

Programação e Desenvolvimento de Software I

Apresentação do Curso

Prof. Héctor Azpúrua
(slides adaptados do Prof. Pedro Olmo)

Introdução

Objetivos

- Introduzir o aluno aos conceitos de:
 - Algoritmos
 - Estruturas de dados
- Noções da organização e funcionamento de um computador
- Noções de linguagens imperativas
- Noções de estruturas de dados

Introdução

Por que aprender programação?

- Porque ***quase todas as profissões*** atualmente interagem com um computador
- Interação cada vez mais sofisticada:
 - Administração: planilhas Excel com macros
 - Física, química, matemática: uso de ferramentas como Octave, Scilab, Matlab, ...
 - Biologia: simuladores para criar novas moléculas

Introdução

Por que aprender programação?

- Será necessário para outras disciplinas
 - PDS II
 - Circuitos digitais (pode ser)
 - Análise numérica
 - Redes de computadores
 - Informática industrial
 - Controle digital
 - Pesquisa operacional
 - Optativas:
 - bancos de dados, redes de computadores, visão computacional, ...

○ nosso curso

- Aulas teóricas e práticas
- Aulas teóricas:
 - Apresentação dos conceitos
- Aulas práticas:
 - Resolução de listas de exercícios
 - Programas a serem desenvolvidos ou exercícios relacionados

O nosso curso

Aulas práticas

- Aulas no laboratório **1009 e 1010 do ICEx**
- Datas marcadas de **azul** no cronograma
- Aulas práticas com avaliação:
 - 1 semana de prazo para entrega da solução
 - Correção automática via VPL

O nosso curso

Avaliações

- 3 provas individuais
 - Primeira: 15 pontos
 - Segunda: 30 pontos
 - Terceira: 30 pontos
- Exercícios práticos
 - 10 pontos de práticas semanais
 - Pode ser em dupla, mas cada um entrega o seu
- Trabalho prático
 - 15 pontos
 - Individual
 - Possibilidade de vários pontos extras

O nosso curso

Provas

- Conteúdo base: livro, transparências e exercícios
- Dou dicas sobre a prova durante as aulas
 - Comentários em sala de aula
 - Exercícios parecidos
- A resposta pode não estar no material, mas a matéria lecionada é a base teórica para resolvê-la

O nosso curso

Provas

- Síncronas (presenciais)
- Duração: 1 hora e 40 minutos
- No papel

O nosso curso

Revisão da correção das provas

- Até duas semanas depois da entrega da nota da avaliação:
 - Evitar choradeira no fim do semestre!
 - Se gostou da nota quando recebeu a correção, tem que gostar dela no fim do semestre também!
 - Importante revisar, pois a correção automática pode levar a erros injustos (ex: na sexta casa decimal)

O nosso curso

Práticas regulares

- Entrega no Moodle: arquivos `.c` e `.h`
- Pontuação do tipo “**entregou, levou**” 😊
- Há vídeos com as correções de todos os exercícios práticos:
 - <https://docs.google.com/document/d/1eb5T6OouGTbBMNeXZfzVXEhrhMXNfeuz45Ijlb-3lBI/edit?usp=sharing>

O nosso curso

Práticas avaliativas

- Avaliativas (treino para a prova)
- Correção automática no VPL do Moodle
- Pontuação depende da correção das questões
- Mesmo formato das provas

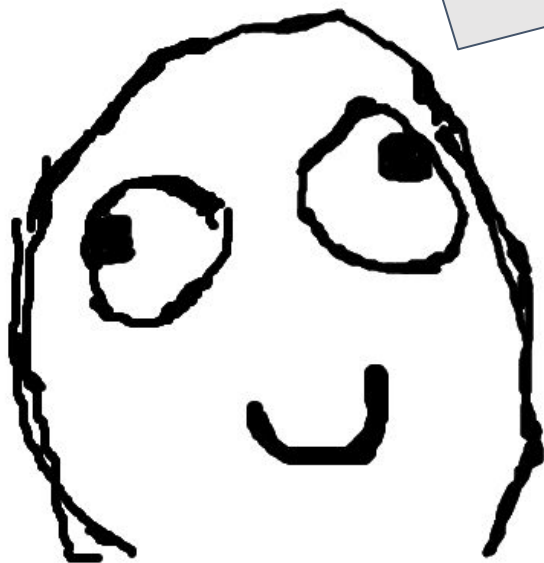
Trabalho prático

- Momento para realizar um projeto mais longo e complexo
- Julgamento do código
 - Comentários, facilidade de leitura, indentação
- Julgamento da documentação
 - Estruturação, clareza e coesão, conteúdo...

Trabalho prático

Algumas ideias sobre o trabalho

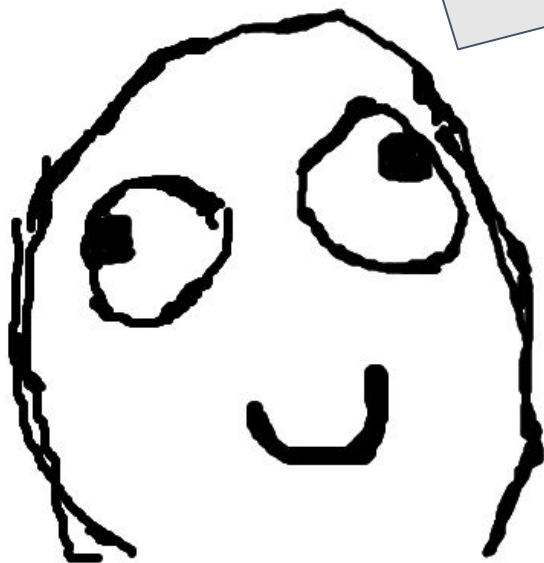
“É uma excelente oportunidade para desenvolver a capacidade de aprender novas ferramentas.”



Trabalho prático

Algumas ideias sobre o trabalho

“Permite que o aluno se divirta desenvolvendo um trabalho prático.”



Trabalho prático

Algumas ideias sobre o trabalho

“Permite o desenvolvimento de um trabalho maior, mais complexo, além de incentivar a criatividade do aluno.”



Trabalho prático

Algumas ideias sobre o trabalho

“TP totalmente não condizente com a realidade, fruto de um devaneio de um profissional frustrado.”



Trabalho prático

Avaliação do trabalho prático

- $\text{nota} = (\text{nota TP} + \text{ponto extra}) * \text{nota av oral}$
- Até 15 pontos para o trabalho básico considerando os itens mencionados anteriormente

Trabalho prático

Avaliação do trabalho prático

- Prova oral consistirá de X (ex: 3) perguntas diretas sobre o código
- Nota da arguição = número de perguntas respondidas corretamente em tempo hábil dividido por X
 - Ex: se $x = 3$, valores possíveis: 1, $2/3$, $1/3$, 0
- Se você sabe o que fez no TP, vai tirar 1.0 na prova oral! 😊
- No entanto, vários alunos já tiraram 0 e $1/3$ na prova oral 😞

Trabalho prático

Avaliação do trabalho prático

- Pontos extras só serão dados aos alunos que ficarem com mais de 50% dos pontos nas provas (mais de 36 pontos)
 - 36 não é $> 50\%$, não adianta chorar!
- Pontos extras **SÓ** servem para aumentar o conceito de:
 - D para C
 - C para B
 - B para A
- Ou seja, **não espere passar com os pontos extras**
 - ~~E para D!~~

