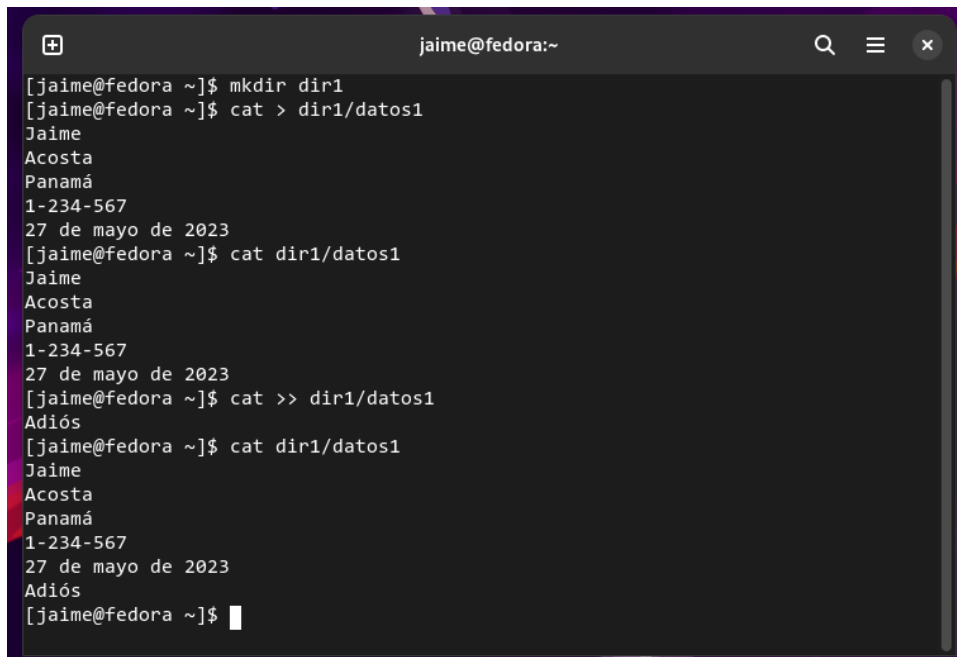
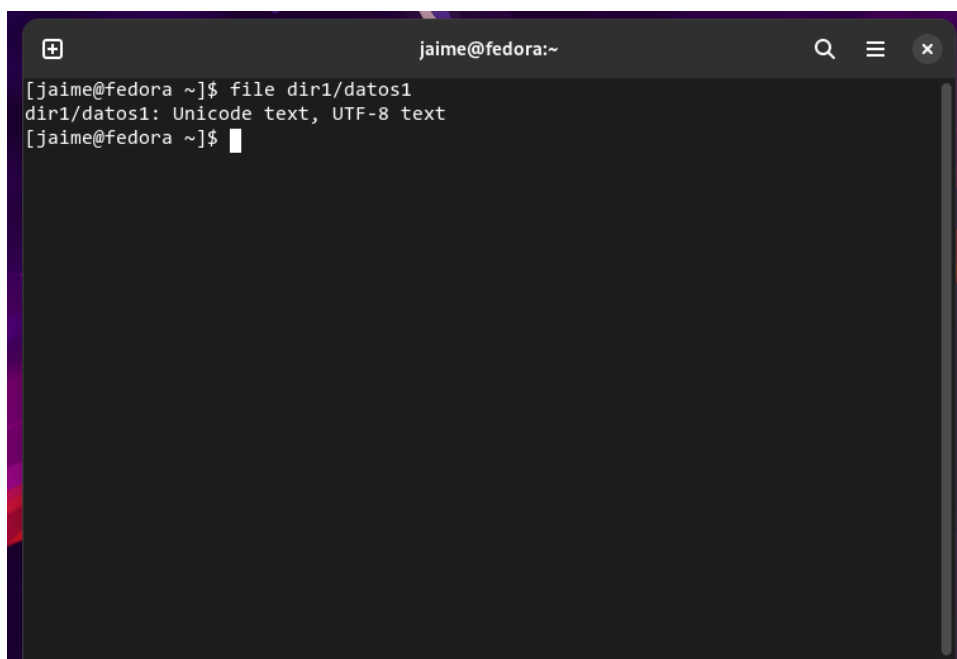


