

Fundamentos

250812-1658

# REXX1

Enterprise IT Duct Tape

Tradução automatizada usando o serviço IBM Globalization

- [FITA ADESIVA DE TI EMPRESARIAL](#)
- [1 INSTALE O CLI ZOWE](#)
- [2 INSTALAÇÃO DO ZOWE NPM NO MACOS](#)
- [3 INSTALAÇÃO DO ZOWE NPM PARA WINDOWS](#)
- [4 FONTE E TERMINAL](#)
- [5 CRIAR UM PERFIL TSO](#)
- [6 DEFINIR PERFIS PADRÃO](#)
- [7 EXECUTE SEU PRIMEIRO REXX](#)
- [8 INICIAR UM ESPAÇO DE ENDEREÇO](#)
- [9 EXECUTAR O MESMO REXX](#)
- [10 ADIVINHE? OUTRO EXECUTIVO](#)
- [11 VEJA COMO O CÓDIGO É BONITO](#)
- [12 BATE-PAPO AO VIVO COM O REXX](#)

# ENTERPRISE IT DUCT TAPE

## O Desafio

Neste desafio, você conhecerá o **Rexx** ("Restructured Extended Executor"), uma linguagem de programação conhecida por sua simplicidade, potência e relativa facilidade de uso. Você aprenderá a executar um programa **Rexx** a partir de uma linha de comando e a iniciar um "espaço de endereço" **TSO** para executar um programa **Rexx** interativo.

Para interagir com **TSO**, você precisará da interface de linha de comando **ZOWE** (**CLI**) instalada.

## Antes De Começar

Esse desafio exige algumas configurações em seu ambiente de terminal.

Certifique-se de seguir as etapas de instalação do site [ZOWE CLI](#).

REXX1|250812-1458

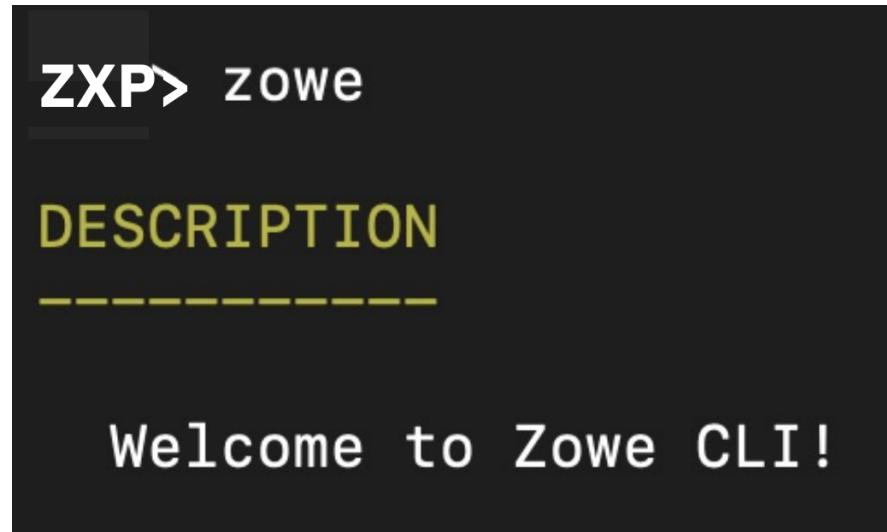
## Investimento

Etapas	Duração
12	60 minutos

# 1 INSTALL THE ZOWE CLI

Até agora, você tem usado o plug-in `Zowe Explorer` para `VSCode`, mas o `Zowe` pode fazer muito, muito mais e é responsável por trazer muito mais para o mainframe.

Para deixar claro, você instalará a interface de linha de comando `Zowe` (`CLI`) em seu próprio computador, *não* no mainframe.



REXX1250812-1658

Você usará o site `Zowe CLI` para trabalhar com o serviço `z/OSMF` que está em execução no mainframe, mas conduzirá a maior parte desse desafio em seu próprio computador.

`Linux` os usuários talvez precisem explorar um pouco para descobrir o que funciona em seu sistema específico, mas isso deve se parecer mais com as etapas do site `Mac` , bastando substituir o arquivo de perfil de shell correto.

`Windows` os usuários talvez precisem modificar a variável de ambiente do usuário `PATH` para poder executar com êxito o `zowe` como um comando - mais sobre isso posteriormente.

# 2 ZOWE NPM INSTALL ON MACOS

## LEIA toda esta seção antes de instalar a CLI do ZOWE

Primeiro, saia do seu aplicativo VSCode - verifique se o VSCode não está em execução.