Trabalho prático 2013-I

Angry birds



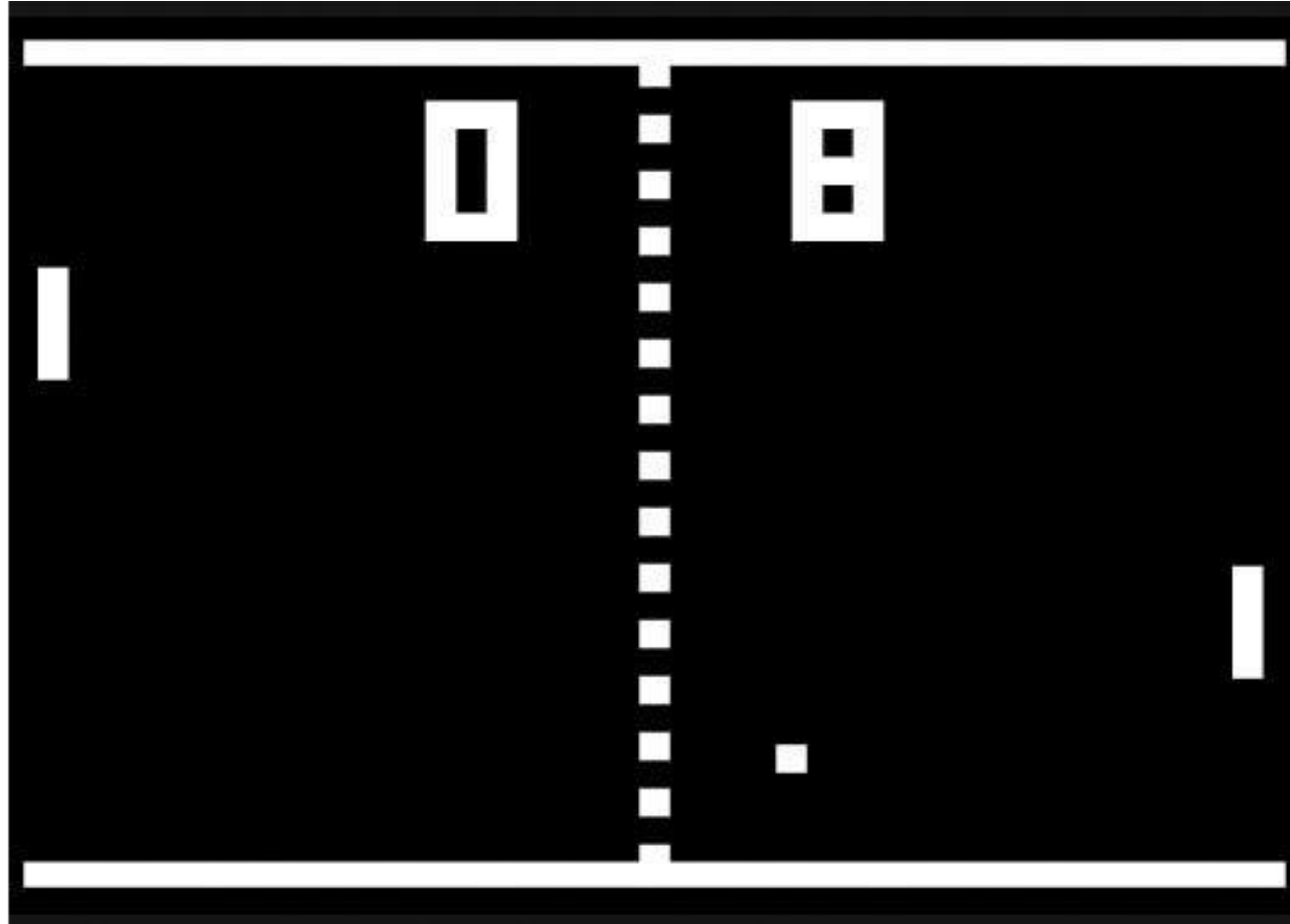
Trabalho prático 2013-2

Frogger



Trabalho prático 2014-I

Pong



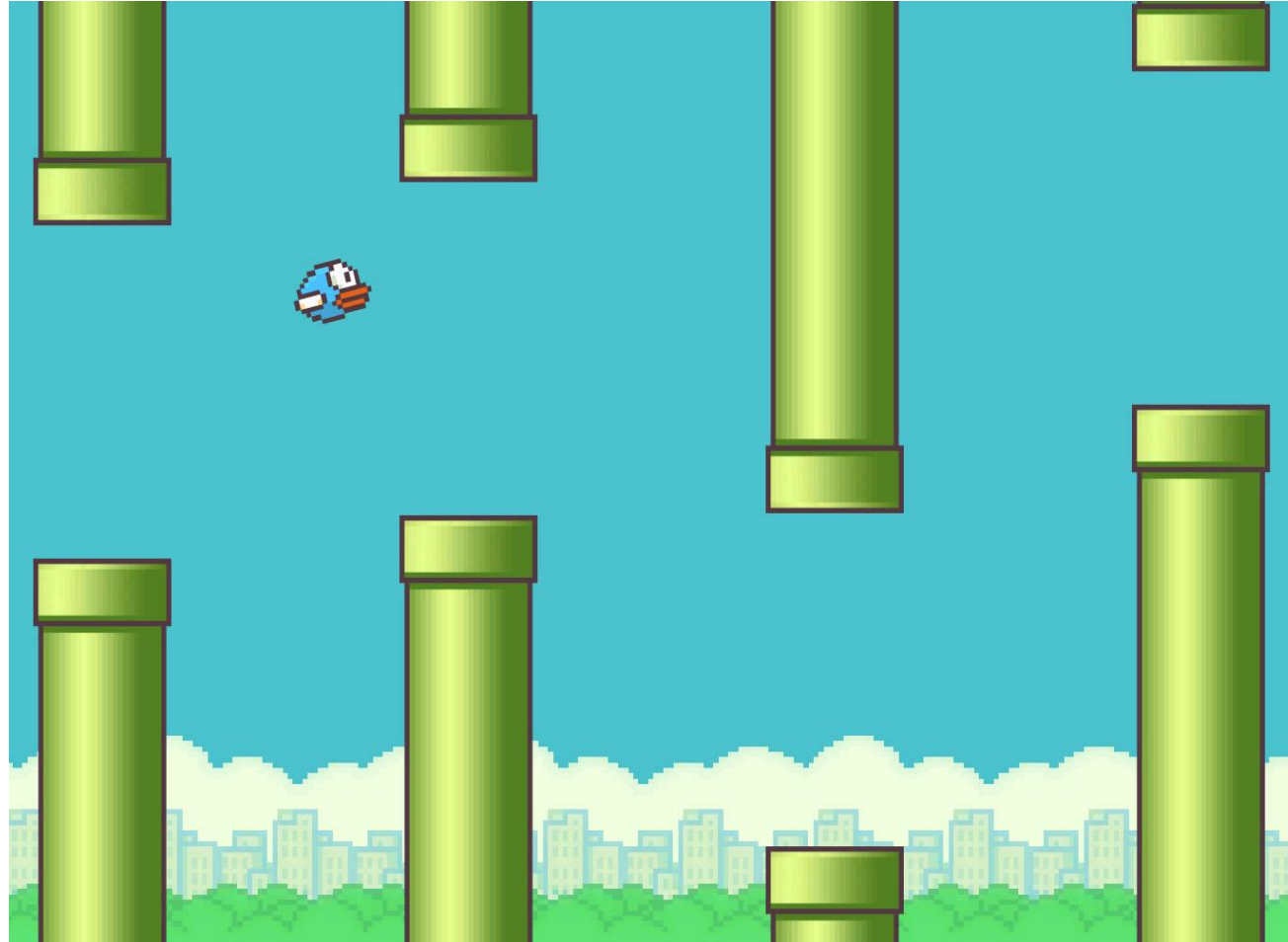
Trabalho prático 2014-2

Air combat



Trabalho prático 2015-I

Flappy Bird



Trabalho prático 2015-2

Zelda



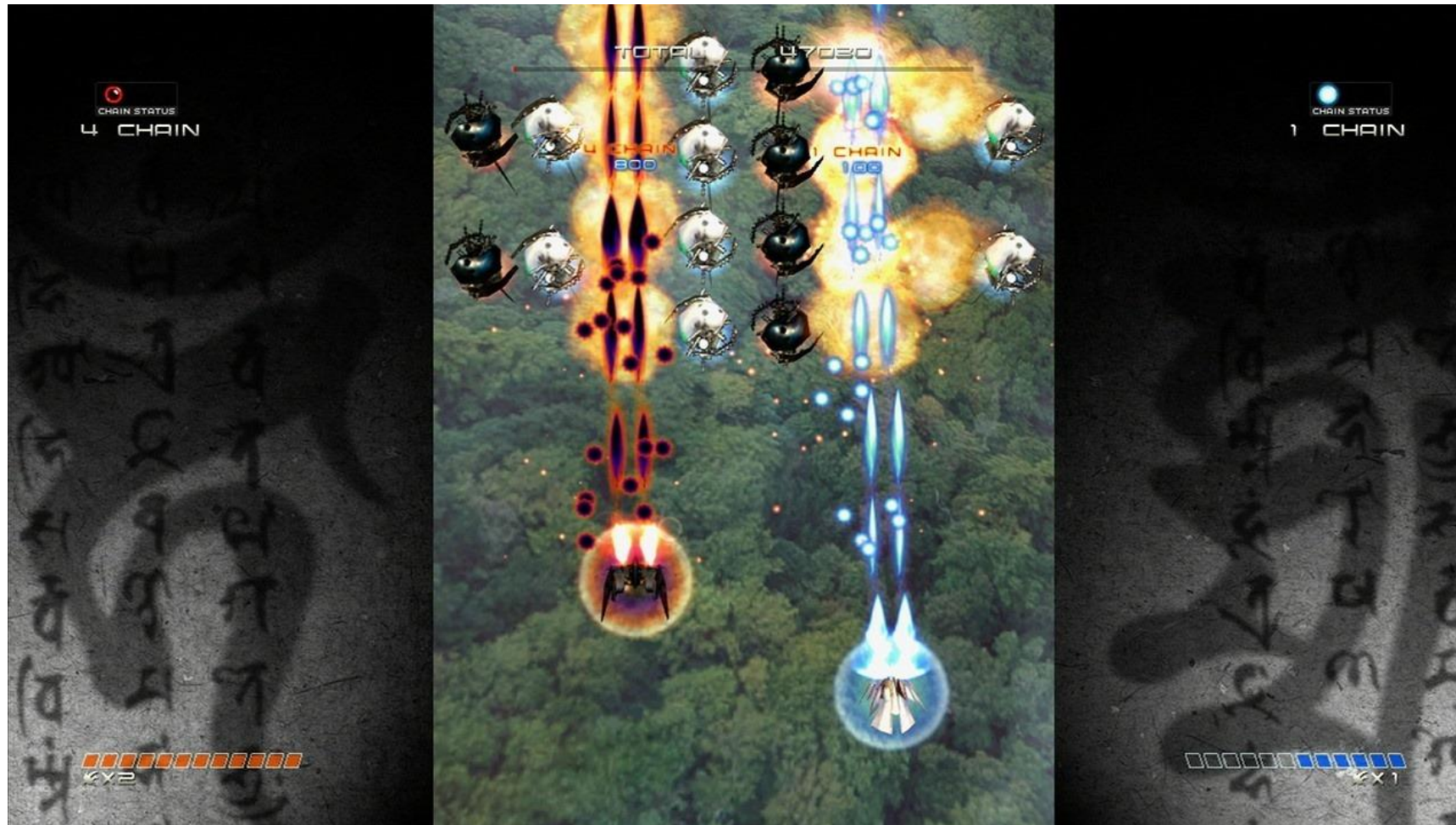
Trabalho prático 2016-I

Bomberman



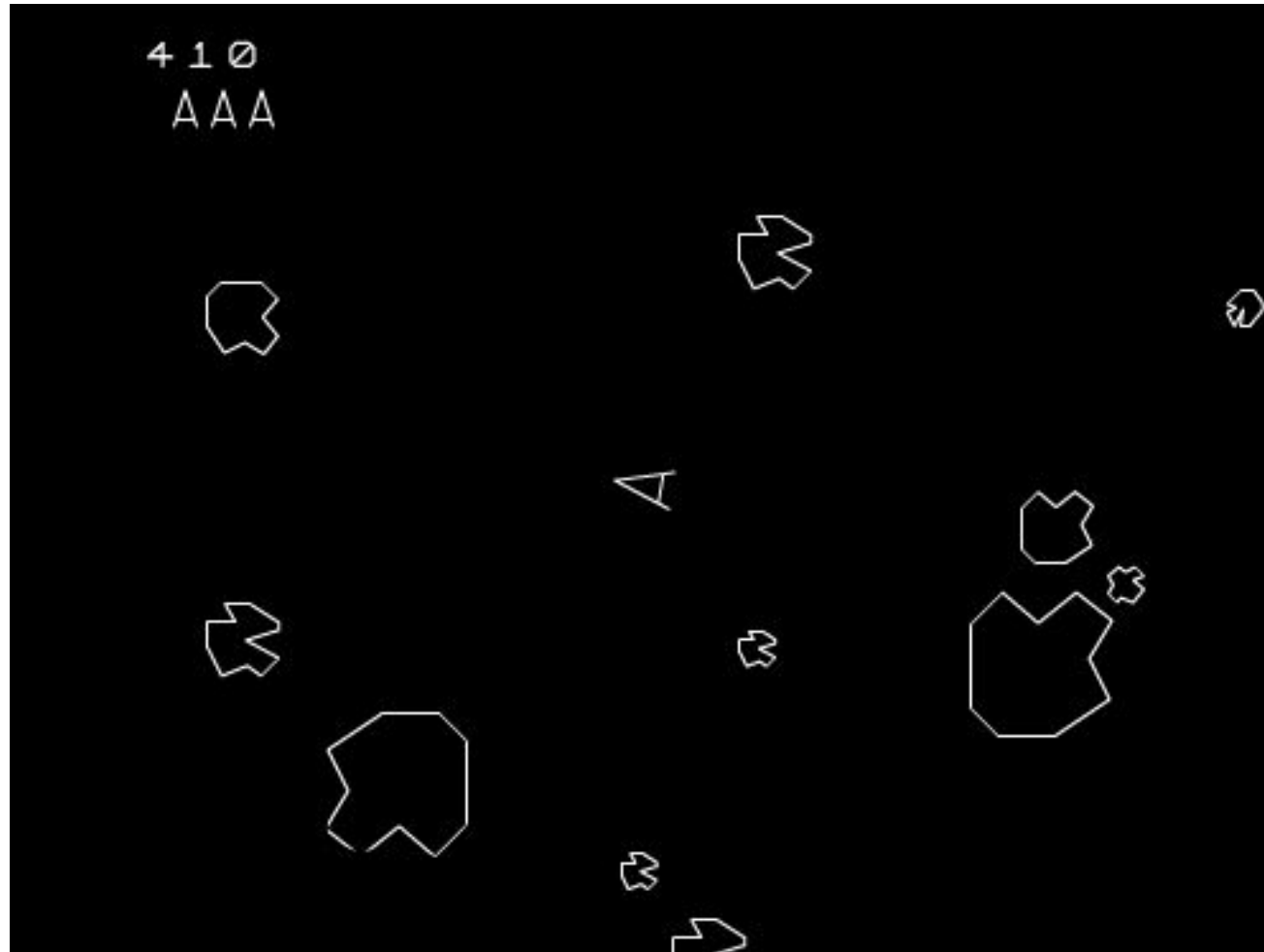
Trabalho prático 2016-2

Ikaruga



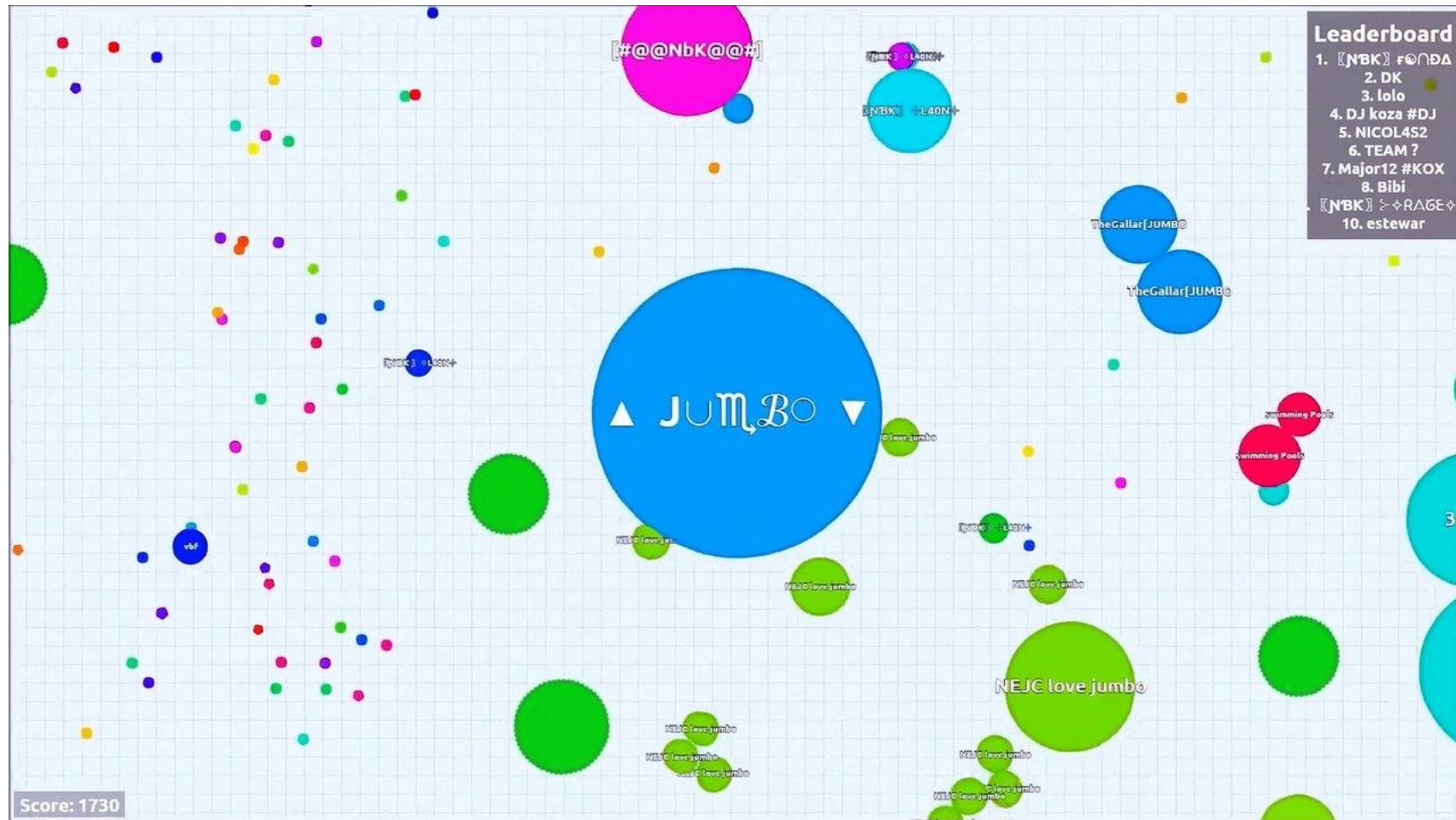
Trabalho prático 2017-I

Asteroids



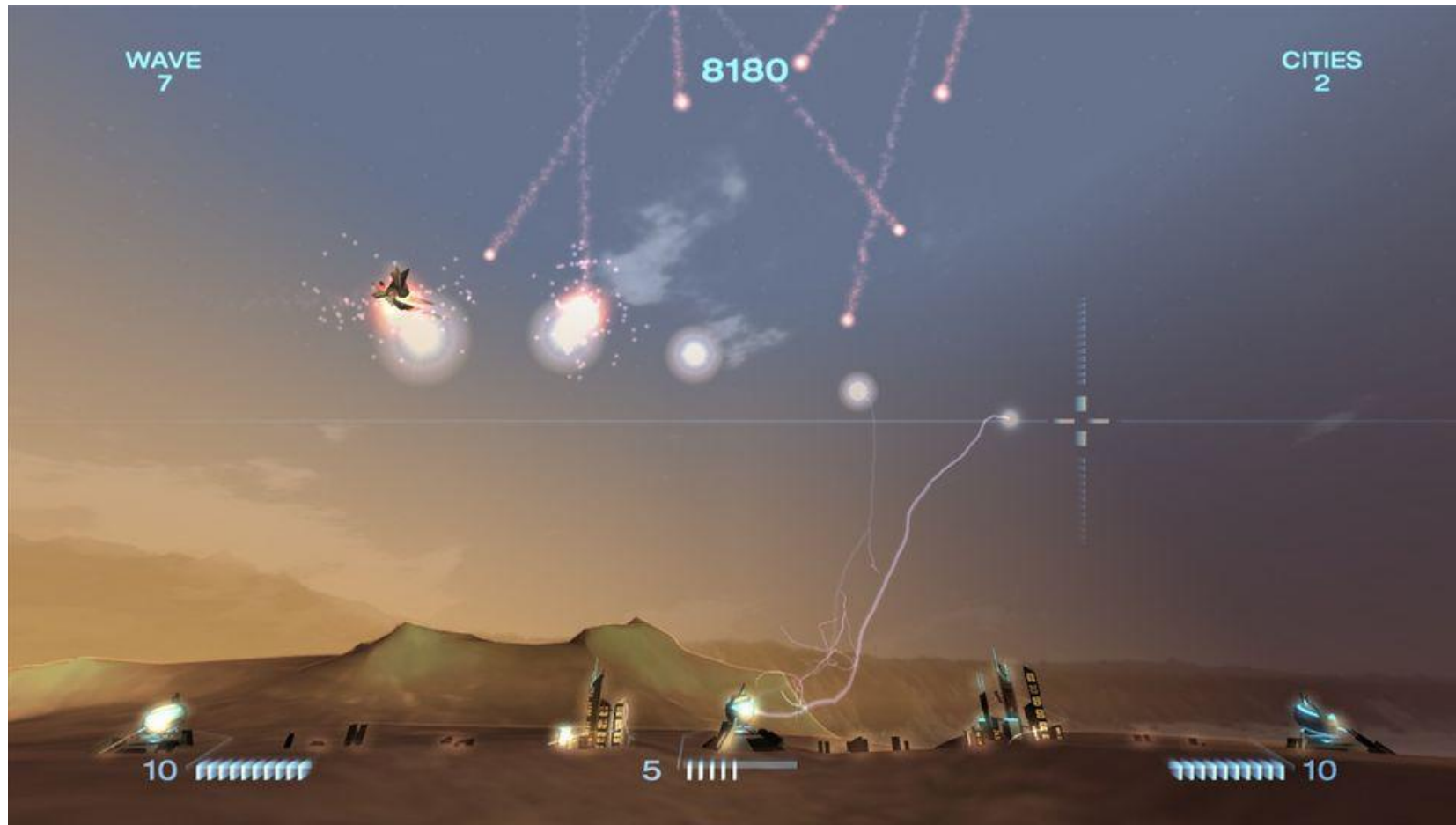
Trabalho prático 2017-2

Agar.io



Trabalho prático 2018-I

Missile Command



Trabalho prático 2018-2

Enduro (Atari)



Trabalho prático 2019-I

Guitar Hero



Trabalho prático 2021-I

Final Fantasy VI



R-Type



Trabalho prático deste semestre

Space Invaders



Trabalho prático deste semestre

Motivação

tec

'Flappy Bird' chega a 50 milhões de downloads e rende US\$ 50 mil por dia

DE SÃO PAULO

08/02/2014 @ 12h10

[Recomendar](#) 178 [Twitter](#) 51 [G+1](#) 4 [OUVIR O TEXTO](#) [Mais opções](#)

Com uma dificuldade absurda disfarçada em gráficos simples, o jogo "Flappy Bird" virou fenômeno.

Primeiro em diversas categorias na App Store e no Google Play, ele já foi baixado 50 milhões de vezes e acumula mais de 47 mil avaliações.

