***Primero se creo la arquitetcura como la propueso el profesor***

martinfranciscocortesbuendia@MacBook-Pro-de-Martin-2 ComputerSecurity % cd honeypot-project

martinfranciscocortesbuendia@MacBook-Pro-de-Martin-2 honeypot-project % mkdir -p prisma src/routes src/controllers src/middleware src/views

martinfranciscocortesbuendia@MacBook-Pro-de-Martin-2 honeypot-project % touch prisma/schema.prisma \

src/app.js \

src/routes/authRoutes.js \

src/routes/adminRoutes.js \

src/controllers/authController.js \

src/controllers/adminController.js \

src/middleware/honeypotProtection.js \

src/views/login.ejs \

src/views/register.ejs \

src/views/user.ejs \

src/views/admin.ejs \

.env \

package.json \

README.md

martinfranciscocortesbuendia@MacBook-Pro-de-Martin-2 honeypot-project % tree .

.

├── package.json

├── prisma

│   └── schema.prisma

├── README.md

└── src

├── app.js

├── controllers

│   ├── adminController.js

│   └── authController.js

├── middleware

│   └── honeypotProtection.js

├── routes

│   ├── adminRoutes.js

│   └── authRoutes.js

└── views

├── admin.ejs

├── login.ejs

├── register.ejs

└── user.ejs

7 directories, 13 files

martinfranciscocortesbuendia@MacBook-Pro-de-Martin-2 honeypot-project %

nstalar dependencias

bash

# Dependencias principales

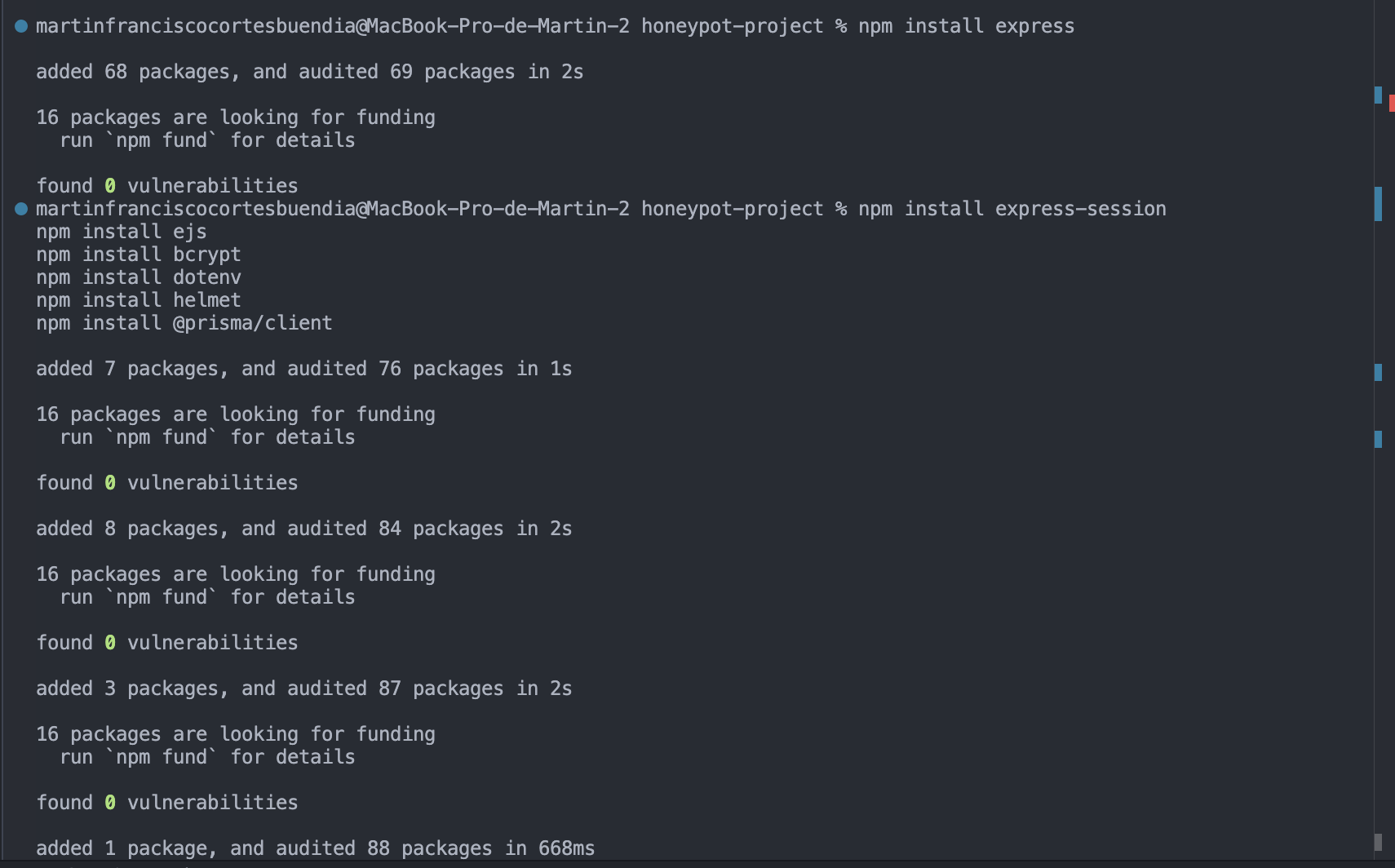
npm install express express-session ejs bcrypt dotenv helmet @prisma/client

