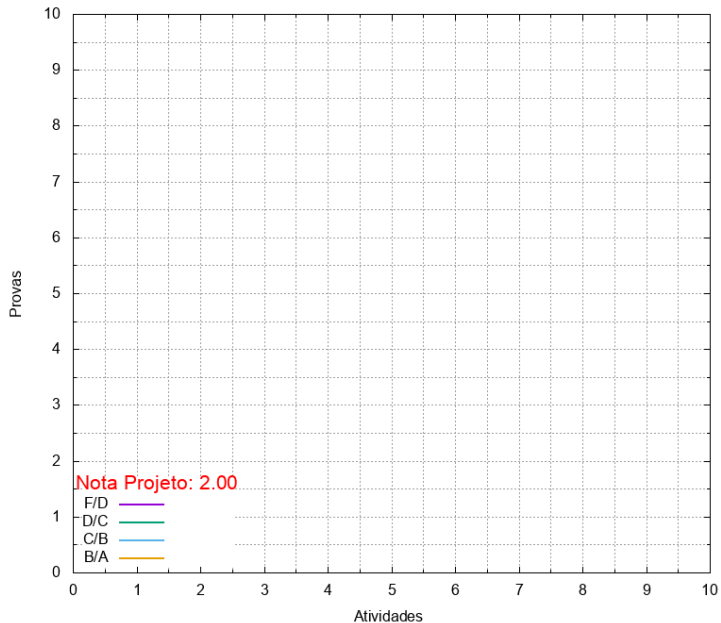
A nota final ( $N_F$ ) será determinada pela média harmônica ponderada de  $N_{\text{Atividades}}$ ,  $N_{\text{Provas}}$  e  $N_{\text{Projeto}}$  com pesos 4, 3 e 3 respectivamente:

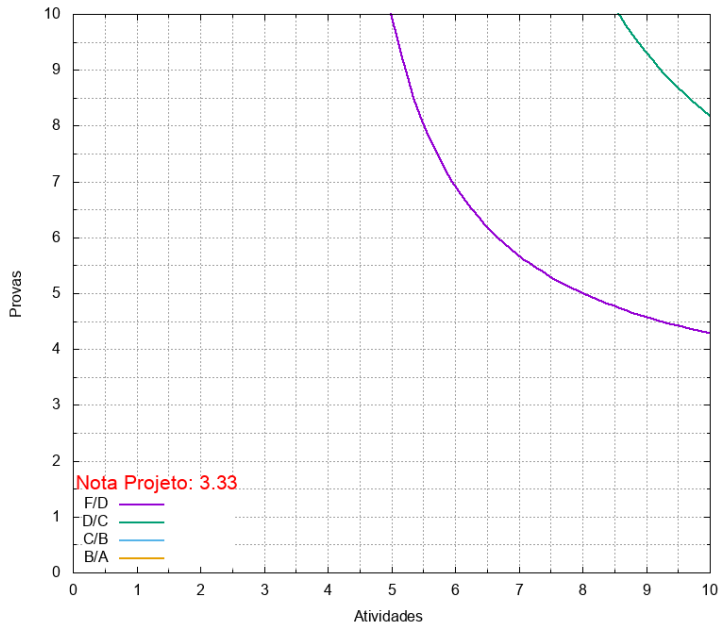
$$N_F = \frac{10}{\frac{4}{\max\{0.1, N_{\text{Atividades}}\}} + \frac{3}{\max\{0.1, N_{\text{Provas}}\}} + \frac{3}{\max\{0.1, N_{\text{Projeto}}\}}}$$