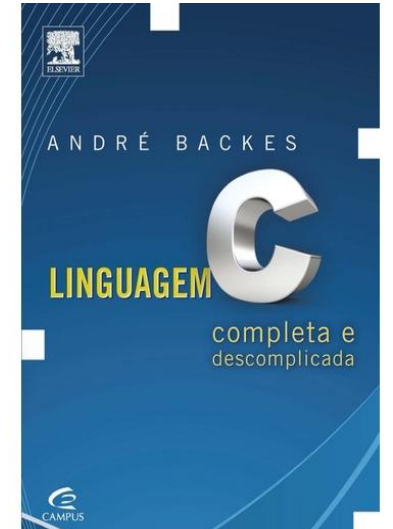
Dong Nguyen, o vietnamita que criou o jogo, diz faturar US\$ 50 mil por dia com publicidade.

O desenvolvedor é independente e atingiu o sucesso sem a ajuda de um grande estúdio. ★ ★ ★



Bibliografia

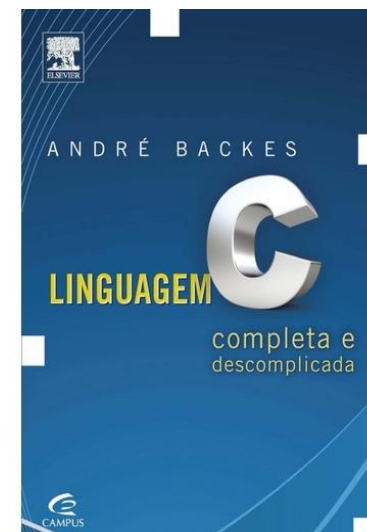
- Livro-texto:
 - **Linguagem C completa e descomplicada**, André Backes
- Outros:
 - **Introdução às Estruturas de Dados**, Waldemar Celes
 - **Projeto de Algoritmos com implementação em PASCAL e C**, 3a edição, Nivio Ziviani
 - **Algoritmos estruturados**, 3a edição, Harry Farrer, Becker, Faria, Matos, dos Santos, Maia



Linguagem do curso

Linguagem C

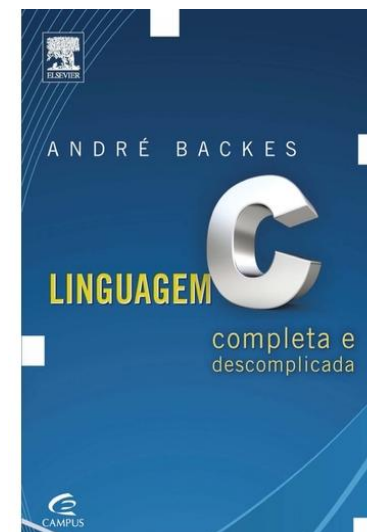
- Criada em 1972
 - 47 anos de idade
 - Para computação → bem velho
- Existem linguagens mais recentes?
 - Sim!!



Linguagem do curso

Linguagem C

- A escolha de uma linguagem para aprender a programar é um problema complicado...
- C serve para entender o funcionamento do computador melhor
 - Baixo nível
- Com o tempo alunos podem usar o conhecimento para aprender outras linguagens
 - Depois de bastante prática, **chavear linguagens é algo simples**
- Plano didático da UFMG é quase todo em C