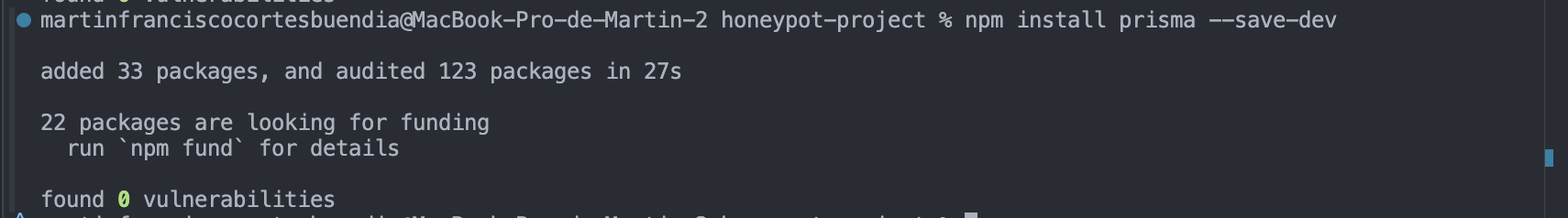
# Dependencias de desarrollo

npm install prisma --save-dev

# Dependencia adicional para process.cwd() en ES modules

npm install path







Paso dos

# 1. Instalar Homebrew (si no lo tienes)

/bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"

