

***Universidad Tecnológica Nacional - Facultad Regional Buenos Aires***

**Sintaxis y Semántica de los Lenguajes**

**2021**

**Trabajo Práctico Integrador N° 2**

**Tema:** Relevamiento

**Curso:** K2054

**Profesor:**Pablo Damian Mendez

Usuario: brian-colman

**Fecha de Estipulada 1° Entrega:** 06/06/2021

|  |  |
| --- | --- |
| **Alumno** | **Legajo** |
| Colman, Brian | 1634689 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Fecha** | **Nota** | **Observaciones** | **Fecha Devol.** | **Firma Docente** |
| **Entrega** |  |  |  |  |  |
| **Corrección 1** |  |  |  |  |  |
| **Corrección2** |  |  |  |  |  |

Ejercicios:

1. Reemplace cada punto del archivo “breve\_historia.txt” por punto y salto de línea generando un nuevo archivo

sed -E 's/\.| \. /&\n/g' breve\_historia.txt > breve\_historia1.txt

1. Borre todas las líneas en blanco

sed -r '/^\s\*$/d' breve\_historia.txt

1. Cree un nuevo archivo: “breve\_historia\_2.txt” con el resultado de las operaciones a y b (redireccionamiento de la salida estándar).

sed -E 's/\.| \. /&\n/g' breve\_historia.txt | sed -r '/^\s\*$/d' > breve\_historia\_2.txt

1. Del archivo “breve\_historia.txt”, liste todas las oraciones que contengan la palabra “guerra” sin distinguir mayúsculas y minúsculas

echo $(sed -E 's/\.| \. /&\n/g' breve\_historia.txt | grep -n -i '.\bguerra\b.')

1. Muestre las líneas que empiecen con “A” y terminen con “s” o “s.” del archivo “breve\_historia.txt”

echo $(sed -E 's/\.| \. /&\n/g' breve\_historia.txt | grep -n -i -E '^A.\*(s|s\.)$')

1. Sobre el mismo archivo del punto anterior, Indique en cuántas oraciones aparece la palabra “peronismo”. Puede usar la opción -c para contar

sed -E 's/\.| \. /&\n/g' breve\_historia.txt | grep -i -c '\bperonismo\b'

1. Muestre la cantidad de oraciones que tienen la palabra “Sarmiento” y “Rosas”.

echo $(sed -E 's/\.| \. /&\n/g' breve\_historia.txt | grep -n -E '\bRosas\b|\bSarmiento\b')

H) Muestre las oraciones que tengan fechas referidas al siglo XIX

echo $(sed -E 's/\.| \. /&\n/g' breve\_historia.txt | grep -n -E '18[0-9][1-9]')

1. Borre la primera palabra de cada línea. Utilice substitución con sed. La sintaxis para sustituir la primera palabra de cada línea por “nada” sería: $sed “s/^[[a-zA-Z]]\*\b//g” nombre\_archivo (La “s” indica sustitución; entre los dos primeros /.../ está la expresión regular que queremos reemplazar, en este caso “/^[[a-zA-Z]]\*\b”; entre el segundo y el tercer “//” se indica la expresión por la cual será reemplazada, en este caso por la palabra vacía. Finalmente la “g” indica que el cambio será en todo el archivo