Notas e frequência

- **Não reprovado por frequência:**
 - **SE** o aluno tiver aproveitamento superior a 60%
- Se for infrequente (frequência < 75%):
 - Não ajuda a mudar de conceito
 - Não ajuda a passar (mesmo que seja por 1 ponto)
 - Não tem direito a exame especial
 - Infrequente e nota menor que 60: conceito F

Notas e frequência

- Listas de presença em todas as aulas
- Não precisa vir à aula
 - Se quiser estudar em casa
 - Se quiser ficar conversando com os colegas
 - **É só tirar 60 pontos ou mais... :-)**
- As aulas são importantes:
 - Posso dar dicas sobre questões de prova
 - Posso propor um exercício parecido ao da prova
 - Posso mencionar algo que não está no livro, mas cai na prova

Exame especial

- **Especial**, como diz o nome:
 - Difícil: matéria do semestre inteiro
 - Muitas questões: preciso avaliar a matéria como um todo
- Sugestão: **evitem fazer o exame especial**
 - É mais fácil passar com as provas e trabalhos
 - Férias começam mais cedo

Extra-classe

- Teremos monitores! 🙏
- Estou disponível fora do horário de aula
 - Marcando horário
 - Por e-mail: melhor para notas/correções
 - hector@dcc.ufmg.br
 - Sala 4326 anexo U do ICEx
- Fórum do Moodle/Minha UFMG: dúvidas sobre a matéria/programação

É fácil aprender programação?