# 2. Instalar PostgreSQL

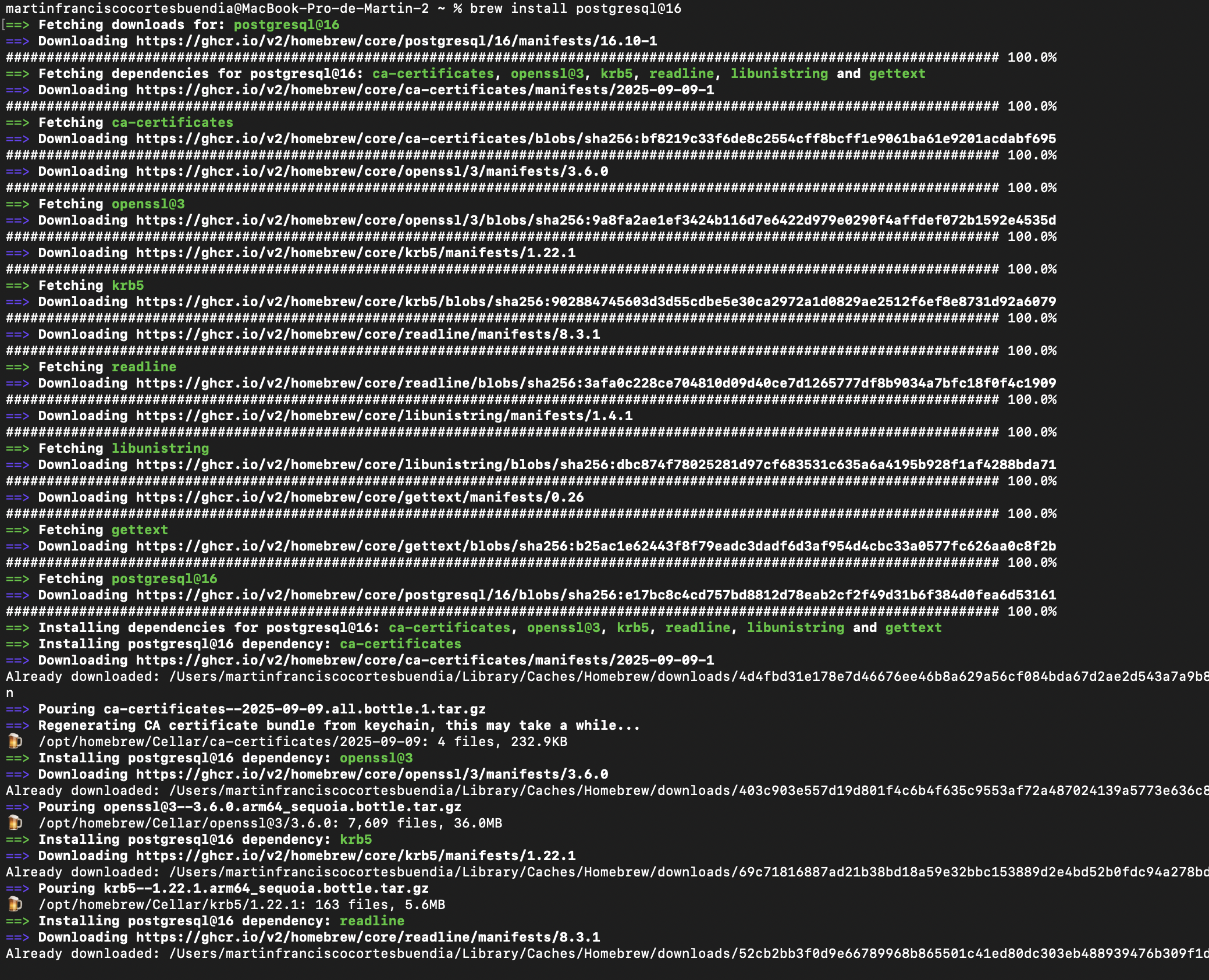
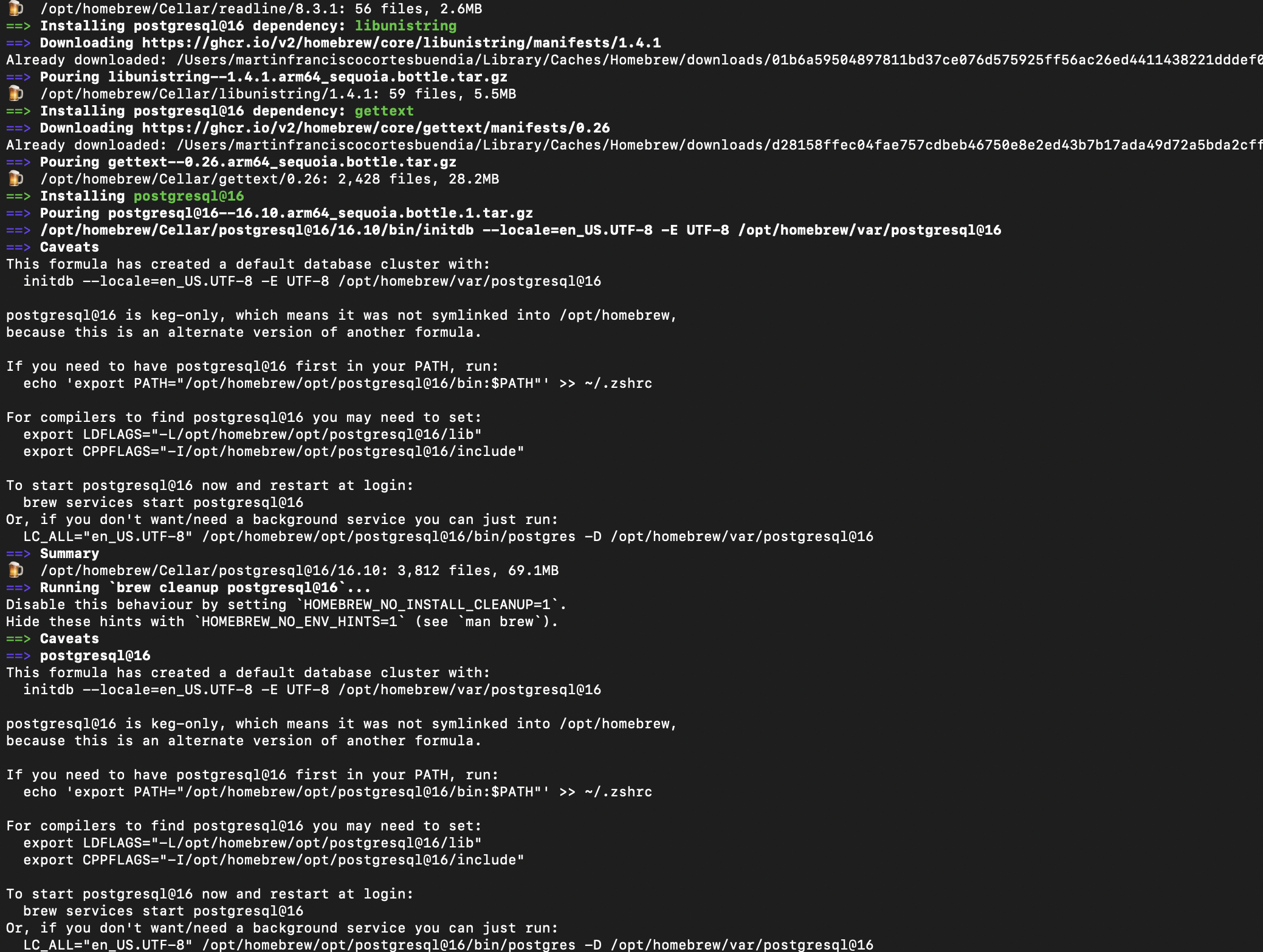