1. Comando cat



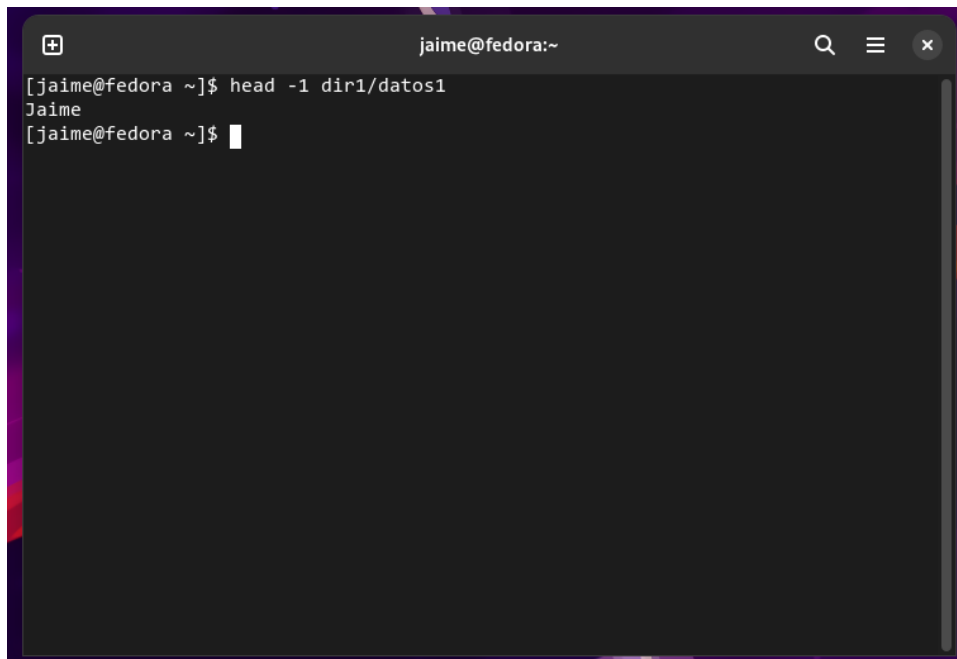
```
jaime@fedora:~  
[jaime@fedora ~]$ mkdir dir1  
[jaime@fedora ~]$ cat > dir1/datos1  
Jaime  
Acosta  
Panamá  
1-234-567  
27 de mayo de 2023  
[jaime@fedora ~]$ cat dir1/datos1  
Jaime  
Acosta  
Panamá  
1-234-567  
27 de mayo de 2023  
[jaime@fedora ~]$ cat >> dir1/datos1  
Adiós  
[jaime@fedora ~]$ cat dir1/datos1  
Jaime  
Acosta  
Panamá  
1-234-567  
27 de mayo de 2023  
Adiós  
[jaime@fedora ~]$
```

