**Agente de Inteligência Artificial com Langchain e MCP | py live #052**

**Transcrito por** [**TurboScribe.ai**](https://turboscribe.ai/pt/?ref=docx_export_upsell)**.** [**Atualize para Ilimitado**](https://turboscribe.ai/pt/subscribed?ref=docx_export_upsell) **para remover esta mensagem.**

Boa noite, rapaziada. Boa noite, galerinha. Estamos ao vivo em mais uma PaiLive, pontualmente às 20 horas.

Boa noite, galerinha que já está no chat aí, desde cedo mandando mensagem aqui no YouTube. Galera do Instagram, boa noite. Hoje vai ser uma live bem bacana.

Eu já comecei aqui... Eu já comecei aqui logo, porque a gente tem bastante conteúdo para ver hoje. O pessoal no Instagram comentou aqui, 50 reais que ele vai começar a live com uma caneca. Cara, eu sempre dou play na live aqui e eu estou tomando água.

Geralmente é água. Geralmente, tá? Depende do dia. E aí quando eu dou play na live, eu estou sempre tomando alguma coisa, cara.

É por causa da voz, né? Para dar conta de duas horas aqui de fala. Então, galerinha do Instagram é o seguinte, tá? Galerinha do Insta, eu deixei um aviso aqui em cima, tá? Eu sempre falo isso. Que eu estou ao vivo no YouTube, né? Por que isso? Porque eu sei que aqui no Instagram é ruim de visualizar a minha tela, meus códigos.

Enfim, as coisas que a gente vai fazer juntos aqui, é ruim de visualizar nesse formato. Então, vai no link da Bill aqui do... da PaiCode. Acesse o perfil da PaiCode e vai no link da Bill.

O primeiro link que eu deixei para vocês já vai cair direto na live lá no YouTube, no aplicativo do YouTube de vocês. Aí, vocês podem jogar para outro lugar e etc. E não precisa ficar assistindo com esse formato, tá? Então, é por isso que eu deixei esse aviso, tá? Primeiro link da Bill, do perfil, cai na live lá no YouTube.

É a mesma live que vocês estão assistindo aqui. Fechou? Galerinha do YouTube, que já está online. Vários nomes conhecidos aí.

Boa noite para a galera. Hoje tem bastante conteúdo aí. O Paitonildo já está online.

O Tércio, meu amigo. Boa noite, meu amigo. Grande amigo, Tércio.

Parceirão das lives da PaiCode. Meu parceiro pessoal também. Tamo junto, cara.

É nóis. Meu parceiro ficou esquisito, né, Tércio? Foi mal, cara. Parceiro de amigo, tá, rapaziada? Vocês não vão achar besteira aí, não.

É faixa, né? É um amigão nosso aí. Amigo pessoal, né? Vai além da... Ó o Fernando. O Fernando mandou um... Ih, o Fernando está online já aí também.

Ah, essa galera é foda. Não dá para falar nada de esquisito que já pega no pé. Andy, meu amigo Anderson.

Está online já aí, cara. E já mandou um superchat. Pronto para mais uma imersão de aprendizado.

Tamo junto, cara. Agradeço aí a participação. Agradeço o superchat também.

Um grande patrocinador oficial da live junto com o Marcelo. Marcelão Batista daqui a pouco aparece aí também. Grande amigo.

Tamo junto. Vamos lá. Hoje tem bastante coisa para ver.

Eu fico às vezes aqui esperando uns minutinhos a galera entrar. Para dar tempo de todo mundo pegar desde o início o conteúdo, né. Então por isso que eu espero uns minutinhos.

A galera vem, recebe a notificação, dá tempo de abrir, ficar à vontade. E para pegar desde o início o conteúdo. Mas hoje tem bastante coisa, então a gente já vai seguir.

A galera que chegar depois pode voltar. Assiste ao vivo, depois volta. Que tamo junto, tá? Vamos nessa? Então, meus amigos.

Eu estava bem ansioso por essa live. E hoje, deixa eu só arrumar aqui. Vamos lá.

Vou arrumar aqui para vocês. E hoje, jovens, vamos falar aqui na PyLive052. Sobre agentes de inteligência artificial com o Lengchen e MCP.