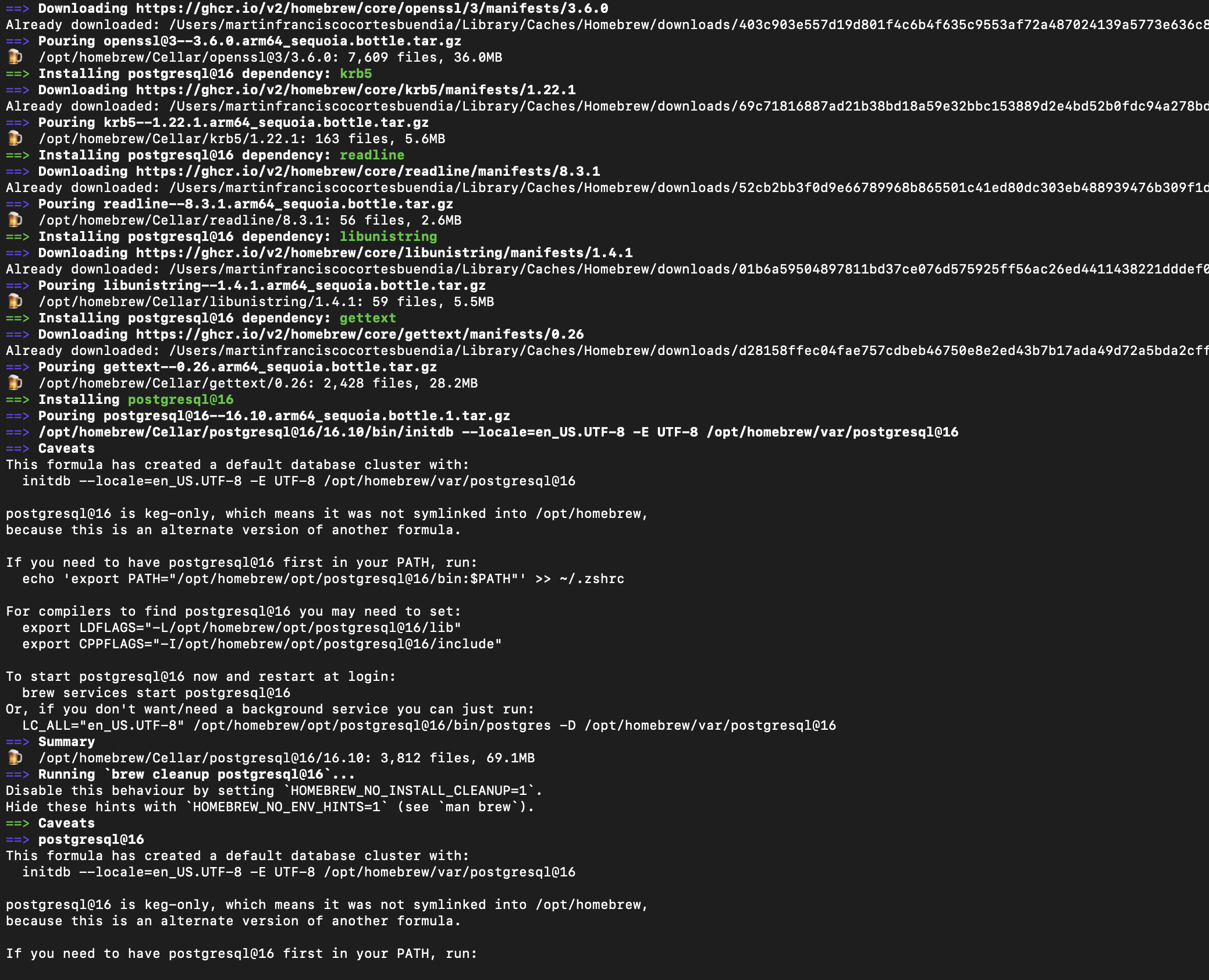
brew install postgresql@16

