

# DCC 096 - Introdução à Computação

---

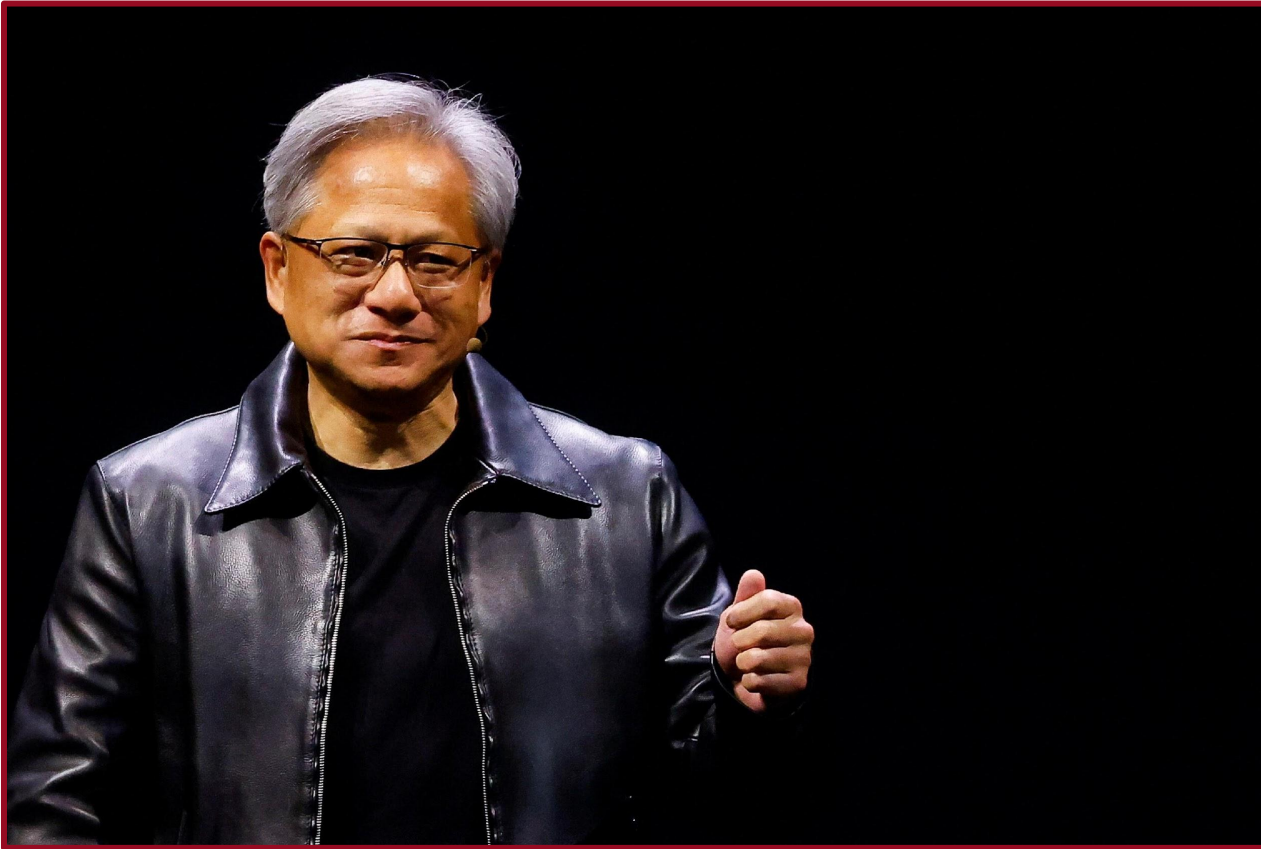
## Aula 03 - Desenvolvimento de software

Prof. Mário S. Alvim

[msalvim@dcc.ufmg.br](mailto:msalvim@dcc.ufmg.br)

[www.dcc.ufmg.br/~msalvim](http://www.dcc.ufmg.br/~msalvim)

# Unsolicited advice?



*“It is our job to create computing technology such that nobody has to program. And that the programming language is human.”*

*“Everybody in the world is now a programmer. This is the miracle of artificial intelligence.”*

Jensen Huang, CEO da Nvidia

# Unsolicited advice?

Fundador da DeepLearning.AI:

