

Fundamentos de Programación y Métodos Numéricos

Visión General de las Bases de Datos

2
0
2
6



Introducción

1

Entrada por teclado

2

Salida por pantalla

3

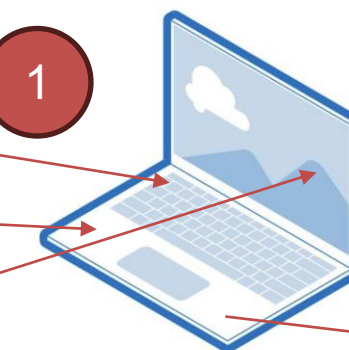
Uso de Memoria Primaria

4

Uso de Memoria Secundaria

Programa en Python

```
1 import math as m
2
3 def cargar_matriz_coeficientes():
4     a = [[0, 0], [0, 0]]
5     for f in range(len(a)):
6         for c in range(len(a[0])):
7             a[f][c] = float(input('Ingrese el coheficiente a' + str(f+1) + str(c+1)))
8     return a
9
10 def cargar_arreglo_constantes():
11     c = [0, 0]
12     for i in range(len(c)):
13         c[i] = float(input('Ingrese la constante c' + str(i+1)))
14     return c
15
16 def error(k1, k1_ant, k2, k2_ant):
17     return m.sqrt(m.pow(k1-k1_ant,2) + m.pow(k2-k2_ant,2)) / m.sqrt(m.pow(k1,2)+m.pow(k2,2))
18
19 a = cargar_matriz_coeficientes()
20 print('Matriz coeficientes: ')
21 print(a)
22 c = cargar_arreglo_constantes()
23 print('Arreglo constantes: ')
24 print(c)
```



Sistema Operativo



Archivo/BD

Necesidades de Almacenar Información

Problema: Contamos con mucha información



.....y en diferentes tipos de almacenamiento:

- Un archivo de texto 
- Varios archivos de texto 
- Base de Datos 



Bases de Datos

Una base de datos es repositorio en la cual se pueden **almacenar datos de manera estructurada** (en tablas), con la **menor redundancia posible**.

Diferentes programas y diferentes usuarios deben poder utilizar estos datos.





Que es un Dato de una Base de Datos?

Un **Dato** es una representaciones simbólicas (numéricas, alfabéticas, lógicas, etc.) de una determinada variable cualitativa o cuantitativa.

Los datos han de ser interpretados (dándole significado) para que se conviertan en **información útil**.

una persona ha nacido en 1965,

el **dato** (1965) va acompañado de su **interpretación** (año de nacimiento de una cierta persona)



Ejemplo de Tabla

MÉDICO

Nombre de la tabla
(relación)

Nombre de la
columna
(atributo)

Esquema de
una tabla o
cabecera -
(comprensión)

Fila (tupla)

Valor o
estado de de
la tabla
(extensión)

Matricula	NyApell	Especialidad	DNI	ClinicaEjerce
234555	Juan Paz	Traumatología	26456678	C. Modelo
345234	Inés Roca	Pediatría	30564865	C. Paz
365478	Pedro Jara	Traumatología	23546987	Cons. Privado
....				

Cada columna tiene un dominio de definición que incluye los valores posibles que puede tomar



Esquemas de Base de Datos

- Una base de datos relacional consiste en un **conjunto de tablas**, a cada una de las cuales se le asigna un nombre exclusivo dentro del esquema de cada usuario
- Cada tabla contiene un conjunto de **filas** de la tabla representa una relación entre un conjunto de valores.
- Cada tabla posee un conjunto de columnas cabeceras (**atributos**) cuyo nombre debe ser único dentro de la tabla
- Para cada atributo hay un conjunto de valores permitidos, llamado **dominio** de ese atributo



Interacción con una BD - Lenguaje SQL

La interacción con la base de datos se realiza a través de sentencia SQL (lenguaje de programación):

