**APRENDA HOJE N8N E AUTOMATIZE QUASE TUDO!**

**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.**

Saber criar automação é chegar a ser uma arte aos olhos de um dev, né? Agora, conseguir criar um fluxo de trabalho bem completinho, que resolva aquelas tarefas repetitivas, é um verdadeiro sonho. Ainda mais quando a gente consegue fazer isso usando uma ferramenta open source gratuita. E é isso que nós vamos te ensinar nesse monocódigo.

Você deve ter visto aí no título, né? Ao usar o Anetian. Já conhece? Você cria workflows com no e low code, podendo conectar APIs, serviços e bancos de dados de forma visual e escalável. Nós vamos criar um workflow para verificar o status de alguns sites e caso algum deles esteja fora do ar, que ele mande uma mensagem para a gente.

E é claro que vamos te mostrar como instalar, configurar e utilizar essa ferramenta. Nesse tutorial, é agora. E seja muito bem-vindo a mais esse vídeo aqui no Código Fonte TV, onde nós vamos explorar o Anetian, que é low code, no code e permite fazer personalizações com JavaScript e Python.

E pode ser executado de forma local ou em servidores próprios. Nós vamos te mostrar como usar local com o Docker e também através da instalação em um VPS com o Docploy. Nesse caso, vamos utilizar o nosso VPS da Hostinger, que, por sinal, já tem um plano especial para o Anetian.

Com um clique, você já instala a ferramenta. Isso porque o VPS já vem pré-instalado com um template com Ubuntu e o Anetian. Dentro de um ambiente Docker, você implanta de forma muito fácil e rápida os fluxos de automação.

E você ainda pode usar um template para configurar o Anetian no modo de fila, para melhorar a escalabilidade. Nós te indicamos justamente porque, além do Anetian, você ainda conta com todas as outras vantagens de utilizar um VPS da Hostinger. Só para citar algumas.

Eles possuem excelente hardware, suporte, backup semanais grátis e automáticos e detector de malware. Você ainda tem até 30 dias para pedir reembolso e os servidores você pode escolher no Brasil. É, mas sobre o reembolso, eu duvido muito que isso aconteça, porque nós estamos usando há anos as soluções da Hostinger e tem sido uma experiência excelente com um preço muito bom.

Então, corre aqui no link do VPS com o Anetian para você conhecer aqui na descrição. E usando o nosso cupom CODIGOFONTE, você ainda ganha um desconto extra. Bora ver então como é que fica para instalar o Anetian local.

Eu criei aqui um arquivo docker-compose. Vou usar a imagem original deles, não vou fazer nenhuma modificação na imagem. Fica tranquilo que o link desse repositório com essas configurações no GitHub a gente vai deixar na descrição também.

Mas na documentação do Anetian também tem as instruções para você usar via docker. Está um pouquinho diferente dessas aqui, mas está valendo também. E vale lembrar que a gente está usando o docker aqui através do próprio VS Code, que está muito fácil de utilizar.

Não, se você já tem a extensão dele aqui, fica moleza. Aí eu configurei as variáveis de ambiente que já vêm na imagem do Anetian, que é o Anetian host, o protocolo, nesse caso aqui a porta eu vou manter padrão. E a gente pode configurar também o host.

Aqui no meu arquivo env, eu já coloquei aqui qual é o domínio, qual é o subdomínio. Nesse caso a gente vai rodar local. O importante, como a gente trabalha também com datas e tudo, é muito importante a gente configurar a timezone.

Isso aqui não deixa de configurar para não dar problema. E aí nesse caso a gente usa as variáveis aqui do nosso docker-compose. Está aqui com as portas.

Uma coisa que eu fiz diferente aqui, é que eu criei uma pasta data, justamente com os meus dados. Nesse caso aqui, vai ser uma lista de links que vai estar aqui. Lista de links que eu coloquei aqui.

Bem simples, num txt puro. É, essa pasta também vai para dentro do docker, para a gente ter acesso a ela. E aí acabou.