Não é...

Desenvolvimento > Linguagens de programação > Rust (linguagem de programação)

Domine C em 30 dias: Bootcamp Intensivo com Projetos Reais

Domine as habilidades mais procuradas no mercado e surpreenda seus colegas de trabalho com o seu conhecimento avançado

3,9 ★★★★★ (6 classificações) 74 alunos

Criado por [Code Puzzle](#)

📅 Última atualização em 12/2023 🌐 Português 🗺️ Português [Automático]



R\$ 28,90 ~~R\$ 128,90~~

78% de desconto

🕒 **Só mais 1 dias** por este preço!

Adicionar ao carrinho

Comprar agora

É fácil aprender programação?

Não é...

Teach Yourself Programming in Ten Years

Peter Norvig

Why is everyone in such a rush?

Walk into any bookstore, and you'll see how to *Teach Yourself Java in 24 Hours* alongside endless variations offering to teach C, SQL, Ruby, Algorithms, and so on in a few days or hours. The Amazon advanced search for [\[title: teach, yourself, hours, since: 2000\]](#) and found 512 such books. Of the top ten, nine are programming books (the other is about bookkeeping). Similar results come from replacing "teach yourself" with "learn" or "hours" with "days."

The conclusion is that either people are in a big rush to learn about programming, or that programming is somehow fabulously easier to learn than anything else. Felleisen *et al.* give a nod to this trend in their book [How to Design Programs](#), when they say "Bad programming is easy. *Idiots* can learn it in 21 days, even if they are *dummies*." The Abtruse Goose comic also had [their take](#).

Translations

Thanks to the following authors, translations of this page are available in:

[Arabic](#)
([Mohamed A. Yahya](#))

العربية

[Bulgarian](#)
([Boyko Bantchev](#))

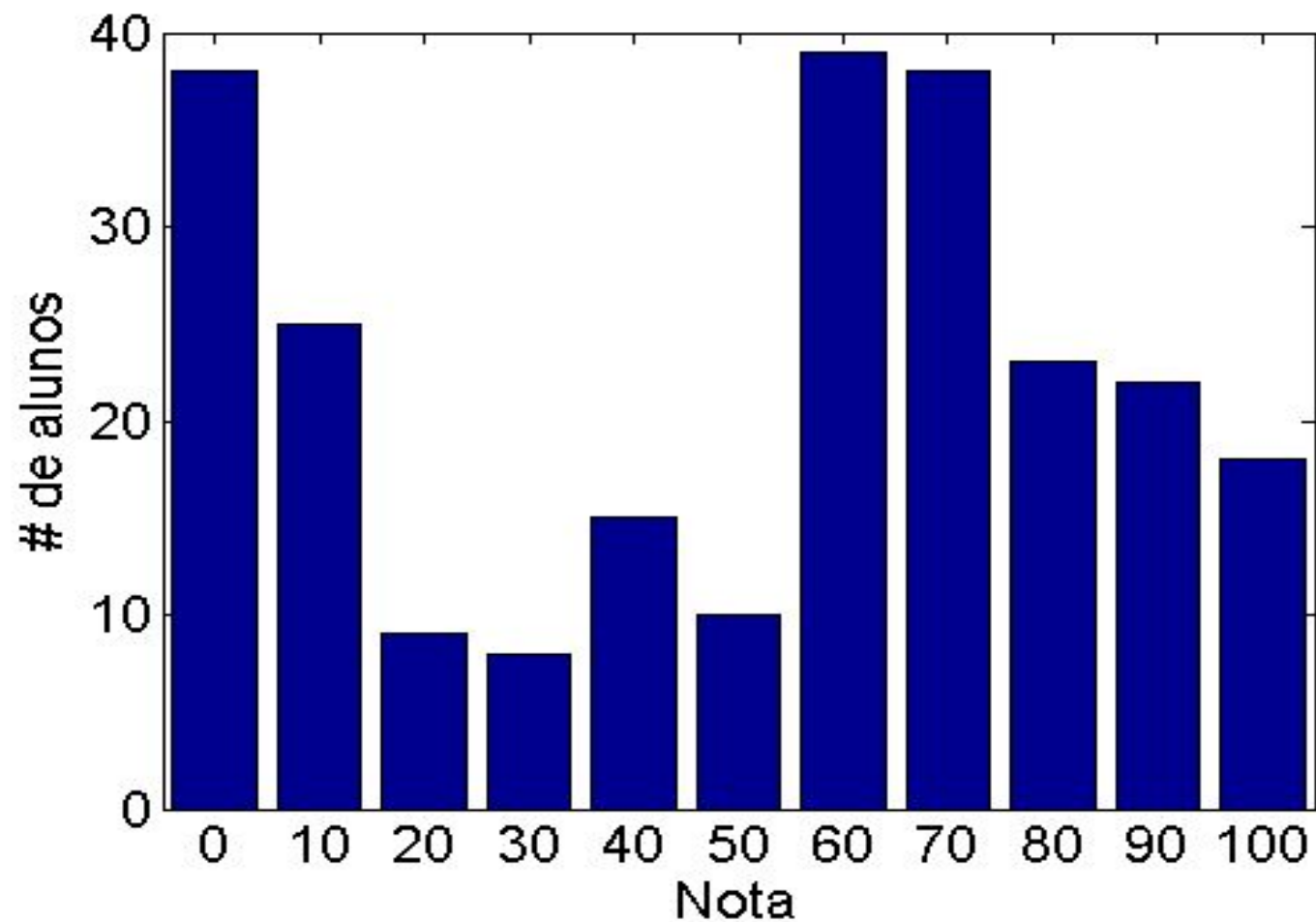
Observações

- O curso não é fácil
- Diferente do segundo grau: **objetivo é formar analistas, não recitadores**
- Programação é difícil de aprender:
 - Diferente de tudo o que vocês já aprenderam antes
 - Matemática + lógica + “Arte”

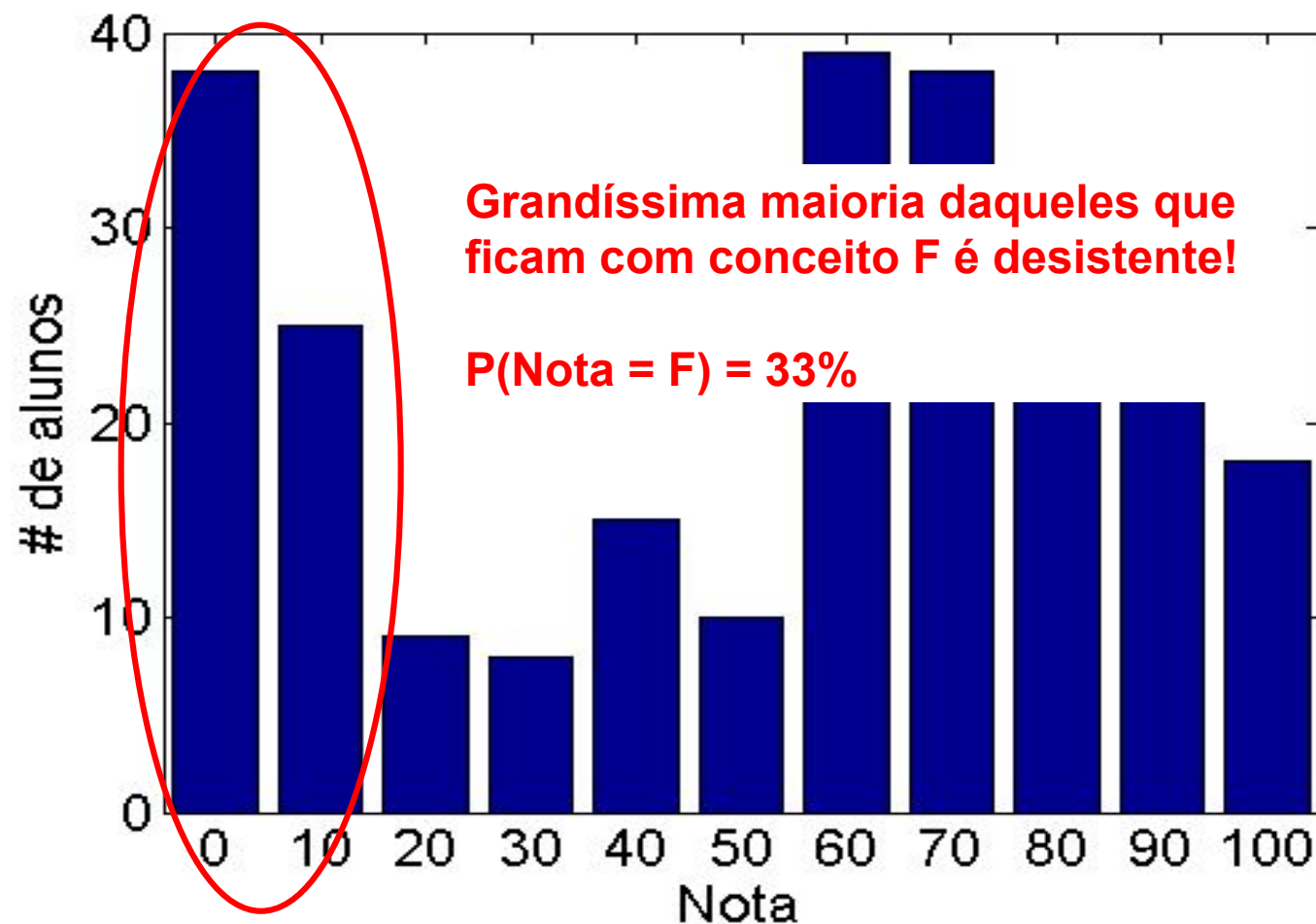
Observações

- Dedicção é fundamental
 - **Só se aprende a programar programando**
 - Trabalhos práticos requerem afinco e muitas horas de esforço
 - Provas e trabalhos exigindo conceitos + raciocínio lógico + análise de problemas