Como o [Zowe CLI](#) usa pacotes de nós, ele exige que uma versão compatível do [Node.js](#) esteja instalada e devidamente acessível.

Para usar os pacotes do node no sistema operacional, será necessário carregá-los em um diretório global '.npm' que possa ser acessado por usuários comuns.

As etapas a seguir configuram isso, informam ao [npm](#) (o Gerenciador de pacotes do Node) para usá-lo e o incluem na lista normal de locais onde o sistema operacional procura programas para executar.

```
ZXP>mkdir ~/.npm-global
ZXP>npm config set prefix '~/.npm-global'
echo "export PATH=~/.npm-global/bin/:$PATH" >> .zprofile
source .zprofile
npm i -g @zowe/cli
ZXP>echo "export PATH=~/.npm-global/bin/:$PATH" >> .zprofile
ZXP>source .zprofile
ZXP>npm i -g @zowe/cli
npm WARN deprecated request@2.88.2: request has been deprecated, see https://github.com/request/request/issues/3142
npm WARN deprecated har-validator@5.1.5: this library is no longer supported
> @zowe/cli@6.33.0 preinstall /Users/rccruicks/.npm-global/lib/node_modules/@zowe/cli
> node ./scripts/preinstall

/Users/rccruicks/.npm-global/bin/bright -> /Users/rccruicks/.npm-global/lib/node_modules/@zowe/cli/lib/main.js
/Users/rccruicks/.npm-global/bin/zowe -> /Users/rccruicks/.npm-global/lib/node_modules/@zowe/cli/lib/main.js

> @zowe/cli@6.33.0 postinstall /Users/rccruicks/.npm-global/lib/node_modules/@zowe/cli
> node ./scripts/validatePlugins

Since you re-installed Zowe CLI, we are re-validating any plugins.
No plugins have been installed into your CLI application.
+ @zowe/cli@6.33.0
added 287 packages from 202 contributors in 19.319s
ZXP>
```

REXX1250812-1458

Para a maioria dos usuários do [MacOS](#), esse conjunto de comandos deve ser suficiente.

1. `mkdir ~/.npm-global`
2. `npm config set prefix ~/.npm-global`
3. `echo "PATH=~/.npm-global/bin/:$PATH" >> .zprofile`

```
4.echo "export PATH" >> .zprofile  
5.source .zprofile  
6.npm i -g @zowe/cli
```

**Observação:** se o seu shell de terminal padrão for `bash` em vez de `Zsh`, essas instruções precisarão ser modificadas para atualizar o arquivo `.bashrc` ou `.bash_profile` em vez do arquivo `.zprofile`.

Depois de instalar com êxito a CLI do ZOWE (ou chegar a um ponto em que esteja claro que não é possível), reinicie o VSCode.

# 3 ZOWE NPM INSTALL FOR WINDOWS

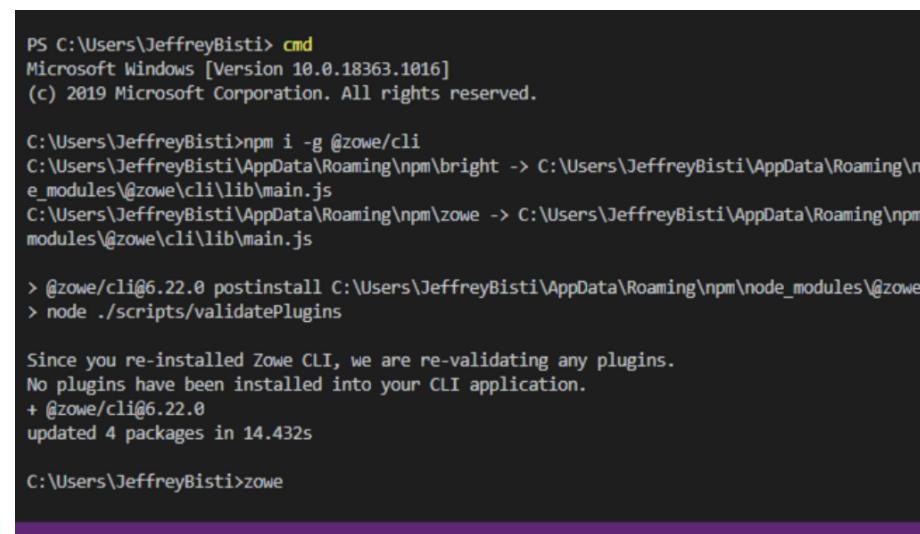
## LEIA toda esta seção antes de instalar a CLI do ZOWE

Primeiro, saia do seu aplicativo VSCode - verifique se o VSCode não está em execução.