- SQL son las siglas de **Structured Query Language** (Lenguaje de Consulta Estructurado)
- SQL permite:
 - crear/borrar/actualizar tablas
 - crear/borrar/actualizar tuplas/registros/datos
 - consultar datos de la tablas



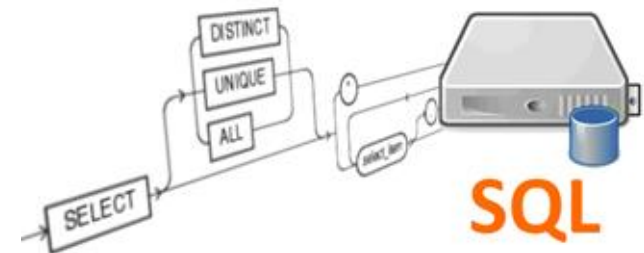
Lenguaje SQL

Por ejemplo algunas sentencias SQL para el manejo de tablas:

CREATE TABLE <nom_tabla> (...); → **creación de una tabla**

ALTER TABLE <nom_tabla> ... ; → **modificación de una tabla**

DROP TABLE <nom_tabla> ; → **eliminación de una tabla**





Structured Query Language

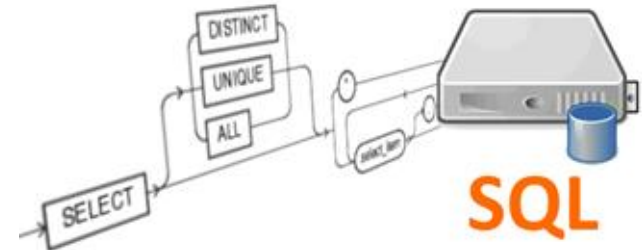
Por ejemplo algunas sentencias para el manejo de tuplas/registros/datos:

INSERT INTO <nom_tabla> (...) **VALUES** (...); → inserción de registros

UPDATE <nom_tabla> SET (...) **WHERE** (...); → modificación de registros

DELETE FROM <nom_tabla> **WHERE** (...); → borrado de registros

SELECT <lista_atrib> **FROM** <lista_tablas> **WHERE** (...); → consulta de datos





Lenguaje de Consulta SQL

- ✓ La sentencia del lenguaje empleada para la recuperación de los datos a partir de las tablas cargadas en la base de datos es el SELECT.
- ✓ La sentencia básica es:

```
SELECT * | { [DISTINCT] columna | expresión [alias],...}  
FROM <lista tablas>
```

- ✓ En una consulta se especifica qué información se requiere, sin especificar cómo obtenerla (no requiere métodos de acceso a los datos).



Lenguaje de Consulta SQL

- ✓ Las cláusulas de la sentencias suelen colocarse en líneas separadas y con sangría, para mejorar la legibilidad
- ✓ Se estila escribir con MAYÚSCULAS las palabras reservadas y con minúsculas el resto
- ✓ Por ejemplo:

```
SELECT mi_atributo  
FROM mi_tabla;
```



Lenguaje de Consulta SQL

```
SELECT * | { [DISTINCT] columna | expresión [alias],...}  
FROM <lista tablas>
```

- **SELECT** identifica las columnas a recuperar – **EL QUE**
- **FROM** identifica la tabla - **DE DONDE** obtener los datos
- El resultado de una consulta es una tabla (si es un número, se considera como una tabla con una fila y una columna).