A gente criou um volume também, ou seja, se eu reiniciar esse docker tudo, ele vai manter todas as minhas configurações lá e não vai perder. Aí para rodar, a gente pode rodar no terminal ou vindo aqui e rodar todos os serviços. Como eu já baixei essa imagem uma vez, então ele só estartou aqui o container.

Eu posso vir aqui, ver quais são os containers que eu tenho. Está aqui, está vendo? Se eu quiser navegar, isso aqui também é legal, dessa extensão. Se eu quiser navegar nas pastas, vamos ver se o home data... Aqui, já está lá o meu arquivo, o meu arquivo txt.

Mas a ideia é a gente rodar isso através do nosso VPS lá na Hostinger, né? Pois é, aqui a gente usa o Dockploy, que é tipo a Vercel, tá? Ele cria tudo através de containers docker, então fica muito simples. Eu já trouxe aqui o NHM, já está criado, mas olha só que legal. Se eu quiser criar um outro tipo de serviço nesse projeto, ele já tem um template para vários tipos aqui.

Eu posso criar um banco de dados, eu posso criar várias coisas e também usar o meu próprio Docker Compose para fazer isso. E no Dockploy, assim como outras formas também que tem para você instalar as aplicações, no VPS da Hostinger, você pode, olha, Grafana, você tem o próprio NHM, Wordpress, vai navegando aqui os templates que já estão prontos, né? Você pode, obviamente, criar o seu, né? Mas isso aqui é muito bom. Facilita muito, né? Porque a instalação fica ali em um clique, né? Clicou, ele já prepara o ambiente.

Olha que legal, dá para dockerizar aqui o macOS e o Windows também, olha só que louco. Então, é muita facilidade, realmente, para você instalar muito rápido, para você poder configurar as suas aplicações e isso deixa a gente desenvolver coisas bem mais rápido, porque você roda local e depois você joga isso aí lá para o seu servidor para você ter acesso aí via web. Uma vez a gente entrando em localhost 5678, na porta, a gente já tem aqui os dados para a gente fazer a configuração.

Só você colocar o seu e-mail, o seu nome e a sua senha e rapidamente você faz login e já está pronto para usar. Uma vez configurado, você já entra nessa página aqui, que é uma espécie de um welcome, né? E aí, vamos fazer um overview aqui. Lembrando que a gente usou a imagem padrão do N8n e ele, por padrão, o banco de dados é um SQLite, mas ele também tem suporte ao Postgre também, tá? Então, se você quiser trocar isso, tem que ir nas configurações, naquelas variáveis de ambiente e para você fazer essa troca, tá? Então, algumas coisas que a gente tem aqui, por exemplo, como variáveis, ele é pago, tá? Então, é uma ferramenta open source, é gratuito, você pode usar, mas tem algumas coisas aqui que já são para o plano pago.

Obviamente, se você está usando bastante, você vai sentir a necessidade depois até de assinar, né? Mas já dá para fazer muita coisa aqui. E ele tem também uma parte aqui de templates. Isso aqui ele já pega da web e aí você já tem coisas prontas aqui que você pode utilizar.

Eu achei muito interessante porque, você vem aqui, por exemplo, em financeiro, aí tem 77 aqui. E aí ele diz, ó, crie o seu primeiro chatbot de análise de dados com inteligência artificial. E aí, quando você clica aqui na documentação, ele abre o modelo, ali o workflow, mas navegável, entendeu? Então, você consegue ver o workflow, não só a imagem, mas se você quiser, poxa, como é que esse cara configurou isso aqui? Você clica, ele abre aqui para você ver realmente como está configurado.

Então, a documentação é sensacional, cara. Dá para aprender muito aqui com essa parte aqui da documentação. E é legal que fica bem prático, né? Porque você acha o workflow, você acredita que seja o que você necessita, pô, visualizando dessa forma, você já mata.