**Em seu laptop ou estação de trabalho Windows , primeiro mude para um terminal CMD em vez do terminal padrão PowerShell e, em seguida, instale o Zowe CLI usando o 'npm', o gerenciador de pacotes do Node.**

Isso deve funcionar para a maioria dos usuários, embora seu resultado possa ser diferente do que você vê na captura de tela.

1. digite cmd (isso mudará o shell de PowerShell para CMD )
2. npm i -g @zowe/cli
3. zowe



```
PS C:\Users\JeffreyBisti> cmd
Microsoft Windows [Version 10.0.18363.1016]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\JeffreyBisti>npm i -g @zowe/cli
C:\Users\JeffreyBisti\AppData\Roaming\npm\bright -> C:\Users\JeffreyBisti\AppData\Roaming\npm\_modules@\zowe\cli\lib\main.js
C:\Users\JeffreyBisti\AppData\Roaming\npm\zowe -> C:\Users\JeffreyBisti\AppData\Roaming\npm\_modules@\zowe\cli\lib\main.js

> @zowe/cli@6.22.0 postinstall C:\Users\JeffreyBisti\AppData\Roaming\npm\node_modules@\zowe\
> node ./scripts/validatePlugins

Since you re-installed Zowe CLI, we are re-validating any plugins.
No plugins have been installed into your CLI application.
+ @zowe/cli@6.22.0
updated 4 packages in 14.432s

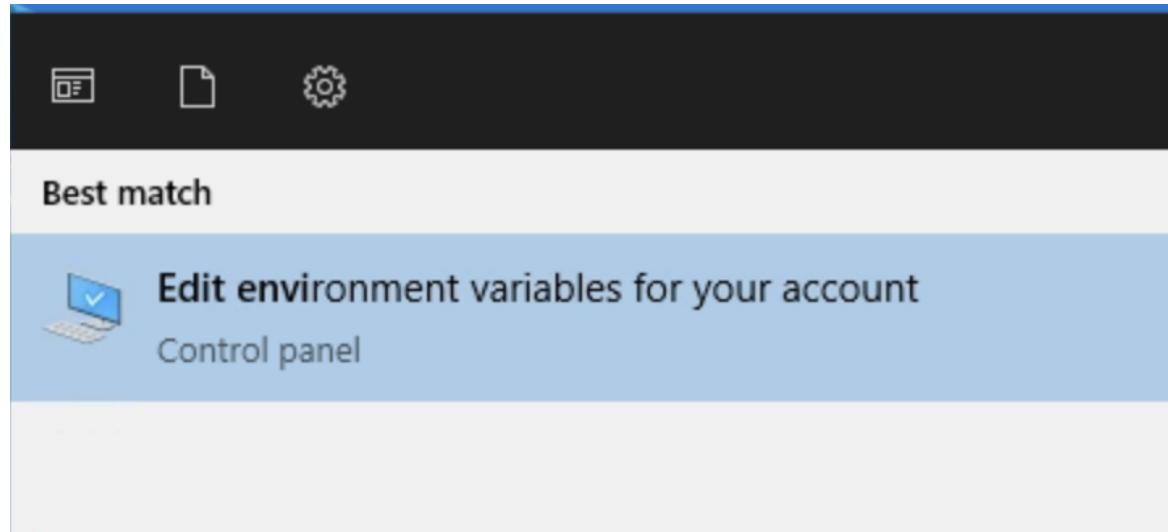
C:\Users\JeffreyBisti>zowe
```

REXX1250812-1458

Se você vir uma resposta do tipo "zowe : command not found", feche a janela do terminal e abra-o novamente. Lembre-se de mudar para o shell/terminal CMD em vez de PowerShell.