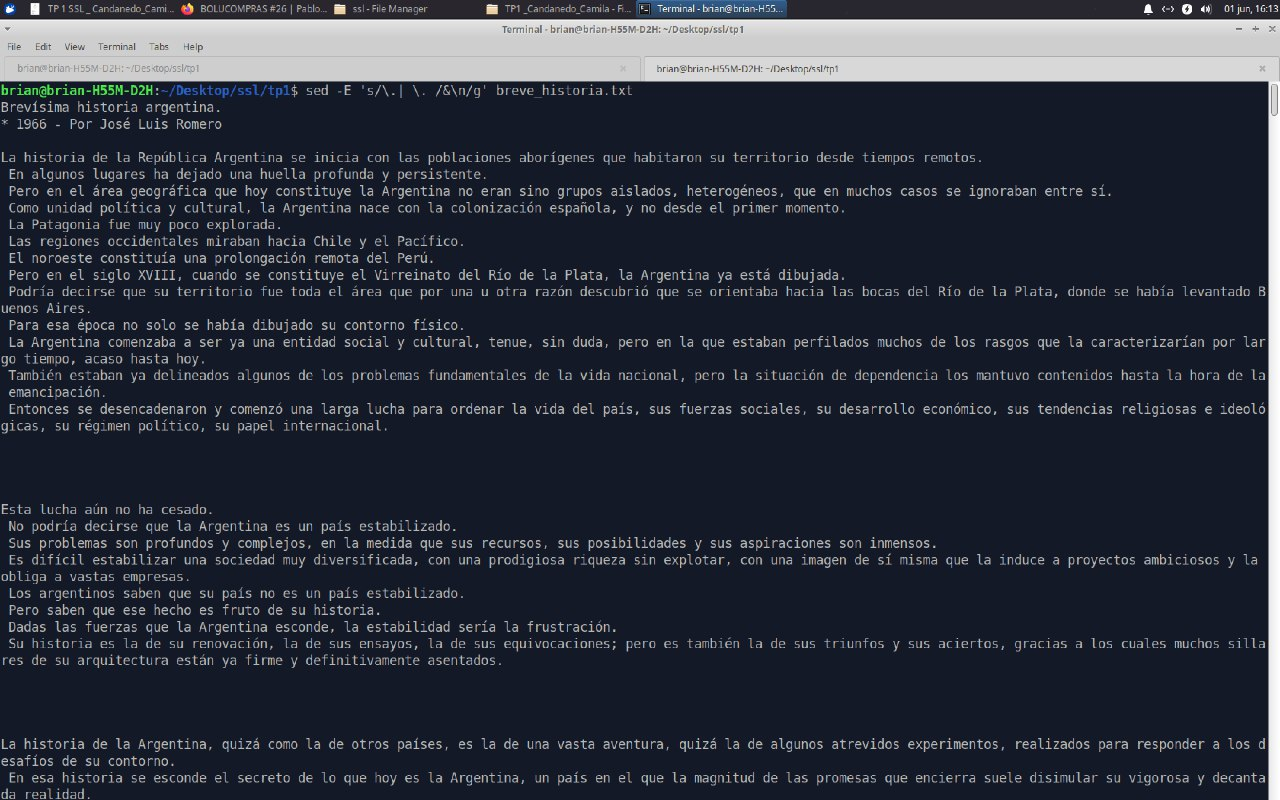
sed 's/^[a-zA-Z]\*\b//g' breve\_historia.txt