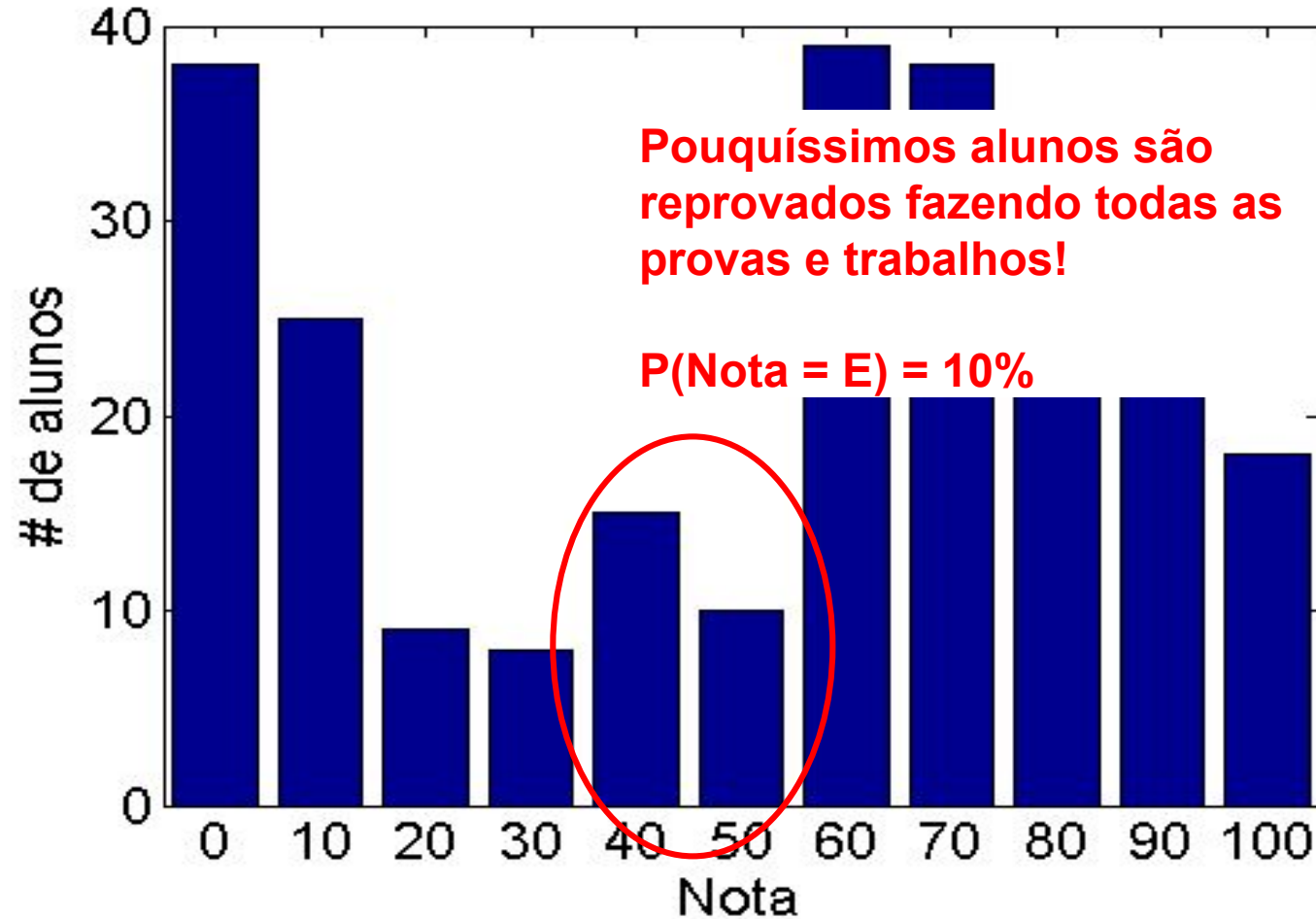
Desempenho dos Alunos



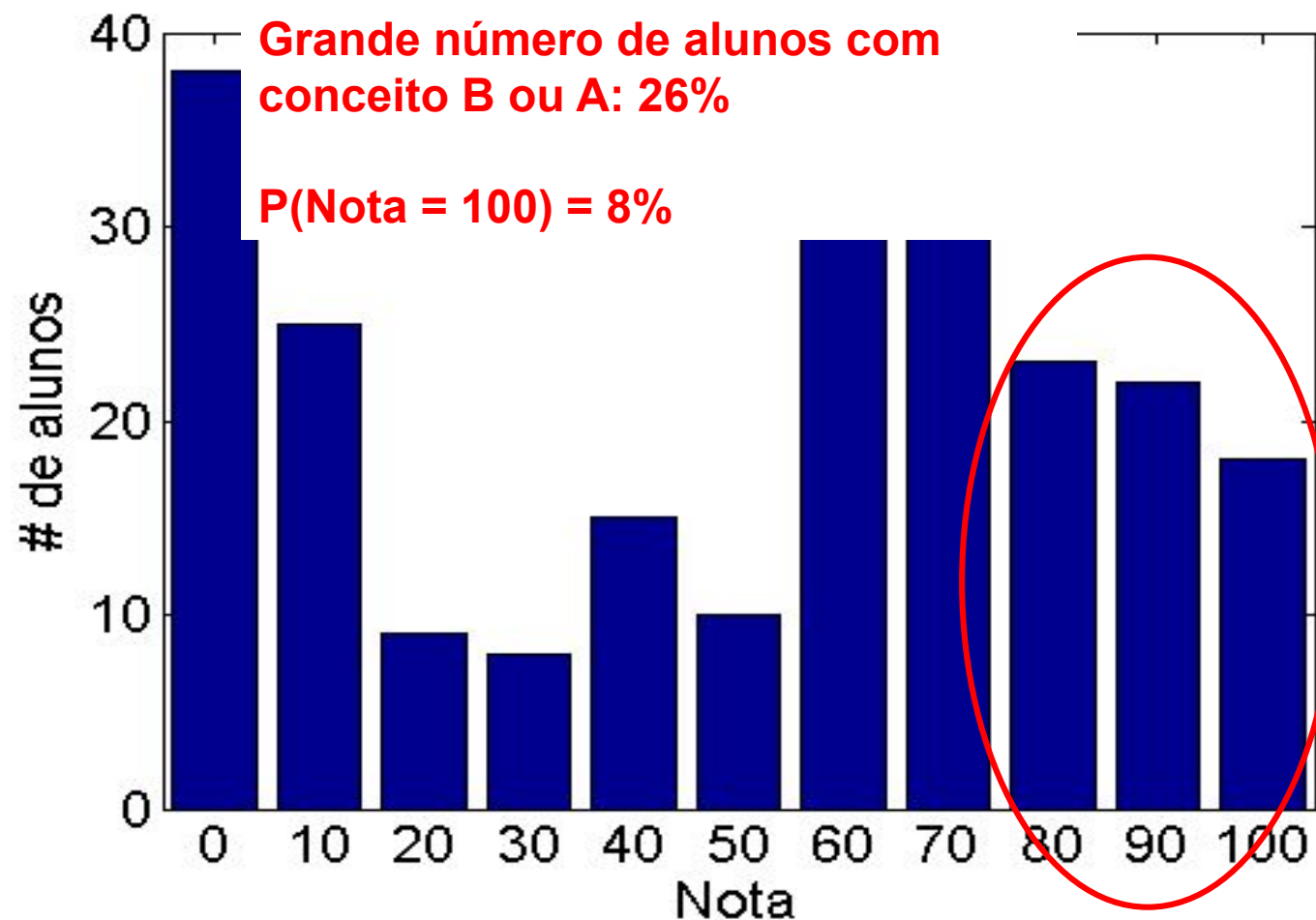
Desempenho dos Alunos



Desempenho dos Alunos

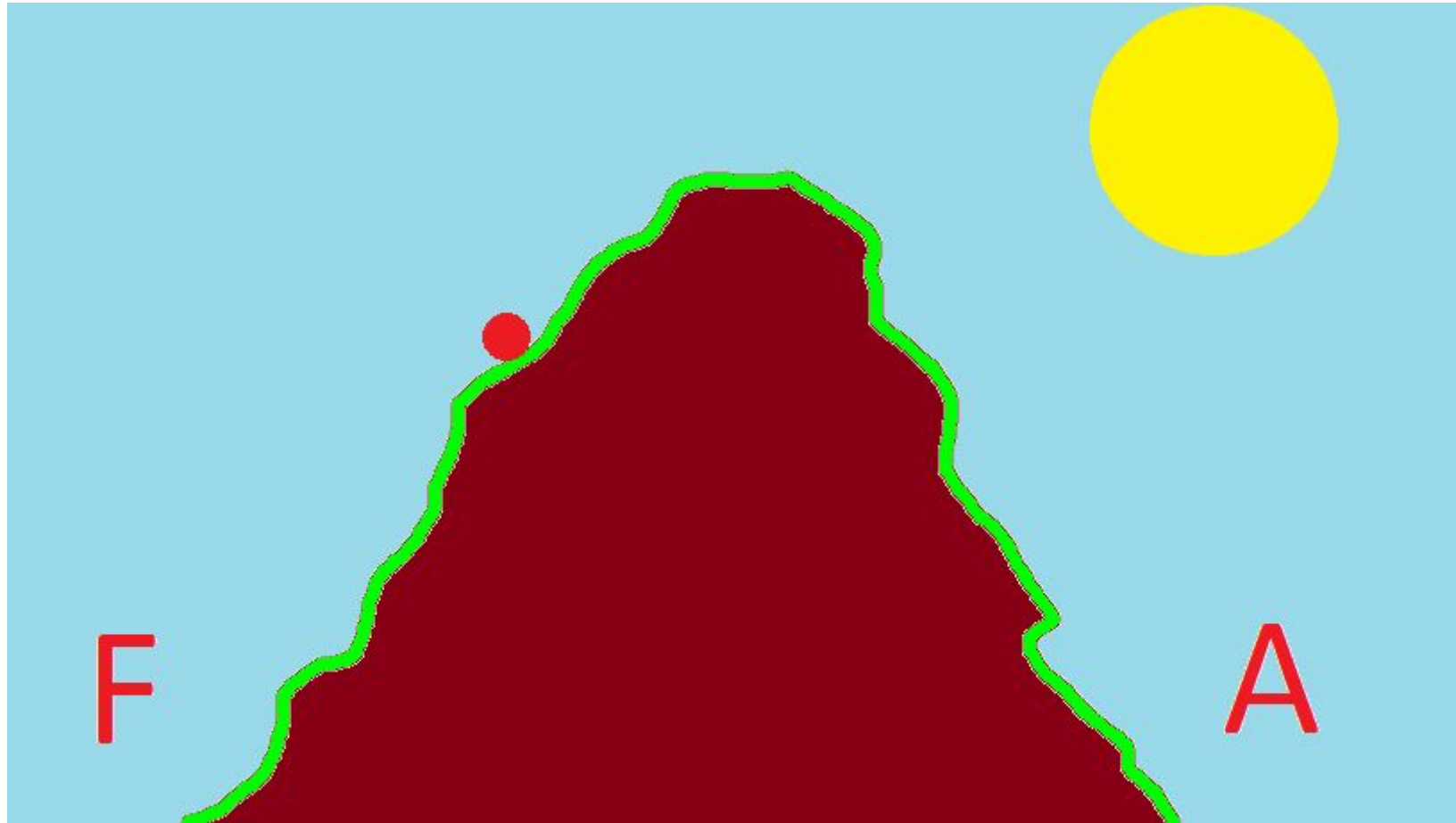


Desempenho dos Alunos



Desempenho dos Alunos

E você?!



Observações

- Gostaria de aprovar todos, mas nem sempre isso é possível:
 - Programação é central em muitos cursos de exatas
- **Vai ser importante para o seu futuro profissional**



Dicas

Práticas de código

WHEN LIFE GIVES YOU FREE STUFF,



TAKE IT AND RUN!!

Dicas

Prova I



Dicas

Prova 2



Dicas

Prova 2



Dicas

Trabalho prático



Dicas

A matéria em geral...

- Muitos desistem pois não conseguem entender NADA que está sendo dado em aula
- CAUSA: Conteúdo é cumulativo, ou seja, para entender a aula x, é preciso ter entendido a aula x-1
- Solução: ir e prestar atenção
- Benefícios colaterais: **menos estudo em casa, menos stress, menos fadiga**