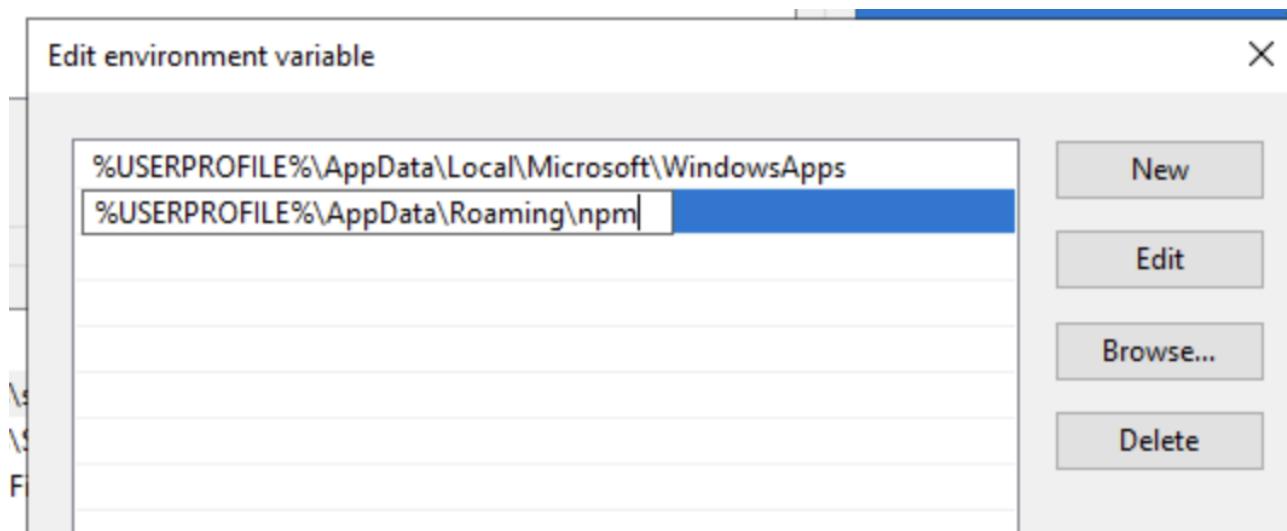
Se *ainda assim* aparecer o erro "command not found" (comando não encontrado), provavelmente será necessário atualizar a variável de ambiente **PATH** para incluir o local do programa **zowe** - geralmente em uma pasta "node\_modules /bin " em **C:\Windows** ou em seu diretório pessoal - geralmente no diretório **.\AppData\Roaming\npm** .

Use a pesquisa do Windows para procurar o aplicativo "Editar variáveis de ambiente para sua conta":



REXX1250812-1658

e adicione uma nova entrada **PATH**, semelhante a:



(Lembre-se de que é necessário verificar o local real do comando `zowe` em seu sistema específico)

Você também pode tentar executar o segundo comando novamente, mas remova a opção "-g":

```
npm i @zowe/cli
```

Faça uma pesquisa na Internet sobre como definir a variável `Windows PATH`, pois ela varia de versão para versão e também pode ser diferente se o laptop tiver sido configurado para perfis de roaming e políticas de grupo.

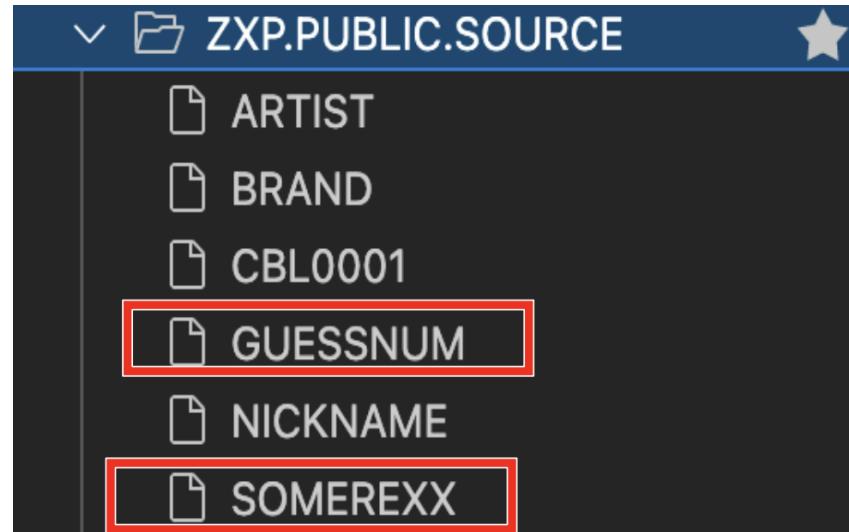
Em geral, se os comandos de código aberto como `zowe` e `ssh` não funcionarem como esperado, tente executar o comando em um terminal CMD, em vez de um terminal `PowerShell`. Se o prompt da janela do terminal em `VSCode` começar com `PS >` ou similar, digite `CMD` para acessar um ambiente de "shell" mais fácil.

Depois de instalar com êxito a CLI do ZOWE (ou chegar a um ponto em que esteja claro que não é possível), reinicie o VSCode.

## 4 SOURCE AND TERMINAL

Comece sua jornada de programação em **REXX** copiando dois membros de **ZXP.PUBLIC.SOURCE** para seu próprio conjunto de dados **SOURCE**.

Especificamente, você está procurando por **SOMEREXX** e **GUESSNUM**



Em seguida, abra uma janela do Terminal, seja em **VSCode** ou diretamente no sistema operacional da estação de trabalho/laptop, como fez no desafio **USS**, mas *NÃO* use **SSH** para se conectar ao mainframe desta vez.

