José Valim (Elixir em Foco + Fronteiras)

Roteiro

24/02 10h00 UTC-3

Links adicionais

<https://hexdocs.pm/lua/Lua.html>

Ep. Roberto <https://fronteirases.github.io/episodios/paginas/52>

Ep. Leonardo <https://fronteirases.github.io/episodios/paginas/41>

Ep. Honey Potion

<https://github.com/lac-dcc/honey-potion>

Tese Lucas Vegi <https://repositorio.ufmg.br/handle/1843/80651>

You have built an Erlang <https://vereis.com/posts/you_built_an_erlang>

Beyond Functional Programming with Elixir and Erlang

<https://blog.plataformatec.com.br/2016/05/beyond-functional-programming-with-elixir-and-erlang/>

(ADOLFO)

Olá, pessoal!

Este é um episódio conjunto dos podcasts Fronteiras da Engenharia de Software e Elixir em Foco, que fazem parte da Rede Emílias de Podcasts.

Eu sou Adolfo Neto, e estou aqui hoje com

Zoey Pessanha do Elixir em Foco.

Tudo bem, Zoey?

(ZOEY)

(ADOLFO) Hermínio Torres, também do Elixir em Foco, não pôde participar hoje.

E estou aqui também com Maria Claudia Emer, do Fronteiras da Engenharia de Software.

Tudo bem, Maria Claudia?

(MARIA CLAUDIA)

(ADOLFO) E eu começo dando as boas-vindas ao nosso entrevistado de hoje: José Valim. Tudo bem, Valim? Obrigado por ter aceito nosso convite!

(VALIM)

(ADOLFO) Eu vou fazer uma rápida apresentação do Valim, pensando principalmente nas pessoas que escutam o Fronteiras da Engenharia de Software, pois quem escuta o Elixir em Foco já deve saber quem você é, Valim.

José Valim é um engenheiro brasileiro formado pela USP, com mestrado pelo Instituto Politécnico de Turim. Ele é fundador e Chief Adoption Officer da Dashbit, empresa que contribui para o ecossistema de Elixir e patrocina a Erlang Ecosystem Foundation.

Antes de criar Elixir, foi um dos principais contribuidores do Ruby on Rails. Em 2011, iniciou o desenvolvimento de Elixir para rodar na BEAM, a máquina virtual de Erlang, enquanto estava na Plataformatec. A primeira versão que se assemelha ao Elixir atual, 0.5, foi lançada em 24/05/2012.

Elixir é amplamente usado em sistemas distribuídos e concorrentes, e empresas ao redor do mundo o adotaram em produção. Na Stack Overflow Survey, figura entre as linguagens mais apreciadas. Diferencia-se de Erlang ao introduzir macros e outras melhorias.

Você tem algo a acrescentar sobre Elixir, Valim?

(VALIM)

(ADOLFO) Vou fazer agora uma pergunta relacionada ao trabalho de Lucas Vegi e Marco Túlio Valente. Lucas Vegi, da Universidade Federal de Viçosa, e Marco Túlio Valente, da Universidade Federal de Minas Gerais, já foram entrevistados no podcast "Elixir em Foco". Marco Túlio Valente também já foi entrevistado no "Fronteiras da Engenharia de Software", e Lucas em breve participará deste podcast para falar sobre seu trabalho, agora que concluiu seu doutorado na UFMG sob a orientação de Marco Túlio Valente.

Para contextualizar, em 2020 entrevistamos o professor Marcelo Maia, da Universidade Federal de Uberlândia, sobre code smells. Na ocasião, perguntei a ele sobre a existência de pesquisas relacionadas a code smells em linguagens de programação funcionais, pois a maioria dos estudos focava em linguagens orientadas a objetos. A resposta dele influenciou a escolha do tema de doutorado de Lucas.

Eles realizaram uma revisão multivocal, coletando dados da chamada "literatura cinza", ou seja, conteúdos produzidos por desenvolvedores de software, como palestras, slides, blogs e livros. Durante essa revisão, identificaram preocupações sobre code smells na comunidade Elixir, incluindo contribuições de Elaine Watanabe, que também foi entrevistada no "Elixir em Foco". Com base nisso, eles propuseram uma classificação de code smells em Elixir e publicaram artigos sobre o tema.

