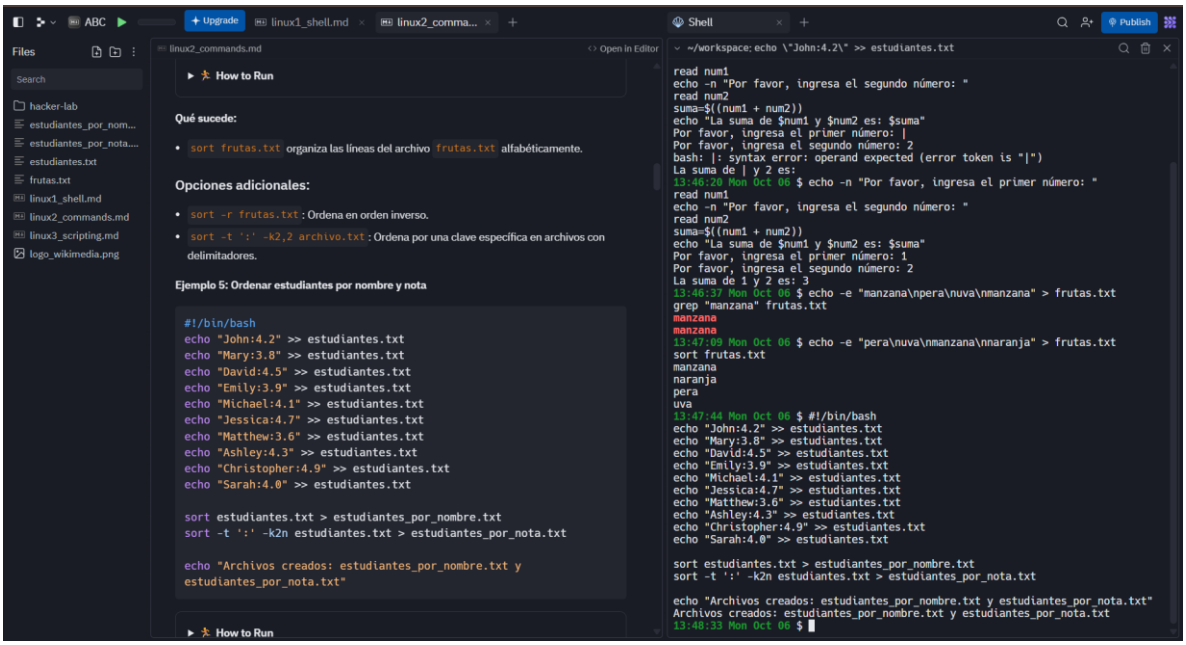
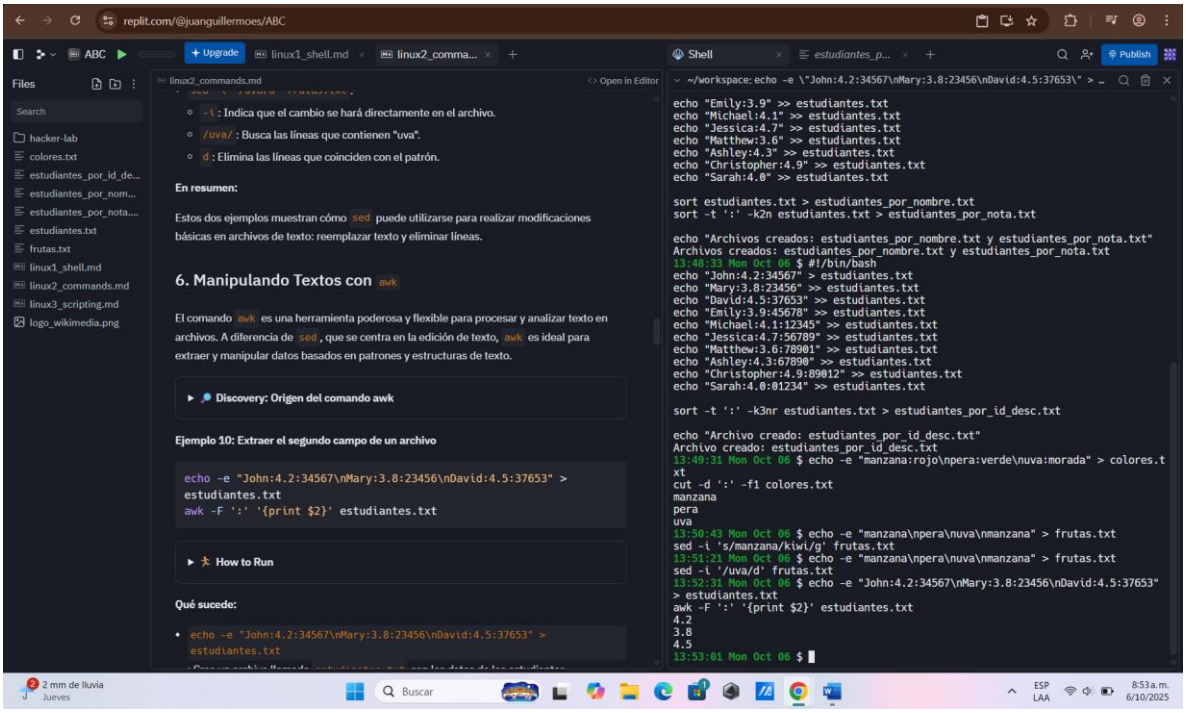
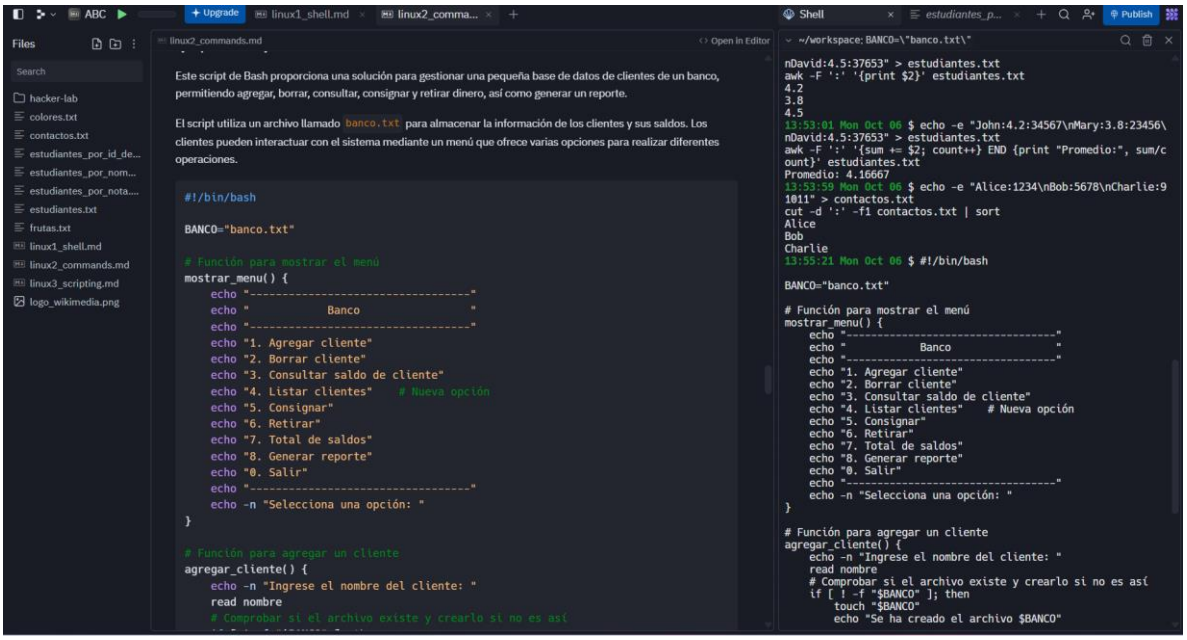
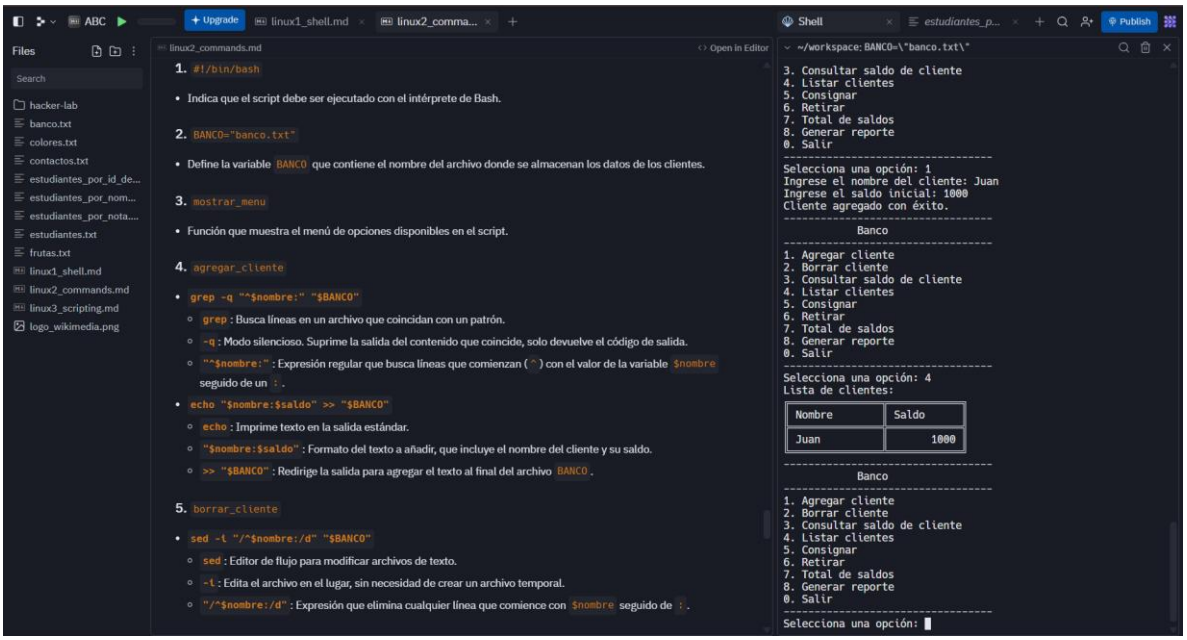

ABC
Upgrade
linux1_shell.md
linux2_comma...
Files
Search
hacker-lab
estudiantes_por_nom...
estudiantes_por_nota....
estudiantes.txt
frutas.txt
linux1_shell.md
linux2_commands.md
linux3_scripting.md
logo_wikimedia.png
linux2_commands.md
Open in Editor
How to Run
Qué sucede:
sort frutas.txt organiza las líneas del archivo frutas.txt alfabéticamente.
Opciones adicionales:
sort -r frutas.txt : Ordena en orden inverso.
sort -t ':' -k2,2 archivo.txt : Ordena por una clave específica en archivos con delimitadores.
Ejemplo 5: Ordenar estudiantes por nombre y nota
#!/bin/bash
echo "John:4.2" >> estudiantes.txt
echo "Mary:3.8" >> estudiantes.txt
echo "David:4.5" >> estudiantes.txt
echo "Emily:3.9" >> estudiantes.txt
echo "Michael:4.1" >> estudiantes.txt
echo "Jessica:4.7" >> estudiantes.txt
echo "Matthew:3.6" >> estudiantes.txt
echo "Ashley:4.3" >> estudiantes.txt
