Possível título:

41: Lean e Verificação de Software, com Leonardo de Moura (AWS)

<https://podcasters.spotify.com/pod/show/fronteirases/episodes/41-Lean-e-Verificao-de-Software--com-Leonardo-de-Moura-AWS-e2brvmj>

<https://youtu.be/fG-au9y4RW0>

Resumo:

Neste episódio do Fronteiras da Engenharia de Software, Adolfo Neto entrevista o pesquisador Leonardo de Moura, Senior Principal Applied Scientist no Grupo de Raciocínio Automatizado na AWS e Chief Architect da Lean FRO. Leonardo é reconhecido por diversas contribuições, entre elas o desenvolvimento do Z3, um SMT Solver, e da Lean, que é ao mesmo tempo uma linguagem de programação funcional e um provador de teoremas.

A entrevista é dividida em partes, começando com a apresentação de Leonardo e de sua trajetória desde o doutorado na PUC-Rio até sua posição atual na AWS. Em seguida, Leonardo explora os conceitos fundamentais sobre verificação de software e como provadores de teoremas, como o Z3, são aplicados nesse contexto.

A terceira parte destaca a evolução do Lean, inicialmente uma ferramenta para matemáticos, para se tornar uma linguagem de programação funcional (Lean 4). Leonardo compartilha insights sobre o sucesso do Lean, sua comunidade ativa e as possibilidades que a linguagem oferece para desenvolvimento e verificação de software.

Dentro do enfoque em Lean, são abordados temas como a mudança para ser uma linguagem de programação funcional, a base em teoria dos tipos dependentes e as potenciais aplicações em pesquisa e desenvolvimento de software. Perguntas do Twitter sobre o uso do Lean fora do meio acadêmico e a Teoria da Homotopia são discutidas, assim como as perspectivas de pesquisa com o Lean 4.

A quarta parte explora temas diversos, como a participação de Lean na Rinha de Backend, onde uma implementação em Lean se destacou, e a criação da Lean FRO, uma organização focada em pesquisa do Lean. Também são discutidos os avanços em IA e como o Lean pode se beneficiar dessas tecnologias.

Na parte final, Adolfo pergunta a Leonardo sobre a próxima fronteira da engenharia de software, proporcionando uma visão do que pode vir a moldar o futuro da área. O episódio encerra com agradecimentos aos ouvintes e a promessa de retornar com mais discussões no próximo episódio do Fronteiras da Engenharia de Software.

Sobre Leonardo de Moura:

* Página <https://leodemoura.github.io/>
* Escavador <https://www.escavador.com/sobre/7442280/leonardo-mendonca-de-moura>
* Scholar <https://scholar.google.com/citations?user=CwazDKgAAAAJ&hl=pt-BR&oi=ao>
* DBLP <https://dblp.org/pid/d/LeonardoMdeMoura.html>
* Linkedin <https://www.linkedin.com/in/leonardo-de-moura-26a27b5/>
* GitHub <https://github.com/leodemoura>

Mais links em <https://fronteirases.github.io/episodios/paginas/41>

Entrevistador: Adolfo Neto (PPGCA UTFPR) [⁠⁠https://adolfont.github.io/⁠⁠⁠⁠⁠](https://adolfont.github.io/)

Nosso site é: [⁠⁠⁠⁠https://fronteirases.github.io/⁠⁠⁠⁠](https://fronteirases.github.io/)

**Opening Song: Extreme Energy (Music Today 80). Composed & Produced by: Anwar Amr Video. Link:**[⁠⁠https://www.youtube.com/watch?v=8ZZbAkKNx7s⁠⁠](https://www.youtube.com/watch?v=8ZZbAkKNx7s)

Links:

* Raciocínio Automatizado com Leonardo de Moura, Pesquisador na Microsoft Research.: <https://youtu.be/bwKFcLaeD1A?si=bIXi6CC0V-OZWcEO>.
* Tese de Doutorado <https://github.com/fronteirases/fronteirases.github.io/blob/master/episodios/material_complementar/01_PhD_moura.pdf>
* Vídeos da comunidade Lean [https://www.youtube.com/@leanprovercommunity5485/videos](https://www.youtube.com/%40leanprovercommunity5485/videos)
* (Nature) Mathematicians welcome computer-assisted proof in ‘grand unification’ theory <https://www.nature.com/articles/d41586-021-01627-2>
* Histórico do site de Lean [https://web.archive.org/web/\*/https://leanprover.github.io/](https://web.archive.org/web/%2A/https%3A//leanprover.github.io/) e [https://web.archive.org/web/\*/http://leanprover.net/](https://web.archive.org/web/%2A/http%3A//leanprover.net/)
* 16 Linguagens em 16 Dias: Minha Saga da Rinha de Backend <https://youtu.be/EifK2a_5K_U?si=xSlY-RFXGvrPah-k>
* "The Economics of Programming Languages" by Evan Czaplicki (Strange Loop 2023) <https://www.youtube.com/watch?v=XZ3w_jec1v8>
* Exploring the Lean4 Language <https://blog.codeminer42.com/exploring-lean4/>
* Lean4: Crafting in an Uncharted Territory <https://blog.codeminer42.com/overcoming-challenges-and-crafting-in-the-uncharted-territory-of-lean4/>
* Exploring Lean4, Sofia Rodrigues <https://www.youtube.com/watch?v=klooP9cJHCs>
* Morph: The personal AI proof engineer - <https://morph.so/blog/the-personal-ai-proof-engineer/>
* Grupo de Estudos em Haskell da UFABC <https://haskell.pesquisa.ufabc.edu.br/>
* Tipos Dependentes <https://haskell.pesquisa.ufabc.edu.br/desenvolvimento-orientado-a-tipos/07.dependenttypes/>
* Elixir em Foco 32: A linguagem de programação Lean, com Algebraic Sofia e Algebraic Gabi <https://www.youtube.com/watch?v=LaN6XzyCqao>
* Lean Brasil - Grupo de pessoas interessadas na linguagem de programação Lean <https://t.me/leanlangbr>
* Site de Lean <https://lean-lang.org/>
* Completion of the Liquid Tensor Experiment <https://leanprover-community.github.io/blog/posts/lte-final/>
* SATurn: SAT Solver-prover in lean 4 <https://github.com/siddhartha-gadgil/Saturn>
* Z3 Theorem Prover <https://en.wikipedia.org/wiki/Z3_Theorem_Prover>

Dia e hora:

20/10 10h00 (horário de Brasília)

Estúdio de gravação:

<https://streamyard.com/sdjr7ya4ep>

Gravação no YouTube (vídeo não listado)

<https://www.youtube.com/watch?v=Wf2YDbLiG5M>

(PARTE 1 - APRESENTAÇÃO DOS HOSTS E DA PESSOA ENTREVISTADA)

(ADOLFO)

Olá,

Este é o Fronteiras da Engenharia de Software

um podcast em que convidamos pesquisadores e pesquisadoras para falarem sobre sua pesquisa e refletir sobre o presente e o futuro da engenharia de software.

Eu sou Adolfo Neto,

professor da UTFPR Curitiba e do Programa de Pós-Graduação em Computação Aplicada (PPGCA).

A professora Maria Claudia Emer, minha colega de bancada, não pôde participar de nossa gravação.

(ADOLFO)

Hoje vamos entrevistar Leonardo de Moura.

Leonardo fez Doutorado em Informática, na área de Engenharia de Software, na PUC-Rio, orientado pelo professor Carlos Lucena, e co-orientado por Edward Hermann Haeusler. O título da tese, defendida em 2000, foi “Um Framework para Análise e Verificação de Programas”.

Depois do Doutorado, Leonardo trabalhou por quase 6 anos no instituto de pesquisa SRI, na Califórnia. Trabalhou então por mais de 16 anos na Microsoft Research, onde foi responsável, entre outros projetos, pelo provador de teoremas Z3.

Atualmente, Leonardo é
*Senior Principal Applied Scientist*

(Cientista Sênior Principal Aplicado)

no Grupo de Raciocínio Automatizado na AWS (Amazon Web Services).

E também

*Chief Architect and Board Member*

*Arquiteto-Chefe e Membro do Conselho*

*Da*

Lean FRO

Organização de Pesquisa Focada em Lean

Leonardo é co-criador, com [Nikolaj Bjørner](https://www.microsoft.com/en-us/research/people/nbjorner/), Lev Nachmanson e Christoph Wintersteiger de

Z3

Um SMT Solver (um tipo de provador de teoremas)

[“usado para verificação de programas, geração de casos de teste entre várias aplicações”.](https://www.microsoft.com/en-us/research/people/nbjorner/)

Z3 ganhou o

Programming Languages Software Award da SIGPLAN

Em 2015 (outros vencedores deste prêmio foram Glasgow Haskell Compiler (GHC), Coq proof assistant, GNU Compiler Collection (GCC), Racket, Scala e OCaml).

[“Nikolaj Bjørner e Leonardo de Moura receberam o Prêmio Herbrand de 2019 por Contribuições de Destaque para o Raciocínio Automatizado em reconhecimento ao seu trabalho na promoção da prova de teoremas com o Z3.”](https://en.wikipedia.org/wiki/Z3_Theorem_Prover)

E criador
Lean

Que é

Um Provador de Teoremas,

Um Assistente de Provas e

Uma Linguagem de Programação Funcional.

Ou, como diz a página de Lean

"Lean é uma linguagem de programação funcional que torna mais fácil escrever código correto e de fácil manutenção."