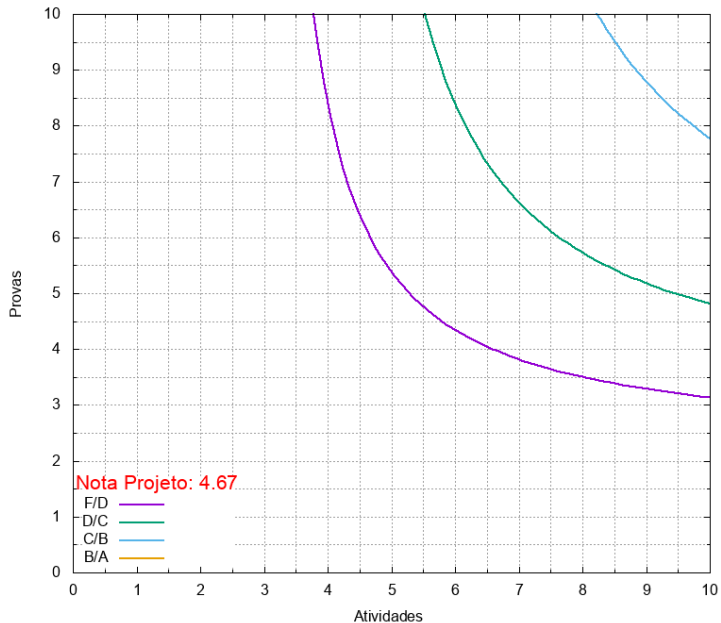
O conceito final ( $C_F$ ) será obtido de acordo com a equação abaixo:

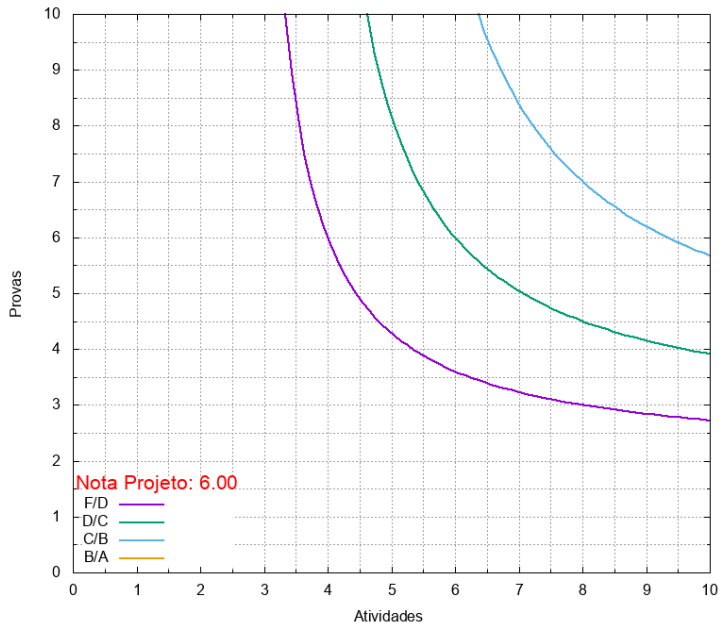
$$C_F = \begin{cases} \text{O,} & \text{se ausência total exceder 25\%} \\ \text{F,} & \text{se } N_F \in [0, 0; 5, 0) \\ \text{D,} & \text{se } N_F \in [5, 0; 6, 0) \\ \text{C,} & \text{se } N_F \in [6, 0; 7, 0) \\ \text{B,} & \text{se } N_F \in [7, 0; 8, 5) \\ \text{A,} & \text{se } N_F \in [8, 5; 10, 0) \end{cases}$$