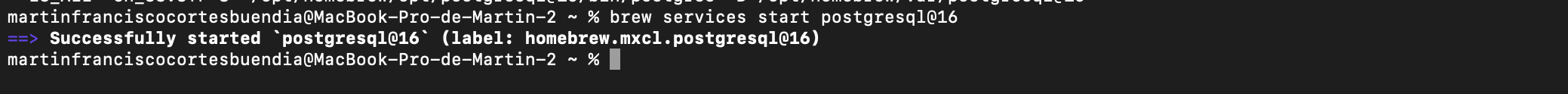
# 3. Iniciar PostgreSQL como servicio

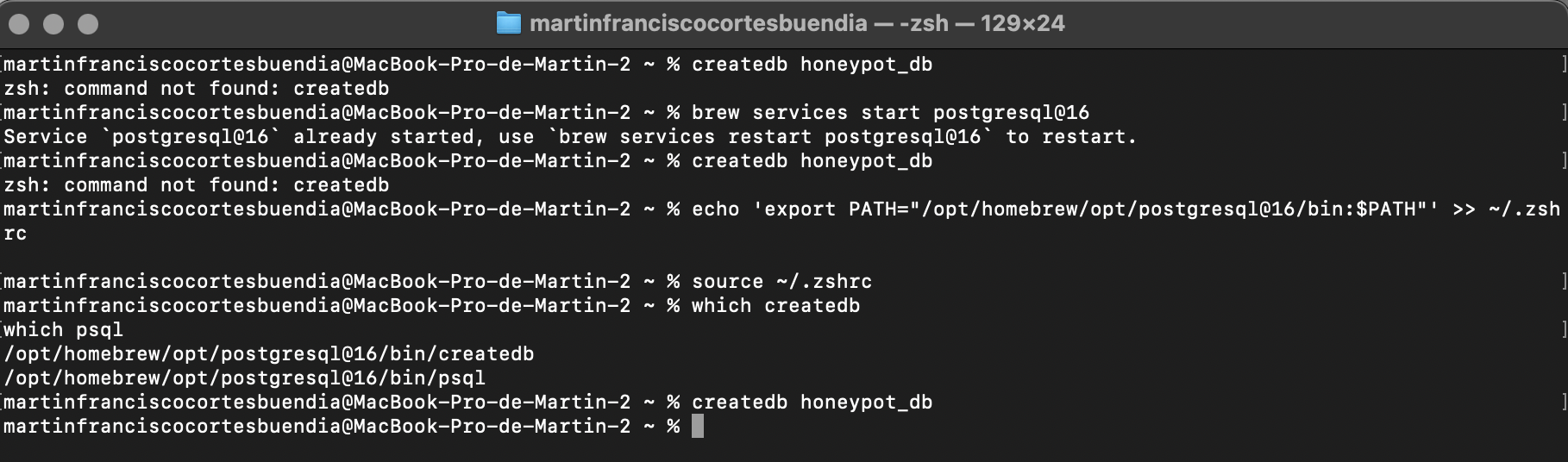
brew services start postgresql@16

# 4. Crear la base de datos

createdb honeypot\_db







A .env le puse   
DATABASE\_URL="postgresql://martinfranciscocortesbuendia:@localhost:5432/honeypot\_db?schema=public"