echo "Christopher:4.9" >> estudiantes.txt
echo "Sarah:4.0" >> estudiantes.txt
sort estudiantes.txt > estudiantes_por_nombre.txt
sort -t ':' -k2n estudiantes.txt > estudiantes_por_nota.txt
echo "Archivos creados: estudiantes_por_nombre.txt y estudiantes_por_nota.txt"
How to Run
Shell
Publish
~/workspace: echo \"John:4.2\" >> estudiantes.txt
read num1
echo -n "Por favor, ingresa el segundo número: "
read num2
suma=$((num1 + num2))
echo "La suma de $num1 y $num2 es: $suma"
Por favor, ingresa el primer número: |
Por favor, ingresa el segundo número: 2
bash: |: syntax error: operand expected (error token is "|")
La suma de | y 2 es:
13:46:20 Mon Oct 06 $ echo -n "Por favor, ingresa el primer número: "
read num1
echo -n "Por favor, ingresa el segundo número: "
read num2
suma=$((num1 + num2))
echo "La suma de $num1 y $num2 es: $suma"
Por favor, ingresa el primer número: 1
Por favor, ingresa el segundo número: 2
La suma de 1 y 2 es: 3
13:46:37 Mon Oct 06 $ echo -e "manzana\npera\nuva\nmanzana" > frutas.txt
grep "manzana" frutas.txt