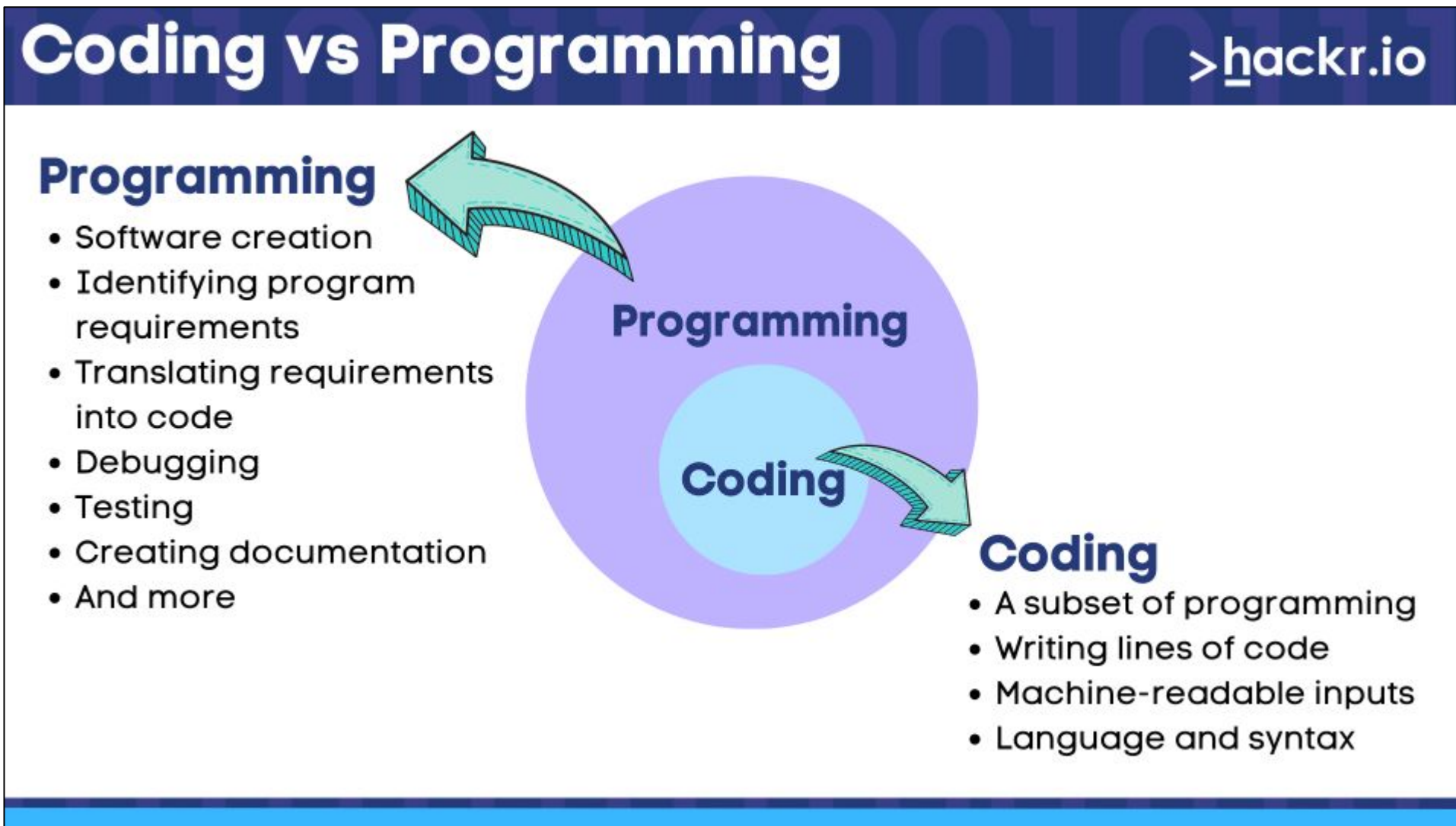
↻ Stanford NLP Group reposted



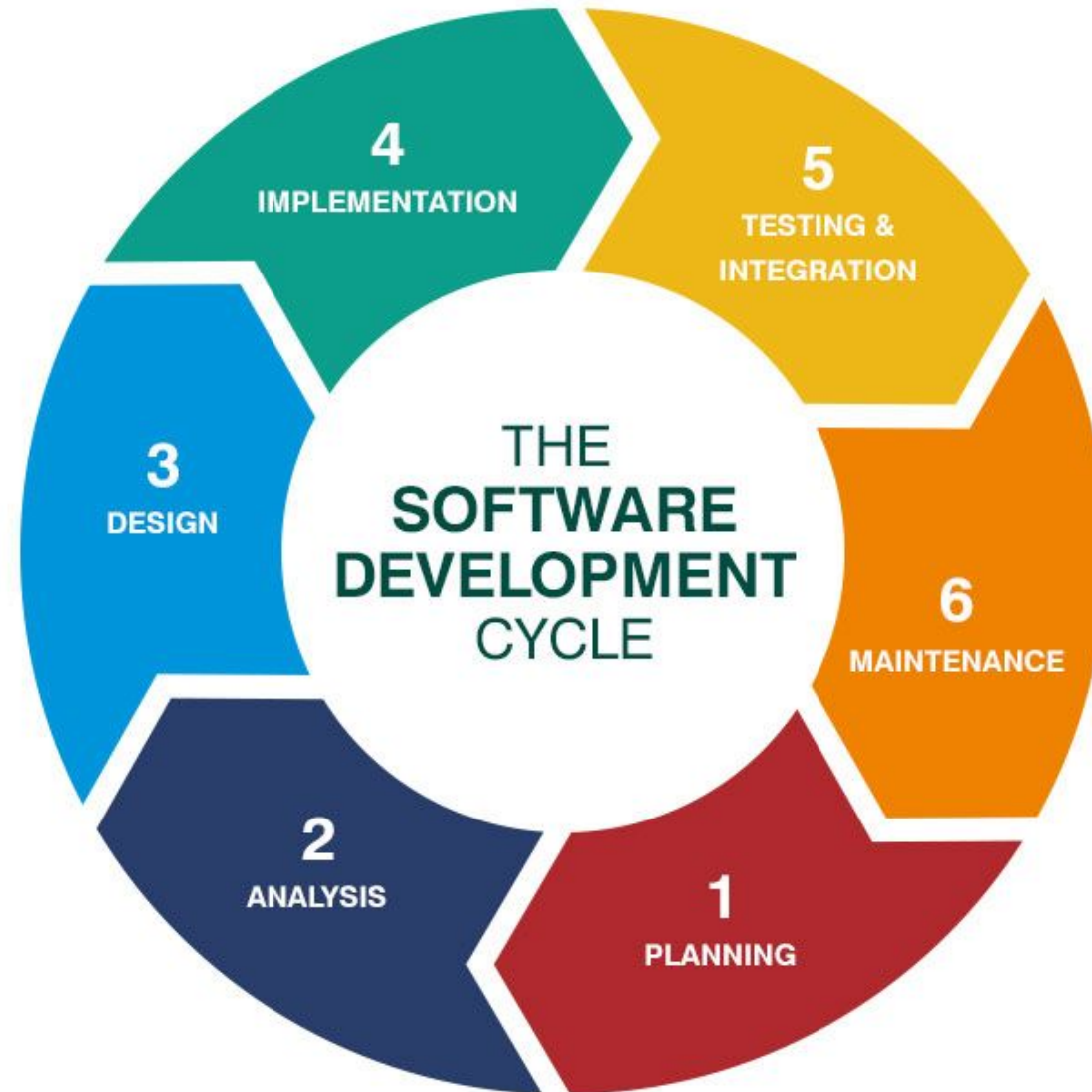
**Andrew Ng**  @AndrewYNg · 2d ...

Some people today are discouraging others from learning programming on the grounds AI will automate it. This advice will be seen as some of the worst career advice ever given. I disagree with the Turing Award and Nobel prize winner who wrote, “It is far more likely that the programming occupation will become extinct [...] than that it will become all-powerful. More and more, computers will program themselves.” Statements discouraging people from learning to code are harmful!

# O que é “programação”?



# Como é o trabalho de programação?



# O que é preciso saber para programar?

## Linguagens

Conjuntos de regras e símbolos que um programador usa para criar instruções compreensíveis para um computador.

## Tecnologias

Ferramentas, linguagens e frameworks usados para desenvolver software, desde a sua concepção até a execução.

## Métodos

Técnicas e abordagens sistemáticas usadas para projetar, escrever e gerenciar o desenvolvimento de software.



# O que são linguagens de programação?

Wikipedia:

Uma **linguagem de programação** é um método padronizado, formado por um conjunto de regras sintáticas e semânticas, de implementação de um código fonte - que pode ser compilado e transformado em um programa de computador, ou usado como script interpretado - que informará instruções de processamento ao computador.

- Quantas linguagens existem?
- De quais linguagens vocês já ouviram falar?