2. Comando file



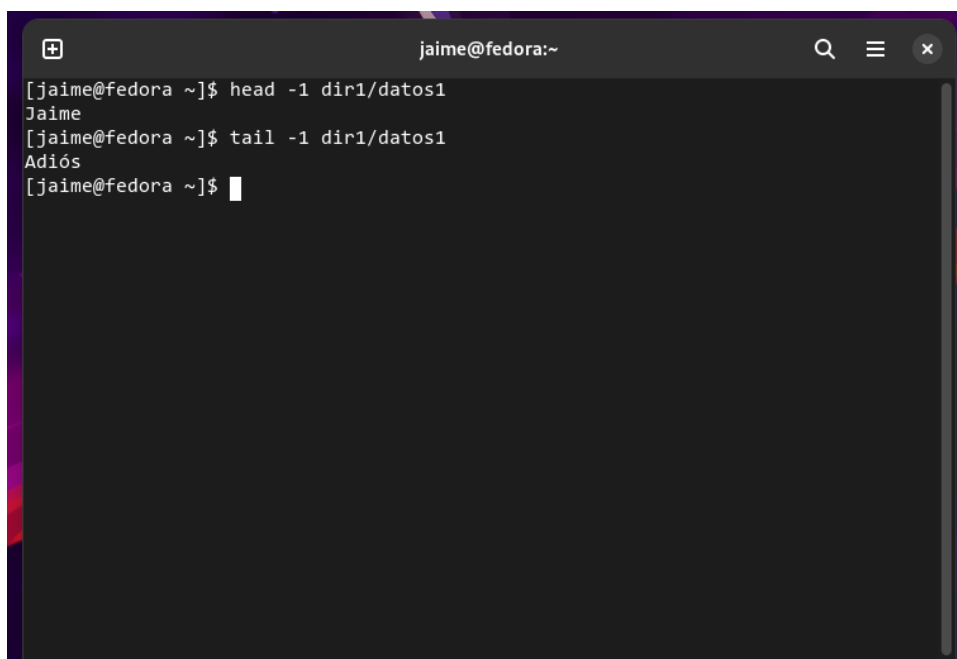
```
jaime@fedora:~  
[jaime@fedora ~]$ file dir1/datos1  
dir1/datos1: Unicode text, UTF-8 text  
[jaime@fedora ~]$
```

3. Comando head

A terminal window titled 'jaime@fedora:~' with search, menu, and close icons. It shows the command 'head -1 dir1/datos1' being executed, which outputs 'Jaime' on the next line. The prompt is ready for the next command.

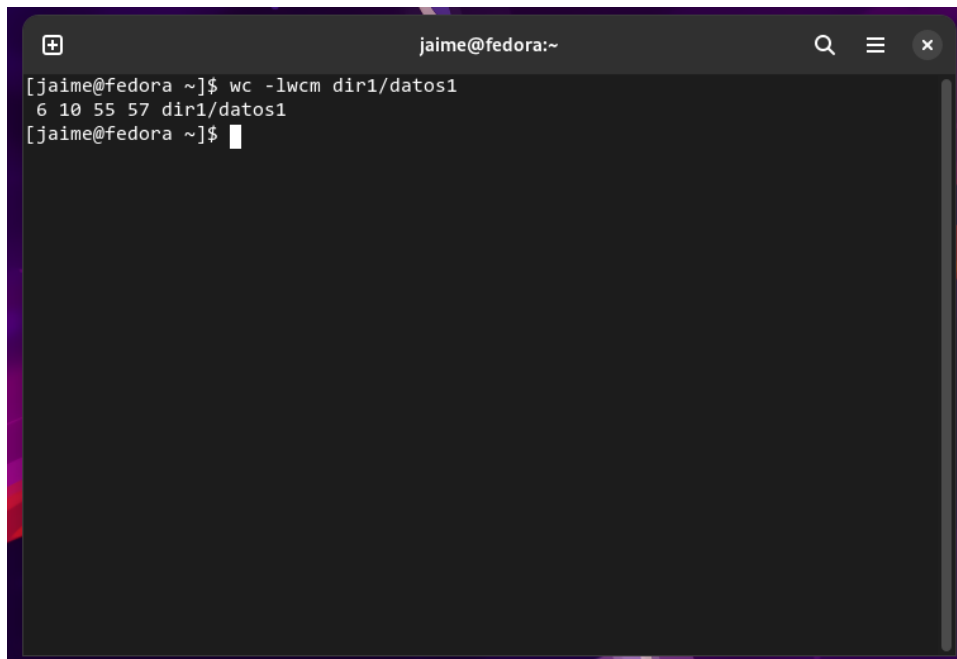
```
jaime@fedora:~  
[jaime@fedora ~]$ head -1 dir1/datos1  
Jaime  
[jaime@fedora ~]$
```

4. Comando tail

A terminal window titled 'jaime@fedora:~' with search, menu, and close icons. It shows two commands: 'head -1 dir1/datos1' which outputs 'Jaime', and 'tail -1 dir1/datos1' which outputs 'Adiós'. The prompt is ready for the next command.