manzana
manzana
13:47:09 Mon Oct 06 $ echo -e "pera\nuva\nmanzana\nnaranja" > frutas.txt
sort frutas.txt
manzana
naranja
pera
uva
13:47:44 Mon Oct 06 $ #!/bin/bash
echo "John:4.2" >> estudiantes.txt
echo "Mary:3.8" >> estudiantes.txt
echo "David:4.5" >> estudiantes.txt
echo "Emily:3.9" >> estudiantes.txt
echo "Michael:4.1" >> estudiantes.txt
echo "Jessica:4.7" >> estudiantes.txt
echo "Matthew:3.6" >> estudiantes.txt
echo "Ashley:4.3" >> estudiantes.txt
echo "Christopher:4.9" >> estudiantes.txt
echo "Sarah:4.0" >> estudiantes.txt
sort estudiantes.txt > estudiantes_por_nombre.txt
sort -t ':' -k2n estudiantes.txt > estudiantes_por_nota.txt
echo "Archivos creados: estudiantes_por_nombre.txt y estudiantes_por_nota.txt"
Archivos creados: estudiantes_por_nombre.txt y estudiantes_por_nota.txt
13:48:33 Mon Oct 06 $


replit.com/@juanguillermoes/ABC
ABC
Upgrade
linux1_shell.md
linux2_comma...
Files
Search
hacker-lab
colores.txt
estudiantes_por_id_de...
estudiantes_por_nom...
estudiantes_por_nota....
estudiantes.txt
frutas.txt
linux1_shell.md
linux2_commands.md
linux3_scripting.md
logo_wikimedia.png
linux2_commands.md
Open in Editor
-i : Indica que el cambio se hará directamente en el archivo.
/uva/ : Busca las líneas que contienen "uva".
d : Elimina las líneas que coinciden con el patrón.
En resumen:
Estos dos ejemplos muestran cómo sed puede utilizarse para realizar modificaciones básicas en archivos de texto: reemplazar texto y eliminar líneas.
6. Manipulando Textos con awk
El comando awk es una herramienta poderosa y flexible para procesar y analizar texto en archivos. A diferencia de sed , que se centra en la edición de texto, awk es ideal para extraer y manipular datos basados en patrones y estructuras de texto.
Discovery: Origen del comando awk
Ejemplo 10: Extraer el segundo campo de un archivo
echo -e "John:4.2:34567\nMary:3.8:23456\nDavid:4.5:37653" > estudiantes.txt
awk -F ':' '{print $2}' estudiantes.txt
How to Run
Qué sucede:
echo -e "John:4.2:34567\nMary:3.8:23456\nDavid:4.5:37653" > estudiantes.txt
Shell
estudiantes_p...