Dicas

A matéria em geral...



Dicas

A matéria em geral...

- Pergunte durante a aula!
 - Fico muito feliz com perguntas
 - Não existe pergunta boba
 - Eu nunca vou caçoar de uma pergunta feita por aluno
 - Então, não tenha medo de fazer perguntas bobas
- Lembre: **A sua dúvida pode ser a dúvida de outro aluno!**

Corrupção

Cola, compra e cópia de trabalhos

■ Cola

- Não preciso pegar colando para identificar uma cola
- Programas são como respostas dissertativas em português: estatisticamente, é impossível ter dois iguais!
- Então, respostas estruturalmente iguais **CARACTERIZAM** uma cola

Corrupção

Cola, compra e cópia de trabalhos

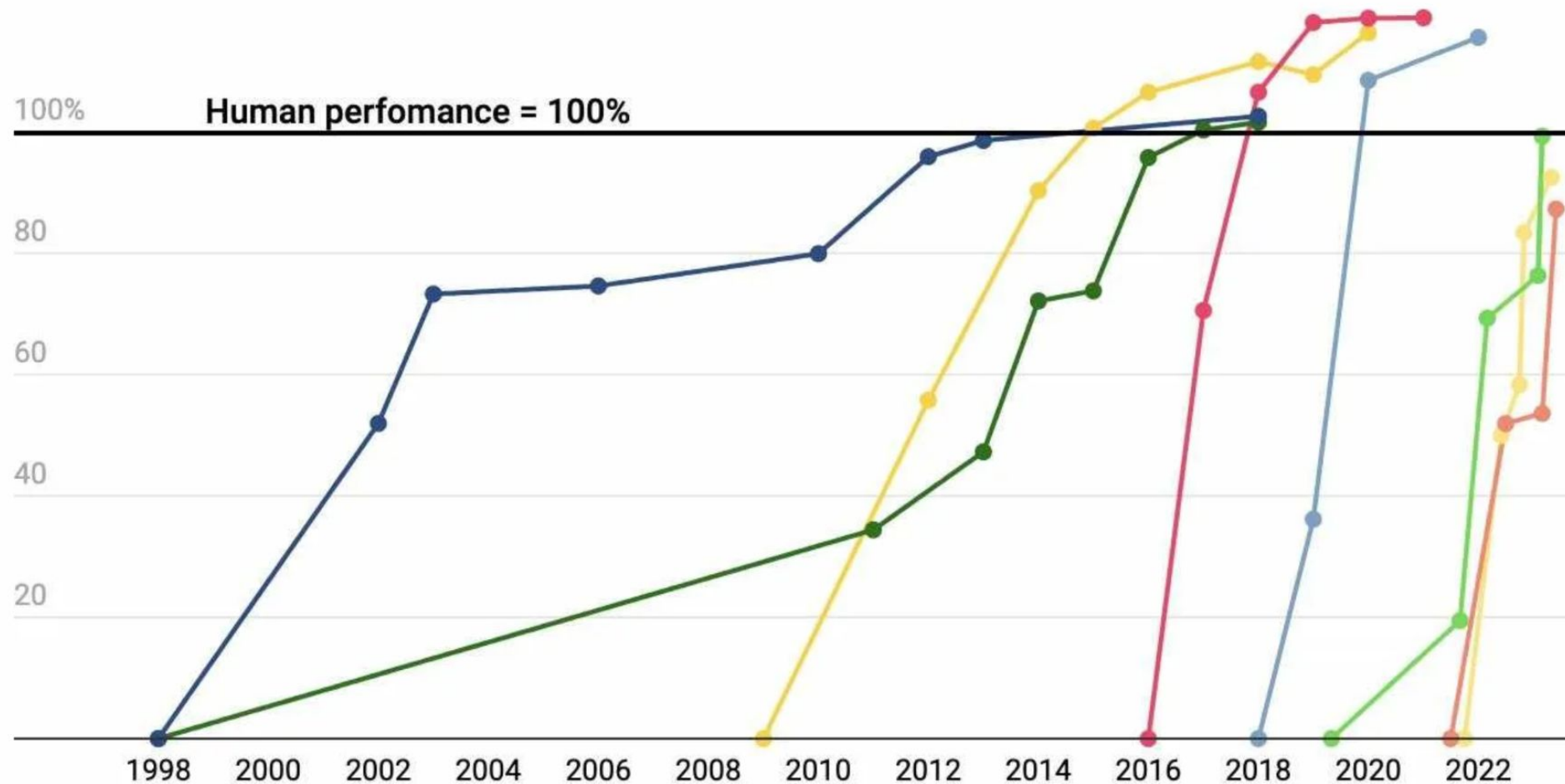
- Compra e cópia de trabalhos
 - Será verificado na prova oral
 - Se você não conseguir responder sobre aspectos básicos do seu SUPOSTO programa, você será punido severamente