SESSION\_SECRET="clave\_segura\_sesiones\_para\_mac\_2024"

PORT=3000

al archivo prisma le puse   
generator client {

provider = "prisma-client-js"

}

datasource db {

provider = "postgresql"

url = env("DATABASE\_URL")

}

model User {

id Int @id @default(autoincrement())

email String @unique

password String

role String @default("user")

createdAt DateTime @default(now())

}

Y despues hacemos migracion  
con lo siguiente

martinfranciscocortesbuendia@MacBook-Pro-de-Martin-2 honeypot-project % npx prisma migrate dev --name init

Environment variables loaded from .env

Prisma schema loaded from prisma/schema.prisma

Datasource "db": PostgreSQL database "honeypot\_db", schema "public" at "localhost:5432"

Applying migration `20251021044342\_init`

The following migration(s) have been created and applied from new schema changes:

prisma/migrations/

└─ 20251021044342\_init/

└─ migration.sql

Your database is now in sync with your schema.

✔ Generated Prisma Client (v6.17.1) to ./node\_modules/@prisma/client in 72ms

martinfranciscocortesbuendia@MacBook-Pro-de-Martin-2 honeypot-project % npx prisma generate

Environment variables loaded from .env

Prisma schema loaded from prisma/schema.prisma

✔ Generated Prisma Client (v6.17.1) to ./node\_modules/@prisma/client in 76ms

Start by importing your Prisma Client (See: https://pris.ly/d/importing-client)

Tip: Interested in query caching in just a few lines of code? Try Accelerate today! https://pris.ly/tip-3-accelerate