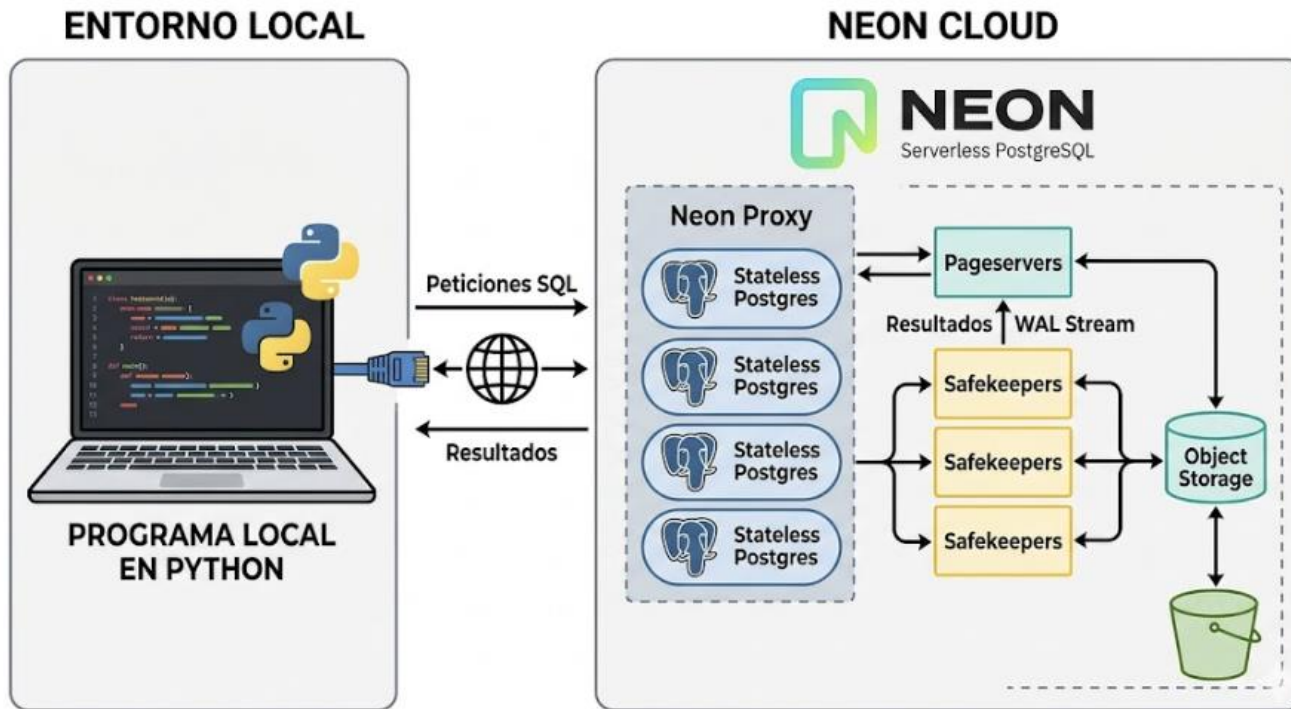
Restringir las Filas Recuperadas

- ✓ La cláusula **WHERE** se usa para realizar las restricciones.
- ✓ Una cláusula **WHERE** contiene una condición lógica, la cual usa
 - operadores de comparación (<, >, =, <=, >=, <> o !=)
 - operadores lógicos (**AND**, **OR**, **NOT**).
- ✓ Las filas recuperadas son aquellas cuyos datos que satisfacen la/s condición/es lógicas

```
SELECT * | { [DISTINCT] columna | expresión [alias],...}  
FROM <lista tablas>  
[WHERE condición/es];
```