Antes de consolidar a proposta, eles apresentaram suas identificações à comunidade para validação. Você mesmo, Valim, contribuiu com o repositório do projeto no GitHub. Eu também ajudei na divulgação com um vídeo, incentivando a participação da comunidade Elixir, que sempre se mostrou aberta a esse tipo de colaboração.

Hoje, existe uma página oficial sobre antipadrões na documentação de Elixir que menciona o trabalho de Lucas Vegi e Marco Túlio Valente. Vale destacar que parte do doutorado de Lucas Vegi foi financiado pelo projeto "Pesquise com Elixir", que também financiou o HoneyPotion (também foi tema de episódio aqui e acaba de ter artigo publicado na CGO).

Diante disso, Valim, gostaria de saber: como surgiu a ideia de criar essa página de antipadrões na documentação de Elixir?

(VALIM)

(ADOLFO)

~~Eu só quero complementar que, em breve, a tese de doutorado do Lucas Vegi estará disponível no site da UFMG. Já vou deixar o link onde vocês poderão encontrá-la, procurando por "Vegi". Alguns artigos resultantes desse doutorado já foram publicados, e outros devem ser publicados em breve. Além disso, há trabalhos em andamento derivados da tese, como, por exemplo, o desenvolvimento de uma ferramenta para refatoração em Elixir, o~~ **~~RefactorEx~~**~~, do Gabriel Pereira, que iremos entrevistar no~~ *~~Elixir em Foco~~*~~. Agora, a próxima pergunta é da Zoey.~~

25:55 meu áudio falhou OK

(ZOEY)

A próxima pergunta é sobre a questão dos tipos em Elixir. Elixir é uma linguagem dinamicamente tipada, assim como Erlang. No entanto, de alguns anos para cá, você e outras pessoas começaram a trabalhar em um sistema de tipos baseado em conjuntos, os *Set-Theoretic Types*. Esse trabalho chegou a ser publicado em um artigo em The Art, Science, and Engineering of Programming, detalhando as ideias por trás do projeto desse sistema de tipos para Elixir.

O que queremos saber é: como surgiu a colaboração com o Giuseppe Castagna, professor de uma universidade na França, e com Guillaume Duboc? Se não me engano, Giuseppe Castagna é italiano, e Guillaume Duboc é francês e doutorando, com uma bolsa patrocinada pela Remote, uma empresa que usa Elixir em produção. Como essa colaboração começou? Como vocês se encontraram para desenvolver esse trabalho, que ainda está em andamento, sobre a adição de tipos em Elixir?

(VALIM)

(ZOEY) Pelas discussões nas redes sociais e pelos seus próprios comentários, percebemos que muitas pessoas estão interessadas nesse sistema de tipos, implementado ainda de forma parcial na linguagem, destacando, por exemplo: "Consegui encontrar esse bug aqui por causa do sistema de tipos de Elixir." O que vocês acharam destes resultados?

(VALIM)

(MARIA CLAUDIA)

Outra pergunta que queremos fazer é sobre sua relação com a pesquisa. Na Elixir Brasil 2018 você ministrou uma palestra chamada *Hughes-Driven Development*, na qual mencionou diversos trabalhos científicos, muitos deles de um pesquisador chamado John Hughes, que trabalha na Suécia. Você falou sobre como esses trabalhos influenciaram a linguagem Elixir. Acredito que, na época, a principal influência estava no formatador de código de Elixir. Nossa pergunta é mais genérica: qual é o papel da pesquisa no trabalho que você e o core team de Elixir realizam no desenvolvimento da linguagem?

(VALIM)

44 CORTE OK

46 CORTE OK

(ADOLFO)

Agora, uma pergunta sobre um tema atual que não podemos deixar de abordar. Muitos desenvolvedores estão preocupados com o futuro do mercado de trabalho e se ainda haverá empregos para profissionais da área. Algumas pessoas acreditam que já perderam seus empregos devido ao uso de IA pelas empresas.