Agente de IA com o Lengchen não é novidade aqui no canal. Aqui na PyCode eu já tenho algumas lives, não foi uma só, trabalhando com o Lengchen. E com inteligência artificial, obviamente.

O que é novidade aqui é o MCP. Que eu nunca trouxe conteúdo sobre. Eu tenho algum tempo que eu uso, e aí a galera me pediu.

Aí eu pensei, pô, por que não, né? Eu vou mostrar um pouco do que eu faço. De como usar essa paradinha, o que que é. Então a grande cereja do bolo aqui, nossa, dessa live vai ser o MCP. Então aqui vocês vão ver que é mais um nome bonito.

Para algo que nem é tão complexo assim. Tudo mais do mesmo, tá? Só que com o IA no meio, aí fica famoso, né? Aí gera hype. Massa? Então a live de hoje a gente vai... Vou explicar algumas coisas teóricas.

É o de praxe, tá? Um pouco de teoria, um pouco de desenho. E aí prática, tá? E a gente vai fazer, vai colocar a mão na massa. Eu vou explicar para vocês o que está acontecendo no código.

Deixa eu só ver se o chat está ok, se a conexão está boa. E aí a gente já começa aqui. Se tem alguma dúvida vocês mandem, tá? Se durante a live vocês foram tendo dúvidas, perguntas, manda por favor.

Que de vez em quando eu pauso, leio aqui. O que eu consegui responder eu respondo. Nem tudo eu consigo, talvez se tiver muitas perguntas eu peço desculpa.

Mas tudo que eu puder responder que for relevante aqui para o assunto da live. Eu vou responder, tá? Importante isso. Tem que ser relevante para o assunto da live as perguntas.

Tamo junto? Olha, a Jaira está online. Grande amiga e aluna. Está online aí, está assistindo a live também.

Tamo junto. Cuidado, hein? Ela é advogada. Cuidado que mando para ela no chat aí, hein? Brincadeira, tamo junto.

Galera, o seguinte. Antes de mais nada, o que é um agente de IA? Muitos de vocês vão estar aí e vocês já vão saber o que é. Porque eu sei que tem alunos aqui do IA Master. E tem gente que me acompanha no Insta.

Já viu posts lá sobre. Então está manjado, mas vamos revisitar um pouco, tá? O agente de IA é tipo um robô virtual. Que ele entende o que tem ao seu redor.

Ele tem ferramentas para fazer as coisas. E baseado no que você pede para ele, a inteligência artificial, que é o cérebro desse agente. Que é o modelo que você estiver usando.

Vai tomar a decisão do que fazer, como fazer e na ordem que ele vai fazer e executar. De forma autônoma, ele decide como ele vai fazer. E o que ele vai usar para fazer o que você pediu.

Basicamente é isso. E é difícil. É difícil a gente imaginar.

Para quem não manja de programação. Ou é iniciante. É difícil conseguir abstrair um agente na sua mente.

O que é? O que é essa coisa? Onde é que eu pego ele? Onde ele está? Fisicamente ele existe? Então, a resposta é que um agente de A. Vocês podem imaginar como uma classe no código. É só isso. Se for em Python, você vai escrever lá, por exemplo.

Uma variável recebe agente. Essa variável é o agente. É isso tudo que a gente está falando.

Então se a gente for abstrair para algo palpável. É isso. Eu estou falando isso porque eu sei que é difícil de abstrair.

De imaginar. Quando a gente explica o que é um agente A. É um robô virtual. Mas cadê? Sabe? A gente pensa.

É uma variável do código. É uma classe. É um objeto no código.

É isso. Primeiro ponto é esse. O grande diferencial aqui.

Até fiz um desenho aqui. Alguns vão conhecer esse desenho já. Um agente faz uso de uma LLM.

Que é um modelo de linguagem. Que é o GPT-4. O Cloud 4 Opus.

O Llama 3. Sei lá. O Gemini 1.5. Sei lá em qual versão está já. Então um agente que é essa sua classe.

Esse seu objeto no código. Essa sua variável no código. No fim das contas é isso.

Ela faz uso. Ela usa. O cérebro dela é um modelo.

De A. Que você escolher. O grande diferencial. O que caracteriza basicamente um agente.