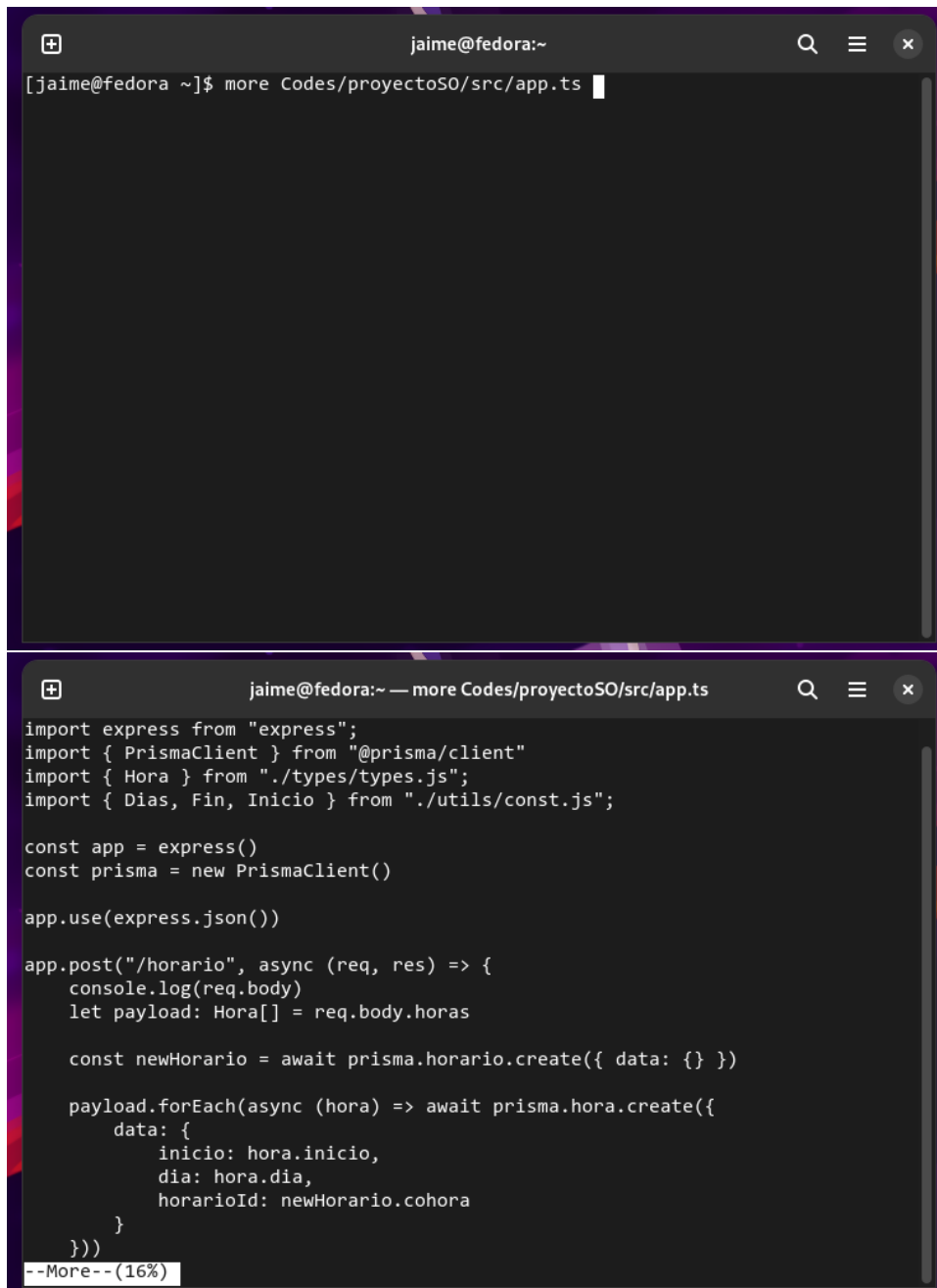
```
jaime@fedora:~  
[jaime@fedora ~]$ head -1 dir1/datos1  
Jaime  
[jaime@fedora ~]$ tail -1 dir1/datos1  
Adiós  
[jaime@fedora ~]$
```