1. Enumere todos los archivos de una carpeta que contengan extensión “.txt”. (tip: pipe con el comando ls).

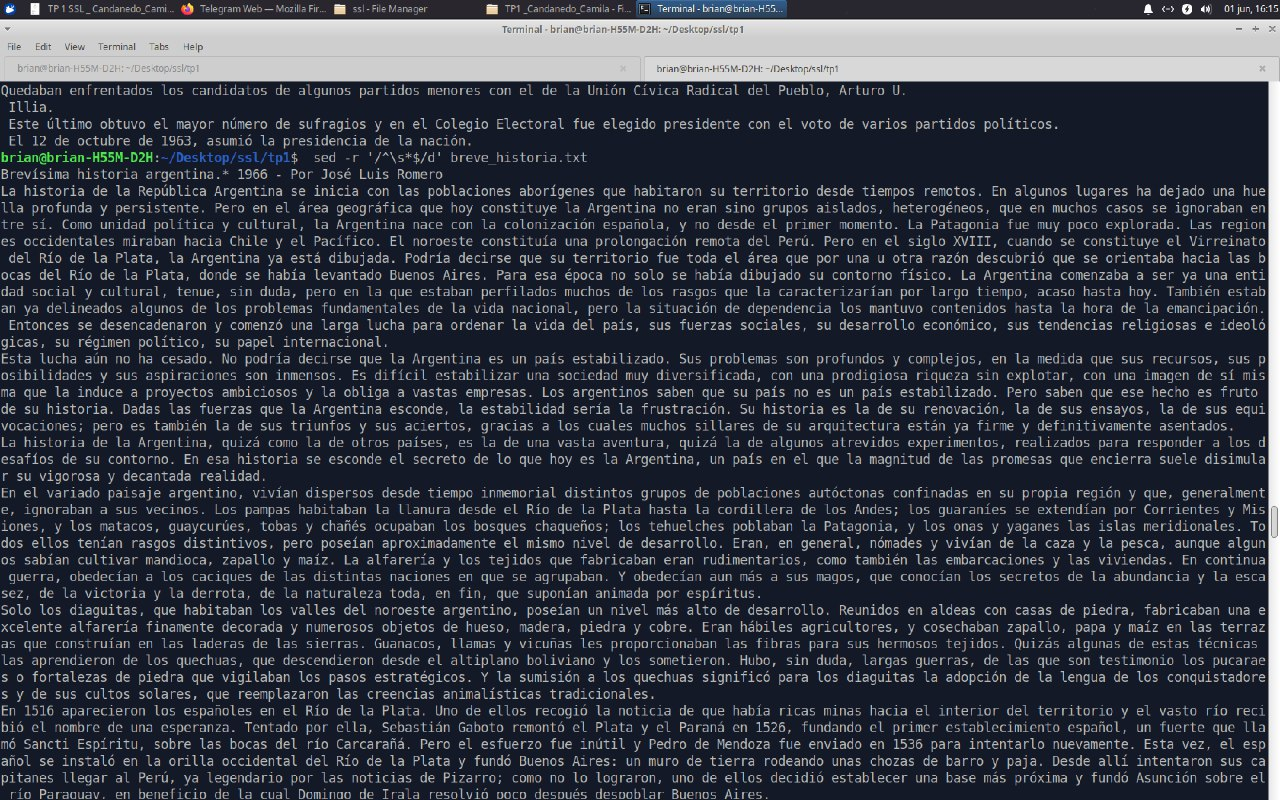
ls \*.txt

Respuestas punto 2 :

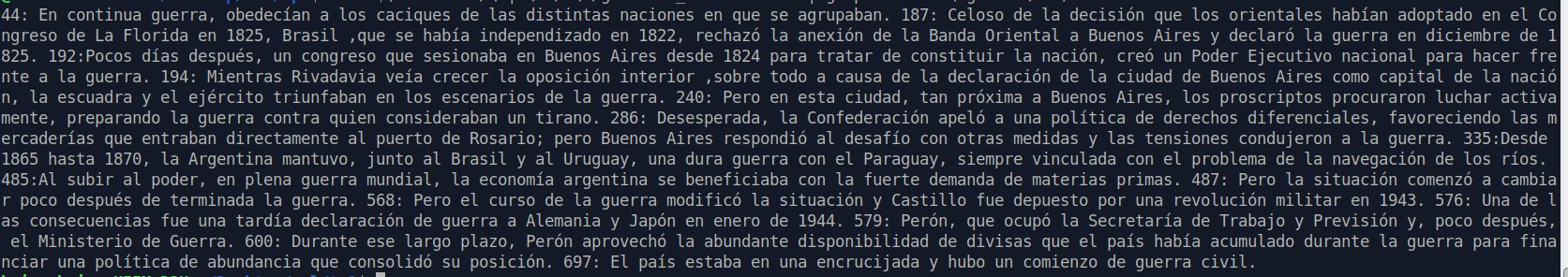
a)

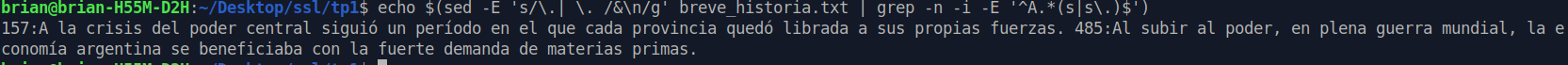


B)



C)