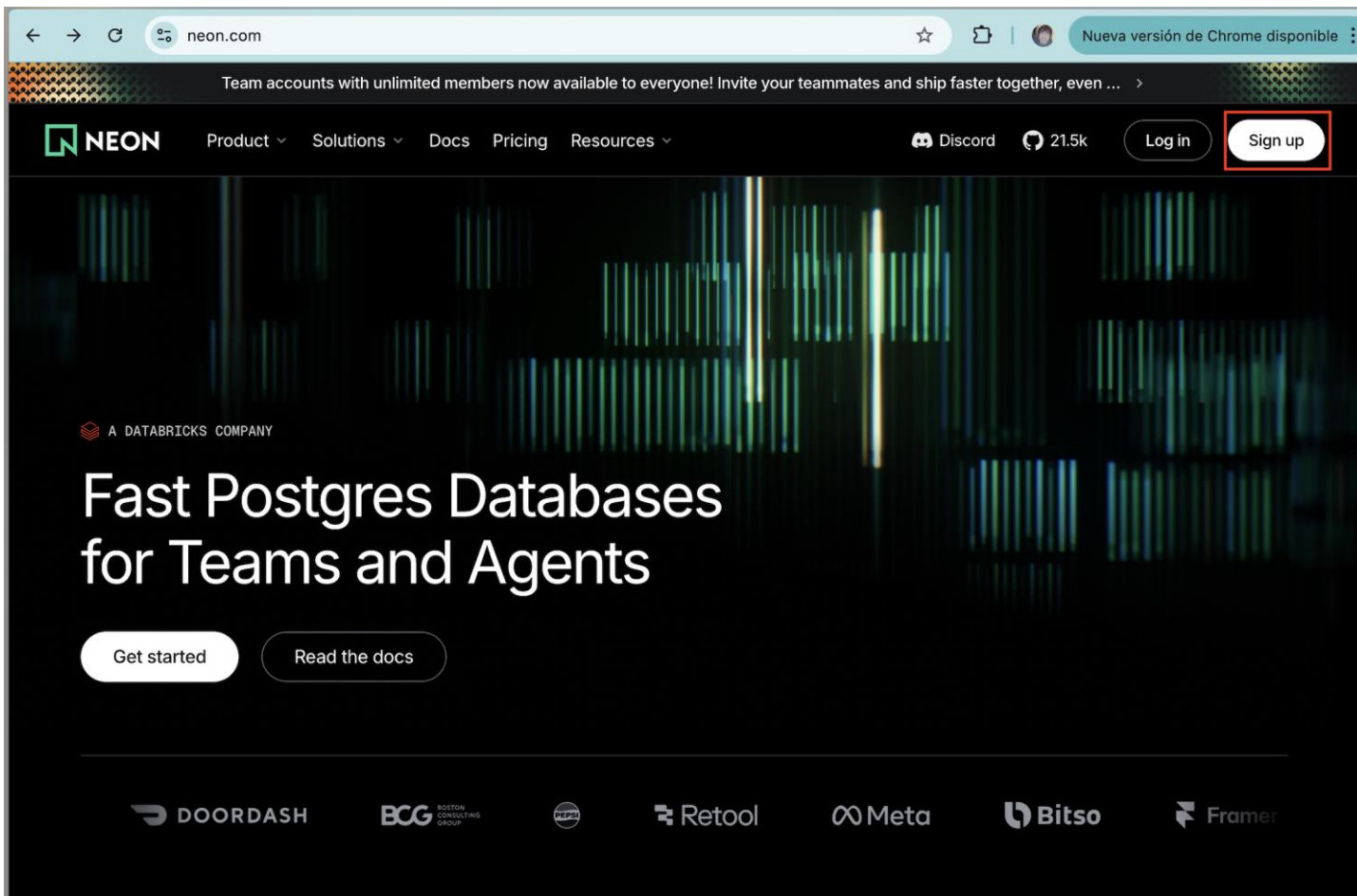
Neon Serverless Postgres — Ship faster

Url: <https://neon.com/>



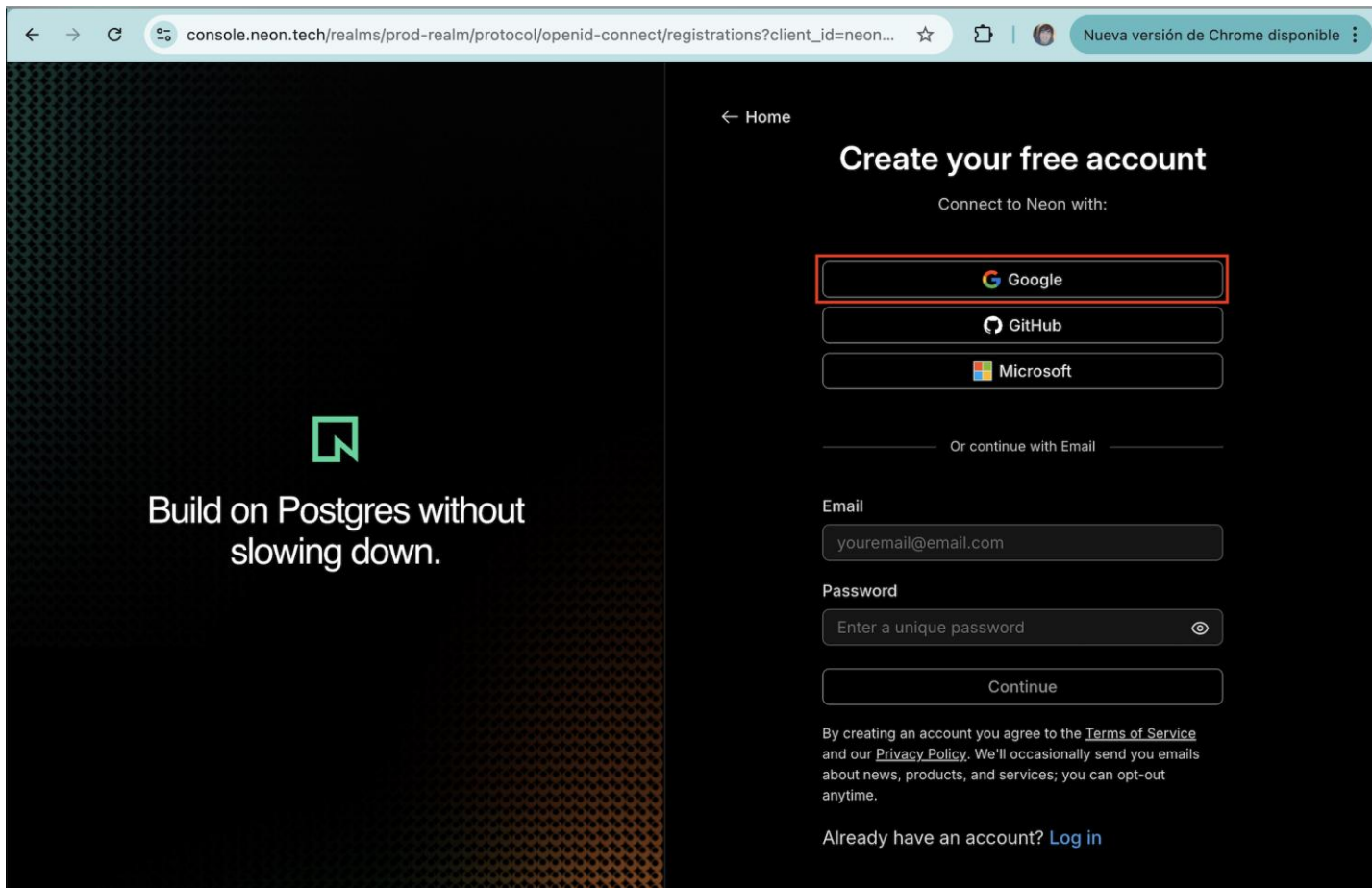
Neon Serverless Postgres — Ship faster

Crear Cuenta:



Neon Serverless Postgres — Ship faster

Crear Cuenta:



console.neon.tech/realms/prod-realm/protocol/openid-connect/registrations?client_id=neon... Nueva versión de Chrome disponible

← Home

Create your free account

Connect to Neon with:

- Google
- GitHub
- Microsoft

Or continue with Email

Email
youremail@email.com

Password
Enter a unique password

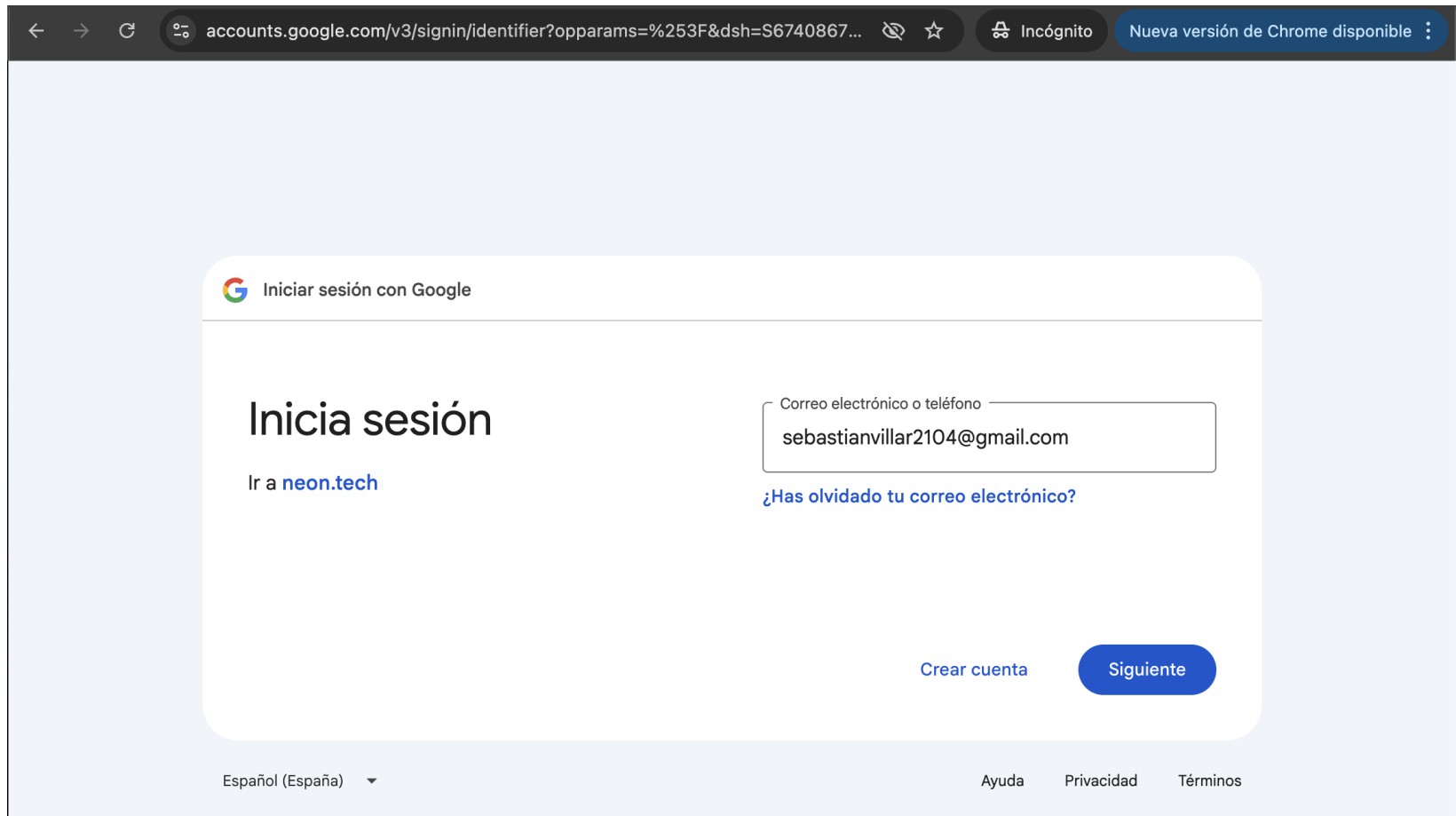
Continue

By creating an account you agree to the [Terms of Service](#) and our [Privacy Policy](#). We'll occasionally send you emails about news, products, and services; you can opt-out anytime.