Publish
~/workspace: echo -e \"John:4.2:34567\nMary:3.8:23456\nDavid:4.5:37653\" > _
echo "Emily:3.9" >> estudiantes.txt
echo "Michael:4.1" >> estudiantes.txt
echo "Jessica:4.7" >> estudiantes.txt
echo "Matthew:3.6" >> estudiantes.txt
echo "Ashley:4.3" >> estudiantes.txt
echo "Christopher:4.9" >> estudiantes.txt
echo "Sarah:4.0" >> estudiantes.txt
sort estudiantes.txt > estudiantes_por_nombre.txt
sort -t ':' -k2n estudiantes.txt > estudiantes_por_nota.txt
echo "Archivos creados: estudiantes_por_nombre.txt y estudiantes_por_nota.txt"
Archivos creados: estudiantes_por_nombre.txt y estudiantes_por_nota.txt
13:48:33 Mon Oct 06 $ #!/bin/bash
echo "John:4.2:34567" > estudiantes.txt
echo "Mary:3.8:23456" >> estudiantes.txt
echo "David:4.5:37653" >> estudiantes.txt
echo "Emily:3.9:45678" >> estudiantes.txt
echo "Michael:4.1:12345" >> estudiantes.txt
echo "Jessica:4.7:56789" >> estudiantes.txt
echo "Matthew:3.6:78901" >> estudiantes.txt
echo "Ashley:4.3:67890" >> estudiantes.txt
echo "Christopher:4.9:89012" >> estudiantes.txt
echo "Sarah:4.0:01234" >> estudiantes.txt
sort -t ':' -k3nr estudiantes.txt > estudiantes_por_id_desc.txt
echo "Archivo creado: estudiantes_por_id_desc.txt"
Archivo creado: estudiantes_por_id_desc.txt
13:49:31 Mon Oct 06 $ echo -e "manzana:rojo\npera:verde\nuva:morada" > colores.t
xt
cut -d ':' -f1 colores.txt
manzana
pera
uva
13:50:43 Mon Oct 06 $ echo -e "manzana\npera\nuva\nmanzana" > frutas.txt
sed -i 's/manzana/kiwi/g' frutas.txt
13:51:21 Mon Oct 06 $ echo -e "manzana\npera\nuva\nmanzana" > frutas.txt
sed -i '/uva/d' frutas.txt
13:52:31 Mon Oct 06 $ echo -e "John:4.2:34567\nMary:3.8:23456\nDavid:4.5:37653"
> estudiantes.txt
awk -F ':' '{print $2}' estudiantes.txt
4.2
3.8
4.5
13:53:01 Mon Oct 06 $
2 mm de lluvia
Jueves
Buscar
ESP
LAA
8:53 a.m.
6/10/2025


Upgrade
linux1_shell.md
linux2_comma...
Publish
Files
Search
hacker-lab
colores.txt
contactos.txt
estudiantes_por_id_de...
estudiantes_por_nom...
estudiantes_por_nota...
estudiantes.txt
frutas.txt
linux1_shell.md
linux2_commands.md
linux3_scripting.md
logo_wikimedia.png
linux2_commands.md
Open in Editor
Este script de Bash proporciona una solución para gestionar una pequeña base de datos de clientes de un banco, permitiendo agregar, borrar, consultar, consignar y retirar dinero, así como generar un reporte.
El script utiliza un archivo llamado banco.txt para almacenar la información de los clientes y sus saldos. Los clientes pueden interactuar con el sistema mediante un menú que ofrece varias opciones para realizar diferentes operaciones.
#!/bin/bash
BANCO="banco.txt"
# Función para mostrar el menú
mostrar_menu() {
echo "-----------------------------------"
echo "Banco "
echo "-----------------------------------"
echo "1. Agregar cliente"
echo "2. Borrar cliente"
echo "3. Consultar saldo de cliente"
echo "4. Listar clientes" # Nueva opción
echo "5. Consignar"
echo "6. Retirar"
echo "7. Total de saldos"
echo "8. Generar reporte"
echo "0. Salir"
echo "-----------------------------------"
echo -n "Selecciona una opción: "
}
# Función para agregar un cliente
agregar_cliente() {
echo -n "Ingrese el nombre del cliente: "
read nombre
Comprobar si el archivo existe y crearlo si no es así
Shell
estudiantes_p...
~/workspace: BANCO=\"banco.txt\"
nDavid:4.5:37653" > estudiantes.txt
awk -F ':' '{print $2}' estudiantes.txt
4.2
3.8
4.5
13:53:01 Mon Oct 06 $ echo -e "John:4.2:34567\nMary:3.8:23456\nDavid:4.5:37653" > estudiantes.txt
awk -F ':' '{sum += $2; count++} END {print "Promedio:", sum/count}' estudiantes.txt
Promedio: 4.16667
13:53:59 Mon Oct 06 $ echo -e "Alice:1234\nBob:5678\nCharlie:91011" > contactos.txt
cut -d ':' -f1 contactos.txt | sort
Alice
Bob
Charlie
13:55:21 Mon Oct 06 $ #!/bin/bash
BANCO="banco.txt"
# Función para mostrar el menú
mostrar_menu() {
echo "-----------------------------------"
echo " Banco "
echo "-----------------------------------"
echo "1. Agregar cliente"
echo "2. Borrar cliente"
echo "3. Consultar saldo de cliente"
echo "4. Listar clientes" # Nueva opción
echo "5. Consignar"
echo "6. Retirar"
echo "7. Total de saldos"
echo "8. Generar reporte"
echo "0. Salir"
echo "-----------------------------------"
echo -n "Selecciona una opción: "
}
# Función para agregar un cliente
agregar_cliente() {
echo -n "Ingrese el nombre del cliente: "
read nombre
# Comprobar si el archivo existe y crearlo si no es así
if [! -f "$BANCO"]; then
touch "$BANCO"
echo "Se ha creado el archivo $BANCO"


Upgrade
linux1_shell.md
linux2_comma...