Como você vê o posicionamento da linguagem Elixir em relação a isso? Ela oferece ferramentas que facilitam o uso desses modelos, especialmente os abertos, ou a integração com APIs de modelos pagos nos programas desenvolvidos em Elixir?

(VALIM)

CORTE 52 OK

(ZOEY)

Além disso, o que pode ser feito de maneira mais específica com IA em Elixir, especialmente no que se refere ao *machine learning* tradicional? Sabemos da existência do **Nx** e já tivemos um episódio no *Elixir em Foco* com Paulo Valente, um dos contribuidores do projeto.  
De forma geral, o que Elixir pode oferecer para pesquisadores em engenharia de software que desejam utilizar inteligência artificial, especialmente *LLMs*? Afinal, hoje em dia, é difícil não usar esses modelos, pois já fazem parte do dia a dia. Como Elixir pode ser uma opção para quem quer integrar *LLMs* e IA em suas pesquisas?

(VALIM)

(ZOEY)

Sua empresa, Dashbit, lançou recentemente o Pythonx, um interpretador de Python em Elixir. Você pode nos explicar um pouco sobre o que é o Pythonx e que problemas ele veio resolver para a comunidade de Elixir?

(VALIM)

(ADOLFO) Esta pergunta foi enviada para nós pelo ouvinte, entrevistado e colaborador do Elixir em Foco Adriano Santos, um dos criadores do projeto Spawn:

“Depois do foco na computação numérica, Machine Learning e do recente esquema de tipos, qual é a próxima grande coisa a vir para o ecossistema de Elixir?

E que projetos fora do movimento ML e Phoenix que você acompanha e vê com entusiasmo realizados pela comunidade?”

(VALIM)

~~// pergunta extra Adolfo se sobrar tempo~~

(MARIA CLAUDIA)

E se alguém estiver ouvindo ou assistindo e quiser, ou fazer pesquisa com Elixir, ou contribuir de alguma forma com a linguagem Elixir, com alguma biblioteca Elixir, o que é que você recomenda para esta pessoa?

(VALIM)

(PARTE 5 - PRÓXIMA FRONTEIRA)

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

(VALIM)

(PARTE 6 - FINAL)