Sobre uso de IA na aula

NO  **GPT**

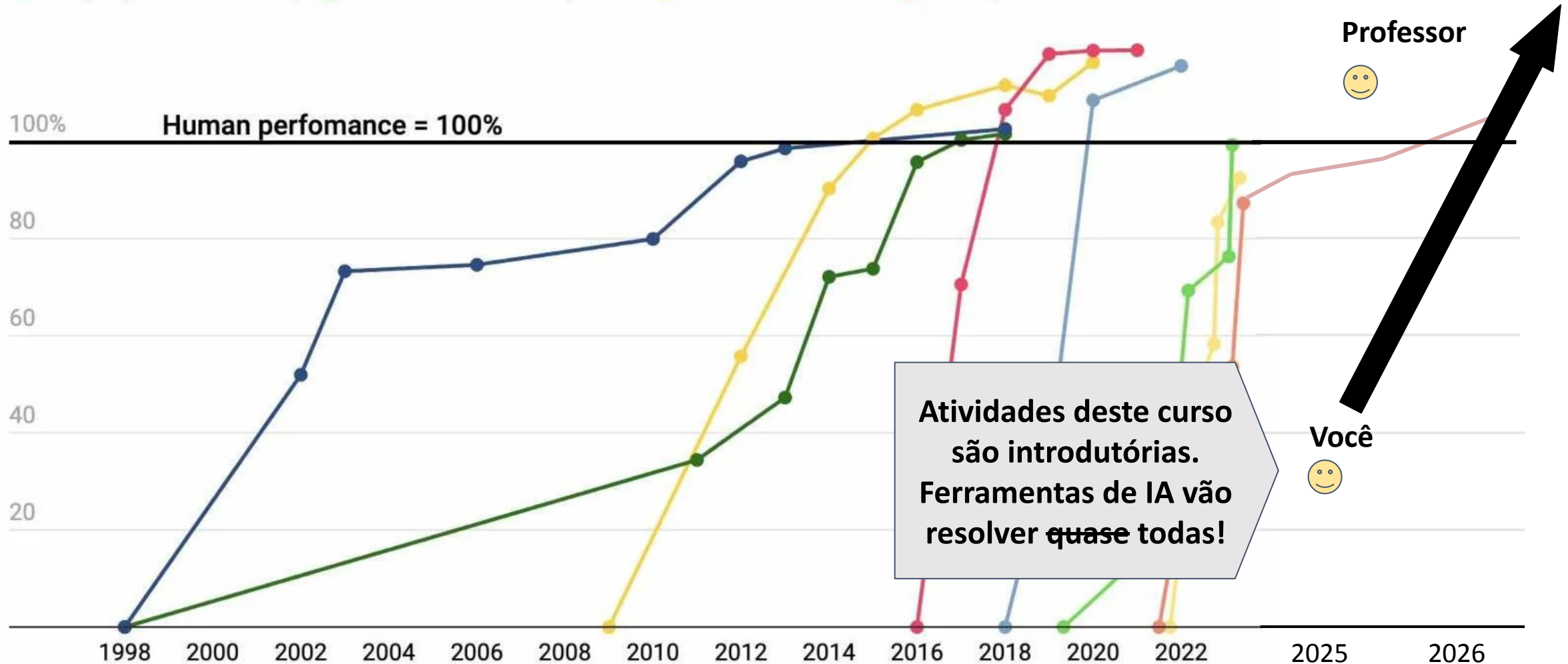
Evolução da IA

● Handwriting recognition ● Speech recognition ● Image recognition ● Reading comprehension
● Language understanding ● Common sense completion ● Grade school math ● Code generation



Evolução da IA

● Handwriting recognition ● Speech recognition ● Image recognition ● Reading comprehension
● Language understanding ● Common sense completion ● Grade school math ● Code generation

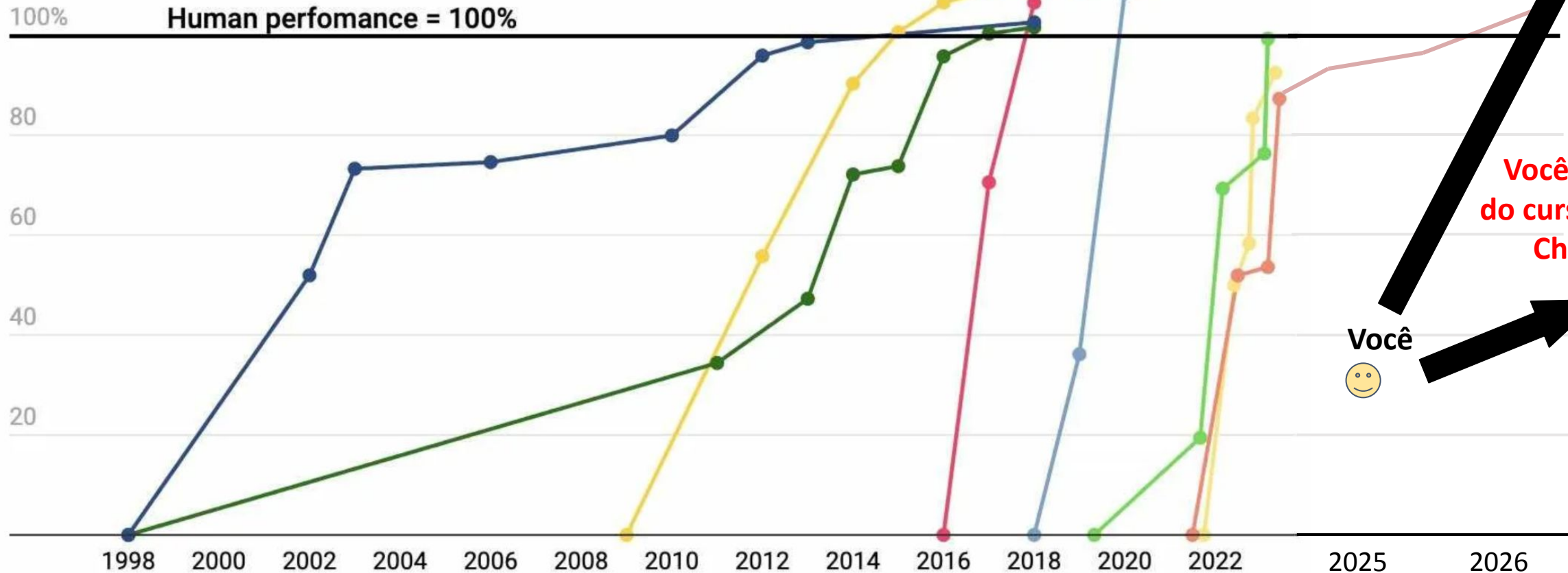


Inteligência artificial é ferramenta

- É essencial aprendermos a utilizar, e **bem...**
- PDS2 é um curso introdutório a programação orientada a objetos
 - Ferramentas de IA “dominam” o conteúdo do curso
- **Você também precisa dominar o conteúdo** introdutório antes do conteúdo avançado nos semestres seguintes

Evolução da IA

● Handwriting recognition ● Speech recognition ● Image recognition ● Reading comprehension
● Language understanding ● Common sense completion ● Grade school math ● Code generation



Não sabemos o que vai acontecer...

EdSource

NEWSAUDIO & VIDEOSPECIAL PROJECTSRESOURCESABOUT

NEWS BRIEF

FRIDAY, SEPTEMBER 6, 2024 — 8:20 AM

Students using artificial intelligence did worse on tests, experiment shows

Aa 1

TECH NEWS


HomeTechnology NewsBusiness NewsNews StoriesGeneral NewsWorld NewsMedia Releases

Tech Business News > Blogs > ChatGPT May Lead To The Downfall Of Education And Critical Thinking

BLOGS

ChatGPT May Lead To The Downfall Of Education And Critical Thinking

SharefXinEmailLinkPrint

 Editorial Desk
Last updated: June 13, 2024 10:47 am

hundrEDInnovationsServicesCommunityPublicationsAbout

22.3.2024 | < Share

How AI is already improving education

February 08, 2023

ChatGPT Is a Plague Upon Education

What winter of 2020 was for COVID-19, winter of 2023 is for ChatGPT—and higher education will never be the same, Jeremy Weissman writes.

By Jeremy Weissman

Usar a IA de forma CRITERIOSA

- Sugestões de como usar:
 - Nunca colocar o enunciado da questão no chat
 - Antes de começar o chat colocar um prompt direcionando a ferramenta

Eu sou um aluno do período XXX do curso de computação e estou aprendendo a programar. Meu objetivo é aprender e dominar os conceitos e fundamentos da programação de computadores. Gostaria que você me ajudasse como um tutor, respondendo perguntas e me ajudando a encontrar lacunas no meu aprendizado. Nunca coloque código em suas respostas, explique de forma que eu consiga aprender e escrever o código por conta própria.

Perguntas?

- E-mail: hector@dcc.ufmg.br
 - Leio somente e-mails de alunos com menos de 30 palavras e sem anexos
 - Outros vão direto para lixeira
- **Não respondo mensagens enviadas pelo Chat do Moodle!**
- Podemos marcar um horário para atendimento individual
- Melhor maneira de tirar dúvidas complexas
- Material da disciplina:
<https://pedroolmo.github.io/teaching/pdsI.html>