Publish
Files
Search
hacker-lab
banco.txt
colores.txt
contactos.txt
estudiantes_por_id_de...
estudiantes_por_nom...
estudiantes_por_nota...
estudiantes.txt
frutas.txt
linux1_shell.md
linux2_commands.md
linux3_scripting.md
logo_wikimedia.png
linux2_commands.md
Open in Editor
1. #!/bin/bash
• Indica que el script debe ser ejecutado con el intérprete de Bash.
2. BANCO="banco.txt"
• Define la variable BANCO que contiene el nombre del archivo donde se almacenan los datos de los clientes.
3. mostrar_menu
• Función que muestra el menú de opciones disponibles en el script.
4. agregar_cliente
• grep -q "^$nombre:" "$BANCO"
◦ grep : Busca líneas en un archivo que coincidan con un patrón.
◦ -q : Modo silencioso. Suprime la salida del contenido que coincide, solo devuelve el código de salida.
◦ "^$nombre:" : Expresión regular que busca líneas que comienzan (^) con el valor de la variable $nombre seguido de un : .
• echo "$nombre:$saldo" >> "$BANCO"
◦ echo : Imprime texto en la salida estándar.
◦ "$nombre:$saldo" : Formato del texto a añadir, que incluye el nombre del cliente y su saldo.
◦ >> "$BANCO" : Redirige la salida para agregar el texto al final del archivo BANCO .
5. borrar_cliente
• sed -i "/^$nombre:/d" "$BANCO"
◦ sed : Editor de flujo para modificar archivos de texto.
◦ -i : Edita el archivo en el lugar, sin necesidad de crear un archivo temporal.
◦ "/^$nombre:/d" : Expresión que elimina cualquier línea que comience con $nombre seguido de : .
Shell
estudiantes_p...
~/workspace: BANCO=\"banco.txt\"