Você deve ser capaz de digitar o comando **zowe** a partir daqui e obter alguma saída útil do programa:

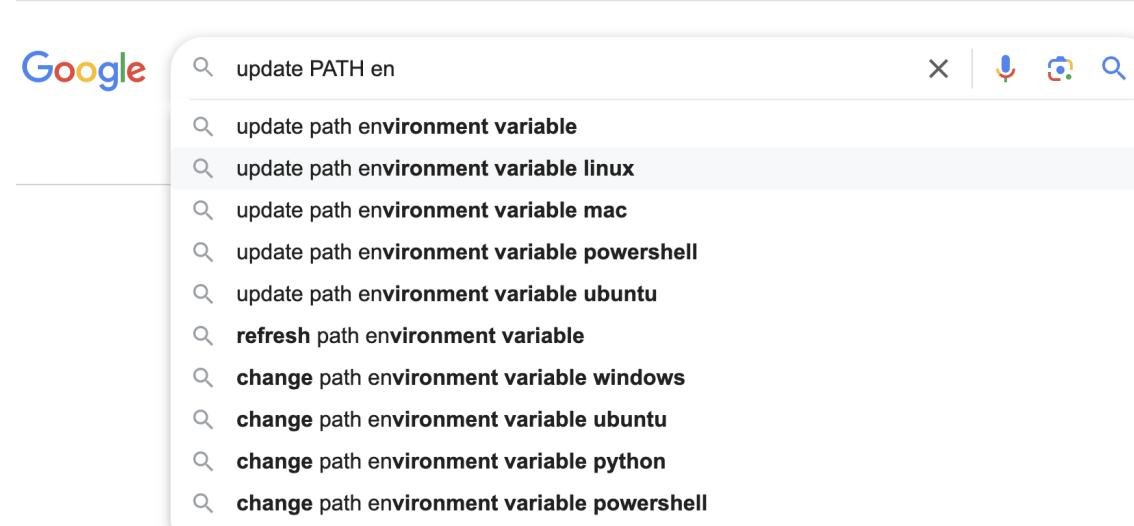
**DESCRIPTION**

-----

Welcome to Zowe CLI!

Zowe CLI is a command line interface (CLI) that provides a simple and streamlined way to interact with IBM z/OS.

Se aparecer uma mensagem como "command not found" (comando não encontrado), você precisará localizar onde o comando `zowe` está instalado e adicionar esse local à sua variável de ambiente `PATH`. O modo como isso é feito depende do sistema operacional que você está usando. Use a pesquisa na Internet para obter instruções que melhor se adaptem ao seu laptop/estação de trabalho:



# 5 CREATE A TSO PROFILE

Até agora, em `VSCode`, você tem usado um perfil `z/OSMF` para fazer conexões com o mainframe

Talvez você também precise criar um perfil TSO `zowe` para emitir comandos `TSO` usando o `Zowe CLI`.

## 5.1 USO DE ARQUIVOS DE CONFIGURAÇÃO DA EQUIPE

Digite o seguinte comando para ver quais perfis já estão definidos:

```
zowe config list
```

Você verá uma impressão que mostra seu perfil `z/OSMF`:

```
zosmfZXplore:  
  type:      zosmf  
  properties:  
    port:      10443  
    host:      204.90.115.200  
    rejectUnauthorized: false  
    user:      (secure value)  
    password:  (secure value)  
  secure:  
    - user  
    - password
```

e o perfil básico `TSO`:

```
tsoZXplore:  
  type:      tso  
  properties:
```

```
account:          FB3
codePage:         1047
logonProcedure:  IZUFPROC
secure:
(empty array)
```

Também deve haver uma seção no final da impressão que mostre os perfis padrão:

```
defaults:
zosmf: zosmfZXPLore
tso:   tsoZXPLore
```

Os nomes dos seus perfis podem ser diferentes - verifique se os nomes na seção `defaults` correspondem aos seus perfis.

# 6 SET DEFAULT PROFILES

**Nesse ponto, você precisa garantir que os perfis corretos sejam definidos como padrão.**

Isso significa que, ao emitir um comando `zowe CLI`, você não precisará fornecer o nome do host, a porta, o ID do usuário, a senha etc. quando o comando for executado, pois todas essas informações estarão disponíveis nos perfis.