Não, é exatamente isso que eu quero, então é só seguir com esse workflow, né? E aí você pode baixar, você pode testar ele também, né? Fora isso, aqui na configuração, a gente vai em settings, você tem muitas configurações, algumas realmente você precisa desbloquear, né? Usando um plano, mas é aquela coisa de os seus dados pessoais, o próprio NHN é uma API também, você pode ativar uma chave, então tudo que você fizer nele, você pode automatizar com ele mesmo também, até para uso externo, né? Tem aqui os secrets, né? Nesse caso aqui, tem que ser com o plano enterprise também e várias outras configurações. Eu acho que todas aqui são para o plano enterprise. Mas o que importa para a gente, para a gente poder configurar as nossas coisas, é o workflow, as credenciais, e as execuções, isso aqui é o log de todas as execuções que foram feitas.

Simples assim, não tem... Não tem firula e essa parte é gratuita. O que seria então as credenciais? Quando a gente vai usar qualquer coisa, vamos automatizar usando o Telegram, vamos usar com o Google Sheets? O Slack. O Slack ou, sei lá, o YouTube, qualquer coisa que você pensar que precisa de API, a gente precisa ter essa credencial já pré-configurada e isso a gente faz justamente aqui.

Então, por exemplo, a gente vai usar, nesse nosso, o Telegram, para enviar as mensagens. Olha só quantos já tem aqui já pré-programados para a gente utilizar. Se a gente quisesse, por exemplo, o Slack, como a Vanessa falou, tem ali.

Se a gente quisesse o WhatsApp, tem ali também. O próprio Google Sheets tem ali também. Então, cara, tem muita coisa.

Nesse caso, a gente vai configurar a credencial do Telegram. Então, a gente seleciona, aqui já vem o token para eu preencher e eu vou salvar. Uma vez que ele salva, ele já faz o teste, dizendo que já está feito, está verdinho aqui, bonitinho.

Vamos combinar, né? Essa parte de pegar as credenciais e tudo mais, acho que é a parte mais chata de tudo isso, né? De todos esses serviços. Cada um tem um jeito, né? O Telegram, você chama um bot master lá e conversa com ele, que ele te passa. O Google é enjoado que só, né? É, cada um tem o seu jeito, né? Então, isso vai levar mais tempo do que a gente criar o próprio workflow, tá? E aqui, quando a gente cria a credencial, a gente tem também uma parte de colaboração, também é paga e você pode ver outras informações aqui, tá vendo? Quando foi criado e etc.

Então, uma vez que a gente tem essa configuração aqui, a gente já consegue usar o nosso workflow. Vamos criar aqui um workflow? A gente pode criar aqui, usando um agente de IA, inclusive, já tem suporte para MCP, inclusive, a gente mostrou esses dias criando um MCP server, né? Dá para usar agora também, usando o NHM. Mas a gente vai criar aqui um workflow do zero, para você ver que a tela do workflow também é super simples, os caras conseguiram realmente fazer algo muito intuitivo, né? Então, aqui eu posso colocar o nome do meu workflow, nesse caso aqui, status sites do código fonte.

Lembrando que, se você já fez o seu workflow ou você quer importar o workflow de alguém, nesse menuzinho aqui, você pode duplicar, fazer download em arquivo JSON, tá? Então, esse mesmo que a gente vai criar, a gente vai deixar também lá no repositório, tá? Você pode importar, você pode importar por arquivo, por URL e você tem as configurações dele aqui, ó. Então, você tem o tipo de motor ali deles, né? Que vai rodar o workflow. Você pode criar um workflow, na verdade, que é o erro. Então, se acontecer algum erro, você roda um workflow.

Olha que louco, né? Tem o timezone, já está tudo aqui configurado e tem outras configurações aqui. Bom, como a ideia é verificar ali se o site está ativo, a gente vai ter que criar meio que um crono, né? Que é justamente para, de tempos em tempos, fazer essa verificação. Então, acho que esse é o primeiro passo ali do nosso workflow.