Already have an account? [Log in](#)

Neon Serverless Postgres — Ship faster

Crear Cuenta:



The screenshot shows a web browser window with the address bar displaying `accounts.google.com/v3/signin/identifier?opparams=%253F&dsh=S6740867...`. The page title is "Iniciar sesión con Google". The main heading is "Inicia sesión". Below it, there is a link "Ir a [neon.tech](#)". To the right, there is a text input field labeled "Correo electrónico o teléfono" containing the email address "sebastianvillar2104@gmail.com". Below the input field is a link "¿Has olvidado tu correo electrónico?". At the bottom right, there are two buttons: "Crear cuenta" and "Siguiente". At the bottom left, there is a language selector "Español (España)". At the bottom right, there are links for "Ayuda", "Privacidad", and "Términos".

← → ↻ accounts.google.com/v3/signin/identifier?opparams=%253F&dsh=S6740867... Incógnito Nueva versión de Chrome disponible

Iniciar sesión con Google

Inicia sesión

Ir a [neon.tech](#)

Correo electrónico o teléfono

sebastianvillar2104@gmail.com

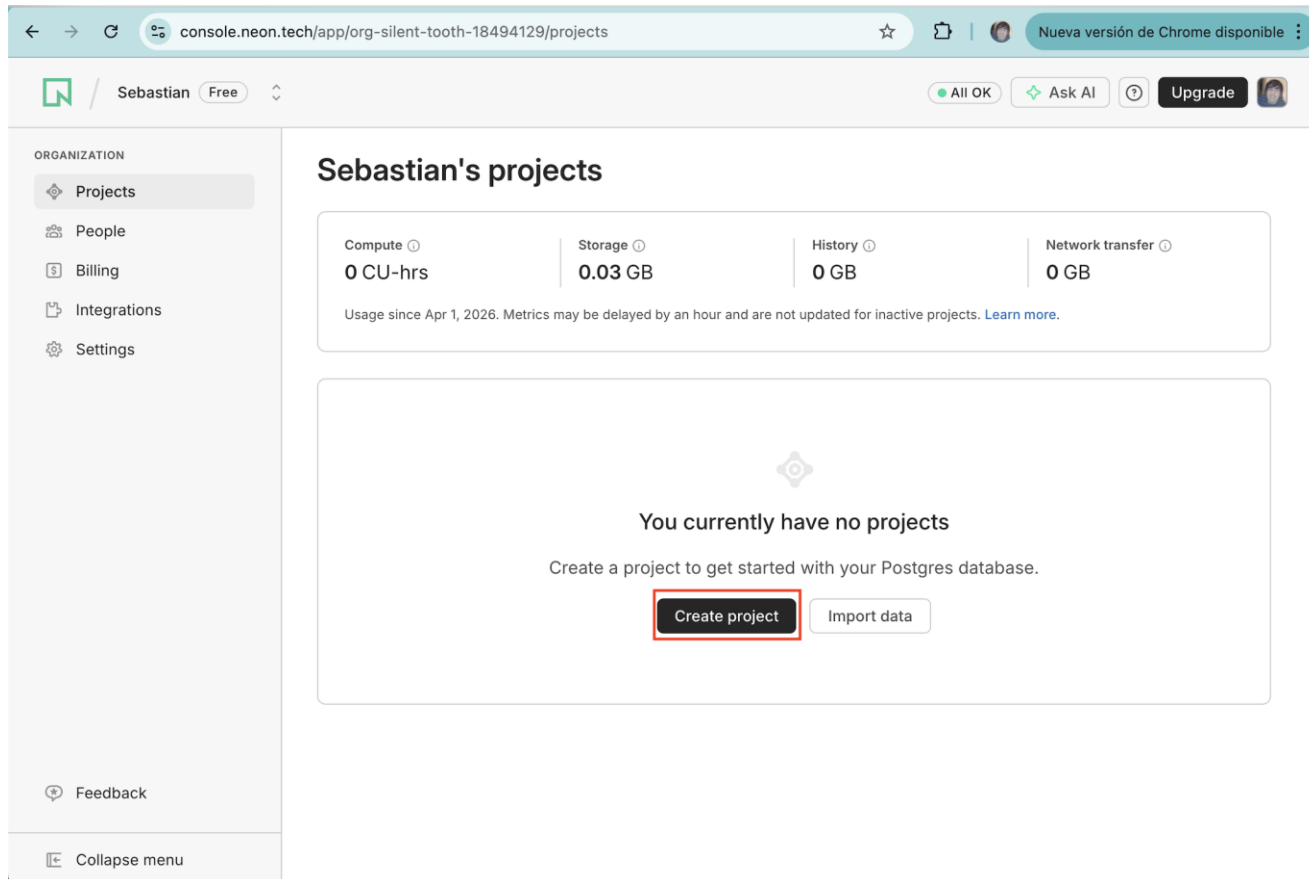
[¿Has olvidado tu correo electrónico?](#)