É que ele vai ter em mãos. Ferramentas. Tools né.

A gente colocou aqui. Deixando mais gringa a coisa. Essas ferramentas podem ser qualquer coisa.

Pode ser uma função do seu código. Pode ser uma conexão com a API. Pode ser uma integração.

E usando esse cérebro dele. O agente vai decidir. Baseado no que você que está aqui.

Ou você pediu para o agente alguma coisa. Ele vai analisar o que você pediu. E ele vai ter capacidade de analisar.

Se ele tem que usar uma ferramenta. Ou outra. Ou tem que chamar uma integração.

Para poder te atender. Qual que ele vai usar. Quando que ele vai usar.

Massa? Então é isso. Não tem grandes mistérios. Isso aqui não é novidade não.

É isso. Ele vai. Ele vai decidir qual ferramenta usar.

Quando usar. Legal? Esse é o conceito de agente. E para a gente construir esses carinhas aqui.

Com o código. Com Python. Porque é a PyCode.

Vocês estão no canal da PyCodeBR. Então vamos falar de Python. Quando a gente fala de criar um carinha desses aqui.

Para fazer alguma coisa que você queira. E para servir ele com ferramentas que você queira dar para ele. A gente usa.

Geralmente. O mais famoso. Que eu indico vocês aprenderem.

É o LangChain. Cara, LangChain é muito simples. Apesar do ecossistema ser bem grande.

Nada mais é do que uma biblioteca em Python. Tem para outras linguagens também. Tem para JavaScript.

Tem para outras linguagens. Mas estamos falando de Python aqui. LangChain é uma biblioteca Python.

Que você instala. E tem um monte de função pronta. Para você construir.

Coisas com inteligência artificial. Dentre essas coisas. Agentes de inteligência artificial.

Ela te ajuda nisso. Acho que tem no desenho aqui. Deixa eu ver se eu acho.

Vamos ver se eu encontro aqui no desenho. Tem tanto desenho aqui. Olha a quantidade de coisas.

Isso aqui é das aulas do IA Master. Então eu estou reaproveitando o material aqui. Mas olha só.

Esse desenho é massa. Tem um modelo de linguagem. GPT, Gemini, Cloud, Llama.

Que é a inteligência artificial em si. E você pode utilizar ela por API. Por exemplo.

Mas isso aqui gera uma dificuldade. Para cada um desses. Para cada um desses.

Você vai ter que usar uma API diferente. O Gemini tem uma API. O Llama da meta tem outra API.

O Cloud tem outra API. O GPT da OpenAI vai ter uma outra API. Você vai ter que conhecer todas.

Se você usa o LangChain. Olha a diferença que fica. Você adiciona uma camada aqui.

No meio. E você escreve um código só. Em teoria.

Você escreve um código. Que ele abstrai para você. Essa dificuldade de conhecer.

Todas as APIs e todos os modelos. Você pode trocar. Os modelos conforme você quer.

E é muito mais do que isso. O LangChain hoje já é um ecossistema gigante. De ferramentas e funções prontas.

Para você construir. Aplicações. Integradas com inteligência artificial.

É o LangChain que nos ajuda a fazer isso. Tem outras bibliotecas. Não é só o LangChain que existe.

Tem o Creo AI. Que inclusive foi um brasileiro que desenvolveu. Agora surgiu o Agno.

Que eu estava dando uma olhada também. Mas o mais consagrado. Que geralmente empresas usam.

Projetos grandes vão usar. É o LangChain com o LangGraph. Aí a gente vai falar.

Daqui a pouco. Mas foquem no LangChain. Legal? Então o LangChain é isso.

Eu até coloco uma observação legal aqui. O LangChain. Ele é o Django.

Da inteligência artificial. Os alunos do Django Master. O framework Django.

É para o desenvolvimento web com Python. Você usa, por exemplo, Django. O Django é um ecossistema inteiro.

De coisas prontas para você criar sistemas. E criar APIs. Desenvolvimento web.

O LangChain é a mesma coisa que o Django. Só que para IA. É uma analogia que eu faço.

Claro que não é assim. Mas eu sempre falo isso. O LangChain é o Django da inteligência artificial.

