**Treinamento TEX – JavaScript DevOps**

User máquina virtual

Fernandorc

Senha

bizo@00tk

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

1ª. Aula

Roque

Renan

Rita

Nicolas

Elton

75% das aulas ao vivo / online assistidas

360 horas no total – e no final tem uma entrega de projeto.

Encontros no Discord, checar o acesso.

Dedicação de 4 horas por día, de segunda a sábado.

Instrutor não estará no Discord.

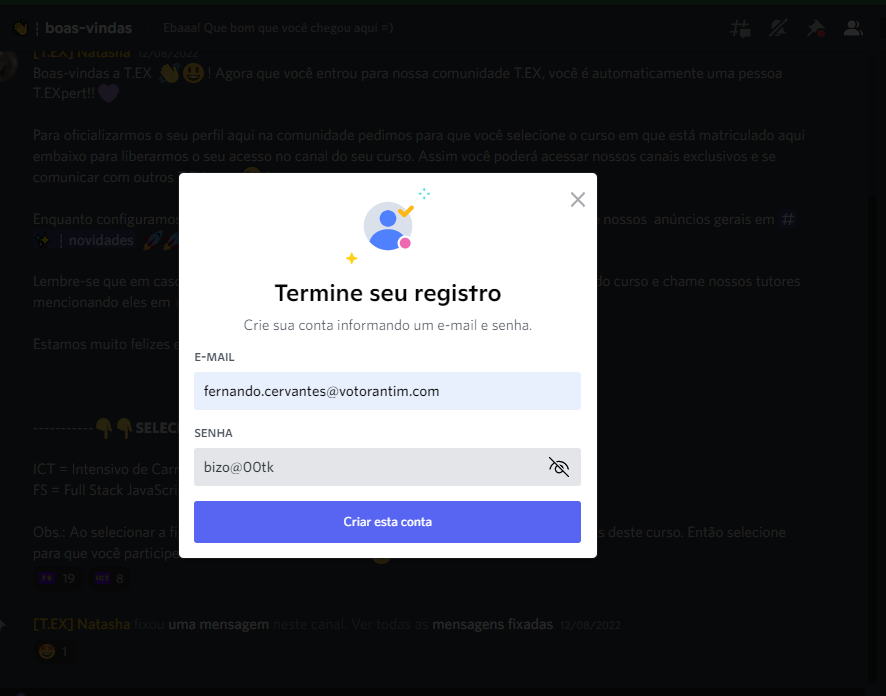
Teremos um recesso no Natal

Tutores das 09:00 as 18:00

Discord –

Siet: https://discord.com/app

Invite <https://discord.gg/tnzyy9dCt8>



Diagrama

Descrição gerada automaticamente com confiança média

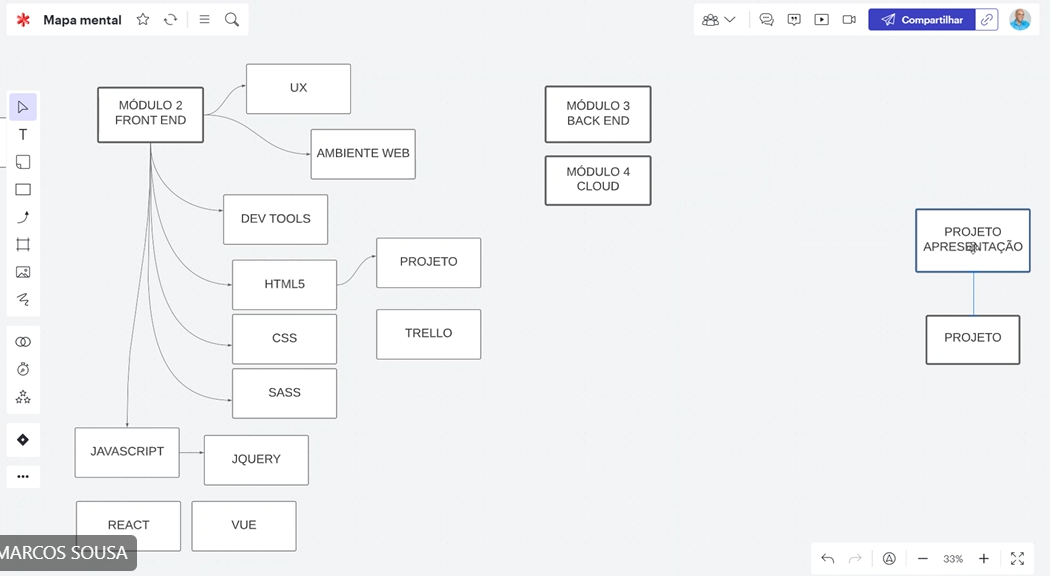
Diagrama

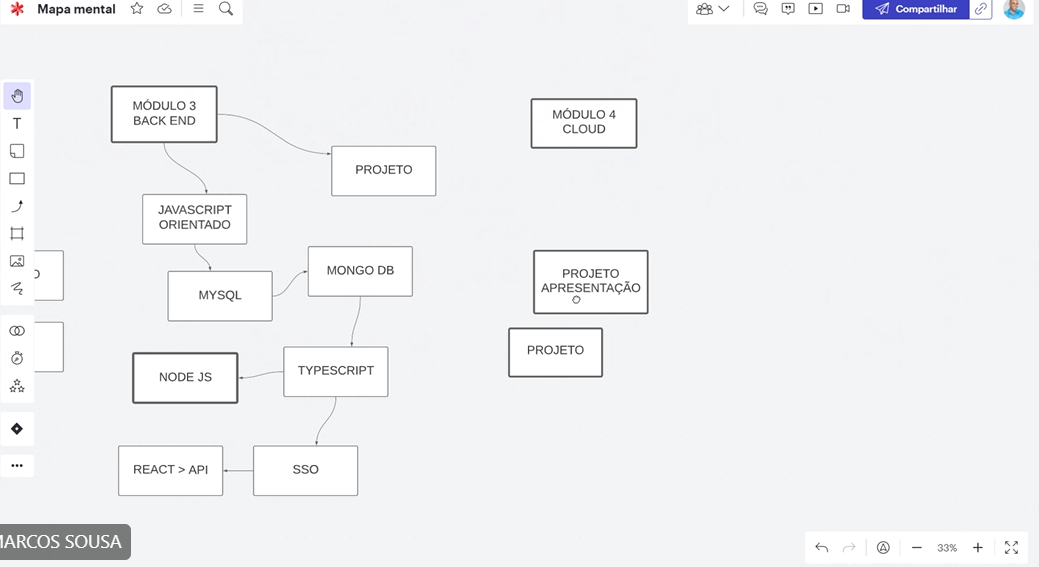
Descrição gerada automaticamente

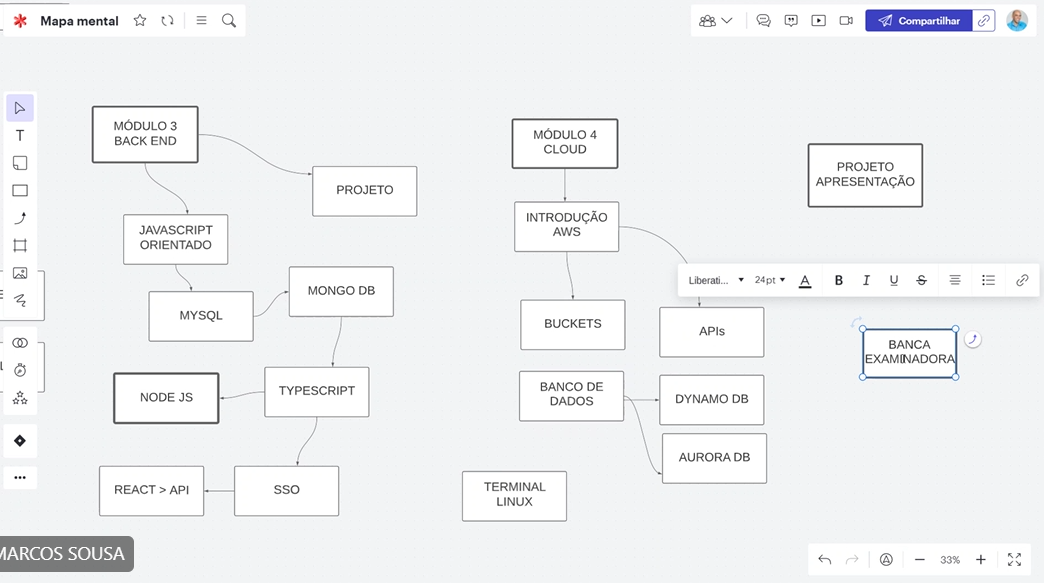
Diagrama

Descrição gerada automaticamente

LucidChart – Mapas Mentais



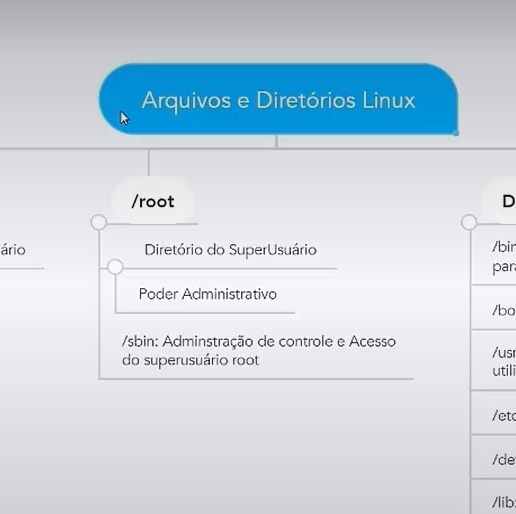




Diretórios Linux