# Quais as linguagens de programação mais populares?

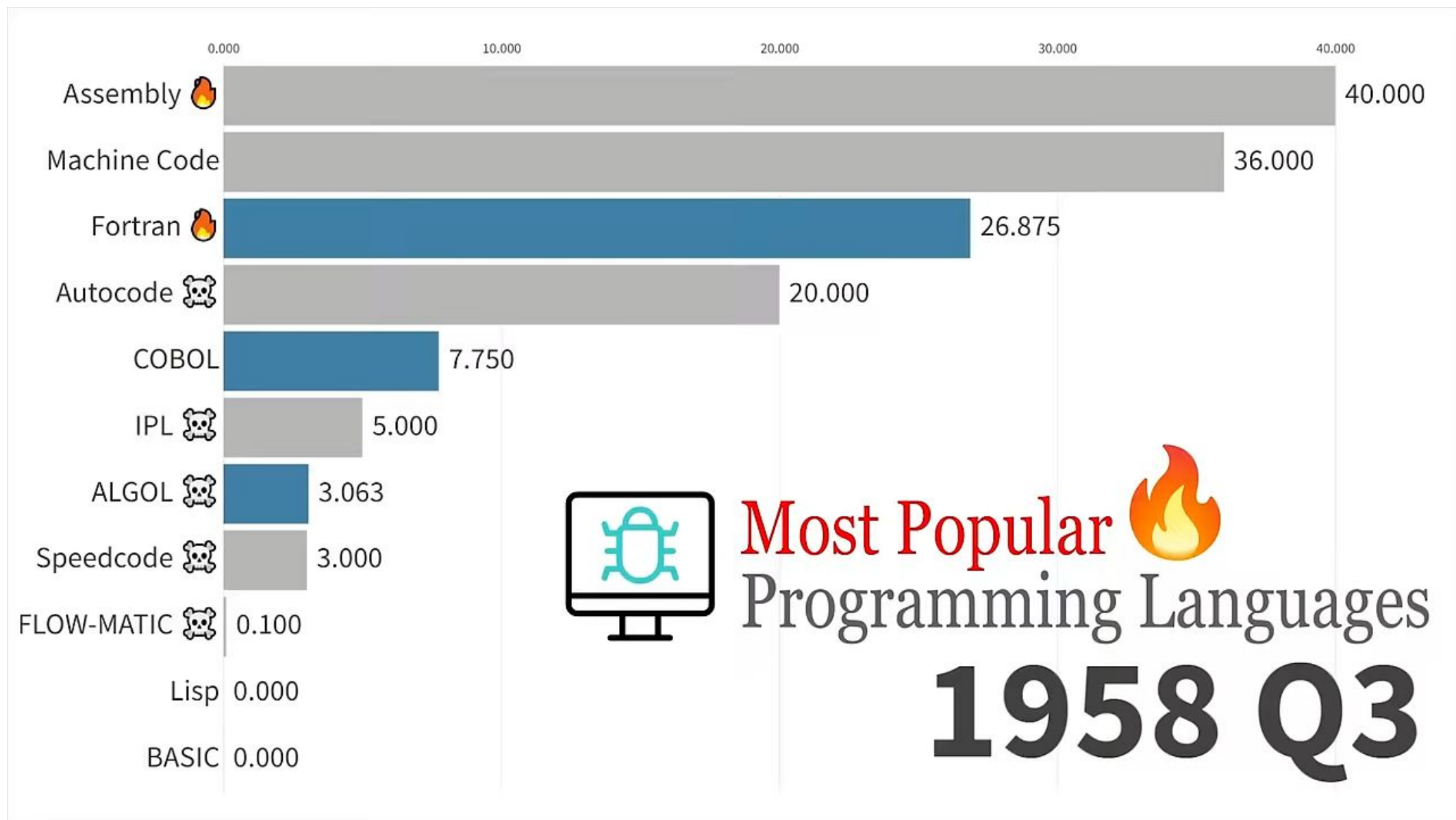


Fonte:

<https://www.darly.solutions/blog/the-most-popular-programming-languages-in-2021>



# Quais as linguagens de programação mais populares?



- Que tipos de tecnologias existem?
  - Compiladores
  - Depuradores
  - Repositórios
  - Servidores web
  - Ambientes de desenvolvimento
  - Bancos de dados
  - Etc, etc, etc
- De quais tecnologias vocês já ouviram falar?

# O que são tecnologias de programação?

## Databases



## Cloud platforms



## Web frameworks and technologies



# Como executar um programa?

## Computador

- Instruction set architecture (ISA)
- Microarchitecture
- Systems design

## Sistema operacional

- Linux
- Windows
- OSX
- IOS
- Android
- AWS
- Docker

## Compilador/ Interpretador

- **Compilador:** Traduz o código-fonte de uma vez só para um arquivo executável que a máquina pode rodar diretamente.
- **Interpretador:** Traduz e executa o código-fonte linha por linha em tempo de execução.

- **Depuração:** É o processo de encontrar e corrigir erros (bugs) no código de um software.
- **Revisão de código:** É a prática de inspecionar o código-fonte de um projeto por outra pessoa para garantir a qualidade, identificar falhas e melhorar o design.
- **Refatoração:** É a alteração da estrutura interna do código sem mudar seu comportamento externo, com o objetivo de melhorar a legibilidade e a manutenibilidade.
- **Testes:** São procedimentos que verificam se o software atende aos requisitos esperados e se comporta conforme o planejado, garantindo que ele funcione corretamente.



- **KISS (Keep It Simple, Stupid):** A simplicidade deve ser prioridade no desenvolvimento de software para evitar complexidade desnecessária e facilitar a manutenção.
- **SOLID:** 5 princípios de design de software - **Single Responsibility, Open/Closed, Liskov Substitution, Interface Segregation, Dependency Inversion.**
- **As Quatro Regras do Design Simples:** O código deve (i) passar nos testes, (ii) revelar a intenção do programador, (iii) evitar duplicação e (iv) ter o menor número de elementos possível.
- **Law of Demeter:** Um objeto deve interagir apenas com seus vizinhos imediatos, limitando a comunicação para evitar acoplamento excessivo.
- **Composition over Inheritance:** Uso de composição de objetos ("tem um") em vez de herança ("é um") para reutilizar código e criar sistemas mais flexíveis e menos acoplados.
- **DRY (Don't Repeat Yourself):** Cada peça de conhecimento no sistema deve ter uma representação única e não ambígua, evitando a duplicação de código.