Eu gosto de dizer que Lean é uma das 3 linguagens de programação criadas por brasileiros e usadas no mundo todo. As outras duas são Lua, que acabou de completar 30 anos, e Elixir, 11 anos.

Nós já conversamos antes em meu podcast pessoal, mas lá o foco maior foi em lógica e hoje o foco maior será em desenvolvimento e verificação de software com Lean.

Tudo bem, Leonardo, você tem algo a complementar nesta sua apresentação?

(Leonardo)

(PARTE 2 - CONCEITOS BÁSICOS)

(ADOLFO)Vamos começar falando sobre o tema principal desta entrevista: Verificação de Software. Algumas das pessoas que nos escutam já devem saber que é Verificação de Software, mas você pode explicar novamente para nossos e nossas ouvintes?

(Leonardo)

(ADOLFO)E como provadores de teoremas ou assistentes de provas são usados na verificação de software? Por exemplo, como o Z3 pode ajudar desenvolvedores a encontrarem bugs em seus programas?

(Leonardo)

(PARTE 3 - LEAN)

(PARTE 3.1 - LEAN COMO SOFTWARE)

(ADOLFO) Eu acho que escutei falar pela primeira vez de Lean quando ainda era Lean 2 ou 3. O que era o Lean até sua versão 3? Um software muito útil para matemáticos fazerem demonstrações?

(Leonardo)

(ADOLFO) E se fosse apenas uma ferramenta para matemáticos, já daria pra dizer que Lean é um grande sucesso:

muitos usuários (para este tipo de ferramenta)

organizados numa comunidade (no Zulip)

que se ajuda e obtém resultados de destaque (mathlib, [Liquid Tensor Experiment](https://leanprover-community.github.io/blog/posts/lte-final/)),

usuários famosos (como Jeremy Avigad e o Medalhista Fields Peter Scholze),

usuários não famosos mas que podem aprender matemática com o Lean e ter suas demonstrações validadas pela ferramenta,

ótimo suporte em um dos editores de código (IDE) mais usados (VS Code).

Como você acha que Lean chegou a tudo isso?

(PARTE 3.2 - LEAN COMO LINGUAGEM DE PROGRAMAÇÃO)

(ADOLFO) A partir da versão 4, Lean se tornou uma linguagem de programação funcional. Por que isto aconteceu?

(Leonardo)

(ADOLFO) E como a utilização de Lean 4 pode ajudar desenvolvedores a criar softwares mais corretos?

(Leonardo)

(ADOLFO) Além de ser uma linguagem de programação funcional, o sistema lógico no qual o Lean 4 se baseia é uma versão da teoria dos tipos dependentes. O que são Tipos Dependentes e por que você escolheu TD para Lean?

(Leonardo)

(ADOLFO) Tivemos duas perguntas feitas pelo Twitter por Tomaz Gomes, orientando de mestrado do professor Haniel Barbosa na UFMG:

“Sobre usar Lean fora do meio acadêmico, já existe alguma empresa interessada em usar em algum projeto?”

(Leonardo)

(ADOLFO)

A outra pergunta do Tomaz foi:

“E o suporte para a Teoria da Homotopia ficou pra trás mesmo?”

(Leonardo)

(ADOLFO) E quais são algumas das possibilidades de pesquisa que já existem ou que você visualiza com Lean 4 em Desenvolvimento e Verificação de Software?

(Leonardo)

[SATurn : SAT Solver-prover in lean 4](https://github.com/siddhartha-gadgil/Saturn)

*“The program itself is checked by lean, so guaranteed to terminate and give a correct answer.”*

(PARTE 4 - OUTROS TEMAS)

(ADOLFO) Recentemente aconteceu um evento na comunidade de pessoas desenvolvedoras, a Rinha de Backend. “[A ideia era que as pessoas participantes trouxessem a implementação de uma API definida nas regras da competição](https://dev.to/leandronsp/rust-go-rinha-e-io-39o2#:~:text=Rinha%20de%20backend,definida%20nas%20regras%20da%20competi%C3%A7%C3%A3o.)”. Duas pessoas, Algebraic Sofia (Rodrigues) e Algebraic Gabi, participaram com Lean e ficaram em quarto lugar. A descrição da submissão foi “Usamos Lean4 e C++ para fazer o servidor, e PostgreSQL para database. O mundo precisa de monads.” O programador e empresário Fabio Akita fez um vídeo descrevendo a Rinha e ficou impressionado com a solução em Lean. Depois disso, Sofia escreveu dois blog posts sobre Lean no blog de engenharia da empresa Codeminer. Como você vê Lean chegando mais a pessoas desenvolvedoras de software apesar do, palavras da Sofia, “ecossistema inexistente de Lean4”?