5. Comando wc

A terminal window titled 'jaime@fedora:~' with search, menu, and close icons. It shows the command '[jaime@fedora ~]\$ wc -lwc dir1/datos1' and its output '6 10 55 57 dir1/datos1'. The prompt '[jaime@fedora ~]\$' is followed by a cursor.

```
[jaime@fedora ~]$ wc -lwc dir1/datos1
6 10 55 57 dir1/datos1
[jaime@fedora ~]$
```

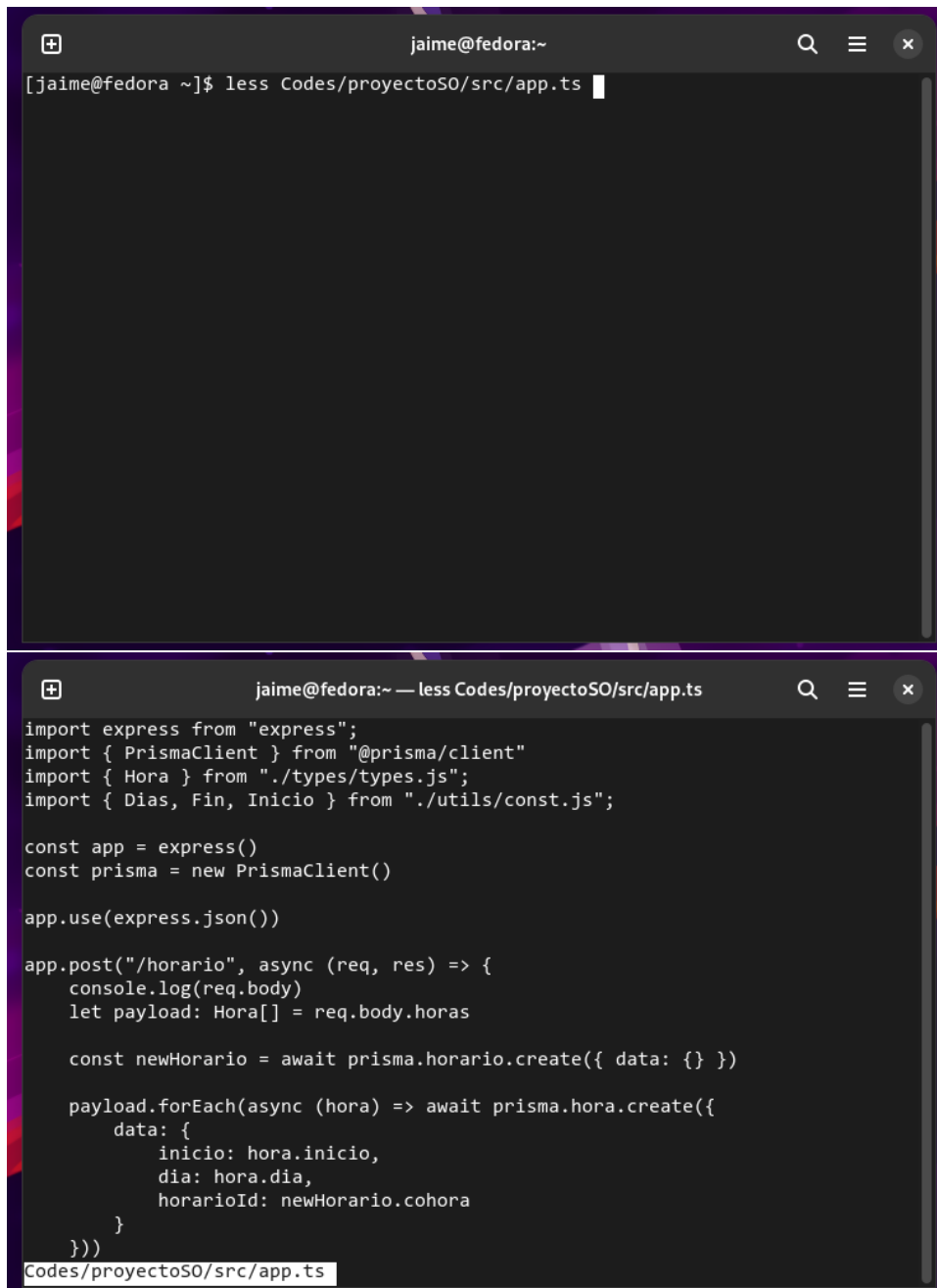
6. Comando more



The image consists of two screenshots of a terminal window. The top screenshot shows the terminal prompt `[jaime@fedora ~]$` followed by the command `more Codes/proyectoSO/src/app.ts`. The bottom screenshot shows the same terminal window with the file content displayed. The file content is a TypeScript file named `app.ts` that imports `express`, `@prisma/client`, and local types and constants. It defines an Express application with a `POST` endpoint `/horario` that creates a new `Horario` entry in the database for each hour in the request body. The bottom of the screenshot shows the `--More-- (16%)` prompt, indicating that the file is being paginated.

```
jaime@fedora:~  
[jaime@fedora ~]$ more Codes/proyectoSO/src/app.ts  
  
jaime@fedora:~ — more Codes/proyectoSO/src/app.ts  
import express from "express";  
import { PrismaClient } from "@prisma/client"  
import { Hora } from "../types/types.js";  
import { Dias, Fin, Inicio } from "../utils/const.js";  
  
const app = express()  
const prisma = new PrismaClient()  
  
app.use(express.json())  
  
app.post("/horario", async (req, res) => {  
  console.log(req.body)  
  let payload: Hora[] = req.body.horas  
  
  const newHorario = await prisma.horario.create({ data: {} })  
  
  payload.forEach(async (hora) => await prisma.hora.create({  
    data: {  
      inicio: hora.inicio,  
      dia: hora.dia,  
      horarioId: newHorario.cohora  
    }  
  }))  
})  
--More-- (16%)
```