- **git:** Sistema de controle de versão que permite a colaboração de equipes no desenvolvimento de software, rastreando todas as alterações do código-fonte de forma segura e eficiente.
- **slack:** Plataforma de comunicação em canais que centraliza as conversas e agiliza a colaboração entre equipes, substituindo o e-mail em muitos ambientes de trabalho.
- **gcc/gdb/clang/lldb:** Compiladores como GCC e Clang traduzem o código em linguagem de máquina, enquanto depuradores como GDB e LLDB ajudam a inspecionar e corrigir erros, sendo ferramentas úteis para o desenvolvimento e a depuração de código.
- **MS Teams:** Ferramenta de colaboração integrada que oferece chat, videoconferências e compartilhamento de arquivos, ideal para empresas que já utilizam o ecossistema da Microsoft.
- **Trello:** Ferramenta visual de gerenciamento de projetos que organiza tarefas em quadros e cartões, tornando o acompanhamento do fluxo de trabalho claro e intuitivo.
- **Ferramentas de IA!**

## **Automação de Tarefas Repetitivas**

- Ferramentas de IA como o GitHub Copilot e o Amazon CodeWhisperer estão cada vez mais eficientes na geração automática de código, testes de unidade e na identificação de bugs.
- Isso pode diminuir a demanda por programadores júnior, que frequentemente realizam essas tarefas.

## **Desvalorização de Habilidades Fundamentais**

- A dependência excessiva de ferramentas de IA pode levar a uma atrofia de habilidades essenciais, como a lógica de programação, o raciocínio crítico para a resolução de problemas e a compreensão profunda das arquiteturas de software.

## **Aumento da Superficialidade**

- A facilidade de gerar código pode incentivar uma abordagem de "copiar e colar" sem a devida análise e compreensão.
- Sistemas podem ficar mais complexos e difíceis de manter a longo prazo, com riscos de segurança e performance.

## GitHub Copilot

**Uso:** Um assistente de codificação que utiliza modelos de linguagem para sugerir código em tempo real, baseando-se no contexto do arquivo em que o programador está trabalhando. Pode gerar funções inteiras, completar linhas e até mesmo escrever a documentação do código.

**Impacto:** Pesquisas mostram que, ao usar o Copilot, os desenvolvedores completam suas tarefas até 55% mais rápido (GitHub, 2022). No entanto, um estudo da Purdue University indicou que 40% do código gerado por IA pode conter vulnerabilidades de segurança (Purdue University, 2023).

## ChatGPT e outros LLMs (Large Language Models)

**Uso:** Utilizados para gerar, refatorar e explicar código. Desenvolvedores usam essas ferramentas para traduzir código entre linguagens, criar scripts complexos e obter explicações sobre conceitos ou bibliotecas.

**Impacto:** Facilita o aprendizado e a prototipagem rápida, mas requer validação humana para garantir a exatidão e a qualidade do código gerado (OpenAI, 2023).

## Amazon CodeWhisperer:

**Uso:** Similar ao Copilot, oferece sugestões de código baseadas em bilhões de linhas de código abertas. Tem um foco especial na segurança, escaneando o código gerado em busca de vulnerabilidades (Amazon, 2023).

**Impacto:** Aumenta a produtividade e a segurança do código, mas o uso massivo pode padronizar soluções, inibindo a inovação e a criatividade.

- **Produtividade Aumentada:**

- Um estudo do McKinsey Global Institute sugere que a IA pode automatizar até 70% das tarefas relacionadas à codificação.
- Isso pode levar a um aumento da produtividade de até 20% (McKinsey, 2023).

- **Reestruturação de Funções:**

- A demanda por programadores com habilidades básicas e repetitivas pode diminuir
- A necessidade de profissionais com expertise em arquitetura de sistemas, engenharia de prompts, e resolução de problemas complexos irá crescer.

- **Novas Oportunidades:**

- Surgem novas carreiras, como engenheiros de IA, especialistas em ética em IA e auditores de sistemas autônomos.
- Essas posições exigem uma combinação de conhecimento em programação e expertise em machine learning e IA.



- **Adaptação Contínua:**

- Programadores precisarão se adaptar constantemente e aprender a colaborar com ferramentas de IA, em vez de competir com elas?
- A proficiência em utilizar assistentes de IA se tornará uma habilidade fundamental?