## 6.1 USO DE ARQUIVOS DE CONFIGURAÇÃO DA EQUIPE

Seus padrões já devem estar definidos, mas você pode verificar usando :

● `zowe config list defaults`

que deve mostrar algo semelhante a:

```
zosmf: zosmfZXPlore  
tso: tsoZXPlore  
ssh: ssh
```

Se você tiver outros perfis de outras atividades ou conexões do mainframe, às vezes é necessário reiniciar o VSCode para que a troca de perfil tenha efeito.

## 6.2 TESTE SUA CONEXÃO PADRÃO

Um teste rápido para garantir que os perfis padrão estejam funcionando - execute um comando simples de lista de conjuntos de dados:

● `zowe files list ds "ZXP.PUBLIC.*"`

isso deve responder com :

ZXP.PUBLIC.CLIST  
ZXP.PUBLIC.DATA  
ZXP.PUBLIC.DB2.CBL  
ZXP.PUBLIC.DB2.DATA  
ZXP.PUBLIC.DB2.JCL  
ZXP.PUBLIC.DB2.SQL  
ZXP.PUBLIC.EXEC  
ZXP.PUBLIC.GLOBAL.SMPPTS  
ZXP.PUBLIC.INPUT  
ZXP.PUBLIC.JCL  
...

Se, em vez disso, for solicitada qualquer opção ausente (host, porta, número da conta etc.), então seus perfis padrão (ainda) não estão configurados corretamente.

Revise as últimas seções e certifique-se de que todas as configurações necessárias estejam em vigor.

## 7 RUN YOUR FIRST REXX

Digite o seguinte comando:

```
zowe tso issue command "exec 'Zxxxxx.SOURCE(somerexx)" --ssm
```

Você deve receber uma mensagem de saudação do mainframe.



Certifique-se de incluir todas as aspas duplas e simples.

Se receber solicitações de "account", "hostname", "userid" ou "password", é quase certo que você não definiu os perfis padrão corretamente - volte algumas etapas e corrija conforme necessário.

*Esta pode ser a última vez que ressaltamos que, sempre que vir Zxxxxx ou Z99994, você precisa inserir seu próprio Z-userid.*

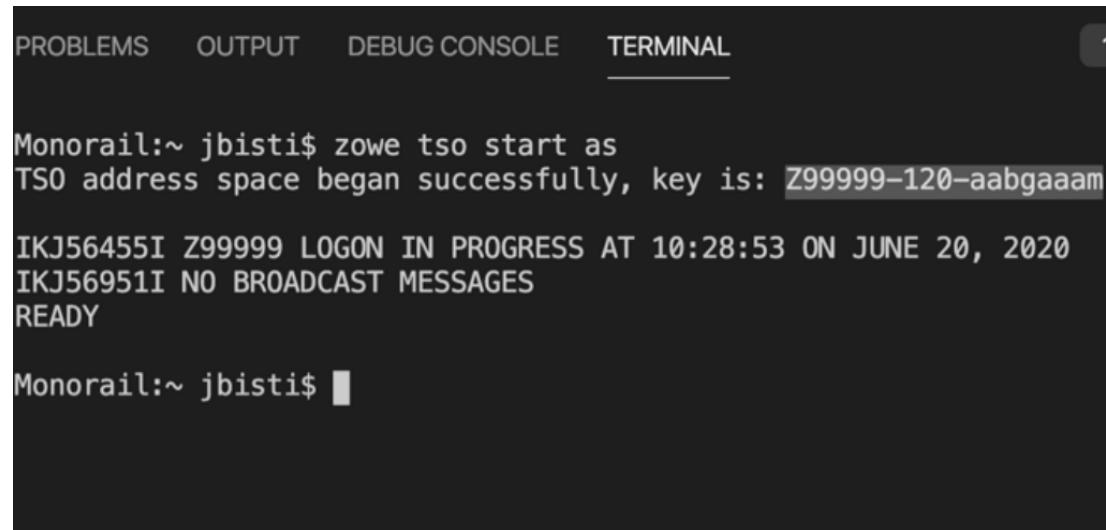
## 8 START AN ADDRESS SPACE

Desta vez, em vez de um único comando, você executará um programa interativo **REXX**; para que isso funcione, você precisará criar um "espaço de endereço" de servidor para iniciar e executar o comando **REXX** até terminar.

Inicie um espaço de endereço com o seguinte comando:

**zowe tso start as**

(aqui **as** é a abreviação de 'address-space')



The screenshot shows a terminal window with tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, and TERMINAL. The TERMINAL tab is selected, displaying the following text:

```
Monorail:~ jbisti$ zowe tso start as
TSO address space began successfully, key is: Z99999-120-aabgaaam
IKJ56455I Z99999 LOGON IN PROGRESS AT 10:28:53 ON JUNE 20, 2020
IKJ56951I NO BROADCAST MESSAGES
READY

Monorail:~ jbisti$ █
```

A small number '1:' is visible in the top right corner of the terminal window.