7. Comando less



The image consists of two screenshots of a terminal window. The top screenshot shows the command prompt `[jaime@fedora ~]$ less Codes/proyectoS0/src/app.ts` with a cursor at the end. The bottom screenshot shows the same terminal window after the file has been opened, displaying the contents of `Codes/proyectoS0/src/app.ts`. The terminal window has a title bar that reads `jaime@fedora:~` and a search icon, a menu icon, and a close icon. The code displayed in the terminal is as follows:

```
import express from "express";
import { PrismaClient } from "@prisma/client"
import { Hora } from "../types/types.js";
import { Dias, Fin, Inicio } from "../utils/const.js";

const app = express()
const prisma = new PrismaClient()

app.use(express.json())

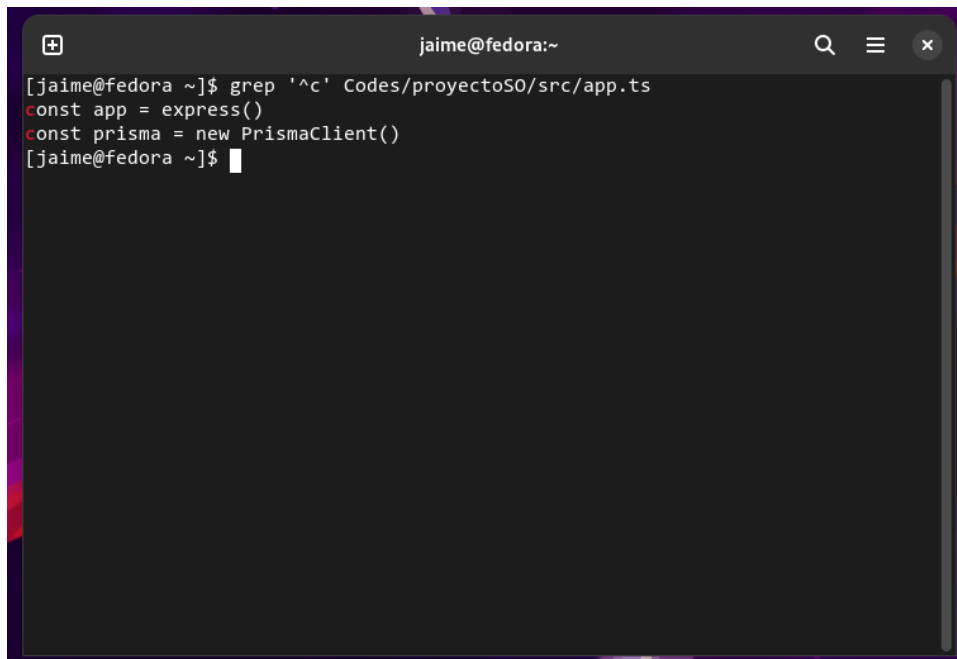
app.post("/horario", async (req, res) => {
  console.log(req.body)
  let payload: Hora[] = req.body.horas

  const newHorario = await prisma.horario.create({ data: {} })

  payload.forEach(async (hora) => await prisma.hora.create({
    data: {
      inicio: hora.inicio,
      dia: hora.dia,
      horarioId: newHorario.cohora
    }
  }))
})
```

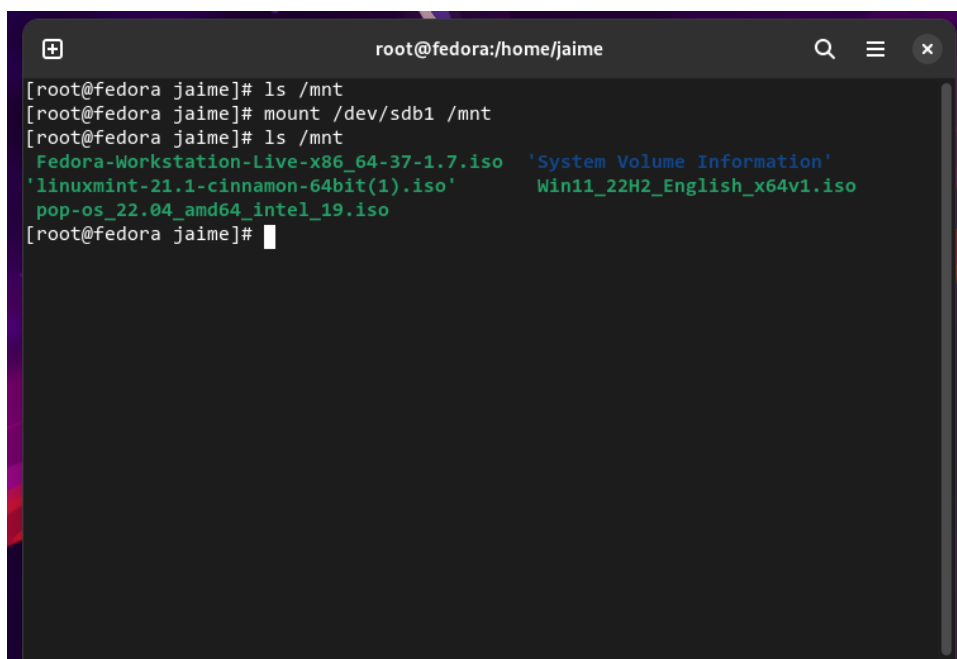
The bottom of the terminal window shows the file path `Codes/proyectoS0/src/app.ts` in the status bar.