- **Foco em Habilidades Humanas:**

- A valorização de habilidades como a criatividade, a colaboração, o pensamento crítico e a capacidade de resolver problemas complexos se tornará ainda mais crucial?

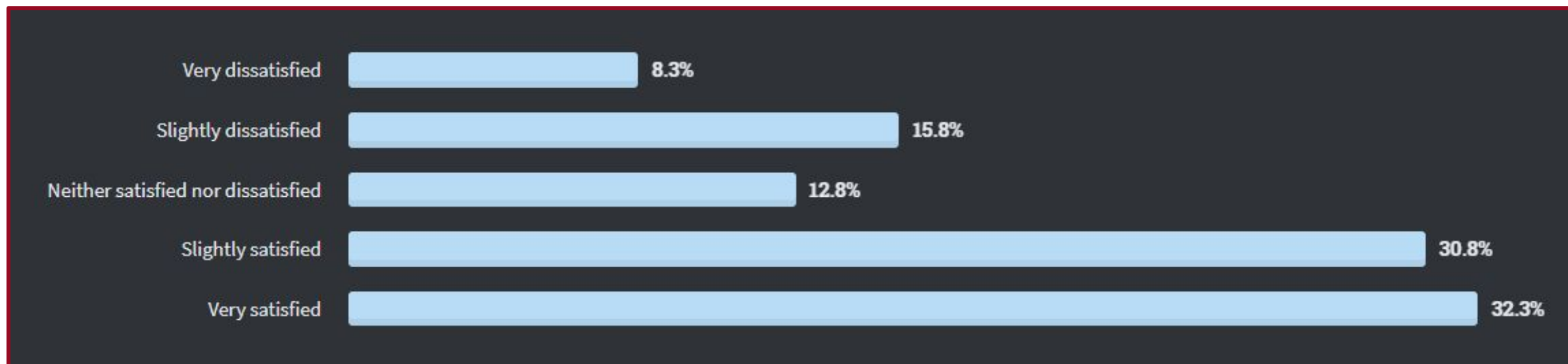
- **Ética e Responsabilidade:**

- O debate sobre a ética na IA e a propriedade intelectual do código gerado por essas ferramentas vai se intensificar.?
- Desenvolvedores precisarão ter um papel ativo na garantia de que o código gerado é seguro, justo e livre de viés?

- **Especialização em Nichos:**

- As carreiras se moverão em direção a nichos mais especializados, como desenvolvimento de sistemas de IA, cibersegurança e integração de IA em diversas áreas?
- O foco será menos em "escrever código" e mais em "projetar soluções"?

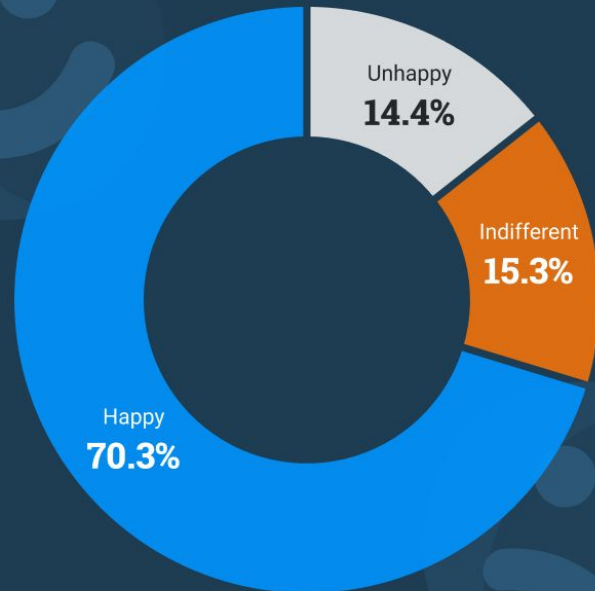
# Programação é um bom trabalho?



# Programação é um bom trabalho?

Developer Happiness

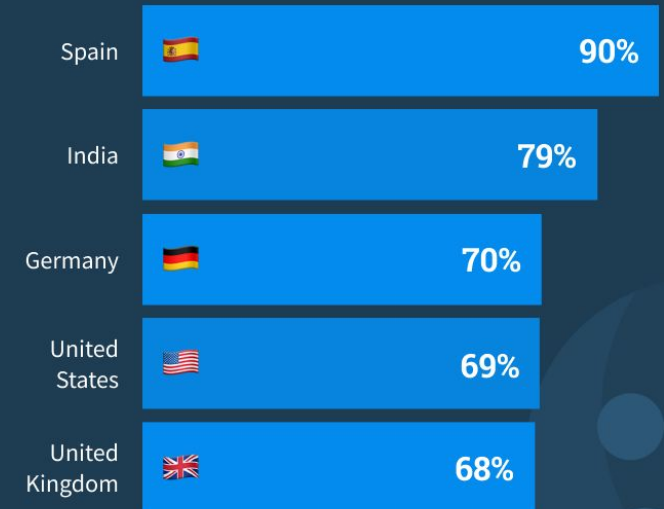
We asked developers — are you **happy** at **work**?