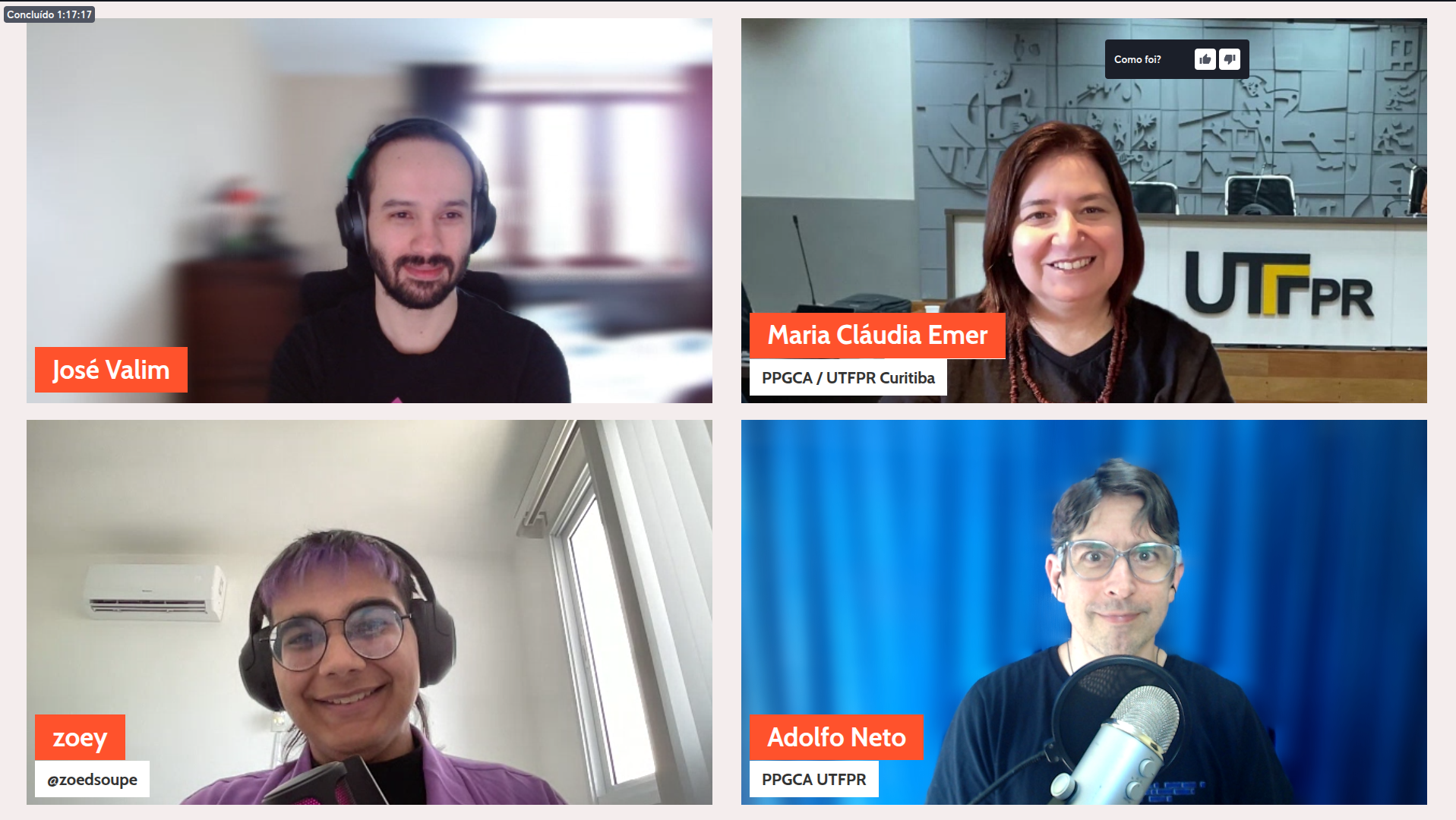
(ADOLFO) Agradece e passa para o(a) entrevistado(a).

*Lembrar que já tivemos aqui Leonardo de Moura (criador de Lean) e Roberto Ierusalimschy (co-criador e arquiteto-chefe de Lua)*

(VALIM)

(MARIA CLAUDIA) 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”

**FAZER PRINT PARA DIVULGAÇÃO EM REDES SOCIAIS**



PERGUNTA EXTRA

SOBRE POSSÍVEIS PESQUISAS DE EXPERIÊNCIA DA PESSOA DESENVOLVEDORA ELIXIR

(ADOLFO)

Versão encurtada:

Muitas pessoas relatam que a experiência do desenvolvedor em Elixir é bastante agradável. Apesar de a instalação não ser sempre simples, especialmente no Linux com **asdf**, depois de configurado, atualizar Elixir se torna fácil.

Elixir oferece um formatador automático e o **mix**, que facilita a criação de projetos. Esses fatores contribuem para uma boa *Developer Experience* (DX), que impacta a produtividade e a satisfação dos desenvolvedores.

Uma questão já foi melhorada com a unificação dos *language servers*, mas o que ainda pode ser aprimorado na experiência do desenvolvedor em Elixir?

**Versão longa:**

Um ponto que já vi muitas pessoas comentarem — sem nenhuma base científica, apenas como uma impressão a partir de interações em redes sociais — é que a experiência do desenvolvedor em Elixir é muito agradável.

Claro, é preciso reconhecer que instalar Elixir nem sempre é simples, dependendo da plataforma. Por exemplo, no Linux, se você usa o \*\*asdf\*\*, precisa primeiro instalar o Erlang, depois instalar o próprio \*\*asdf\*\*, adicionar o plugin do Erlang, baixar e instalar o Erlang, depois adicionar o plugin do Elixir e finalmente baixar e instalar o Elixir. No entanto, depois de passar por esse processo inicial, atualizar para novas versões se torna muito fácil.