Eu ouvi isso de um amigo meu. E eu tento explicar dessa maneira. Então, o que é agente? Acabamos de entender o que é um agente de IA.

Só para relembrar vocês aqui. E vamos usar o LangChain para construir nossos agentes. E vamos fazer aqui.

Um agente bem simples. Para começar só. Eu nem falei de MCP ainda.

A gente precisa ir com calma aqui. Vocês peguem os conceitos das coisas. Vocês entenderam o que é um agente.

Vocês entenderam o papel do LangChain. A gente vai fazer agora. Vamos praticar aqui.

Deixa eu botar aqui para o lado. Vamos praticar com um pequeno agente. E a gente vai adicionando coisas nele.

Vamos fazer um outro. Depois aí sim entra MCP. Vai entrar os caralho tudo.

Lá ele. Fica comigo aqui que vai dar bom. Eu prometo.

Então, legal. Eu criei uma pastinha aqui para o nosso projeto. MCPAgent.

Eu estou chamando. Criei um ambiente virtual. Não tem nada instalado.

Está em branco ainda. Eu só criei aqui para adiantar na live mesmo. Para não precisar fazer coisas do tão do zero aqui.

E legal. A gente vai ter que instalar aqui. Algumas bibliotecas.

Dentre elas. O LangChain. Eu vou usar aqui como modelo de inteligência artificial.

Eu vou usar a OpenAI. Porque eu estou com um cascalho aqui. Eu estou com restinho de saldo.

Aqui. Deixa eu ver quanto eu tenho. Eu tenho uma undoleta e trinta e dois.

Vai dar. Vai dar para fazer a live tranquila. Então, eu vou usar a OpenAI.

Porque eu tenho um crédito aqui. Você pode usar Antropic, Cloud. Qual você quiser.

Mas o LangChain. Ele tem. Por exemplo.

A OpenAI tem o forte do LangChain deles. Específico para a OpenAI. Então, quando a gente instalar aqui.

A gente vai ter que fazer. pip install. LangChain.

Underline. OpenAI. Porque eu vou usar a OpenAI com o LangChain.

Então, tem uma biblioteca separada para isso. LangChain. OpenAI.

Se você vai usar o Antropic. Que é os modelos Cloud. Você vai ter que instalar o LangChain.

Antropic. Enfim. Por aí vai.

Entendeu? Tem uns sub-pacotes aqui. Para você instalar. Então, eu instalei aqui o LangChain.

OpenAI. Eu vou precisar instalar. O LangGraph.

Então, vamos instalar aqui. O LangGraph. Eu vou usar alguns recursos deles.

De criação de agentes. E o Python.env. Para variáveis de ambiente. Vai ser necessário nesse projetinho aqui.

Acho que por enquanto é isso. Que eu lembro. Tá? Tá em branco o projeto.

Mas a gente vai ter que criar aqui. Um arquivo .env. Para armazenar a nossa variável de ambiente. Da OpenAI.

Então, eu vou ter que armazenar aqui. A API Key. Da OpenAI.

Que eu. Aqui ó. 1, 2, 3, 4, 5. Que eu vou pegar aqui. No meu painel.

Da OpenAI. Isso aqui. Vem aqui em API Keys.

Eu já criei uma aqui. Já tá no meu. Nas minhas variáveis de ambiente.

Né? Eu já adiantei. Mas você vem aqui. Cria essa variável de ambiente.

Dá um CTRL C nela. E cola aqui. Tá? Esse aqui é um exemplo.

Esse .env. Exemple. É como o nome diz. É um exemplo.

As minhas variáveis de ambiente. Estão aqui no .env. Já prontas. Tá? Então, eu só criei de exemplo mesmo.

E agora a gente pode começar. A construir aqui no main. O nosso agente.

Vai ser um agente bem simples. Então, a primeira coisa que eu tenho que fazer. Eu vou importar.

O .env. Que eu acabei de instalar o Python.env. Né? Eu vou fazer um load.env. Pra ele. Carregar nas variáveis de ambiente. Essa variável aqui da OpenAI.

Né? Então, é só isso que ele faz. Isso aqui não tem relação com o agente. De AI em si.

É um setup. Do nosso projeto. De LangChain OpenAI.