(Leonardo)

(ADOLFO) Você e algumas outras pessoas fazem parte do Conselho da **Lean FRO**, a organização focada em pesquisa do Lean. O Tomaz até comentou que “Eu queria dizer que estou super impressionado com o tanto de ferramenta nova e interessante que está vindo com o Lean FRO :)”. Por que surgiu o Lean FRO?

(Leonardo)

// pretende pagar bolsas de M&D?

(ADOLFO) Recentemente a Lean FRO divulgou o "Morph Prover, o primeiro modelo de código aberto treinado como assistente de conversação para usuários do Lean."

Como Lean e seu desenvolvimento beneficia ou pode se beneficiar dos avanços em IA, Large Language Models, ChatGPT, CoPilot, etc.?

/[/ ontem mesmo uma colega, a professora Carla Silva da UFPE, (re)compartilhou este artigo: “Security Weaknesses of Copilot Generated Code in GitHub”](https://arxiv.org/abs/2310.02059#nwit)

(PARTE 5 - PRÓXIMA FRONTEIRA)

(ADOLFO) Para você, qual é a próxima fronteira da engenharia de software? (pode ser algo que você acha que vai acontecer ou que você gostaria que acontecesse em nossa área)

(Leonardo)

(PARTE 6 - FINAL)

[ADOLFO] Agradece e passa para o(a) entrevistado(a).

(Leonardo)

(MARIA) Fecha o episódio com algo como “Agradecemos a todos os nossos e nossas ouvintes, e até o próximo episódio do Fronteiras da Engenharia de Software”

(Leonardo)

**FAZER PRINT PARA DIVULGAÇÃO EM REDES SOCIAIS**



Lex Friedman

Meu

Flow

All In Bay Area

Ciencia (vai mandar)

Marcos Fontoura CTO Stone irmão acadêmico

 VP MSoft

// LEAN + ELIXIR?

IDEIAS PARA AUMENTAR A PRESENÇA DE LEAN NO BRASIL

Leonardo de Moura

* Página <https://leodemoura.github.io/>
* Lattes <https://www.escavador.com/sobre/7442280/leonardo-mendonca-de-moura>
* Scholar <https://scholar.google.com/citations?user=CwazDKgAAAAJ&hl=pt-BR&oi=ao>
* DBLP <https://dblp.org/pid/d/LeonardoMdeMoura.html>
* Twitter
* Linkedin <https://www.linkedin.com/in/leonardo-de-moura-26a27b5/>
* GitHub <https://github.com/leodemoura>

Links:

* Raciocínio Automatizado com Leonardo de Moura, Pesquisador na Microsoft Research.

YouTube: <https://youtu.be/bwKFcLaeD1A?si=bIXi6CC0V-OZWcEO>. Podcast: <https://podcasters.spotify.com/pod/show/adolfont/episodes/Raciocnio-Automatizado-com-Leonardo-de-Moura--Pesquisador-na-Microsoft-Research-e1anc8j>

* Tese de Doutorado <https://github.com/fronteirases/fronteirases.github.io/blob/master/episodios/material_complementar/01_PhD_moura.pdf>

Artigos:

Outros materiais:

* Vídeos da comunidade Lean [https://www.youtube.com/@leanprovercommunity5485/videos](https://www.youtube.com/%40leanprovercommunity5485/videos)
* (Nature) Mathematicians welcome computer-assisted proof in ‘grand unification’ theory <https://www.nature.com/articles/d41586-021-01627-2>
* Histórico do site de Lean [https://web.archive.org/web/\*/https://leanprover.github.io/](https://web.archive.org/web/%2A/https%3A//leanprover.github.io/) e [https://web.archive.org/web/\*/http://leanprover.net/](https://web.archive.org/web/%2A/http%3A//leanprover.net/)
* 16 Linguagens em 16 Dias: Minha Saga da Rinha de Backend <https://youtu.be/EifK2a_5K_U?si=xSlY-RFXGvrPah-k>
* "The Economics of Programming Languages" by Evan Czaplicki (Strange Loop 2023) <https://www.youtube.com/watch?v=XZ3w_jec1v8>
* Exploring the Lean4 Language <https://blog.codeminer42.com/exploring-lean4/>
* Lean4: Crafting in an Uncharted Territory <https://blog.codeminer42.com/overcoming-challenges-and-crafting-in-the-uncharted-territory-of-lean4/>
* Exploring Lean4, by Sofia Rodrigues, November 9th, 12:30 PM (GMT-3), <https://www.youtube.com/watch?v=klooP9cJHCs>
* <https://morph.so/blog/the-personal-ai-proof-engineer/>
* <https://haskell.pesquisa.ufabc.edu.br/>
* <https://haskell.pesquisa.ufabc.edu.br/desenvolvimento-orientado-a-tipos/07.dependenttypes/>