Isso criará um espaço de endereço para você e informará sua **chave**, que começará com seu ID de usuário (como você pode ver acima).

Você precisará dessa chave para as próximas etapas - neste documento, ela será chamada de "`{my-as-key}`".

Às vezes, depois de não ser usado por um tempo, um espaço de endereço TSO desaparece. Se isso acontecer, basta criar outra usando o mesmo comando **zowe tso start as** - **observe a nova chave que ele retorna**.

Você pode optar por interromper um espaço de endereço com o comando:

```
zowe tso stop as {my-as-key}
```

REXX1|250812:1458

## 9 RUN THE SAME REXX

```
Monorail:~ jbisti$ zowe tso send as Z99999-120-aabgaaam --data "exec 'Z99999.source(somerexx)'"  
G R E E T I N G  
F R O M  
D E V E  
> < > <  
R E V E  
READY  
Monorail:~ jbisti$
```

Execute o mesmo programa **REXX** da etapa 7, mas, desta vez, direcione a entrada para o espaço de endereço (e lembre-se de que você provavelmente pode pressionar a seta para cima para recuperar os comandos anteriores)

```
zowe tso send as {my-as-key} --data "exec 'Zxxxxx.SOURCE(somerexx)'"
```

*Observações :*

1. Esse comando está todo em uma linha
2. {my-as-key} é a chave do espaço de endereço que você acabou de iniciar
3. dessa vez, não use a opção '-ssm'

Você deve receber exatamente a mesma resposta de antes.

A grande diferença aqui é que agora você está emitindo esses comandos para um espaço de endereço TSO semi-persistente, o que fará mais sentido na próxima etapa.

## "TSO? ESPAÇO DE ENDEREÇO?"

TSO (Time Sharing Option) é outra maneira pela qual o z/OS permite que muitos usuários tenham acesso a conjuntos de dados, executem programas e vejam os resultados. Basicamente, é a interface de linha de comando para z/OS (quando você não estiver usando o USS para acessar o lado UNIX das coisas).

Pense em um "espaço de endereço" como um bilhete que permite que você comece a usar a memória do sistema. Um espaço de endereço representa uma enorme quantidade de memória, embora o sistema ainda controle o que fica na memória real do chip e o que é movido (ou paginado) para o disco.

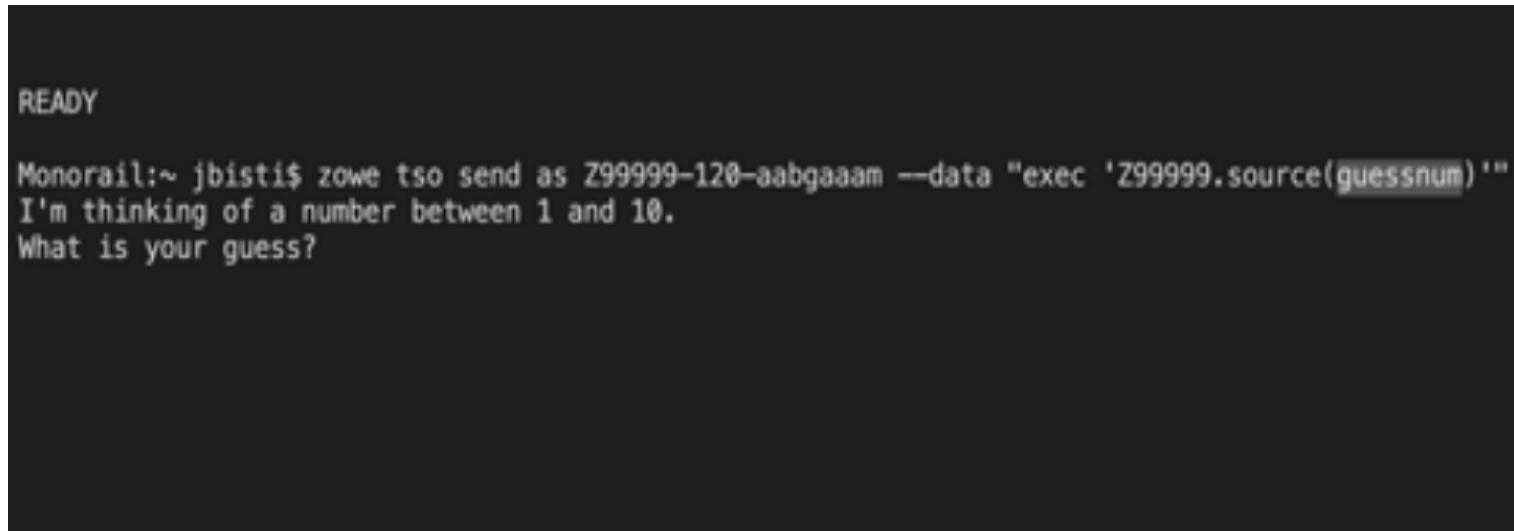
A origem da memória do seu programa depende de sua importância, de como ele deve ser usado e se será compartilhado com outros programas.