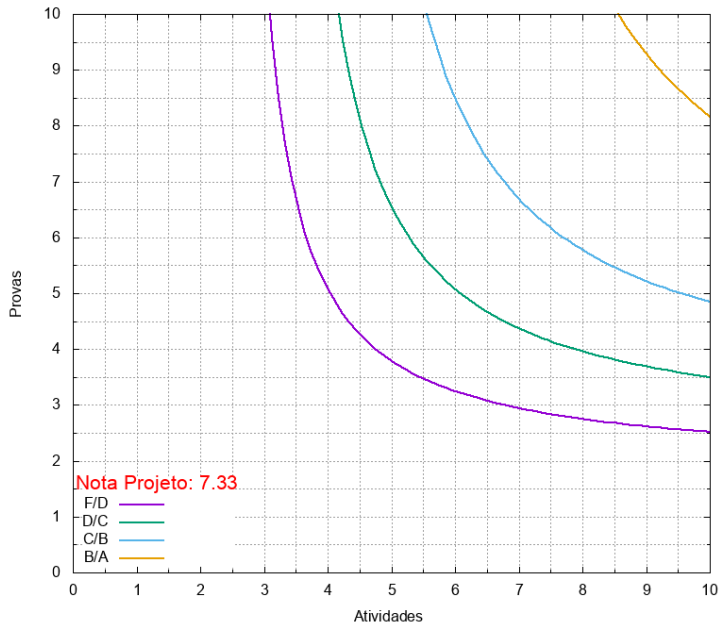


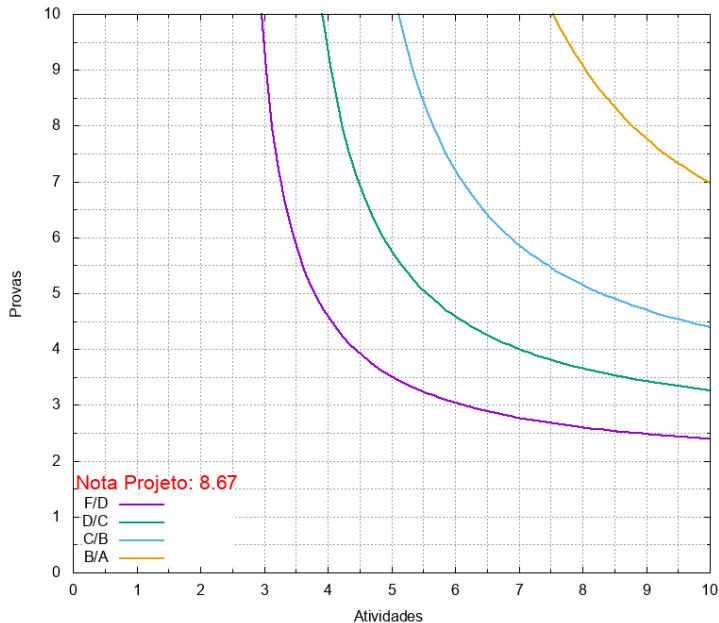


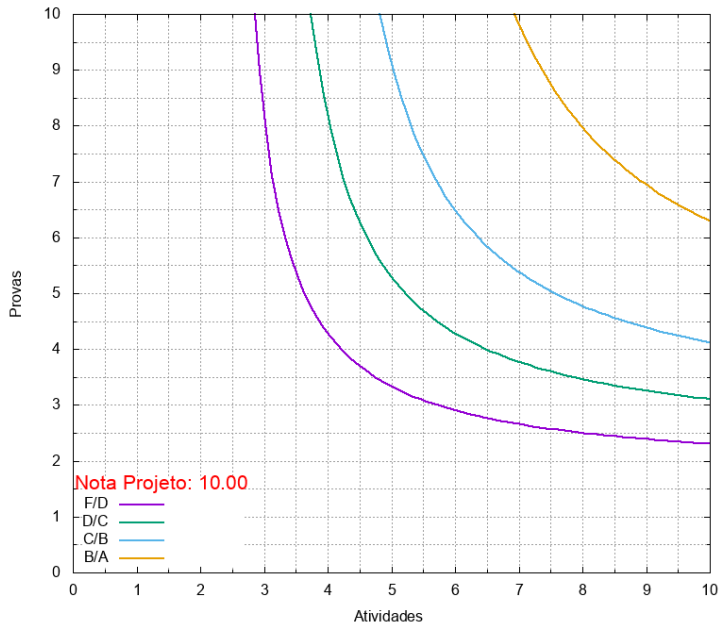












Conforme Resolução ConsEPE nº 182 fica assegurado a todos os alunos com  $C_F$  igual a **D** ou **F** o direito a fazer uso de mecanismos de recuperação.