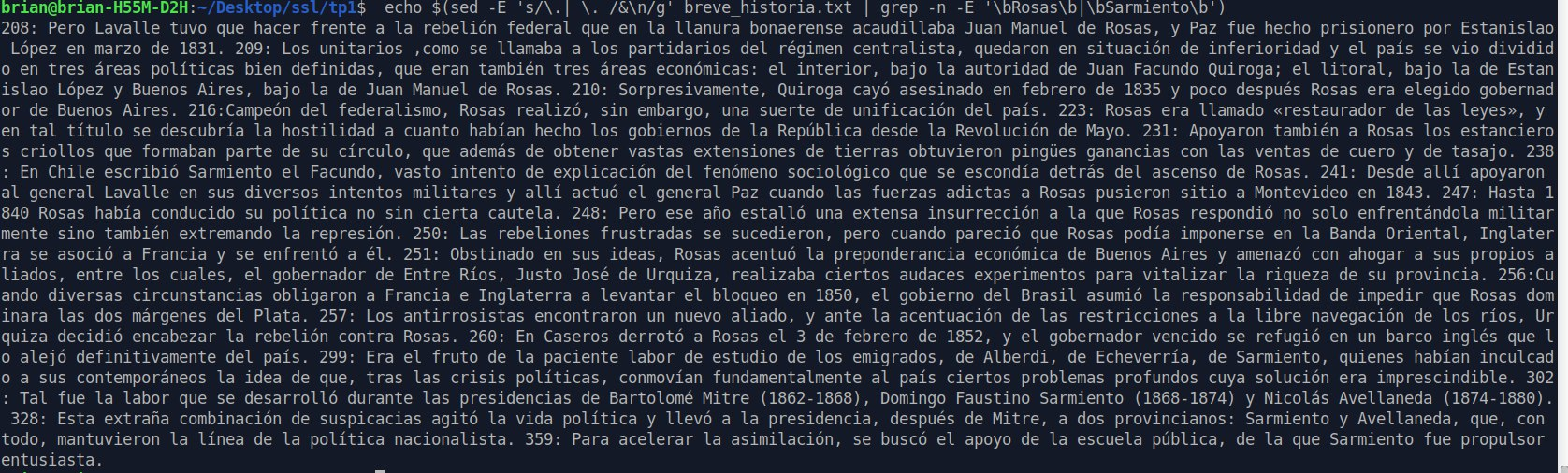


E) 

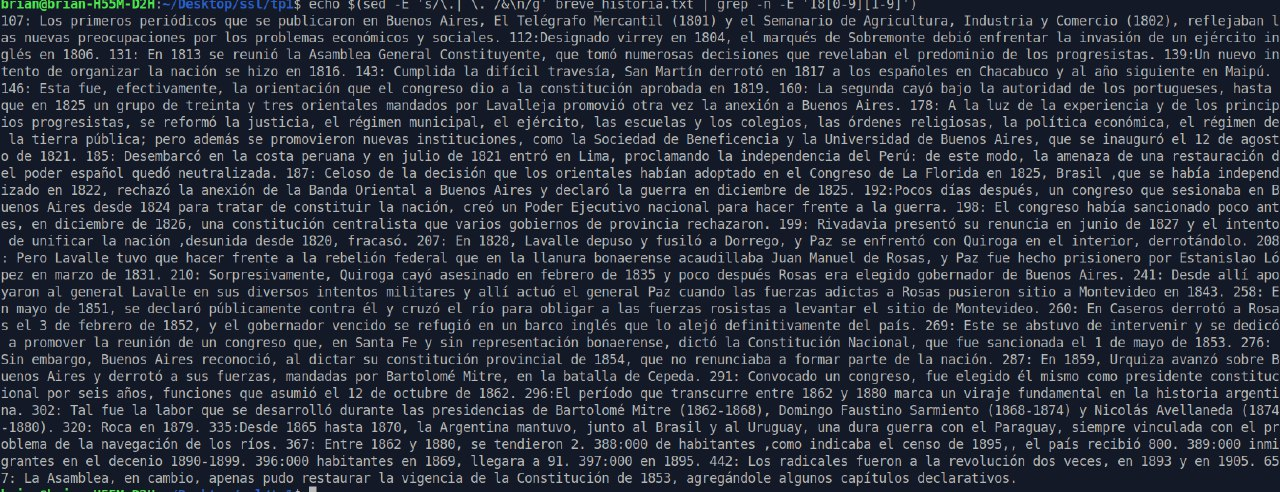
F )