3. Consultar saldo de cliente
4. Listar clientes
5. Consignar
6. Retirar
7. Total de saldos
8. Generar reporte
0. Salir

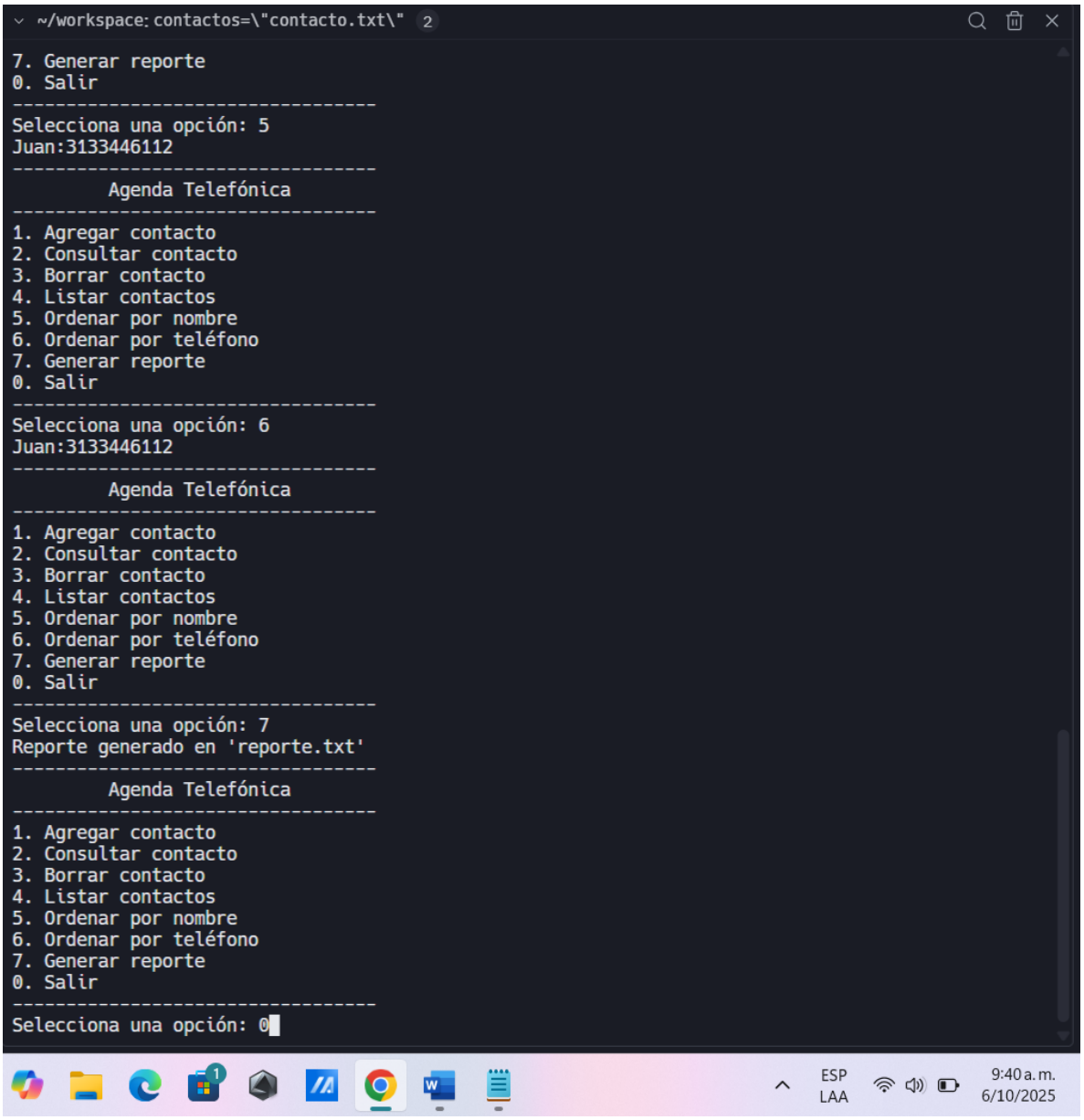
Selecciona una opción: 1
Ingrese el nombre del cliente: Juan
Ingrese el saldo inicial: 1000
Cliente agregado con éxito.

Banco
----
1. Agregar cliente
2. Borrar cliente
3. Consultar saldo de cliente
4. Listar clientes
5. Consignar
6. Retirar
7. Total de saldos
8. Generar reporte
0. Salir

Selecciona una opción: 4
Lista de clientes:
Nombre Saldo
Juan 1000

Banco
----
1. Agregar cliente
2. Borrar cliente
3. Consultar saldo de cliente
4. Listar clientes
5. Consignar
6. Retirar
7. Total de saldos
8. Generar reporte
0. Salir

Selecciona una opción:


~/workspace: contactos=\"contacto.txt\" 2
7. Generar reporte
0. Salir
Selecciona una opción: 5
Juan:3133446112
Agenda Telefónica
1. Agregar contacto
2. Consultar contacto
3. Borrar contacto
4. Listar contactos
5. Ordenar por nombre
6. Ordenar por teléfono
7. Generar reporte
0. Salir
Selecciona una opción: 6
Juan:3133446112
Agenda Telefónica
1. Agregar contacto
2. Consultar contacto
3. Borrar contacto
4. Listar contactos
5. Ordenar por nombre
6. Ordenar por teléfono
7. Generar reporte
0. Salir
Selecciona una opción: 7
Reporte generado en 'reporte.txt'
Agenda Telefónica
1. Agregar contacto
2. Consultar contacto
3. Borrar contacto
4. Listar contactos
5. Ordenar por nombre
6. Ordenar por teléfono
7. Generar reporte
0. Salir
Selecciona una opción: 0
ESP
LAA
9:40 a. m.
6/10/2025