Todo workflow, ele precisa de um trigger, ou seja, é um gatilho para ativar. E tem vários triggers que você pode utilizar, tá? Cada passo desse workflow, eles chamam de nó, tá? Então, essa nomenclatura é importante para a gente saber. Se a gente vai aqui em adicionar, ou aqui também, ele vai dizer, ó, adicionar trigger, já que não tem nada, né? Manualmente, a partir de um evento de alguma aplicação ou agendado, que é esse aqui que é o crono, ou seja, eu posso dizer de quanto e quanto tempo ele vai ser executado.

A partir de um webhook, ou seja, aconteceu algo numa URL, ele ativa ali o workflow. Ao submeter um formulário, cara, e muita coisa, uma mensagem de chat e outras coisas, alteração em arquivo, você pode criar o trigger que você quiser. Então, por exemplo, eu vou criar um desse aqui, que é um manual, ou seja, se eu clicar aqui, ele já executa.

E eu posso criar também, ao mesmo tempo, como já tem um trigger, ele já abriu outras opções agora. Eu quero voltar no trigger, e aqui eu coloco um trigger de cronjob, né? Então, por exemplo, se eu quiser verificar, vamos verificar site, vamos verificar aqui de 10 em 10 minutos, por exemplo. E aí não precisa nem explicar muito, que está tudo bem intuitivo.

Colocou ali 10 minutos, você adiciona essa regra, a configuração desse pedaço, acho que já está pronto. Se eu quiser adicionar outra regra aqui, mas eu quero também executar uma vez por mês, duas vezes por semana, ele acrescenta também isso, você pode fazer. Então, colocamos aqui, olha, ele já está aqui.

São dois triggers que nós teremos, então, no nosso workflow. Mas a gente vai precisar ler um arquivo, não é? Então, a gente pode vir agora. Aqui está tudo categorizado por várias formas diferentes de você executar, né? Então, já tem aqui de IA, OpenAI, muita coisa, classificação de texto, tudo.

Então, você tem que explorar isso aí, para você ver a quantidade de coisas que dá para fazer, é muita coisa. Mas, como a gente quer ler um arquivo, então, eu vou botar aqui, Read and Write File. Então, nesse caso aqui, eu vou ler um arquivo.

Lembra o caminho do nosso arquivo? Home, Data, então, é meus. Qual é a vantagem disso? Se ele não achar o arquivo, ele vai dizer aqui. Então, eu posso tocar aqui em testar esse passo, olha lá.

Ele leu o arquivo e ele já mostrou os dados dele aqui. Nesse caso, você vê que voltou um binário. Se eu colocar em JSON, ele não volta os dados.

Então, é um arquivo de texto, as informações estão em binário. Tem que guardar essa informação, tá? No caso de tabela, você vê, olha. Então, você consegue ver um output com várias informações, nesse caso aqui, com, por exemplo, quando a gente coloca para visualizar o arquivo, aí ele mostra aqui o conteúdo.

O conteúdo que está no arquivo, né? Mas, senão, ele já traz todas as informações do arquivo em si. E você vê aqui, olha, o input dele foi o trigger. Tem esse aqui que a gente está configurando e a saída.

Tem outras opções que a gente pode colocar, como o nome do arquivo, nesse caso aqui, né? Então, se a gente quiser especificar certinho o nome do arquivo, a extensão, o mintype, dá para a gente fechar essa configuração. Se a gente for aqui em settings, praticamente todos têm isso também. A gente pode, por exemplo, se o arquivo não existir e deu problema.

Então, eu posso forçar sempre voltar algum tipo de dado. Por exemplo, se der um erro, nesse caso aqui, ele para o workflow. Ou ele continua fazendo o erro ou o acerto, né? Ele tem essas duas opções.

Ou simplesmente continua e se dane. Esse passo aqui não seria obrigatório. Então, você vê que ele está aqui, eu posso colocar ele aqui, olha.

Se eu quiser testar o workflow, eu posso clicar aqui, olha, e ele testa aqui, olha, testa aqui e foi. Se eu vier por aqui, ele também vai, olha. Então, nesse caso, olha, se a gente vier aqui em execução, a gente consegue ver aqui já o histórico de vezes que a gente executou, ver o que aconteceu em cada execução.