martinfranciscocortesbuendia@MacBook-Pro-de-Martin-2 honeypot-project %

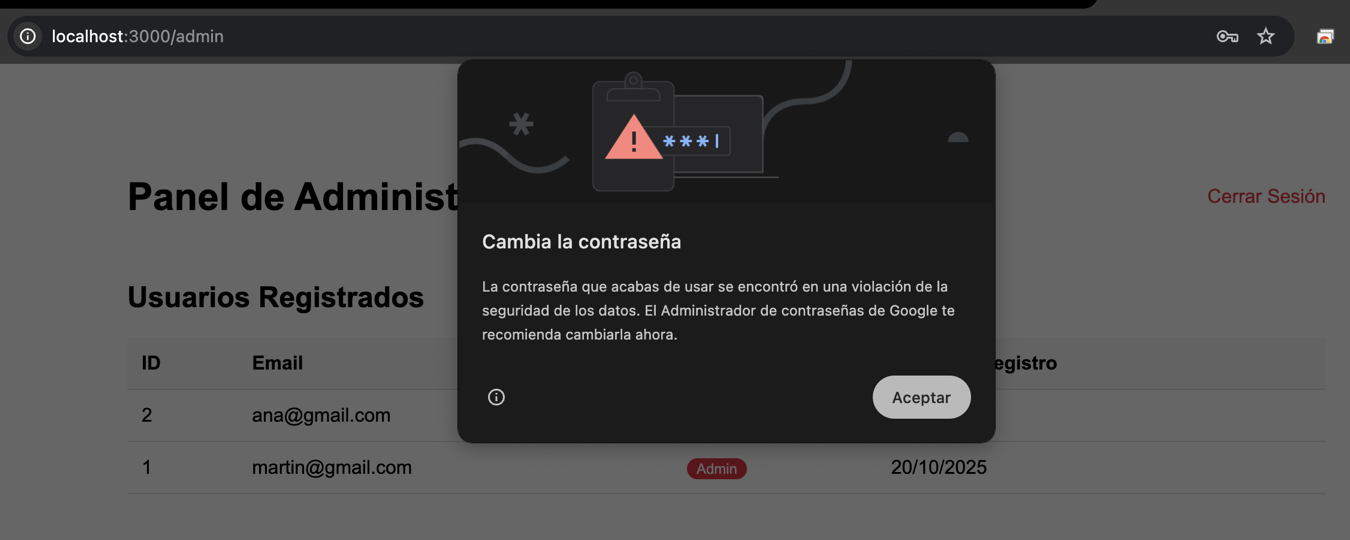
despues le damos a

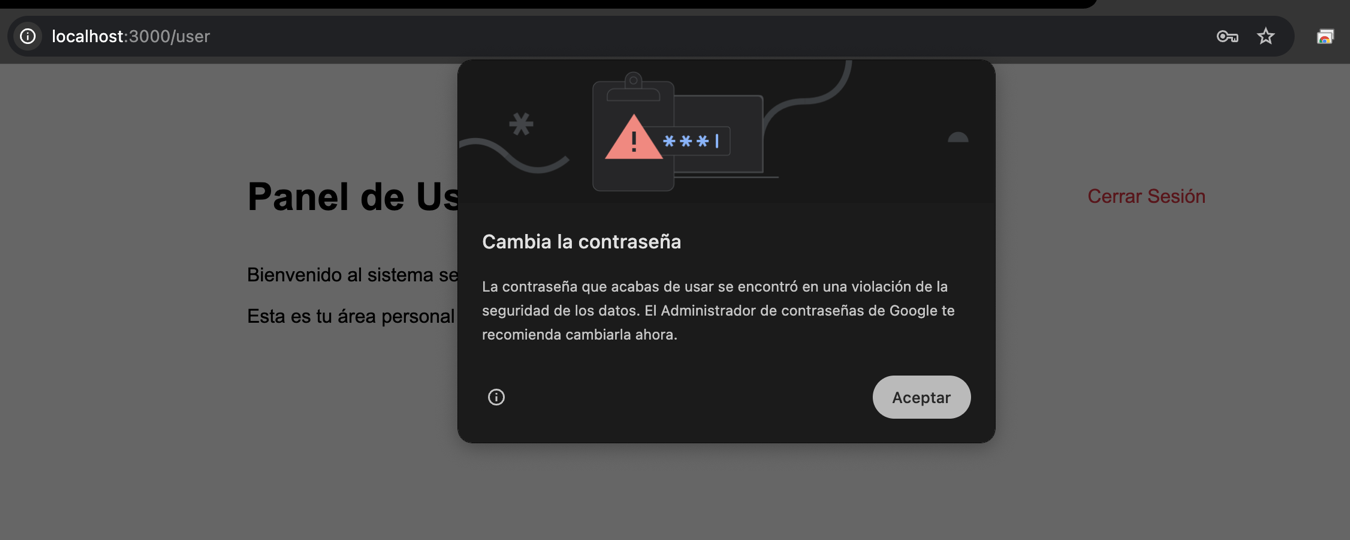
martinfranciscocortesbuendia@MacBook-Pro-de-Martin-2 honeypot-project % npm start

y de ahí nos vamos a localhost en el puerto 3000

<http://localhost:3000/login>

y si le damos a registrar y enviamos confirmacion nos da lo siguiente





**2.**Prueba el flujo normal:

Registra el primer usuario (será automáticamente administrador)

Inicia sesión

Accede al panel de administración (si eres admin)

Cierra sesión

 Prueba la protección Honeypot:

Opción A: Con curl (simulando bot)

bash

curl -X POST http://localhost:3000/register \

-d "email=bot@test.com&password=123456&hiddenField=soy\_un\_bot"

Deberías recibir: "Acceso bloqueado por actividad sospechosa."

Opción B: Con el navegador

Abre las herramientas de desarrollador (F12)

Ve a la pestaña "Elements"

Busca el campo honeypot (tiene style="display: none")

Cambia el estilo a display: block

Llena el campo y envía el formulario

4. Verifica en la terminal del servidor:

Deberías ver el mensaje: " Bot detectado - campo honeypot lleno: soy\_un\_bot"

Estructura final que se tiene funcionando:

* Servidor Express con seguridad Helmet  
  Base de datos PostgreSQL con Prisma ORM  
  Autenticación segura con bcrypt  
  Sesiones protegidas con cookies HttpOnly  
  Sistema de roles (admin/user)  
  Protección Honeypot contra bots  
  Vistas EJS dinámicas

Comandos útiles para continuar:

bash

# Para desarrollo con auto-recarga

npm run dev

# Para ver los logs en tiempo real

# (mantén esta terminal abierta)