stackoverflow

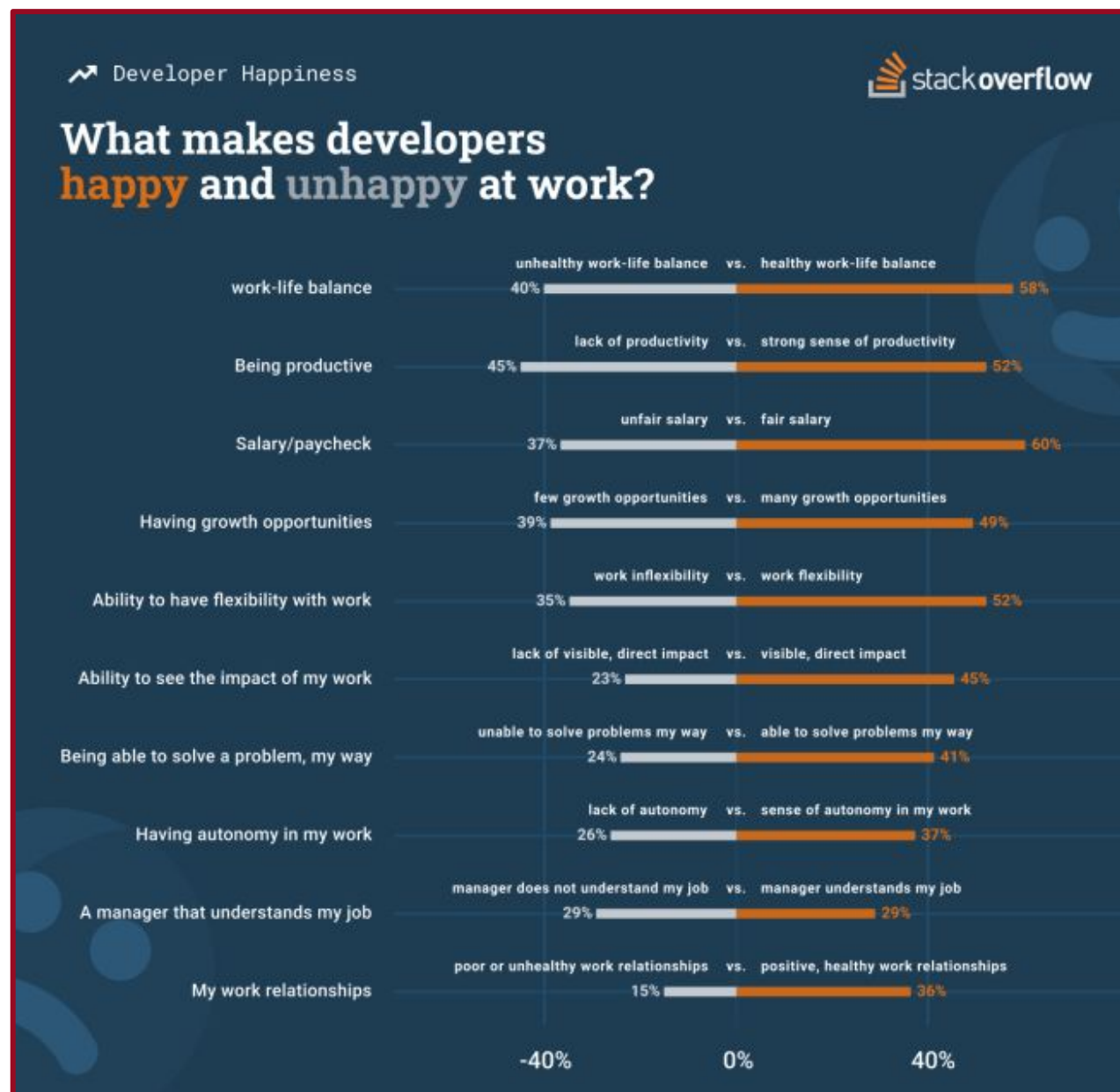
Developer Happiness

What country has the **highest percent of happy** working developers?