De qualquer forma, muitas pessoas dizem que, uma vez instalado, Elixir é uma linguagem muito boa para se trabalhar. Ele conta com um formatador automático e o \*\*Mix\*\*, que facilita a criação de projetos já pré-configurados com diversas funcionalidades. Tudo isso faz parte do conceito de \*Developer Experience\* (Experiência do Desenvolvedor), que em Elixir parece ser bastante positiva.

Developer Experience (DX), ou Experiência do Desenvolvedor, refere-se ao conjunto de ferramentas, práticas e interações que impactam a produtividade e a satisfação dos desenvolvedores ao trabalhar com uma tecnologia, como uma linguagem de programação, framework ou plataforma.

Eu, que nunca trabalhei profissionalmente com Erlang ou Elixir, tenho a impressão de que a experiência no Elixir é até melhor do que no próprio Erlang. Mas minha pergunta é voltada para quem quiser pesquisar sobre esse tema: **o que ainda pode ser melhorado na experiência do desenvolvedor em Elixir?**

Sabemos que uma questão recorrente era a existência de três \*language servers\* diferentes para Elixir. Agora, as pessoas responsáveis por esses servidores se uniram para criar um \*language server\* oficial. Mas, além disso, o que mais pode ser feito para melhorar a experiência dos desenvolvedores que trabalham com Elixir?

Camilo, Lee, GPTs, Heron

Pergunta no Slack

PERGUNTA SOBRE CODE SMELLS E REFATORACAO

<https://bsky.app/profile/lucasvegi.bsky.social/post/3l5iksqkwxw2h>

*PERGUNTA SOBRE TIPOS*

PERGUNTA SOBRE IMPACTO DA PESQUISA NO DESENVOLVIMENTO DE ELIXIR

PERGUNTA SOBRE ELIXIR E IA

PERGUNTA SOBRE PYTHONX

CONVITE A COLABORADORES

PERGUNTA SOBRE A PRÓXIMA FRONTEIRA

incluir link?

LINKS CHATGPTS DO LEE

aquela playlist spotify

episodio bill Gates

Tese Lucas Vegi

<https://ppgcc.dcc.ufmg.br/teses/>

Artigos Lucas Vegi

<https://scholar.google.com/citations?hl=pt-BR&user=N6KnVK8AAAAJ&view_op=list_works&sortby=pubdate>

Pessoas

Guillaume Duboc <https://gldubc.github.io/>

PhD student at Université Paris Cité, under the supervision of Giuseppe Castagna <https://www.irif.fr/~gc/>

Artigos:

Guard analysis and safe erasure gradual typing: a type system for Elixir

Giuseppe Castagna, Guillaume Duboc

2024/8/26 arXiv preprint

<https://arxiv.org/abs/2408.14345>

Giuseppe Castagna, Guillaume Duboc, José Valim

The Design Principles of the Elixir Type System

The Art, Science, and Engineering of Programming - Best Paper of the Year Award

<https://www.irif.fr/_media/users/gduboc/elixir-types.pdf>

Projects

Elixir Type Checker

A (prototype) type checker for Elixir based on set-theoretic type systems.

<https://typex.fly.dev/>

Palestras

Bringing Types to Elixir by Giuseppe Castagna and Guillaume Duboc | ElixirConf EU 2023

<https://www.youtube.com/watch?v=gJJH7a2J9O8>

Quem é José Valim? Respostas de vários LLMs

<https://gist.github.com/adolfont/a95b7e37867cc1b2e24cd0e372727d8c>

Honey Potion

<https://www.youtube.com/watch?v=CoFNns01VjA>

RefactorEx

<https://github.com/gp-pereira/refactorex>

Jido framework

https://github.com/agentjido/jido

// em breve

Roteiro inicial

Apresentação de Elixir para dois públicos do Fronteiras

* + Profissionais de outras stacks
  + Pesquisadores

Qual é a próxima fronteira

da ES

da programação

do desenvolvimento de software