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente



**Comandos Terminal Linux**

fernandorc@fernandorc-VirtualBox:~/Documentos$ echo {1..15}

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

fernandorc@fernandorc-VirtualBox:~/Documentos$ echo ${USER}

fernandorc

fernandorc@fernandorc-VirtualBox:~/Documentos$ echo -e "\nBem vindo ao treinamento Linux" >> leiame.txt

fernandorc@fernandorc-VirtualBox:~/Documentos$ echo -e "Hello World \\ \"Developers\" \nLinux"

Hello World \ "Developers"

Linux

fernandorc@fernandorc-VirtualBox:~/Documentos$ echo -e "\033[0;31mLinux - Fernando R Cervantes"

Linux - Fernando R Cervantes

fernandorc@fernandorc-VirtualBox:~/Documentos$ echo -e "\033[0;32mLinux - Fernando R Cervantes"

Linux - Fernando R Cervantes

fernandorc@fernandorc-VirtualBox:~/Documentos$ echo -e "\033[0;33mLinux - Fernando R Cervantes"

Linux - Fernando R Cervantes

fernandorc@fernandorc-VirtualBox:~/Documentos$

Interface gráfica do usuário, Texto

Descrição gerada automaticamente

fernandorc@fernandorc-VirtualBox:~/Documentos$ echo [1+2]

[1+2]

fernandorc@fernandorc-VirtualBox:~/Documentos$ echo {1+2}

{1+2}

fernandorc@fernandorc-VirtualBox:~/Documentos$ x=10

fernandorc@fernandorc-VirtualBox:~/Documentos$ echo $x

10

fernandorc@fernandorc-VirtualBox:~/Documentos$ y=30

fernandorc@fernandorc-VirtualBox:~/Documentos$ echo $y

30

fernandorc@fernandorc-VirtualBox:~/Documentos$ echo $((x+y))

40

fernandorc@fernandorc-VirtualBox:~/Documentos$ z=$(($x\*$y))

fernandorc@fernandorc-VirtualBox:~/Documentos$ echo $z

300

fernandorc@fernandorc-VirtualBox:~/Documentos$ echo $((RANDOM%100+1))

55

fernandorc@fernandorc-VirtualBox:~/Documentos$ echo $((RANDOM%100+1))

52

fernandorc@fernandorc-VirtualBox:~/Documentos$ echo $((RANDOM%100+1))

52

fernandorc@fernandorc-VirtualBox:~/Documentos$ echo $((RANDOM%1000+1))

665

fernandorc@fernandorc-VirtualBox:~/Documentos$ echo $((RANDOM%1000+1))