Agora, é interessante que você pode criar também um nó de configuração, que aí justamente ajusta. Por exemplo, ali a gente colocou o nome do arquivo e isso poderia estar em uma espécie de uma variável. A gente vai utilizar também as informações do Telegram, que também podem estar nesse nó de configuração, né? Até porque se você precisar reutilizar ele durante o seu fluxo, você só tem ali um lugar onde tem as variáveis.

Então, a gente tem essa opção aqui do Edit Fields. Olha que interessante, a gente pode trazer um JSON com todas as configurações ou criar um mapeamento dos campos manualmente. Se eu colocar aqui, por exemplo, arquivo links, e aí eu posso dizer justamente onde que ele está.

Então, está aqui, vou tocar o teste aqui, está funcionando. Só que o que acontece, eu preciso desse cara aqui antes, senão eu não vou conseguir pegar, né? Então, se eu colocar ele aqui, vou excluir, coloco ele aqui, excluo aqui, e ele vai vir conectado aqui. Olha só o que acontece agora.

Quando eu venho aqui para ler o nome do arquivo, ao invés de colocar fixo, eu posso colocar uma expressão. Aí, eu vindo aqui, eu venho, olha, aqui em Edit Fields, eu consigo pegar justamente o nome desse arquivo. Eu arrastei e ele já me dá até o resultado, qual é o nome do meu arquivo.

Se eu só conferir se é isso mesmo, é traço, não underscore. Então, eu posso vir aqui, atualizar e colocar um traço. Agora, eu posso até testar o meu workflow.

Olha lá, ele conseguiu ler o arquivo, né? Mas lembra que eu falei que o arquivo txt está em binário? Então, a gente precisa converter esse binário para um tipo de dados, nesse caso aqui, para uma string. Como é que a gente pode fazer isso? Com código, que eu acho que, na minha opinião, é o melhor. Ou dá para fazer de alguma forma aqui, eu acho que de conversão de dados, mas eu prefiro código.

Então, a gente tem aqui uma área para rodar código e o que eu fiz aqui foi bem simples, deixa eu abrir aqui para ver melhor. Eu peguei o item 0, ele é binário, tá vendo? E aí, eu peguei o buffer dele e converti ele em string UTF-8. Estou com os dados aqui.

Agora, eu usei uma expressão regular que quebra a linha. Eu pego os dados aqui e quebra a linha de cada um e depois faz um trim se tiver algum espaço em branco para não ter problema com links, né? O que importa aqui no fim é que a gente retorna um JSON com as URLs, né? Faz um map aqui dessas URLs e retorna um JSON disso. Ou seja, a gente já preparou todos os dados que a gente precisa para fazer a verificação de cada um dos sites, né? E aí, você está vendo que eu chamei ele de links aqui, tá vendo? Esse aqui é o nome da propriedade.

Eu tenho que voltar aqui na leitura do arquivo e eu seleciono qual é o campo de saída dele. Então, eu vou colocar links. Vou testar aqui.

Aqui funcionou. Posso vir aqui agora e testar também. Aí, funcionou.

Está vendo? A saída já é aqui os links. Então, por exemplo, aqui a saída eu posso colocar, olha, em formato de tabela. Deixa eu mexer aqui, ó. Fiz um ajuste aqui na minha expressão regular.

Agora sim, ó, ele está voltando a tabela, o JSON, tá vendo? Estão todas as informações certinhas aqui dos links. Bom, agora nós já temos ali a lista de sites que a gente quer verificar. O próximo passo é justamente fazer essa requisição para ver se eles estão voltando ali com status 200 ou qualquer outro que a gente vai mandar mensagem, né? Então, vimos aqui a HTTP request.

A gente vai fazer justamente um GET e na URL, a gente vai usar uma expressão. Aqui no código, a gente tem a URL, ó. Se a gente arrastar para cá, ele já vai trazer quais são as URLs que a gente vai testar. E aí, se a gente quiser, olha, testar aqui, a gente já pode fazer um teste para ver o que ele vai trazer.