Que eu instalei. Eu vou importar aqui. O ChatOpenAI.

Já vai fazer sentido. Daqui a pouquinho vai fazer sentido. Eu vou precisar aqui também.

Do LangGraph. Que eu acabei de instalar. Uma memória.

Que está em checkpoint. Memory. E eu vou explicar pra vocês tudo.

Isso aqui. Não se preocupem. Tá? Eu vou pegar aqui o Memory Saver.

E do LangGraph também. Isso aqui. É interessante.

Essa... Essa função aqui do LangGraph. Ah! CreateReactAgent. E React não é da biblioteca JavaScript não.

De front-end. É de Reason and Action. É um agente que pensa e reage.

Ele pensa e age. Por isso, ReactAgent. Se vocês verem por aí.

Esse CreateReactAgent. Antes ele ficava na LangChain. Na biblioteca LangChain.

E agora tem na LangGraph. Eu acho que ainda tem na LangChain. Por questão de manter o legado.

Mas acabou que ele ficou na LangGraph. Por motivos que se eu citar aqui. Dá uma live inteira.

Mas é isso. Legal? Importamos o que a gente precisa. O que que um agente.

Um agente. Pra gente poder criar aqui. Um agente de inteligência artificial.

Lembram do desenho? Vamos lá. Ele precisa de um modelo. Ele precisa que eu.

Forneça pra ele um cérebro. Um modelo. No nosso caso aqui vai ser.

Pode ser o GPT-4 ou a Mini. Por exemplo. Da OpenAI.

E ele precisa de ferramentas. E precisa de prompt. Precisa de comandos.

Que eu escrevo o que eu quero. O que eu quero dele. Um agente.

Ele precisa disso pra trabalhar. Ele precisa de contexto. Ele precisa das coisas dele.

Pra ele poder trabalhar pra nós. Vamos começar com o cérebro dele aqui. Que é o mais importante.

Então eu vou criar uma variável. Chamada Model. Que vai ser o Chat OpenAI.

E o Model. Que eu vou usar. Eu vou usar o GPT-4.

Ou a Mini. Tem um custo-benefício bem legal. Pode usar o 4O.

Ele é melhor. Mas é mais caro. Mas eu vou usar aqui.

O GPT-4 ou a Mini. Que ele tá. Tá bem bacana.

O agente. Não tem no desenho aqui. Tá.

Mas é legal. Se o agente for trabalhar em formato de chat. Que você vai conversar com ele.

É legal que ele tenha uma memória. Que ele armazene o histórico da conversa. Senão a cada vez que você pedir uma coisa pra ele.

É como se você estivesse pedindo algo novo. Do zero. Então é legal você ter uma memória.

Aí entra esse Memory Saver. Que a gente tem aqui. Então eu vou criar aqui.

Uma memória. Que vai ser um Memory Saver. Do Languagraph.

Que já faz toda a gestão de memória. Pra você por baixo dos panos. Não precisa se preocupar.

Ah. Como isso aqui funciona por baixo dos panos. Daria uma live inteira.

Mas isso aqui já vai resolver. Esse Memory Saver. Ele é uma memória mais simples.

Ele só fica em tempo de execução. Na memória RAM mesmo. Mas tem aqui também.

Se eu não me engano. É. Pra você que quiser armazenar memória. Em um SQLite por exemplo.

Em um banco de dados. É. Memory. Deixa eu ver se tem aqui o. O SQLite.

Não. Então não é. No Checkpoint. É. Languagraph.

Cara. Eu esqueci a importação. É. Vamos ver aqui.

Porra. Eu esqueci. Checkpoint.

Tem alguma importação aqui. Eu vou pesquisar depois pra vocês. Porque não tava no meu roteiro aqui.

Mas. Que é uma memória que você armazena. No.

No SQLite. E aí fica persistido. Enfim.

Depois a gente vê isso. Ele vai precisar. De um.

Prompt. A gente chama de. System Prompt de um agente.

São as diretrizes dele. É tipo assim. Você é alguma coisa.

Eu vou colocar um bem simples aqui. Só pra gente testar. A gente pode elaborar melhor depois.

Vou colocar aqui. Você é um agente. E deve utilizar suas ferramentas.

Pra responder o usuário. Ponto. Você é um agente.