Os espaços de endereço são uma parte essencial do z/OS e você deve ler mais sobre eles quando tiver um minuto:

[https://www.ibm.com/support/knowledgecenter/zosbasics/com.ibm.zos.zconcepts/zconcepts\\_82.htm](https://www.ibm.com/support/knowledgecenter/zosbasics/com.ibm.zos.zconcepts/zconcepts_82.htm)

# 10 GUESS WHAT? ANOTHER EXEC

Supondo que o espaço de endereço **TSO** esteja ativo, execute o programa **GUESSNUM** usando o mesmo comando (apenas altere o nome do membro de **SOMEREXX** para **GUESSNUM**).



```
READY
Monorail:~ jbisti$ zowe tso send as Z99999-120-aabgaaam --data "exec 'Z99999.source(guessnum)'" I'm thinking of a number between 1 and 10.
What is your guess?
```

Esse é um programa que gera um número aleatório entre 1 e 10, e você precisa adivinhar esse número.

Pode haver outros jogos que você preferiria estar jogando agora, mas nem todos eles o ensinarão sobre **REXX** e **TSO**, portanto, tenha isso em mente se estiver olhando para o seu **XBOX**!

# 11 SEE HOW CUTE THE CODE IS

Se ainda não tiver feito isso, abra o código desse programa em seu editor [VSCode](#).

Não se trata de um programa complicado de forma alguma, mas você deve estar percebendo como esse código [REXX](#) é realmente simples.

16 linhas de código, incluindo um comentário e uma linha em branco; e sim, esses comandos "say" e "pull" realmente fazem o que você pensa que fazem.

Você pode ver por que as pessoas adoram esse idioma.

Ele também tem o que considero o melhor logotipo do mundo para uma linguagem de programação.

Dê uma olhada nisso. Simplesmente. Veja. Em. Ele.

REXX1|250812-1458



## 12 LIVE CHAT WITH REXX

Agora você pode enviar seus palpites para o programa substituindo tudo entre aspas duplas por um número.

```
Monorail:~ jbisti$ zowe tso send as Z99999-120-aabgaaam --data "exec 'Z99999.source(guessnum)'"  
I'm thinking of a number between 1 and 10.  
What is your guess?  
  
Monorail:~ jbisti$ zowe tso send as Z99999-120-aabgaaam --data "1"  
That's not it. Try again  
What is your guess?  
  
Monorail:~ jbisti$ zowe tso send as Z99999-120-aabgaaam --data "2"  
That's not it. Try again  
What is your guess?  
  
Monorail:~ jbisti$ zowe tso send as Z99999-120-aabgaaam --data "3"  
That's not it. Try again  
What is your guess?  
  
Monorail:~ jbisti$ zowe tso send as Z99999-120-aabgaaam --data "4"  
You got it! And it only took you 4 tries!  
READY
```

Você pode ver que os comandos usam a chave de espaço de endereço para garantir que sua entrada continue indo para o espaço de endereço TSO correto, que está lá esperando a próxima entrada.

Se você acertar na primeira tentativa, parabéns!

Sinta-se à vontade para iniciar o programa novamente e certifique-se de que você pode vê-lo passar pelas etapas de "Try again".

Agora envie seu cheque de conclusão - [CHKREXX1](#) de [ZXP.PUBLIC.JCL](#)

Bom trabalho - vamos recapitular	A seguir.
<p>Agora você está entrando no ritmo das coisas. Você está interagindo com um programa <b>Rexx</b>, executado em um espaço de endereço <b>TSO</b>, e está fazendo isso por meio do comando <b>zowe</b>.</p> <p>Provavelmente, você também aprendeu um pouco sobre a linguagem Rexx e pode até ter lido um pouco mais sobre Address Spaces. Tudo aqui o ajudará a se tornar um profissional de mainframe mais qualificado.</p>	<p>O ferro está definitivamente queimando e você provavelmente está se tornando um fã do Rexx. Por enquanto, porém, vamos continuar com o restante dos desafios em Fundamentos.</p>