160

fernandorc@fernandorc-VirtualBox:~/Documentos$ echo $x

10

fernandorc@fernandorc-VirtualBox:~/Documentos$ echo $y

30

fernandorc@fernandorc-VirtualBox:~/Documentos$ echo $(($x > $y))

0

fernandorc@fernandorc-VirtualBox:~/Documentos$ echo $(($x < $y))

1

Texto

Descrição gerada automaticamente

sort -r / sort -R / sort -n – checar sort –help

tail -5 numeros.txt | sort -R

Checar uso do pipe

Interface gráfica do usuário, Texto, Aplicativo, Email

Descrição gerada automaticamente

sudo apt update

sudo apt upgrade

**Resumo comandos**

**Comandos do terminal Linux:**

* whoami: exibe o usuário em atividade.
* clear: limpa o conteúdo da tela do terminal.
* Copiar/Colar: ctrl+shift+c/ctrl+shift+v.
* Para navegar entre os comandos recentes, utilize as setas do teclado para cima e para baixo.
* history: exibe o histórico de comandos executados no terminal.
* exit: encerra o terminal.
* touch: possibilita gravar um arquivo em um diretório.
* echo: permite a inserção de textos no terminal e em arquivos.
* zip | unzip: compacta e descompacta arquivos **“.zip”**.
* shutdown: solicita o desligamento da máquina.
* sort: consulta um arquivo com sequências numéricas ou não.
* pwd: exibe a referência do diretório.
* cd: usado para navegação entre os diretórios.
* alias: cria uma variável para disparo de comandos.
* ls: listar arquivos de diretório.
* tail: exibe as linhas finais de um arquivo.
* wc: contador de linhas, palavras e caracteres de um arquivo.
* bc: processador aritmético e lógico.
* whatis: descreve o objetivo do comando.
* mkdir: capaz de criar um diretório.
* rm: remove arquivos ou diretórios.
* apt update: baixa pacotes de fontes configuradas.
* apt upgrade: atualiza os pacotes do sistema (/etc/apt/sources.list).
* O comando **cat** é capaz de mostrar o que há dentro do arquivo.
* É possível editar o arquivo através do comando **nano**.
* Através dos comandos **cp** e **mv** é possível copiar e colar diretórios e arquivos.
* O comando **mv** também é capaz de renomear arquivos.
* O comando **find** pode ser utilizado para encontrar arquivos, ainda possui capacidade de filtrar arquivos específicos.

Texto

Descrição gerada automaticamente

**Principais pontos da aula:**

* O Ubuntu é um sistema operacional multiusuário, no qual cada usuário pode possuir permissões diferentes.
* A hierarquia de permissões vai de 0 a 7, sendo 0 a menor permissão, e 7 a maior delas.
* No Linux, as letras R,W,X, referem-se, nesta ordem, às permissões de ler, escrever e executar (read, write e execute).

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

UGO – user, group e others

Commando CHMOD

CHMOD (+/-) (U / G / O) mais permissão (RWX) para ajustar as permissões de leitura, gravação e execução

**Principais pontos da aula:**

* O comando **“sudo”** é um comando de acesso temporário.
* O usuário **root** possui privilégios ilimitados, diferente do usuário criado durante o setup da máquina.
* É possível intercalar entre usuários durante a utilização do sistema operacional.
* Com o comando **“id nG usuário”** é possível verificar a quais grupos o usuário pertence.
* Com o comando **“useradd -m usuário -s /bin/bash”** é possível adicionar um novo usuário ao sistema.
* Com o comando **“deluser usuário –remove-home”** é possível apagar um usuário.

sudo -i > muda para admin

getent group

id -nG fernandorc

sudo usermod -aG sudo profroque >> mudando o grupo do user profroque, incluindo o mesmo no grupo “sudo”

Texto

Descrição gerada automaticamente

fernandorc@fernandorc-VirtualBox:~$ sudo useradd -m admin -s /bin/bash

fernandorc@fernandorc-VirtualBox:~$ sudo passwd admin

fernandorc@fernandorc-VirtualBox:~$

Remove o user e o diretório home

sudo deluser profroque --remove-home

Cria um grupo novo

sudo groupadd devs

Inclui admin no novo grupo

$ sudo usermod -aG devs admin

Texto

Descrição gerada automaticamente

id – nG

**Principais pontos da aula:**

* Dentro do Shell Script é possível executar comandos do terminal e realizar ainda a leitura de dados e a utilização desses dados dentro do próprio script.