E ele vai trazer, na verdade, se eu colocar em um JSON, os dados do site, todos. Só que só trouxe os dados. Na verdade, eu quero todas as informações, né? Então, nesse caso, para não voltar só os dados, eu preciso adicionar uma opção.

Eu vou aqui em Response. E nesse caso, eu vou voltar não só os dados, mas também o status, ó. Então, eu posso testar novamente. E agora a gente tem, ó, status code 200.

É isso que a gente quer. Mas só complementando, se a gente fizer uma chamada e voltar algo diferente de 200, na verdade, ele vai voltar um erro. Então, o que a gente precisa configurar aqui é que, ao dar o erro, a gente continua.

Olha só o que ele vai fazer. Ele vai voltar, olha, um caminho para o sucesso e um caminho para o erro. Então, a gente vai precisar configurar justamente o erro.

A gente vai testar isso daqui a pouco. A gente tem a saída desse nó para quando as informações tiverem acabado. Tiverem corretas e quando tiver um erro, né.

Nesse caso, quando a URL ali não for encontrada. Eu adicionei uma URL que não existe, justamente para a gente testar esse nó. E aí, a gente vai precisar utilizar um nó de IF, justamente para verificar esses dados, ó. Então, como eu já tenho aqui um nó que não existe, ó, e o que eu preciso é justamente o status, eu posso colocar ele aqui e verificar.

Se ele não for igual a 200, eu sei que teve um problema. Nesse caso aqui, ele passou, quando eu testei. Então, realmente, teve um problema.

Aí sim, a gente envia a mensagem via Telegram. Escolhemos o Telegram e vamos enviar uma mensagem de texto. Agora sim, olha, já pega aqui a credencial que a gente testou e vamos mandar essa mensagem.

Então, para a gente configurar, a gente tem o ID que a gente vai configurar aqui na parte de config, lembra? Então, eu posso criar aqui um campo chamado Telegram ID. Eu preencho aqui com o valor do Chat ID e aí a gente já consegue utilizar o nó. Então, para configurar o Chat ID, a gente pode, justamente, vir aqui, ó, no nó de Edit Fields.

Que é justamente onde a gente colocou as variáveis ali de configuração. E aí, ele já traz aqui o valor. E aqui a gente pode colocar o chat.

O que a gente vai escrever na tela? Bom, aqui é abrir aqui essa telinha um pouco maior. E aí eu escrevo um texto. Nesse caso, usando aqui as variáveis, olha, URL e eu colocando aqui a data, né? Interessante que quando a gente escreve esse aqui, ele já até mostra aqui, olha, como vai ficar.

Se a gente escrever alguma coisa errada, que não existir, ele já diz aqui que está indefinido. Agora sim, espero que você tenha conseguido entender, vamos fazer um teste. Viu que ele executou, encontrou um erro.

Os outros sites aqui estão no ar, ainda bem. Testando aqui no meu Telegram, olha só, já está o meu bot lá e ele já trouxe a mensagem, olha. Ó, lindo, só não funcionou a quebra de linha.

Quebra de linha, a gente vai ter que ver o que a gente fez aqui de errado. Agora, fala a verdade. É simples de fazer esse workflow, não é não? Gostou? Então, o que fica de dica? É você ter as suas credenciais e saber exatamente o que você quer antes de fazer.

Dá uma navegada, vê quais são as possibilidades, tem muita coisa. E se você curtiu, pede para a gente mais automações que a gente vai criar e vai mostrar para vocês. Eu quero muito testar MCP, testar com o OpenAI.

Eu já testei, na verdade, tá? Mas se você quiser coisas mais complexas, diz aqui nos comentários o que você precisa e a gente vai elaborar junto aqui para trazer para você também. Espero que você tenha curtido aí esse tutorial, né? Ficou um pouquinho grande, mas valeu a pena, viu? E a gente se vê no próximo vídeo. Tchau! Tchau!

**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.**