8. Comando grep



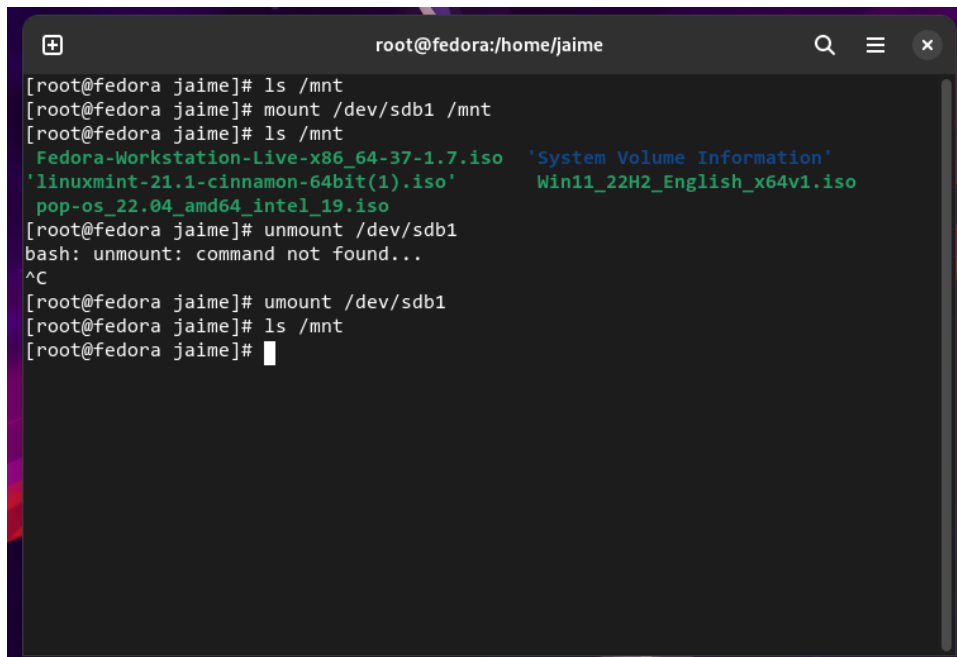
```
jaime@fedora:~  
[jaime@fedora ~]$ grep '^c' Codes/proyectoS0/src/app.ts  
const app = express()  
const prisma = new PrismaClient()  
[jaime@fedora ~]$
```

9. Comando mount



```
root@fedora:/home/jaime  
[root@fedora jaime]# ls /mnt  
[root@fedora jaime]# mount /dev/sdb1 /mnt  
[root@fedora jaime]# ls /mnt  
Fedora-Workstation-Live-x86_64-37-1.7.iso 'System Volume Information'  
'linuxmint-21.1-cinnamon-64bit(1).iso' Win11_22H2_English_x64v1.iso  
pop-os_22.04_amd64_intel_19.iso  
[root@fedora jaime]#
```

10. Comando umount

A terminal window titled 'root@fedora:/home/jaime' with search, menu, and close icons in the title bar. The terminal shows the following commands and output:

```
[root@fedora jaime]# ls /mnt
[root@fedora jaime]# mount /dev/sdb1 /mnt
[root@fedora jaime]# ls /mnt
Fedora-Workstation-Live-x86_64-37-1.7.iso  'System Volume Information'
'linuxmint-21.1-cinnamon-64bit(1).iso'    Win11_22H2_English_x64v1.iso
pop-os_22.04_amd64_intel_19.iso
[root@fedora jaime]# umount /dev/sdb1
bash: umount: command not found...
^C
[root@fedora jaime]# umount /dev/sdb1
[root@fedora jaime]# ls /mnt
[root@fedora jaime]#
```