2ª Aula

Pilar do SCRUM – transparência

[50+ Basic Ubuntu Commands Every Beginners Should Know (ubuntubeginner.com)](https://www.ubuntubeginner.com/basic-ubuntu-commands-for-beginners/)

Para a próxima aula assistir até o vídeo

- acessar o Discord para confirmar

- assistir até operadores aritméticos

1.2 Lógica de Programação

**Principais pontos da aula:**

* É necessário compreender a diferença entre o pensamento comum analítico e o pensamento lógico, orientado à resolução de problemas.
* Algoritmos são sequencias de passos necessárias para atingir algum resultado específico.
* Se você adquirir bons conhecimentos de lógica de programação, poderá se aprofundar em qualquer linguagem que desejar.

Aplicativo

Descrição gerada automaticamente com confiança baixa

Interface gráfica do usuário

Descrição gerada automaticamente com confiança média

Passar Café

1 - Tem água na chaleira?

2 - Se não, colocar água na chaleira

3 - Se sim colocar chaleira no fogão

4 – Água na chaleira está quente?

5 - Se não, aguardar água ferver

6 - Se sim, tem filtro na jarra?

7 - Se não, colocar filtro na jarra

8 - Se sim, tem café no filtro?

9 - Se não, colocar café no filtro

10 - Se sim, jogar água quente da chaleira no filtro

Texto, Aplicativo

Descrição gerada automaticamente

Pegar os ovos

5 - Partir os ovos e colocar em um recipiente

6 - Colocar uma pitada de sal

4 - Bater os ovos

1 – Colocar os ovos batidos numa frigideira quente

3 – Fritar e dobrar

2 - Servir

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

1 – ir até o caixa eletrônico

2 – verificar se caixa está ativo

3 – se sim, verificar se caixa tem recursos (dinheiro)

4 – Se não, procurar outro caixa

5 – Se sim, introduzir cartão

6 – se cartão não for aceito, procurar outro caixa

7 – Se cartão aceito, entrar com a senha / biometria

8 – Se senha / biometria não aceita, procurar outra caixa ou se já fez isso dirigir-se à um atendente humano do banco

9 – Se senha / biometria for aceita, informa valor

10 – Se valor não disponível, procurar outro caixa

11 – Se valor disponível efetuar a transação, contar as cédulas

12 – Se cédulas não igual valor solicitado, dirigir-se à um atendente humano do banco

13 – Se cédulas = valor solicitado, retirar cartão e encerrar operação.

Declaração de variáveis

let <var> = “alguma coisa”

var <var> = “outra coisa>

const <constante> = “x”

let x,y,z,w,k,soma,resultado,total,ac

x = 10

y = 5

z = false

w = 12

k = 13

soma = x+w

ac = 0

resultado = x/y //2

ac = ac + resultado

resultado = x\*y //50

ac = ac + resultado

total = 2 + 5 \* 3 + 8 //25

total2 = (2 + 5) \* (3 + 8) //77

// Acumulador //

console.log(“x igual y : “ + (x == Y)) // false

console.log(“x maior ou igual y : “ ${(x >= Y)}”) // true

Texto

Descrição gerada automaticamente

console.log(y <= x) // true

console.log(x >= y && z ) // false && AND

console.log(x >= y || z ) // true || OR

console.log(‘soma: ${soma}’) // 22

console.log(‘resultado: ${resultado}’) // 25

console.log(‘total: ${total}’) // 77

console.log(`acumulador do resultado: ${ac}`)

console.log(`incremento de x: ${x++}`) //10

console.log(`incremento de x: ${x}`) //11

console.log(`decremento de y: ${--y}`) //4

console.log(`decremento de y: ${--y}`) //4 alternativa para y=y+1 ou y=y-1

console.log(x % 2) // modulo 2 – resto da divisão de x por 2

console.log(k % 2) // modulo 2 – resto da divisão de k por 2

Texto

Descrição gerada automaticamente

Aula de Logica: condicionais

Referências

Livro - Refatoração de martin fowler