Sei lá. Que faz alguma coisa. Legal? Aqui é o prompt.

Aí você pode elaborar. Colocar prompts. Melhores.

Enfim. E ele precisa das ferramentas. Então vou criar aqui uma variável.

Tools. Vazia. Não tem nenhuma ferramenta ainda.

Legal. Agora a gente pode criar. O nosso.

A gente chama de. Agent Executor. Que é de fato.

O agente que vai ser executado. A gente chama. A create.

React. Agente. Que a gente importou do Languagraph.

E a gente passa. As coisas do agente. Pro agente.

O modelo que a gente criou. As tools que a gente criou. Que não tem nenhuma ainda.

Tá vazia. Mas eu já vou passar aqui. O prompt.

Que a gente criou. E o checkpointer. Que é a memória.

A memory. Que a gente criou. Pronto.

Criamos o nosso agente. Lembra que eu falei. Que é difícil.

Quando a gente fala. O agente é um robô virtual. É difícil a gente abstrair.

Você imaginar. Nossa. Mas o que é isso? O que é um agente de A palpável? Como eu imagino algo.

Que eu possa. Que seja. Existente.

Como você abstrai. Isso na sua mente. No seu pensamento.

O que é. É uma variável. No seu código. Essa variável.

Agent Executor. É o nosso agente. Que tem tudo.

Tudo que a gente precisa. Para chamar ele. E ele vai trabalhar.

Então. Quando a gente resume isso. Em código.

Quando falam de agente de A. A gente falando de programação. Isso aqui é um agente. Na linha 20.

Essa variável. Esse objeto. Essa classe.

Que foi criada. Com a função. Create React Agent.

Ela me retornou. Um agente. De A. Prontinho.

Para fazer o que eu preciso. Configurado. Com tudo que eu passei para ele.

E o que eu passei para ele? Eu passei um cérebro para ele. Que é o modelo. Que eu criei aqui.

Eu passei as tools. Para ele. Que por enquanto está em branco.

A gente vai trabalhar isso. Vai criar. Ferramentas para ele ter.

Ferramentas para ele poder usar. Eu passei. As instruções para ele.

O prompt de sistema. Tipo. Você é. Tal coisa.

Faça tal coisa. E eu passei uma memória. Para ele lembrar.

Que eu estava conversando com ele. Para ele lembrar. Do histórico das mensagens.

Pronto. Isso aqui é um agente. Assim.

Da forma mais simples possível. Que existe. Isso aqui.

É um agente de inteligência artificial. Usando Python e LangChain. Certo.

Você cria as coisas. E dá para ele as coisas. E há uma variável no seu código.

Uma classe. Um objeto. É isso.

Se você for abstrair isso. For abstrair. A ideia de um agente.

Está aqui. Entende? Legal. Deu para sacar a ideia.

Às vezes eu viajo. Às vezes eu. Parece que.

Que fuma uma jamba. Mas é. A programação é isso. É para doido mesmo.

É para maluco mesmo. Legal. Mas.

Como que eu chamo esse cara agora? Na documentação do LangChain. Tem formas de chamar esse agente. Eu posso fazer a forma mais simples.

Que eu não gosto muito. Que é só um Invoke. Se eu fizer esse Invoke aqui.

Eu passo um input para ele. Que é o que eu quiser. Se eu fizer aqui.

Sei lá. Faça x coisa. Isso aqui vai retornar.

Um resultado. Um texto. Ele vai responder.

E aí eu dou um print. No result. Pronto.

Eu vou ter a resposta. Do agente aqui. Mas eu gosto daquela.

Experiência de string. Que ele vai respondendo aos pouquinhos. Sabe? E que vai mapeando tudo.

O LangChain atualizou. Uns meses. A versão 3. Tem uma coisinha muito legal.

Que eu quero fazer aqui com vocês. Que é bem massa. Está na documentação.

Segunda página da documentação do LangChain. Tem isso aqui que eu vou fazer. Não é mistério nenhum.

Primeiro a gente tem que criar uma configuração. De memória. Vou mostrar para vocês já como funciona.

Então você vai passar aqui. Uma configuração. Configurable.

Vocês já vão entender do que se trata. Thread. ID.