stackoverflow

# Programação é um bom trabalho?

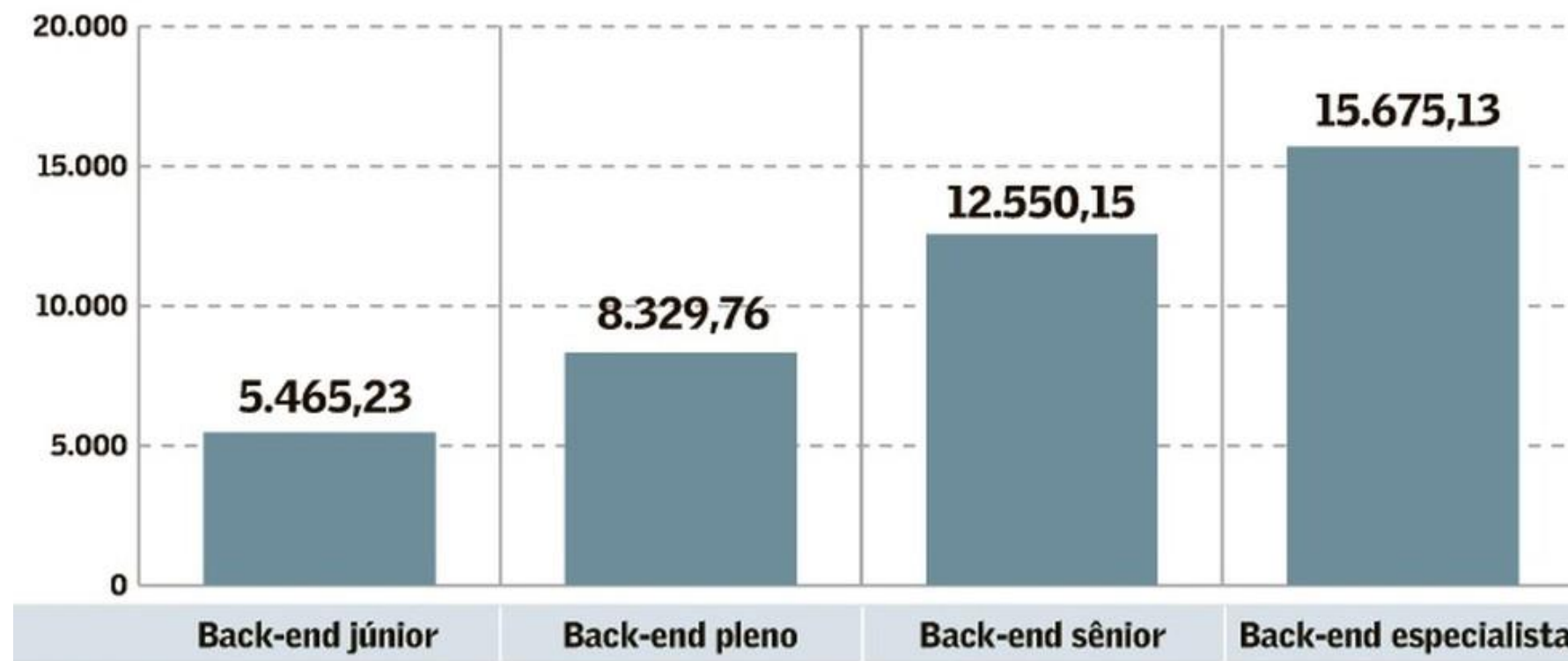




# Quanto ganha um programador?

## Salário dos desenvolvedores

Média salarial oferecida pelas empresas no país - em R\$



Fonte: Intera Back-End Report 2022

- A carreira em tecnologia é vasta e cheia de oportunidades!
  - Grandes empresas, startups, etc – aqui é apenas o começo
- Desenvolver software é construir o futuro.
  - Transformar ideias em soluções reais que impactam o mundo.
- Aprender a programar é aprender a pensar.
  - “Primeiro, resolva o problema. Depois, escreva o código.”

# Relatório

---

1. Os **enunciados dos relatórios são passados em cada aula expositiva**, conforme cronograma da disciplina.
2. Atente para o prazo de entrega de cada relatório determinado no cronograma de atividades do curso. Submissões serão aceitas **no Moodle até as 23:59 da segunda-feira imediatamente seguinte à aula correspondente ao relatório**. Não serão aceitos envios fora do prazo.
3. Você deve escrever suas respostas à mão, seja no papel para depois digitalizá-las ou fotografá-las, seja diretamente em algum aplicativo que permita desenho à mão livre. **Suas respostas devem ser dadas no modelo de relatório entregue em sala ou disponível na seção de material da disciplina.**
4. Cada relatório, mesmo que tenha múltiplas folhas, deve ser submetido em **um único arquivo necessariamente em formato PDF**. Outros tipos de arquivos não serão considerados para avaliação (por exemplo, fotos em .jpeg ou .png, ou múltiplos arquivos para uma mesma lista de exercícios).

## Questão 01

Quando você ouve o termo 'programação', o que vem à sua mente?

Você associa programação a resolver problemas, criar soluções tecnológicas, ou a algo mais técnico e complicado?

## Questão 02

Quais habilidades você acha que são importantes para alguém que trabalha com desenvolvimento de software?

Você acredita que já possui algumas dessas habilidades ou gostaria de desenvolvê-las?

## Questão 03

O que você espera aprender ao se aprofundar na área de desenvolvimento de software?

Há algo específico que te interesse, como criação de jogos, aplicativos móveis ou soluções empresariais?