A recuperação será feita através de uma prova  $P_R$ , sem consulta, e a sua nota será utilizada para compor a o conceito pós-recuperação  $C_R$  conforme as equações abaixo:



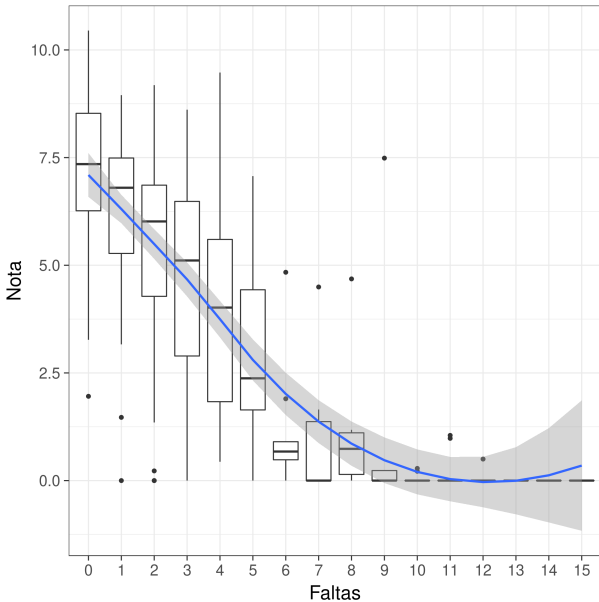
$$N_R = \frac{P_R + N_F}{2}$$

Caso 1  $C_F = D$ :

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

Caso 2  $C_F = F$ :

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



- Prova 1 - 19/07/2019
- Prova 2 - 23/08/2019
- Projeto - 25/08/2019
- Prova Substitutiva - 28/08/2019
- Prova de Recuperação - A definir (Q3 2019)

- Resolução ConsEPE nº 227 - Regulamenta a aplicação de mecanismos de avaliação substitutivos
- Resolução ConsEPE nº 182 - Regulamenta a aplicação de mecanismos de recuperação
- Código de Ética da UFABC
- Resolução ConsUni nº 63 - Regimento Geral da UFABC

# Recursos Online

---

- Grupo de Estudos em Haskell da UFABC
  - ▶ Não esqueça de assinar a lista de discussão: <https://groups.google.com/d/forum/haskell-brasil>
- 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

- MCTA016-13: Paradigmas de Programação (em Haskell), UFABC. [2018](#).
- CR062-2018: Programação Funcional em Haskell, UFABC. [2018](#).
- 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](#)

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



Para os primeiros exercícios semanais, os ambientes online podem ser suficientes. Mais adiante no curso você provavelmente vai querer um ambiente local na sua máquina

- Online

- ▶ repl.it - <https://repl.it/>
- ▶ CodeWorld - <https://code.world/haskell>
- ▶ Try Haskell - <http://tryhaskell.org/>

- Local

- ▶ Ambiente de programação (Linux)
- ▶ The Haskell Tool Stack
- ▶ Haskell Platform (Mac OS X, Windows, Linux)

- 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](#)

# Bibliografia

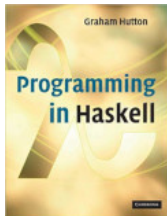
---

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

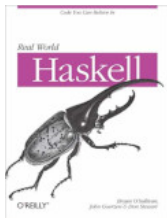


- **Programming in Haskell.** 2nd Edition.
  - ▶ Por *Graham Hutton*.

A primeira edição (antiga), 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](http://biblioteca.ufabc.edu.br/index.php?codigo_sophia=15287)

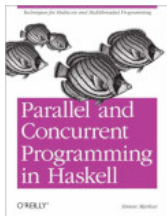


## ■ Real World Haskell.

- ▶ 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/9781449335939/>