Pode ser 1. Isso aqui. Thread ID. Eu preciso passar aqui.

Como se fosse o ID da conversa. Sabe quando você vai no chat GPT. E dá uma nova conversa lá.

Ele gera um ID novo para aquela conversa. Para o agente olhar para o histórico. Apenas daquela conversa.

Eu vou colocar aqui um ID 1. Poderia ser qualquer coisa aqui. É um ID único. De cada conversa.

Ele precisa disso. Então a gente vai criar aqui. E agora.

Eu quero que no terminal. O meu agente fique me perguntando. Fique aparecendo sempre.

Para eu digitar mais coisas. Então a gente vai fazer. Um IOTrue aqui.

Um laço. Para a gente ficar. Perguntando coisas para a gente.

E o input do usuário. A pergunta que eu vou fazer. Para o agente.

A gente vai estruturar aqui. Essa role de mensagem aqui. Vai ser do usuário.

User. Legal. E o conteúdo da pergunta.

Da mensagem que eu vou mandar. Vai ser um input do Python. Porque eu vou digitar no terminal.

E eu vou botar aqui. Digite. Para eu mesmo saber que tem que digitar.

Vocês já vão entender. Já vai ficar bem claro. Agora que vem aquela parte.

De ele responder em string. E eu poder ver no terminal. A gente vai fazer um for.

Nos passos que o agente vai trabalhar. No agente executor. A. String.

Cadê aqui? Não. Ponto string. Aqui.

Não vai ser async. Esse. Por enquanto.

E aqui. Eu vou passar. As mensagens.

Que são. Input message. Input message.

Vou passar config. Opa. E a string.

Pera aí que eu fiz errado aqui. Vou passar config. E a string mode dele.

Vai ser. Values. Legal.

Já vou explicar para vocês. Do que se trata. Isso aqui é um for.

Que eu estou fazendo. Apesar da sintaxe estar. Um pouco esquisita.

Está na documentação. Do próprio linkchain. Segunda.

Terceiro exemplo. E a gente vai. Vai debugar esse código.

Vai entender junto. Só estou terminando ele aqui. Legal.

Que desgraça está fazendo aqui. Até aqui vocês entenderam. Até aqui.

Criamos um agente. Com as coisas que ele tem que ter. De agente.

Aqui eu só estou fazendo. Uma gambiarra toda. Uma volta toda.

No código. Com Python. Para poder ficar perguntando coisas.

Para ele no terminal. Estou fazendo um laço infinito aqui. Para ele sempre.

Eu mando uma mensagem. Aqui tem um input. Está vendo? Eu digito alguma coisa no terminal.

Dou um enter. Ele vai invocar o agente. Agente executor.

String. Está vendo? Isso aqui vai invocar. O agente.

Vai chamar o meu agente. Vai botar o agente para trabalhar. E eu vou passar para o agente.

A mensagem aqui do input. Que eu perguntei. A pergunta que eu fiz.

As configurações. Para ele saber qual é a thread id. Que eu estou.

Qual é o id da conversa. Que eu estou. E o string mode.

Values. Para ele responder. Cada passo a passo.

Do que ele está fazendo. Ele vai. Ficar respondendo.

E eu vou ficar dando um pretty print. Esse pretty. Pretty.

Ou pretty. Print. É uma função.

Do. Lang chain. Para fazer um print bonito.

Como o próprio nome está dizendo. É um print. Bonito.

No terminal. Para eu ficar vendo. O que o agente está fazendo.

Então. Apesar da sintaxe ser estranha. Isso aqui tudo poderia ficar.

Para ficar menos estranho. Ficaria em uma linha só. Olha só.

É um for. Isso aqui. É um laço de repetição.

Tá. E isso aqui. Poderia ficar em uma linha só.

Estou chamando. Que o agente executa o string. E passando os parâmetros para ele.

Poderia ficar em uma linha só.

**Este arquivo é mais longo que 30 minutos.**

[**Atualize para Ilimitado**](https://turboscribe.ai/pt/subscribed?ref=docx_export_upsell) **em** [**TurboScribe.ai**](https://turboscribe.ai/pt/?ref=docx_export_upsell) **para transcrever arquivos de até 10 horas.**