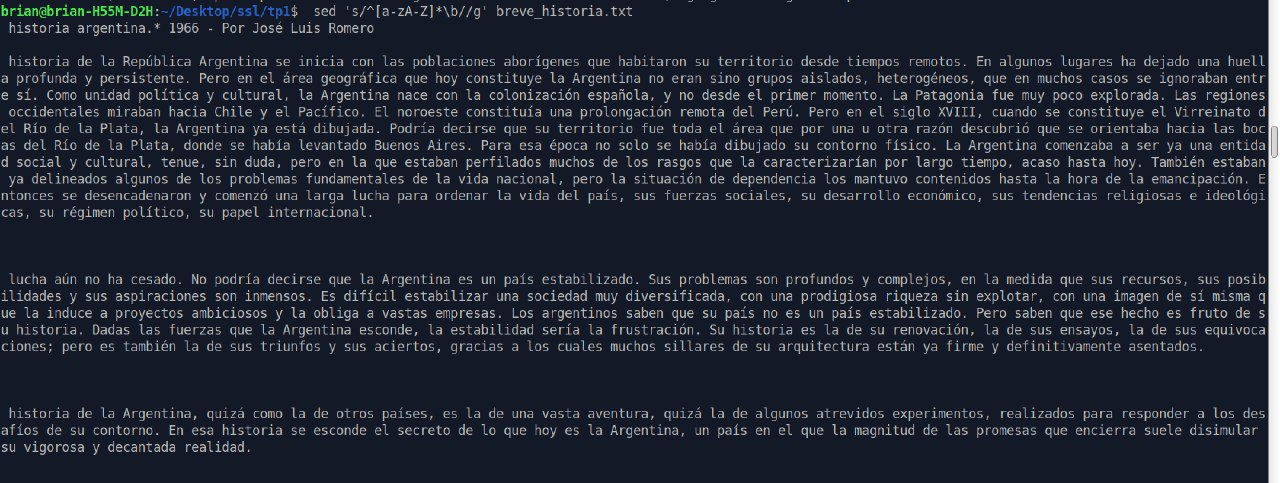


G)

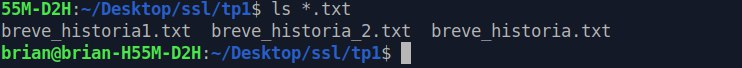


H )



I ) 

J)



Ejercicio 3 :

# Variables

# Se usan con letras mayusculas

# La igualdad siempre va pegado a la variable

````

MENSAJE="Hola Mundo"

echo $MENSAJE

MENSAJE="\"Hola Mundo ...\""

echo "El valor de la variable \$MENSAJE es $MENSAJE"

# Variables numericas

let A=100

echo "A: $A"

#-----------------------------------------

# Sentencias Condicionales

# con $1 se piden parametros en consola ./nombre param

# -eq equal (igual)

# -ne not equal (distinto)

# -gt greater than (mas grande que)

# -ge greater or equal (>=)

# -lt less than (<)

# -le less or equal (<=)

if [[ $1 -eq 1 ]]; then

echo "1 was passed in the first parameter"

elif [[ $1 -gt 2 ]]; then

echo "2 was not passed in the first parameter"

else

echo "The first parameter was not 1 and is not more than 2."

fi

#-----------------------------------------

# Bucles for

# Muestra todos los elementos uno por uno con el comando ls

for i in $(ls); do

echo $i

done

# Como C

for ((i = 0 ; i < 100 ; i++)); do

echo $i

done

# Con rangos

for i in {1..5};do

echo "hola jugador $i"

done

#-----------------------------------------

# While

# Estructura

# while [condition];do

# commands

# done

number=10

while [ $number -gt 5 ];do

echo $number

number=$(($number-1))

done

#-----------------------------------------

# Funciones

# [function]NombreFunción ()

# {Bloque ... [ return [Valor] ] ... }

myfunc() {

echo "Using functions"

}

myfun

# Con return

myfunc2() {

read -p "Enter a value: " value

echo "adding value"

return $(($value + 10))

}

myfunc2

echo "The new value is $?"

myfunc3() {

read -p "Enter a value: " value

echo $(($value + 10))

}

result=$(myfunc3)

echo "The value is $result"