[Crear cuenta](#) [Siguiente](#)

Español (España) ▼ Ayuda Privacidad Términos

Neon Serverless Postgres — Ship faster

Crear Proyecto:



The screenshot shows the Neon console interface for a user named Sebastian. The left sidebar contains navigation links for Organization, Projects, People, Billing, Integrations, and Settings. The main content area displays 'Sebastian's projects' with a summary of usage: 0 CU-hrs, 0.03 GB Storage, 0 GB History, and 0 GB Network transfer. Below this, a message states 'You currently have no projects' and 'Create a project to get started with your Postgres database.' A red box highlights the 'Create project' button, which is next to an 'Import data' button. The bottom of the sidebar includes 'Feedback' and 'Collapse menu' options.

Sebastian's projects

Compute ⓘ	Storage ⓘ	History ⓘ	Network transfer ⓘ
0 CU-hrs	0.03 GB	0 GB	0 GB

Usage since Apr 1, 2026. Metrics may be delayed by an hour and are not updated for inactive projects. [Learn more.](#)

You currently have no projects

Create a project to get started with your Postgres database.

[Create project](#) [Import data](#)

FIO Neon Serverless Postgres — Ship faster

Crear Proyecto:

Create project

Project name

FIO-UNICEN

Postgres version

17

Region

AWS US East 1 (N. Virginia)

Select the region closest to your application.

Enable Neon Auth

☐ Neon Auth is a managed authentication service that stores users and sessions directly in your database.



Region

AWS US East 1 (N. Virginia)

Your free plan includes:

0.5 GB storage

Scales to zero when inactive

Autoscaling up to 2 CU

10 branches per project

Cancel

Create

Neon Serverless Postgres — Ship faster

Utilizar el Dashboard de Neon:

The screenshot shows the Neon Project dashboard for a project named 'FIO-UNICEN'. The interface includes a top navigation bar with the user 'Sebastian' and account status 'Free'. The left sidebar contains navigation links for 'PROJECT' (Dashboard, Branches, Integrations, Settings) and 'BRANCH' (production). Under 'BRANCH', the 'SQL Editor' link is highlighted with a red box. The main content area is titled 'Project dashboard' and includes buttons for 'Connect', 'Import data', and 'Share'. A section titled 'Get connected to your new database' offers three options: 'Connection string', 'Neon init', and 'IDE extension'. Below this, a metrics section displays 'Branches: 1 / 10', 'Compute: 0.28 / 100 CU-hrs', 'Storage: 0.03 / 0.5 GB', and 'Network transfer: 0 / 5 GB'. The 'Monitoring' section shows the 'production' branch with the compute state set to 'Idle'. A '1 Branch' summary card at the bottom right shows the 'production' branch with a compute state of 'Idle' and a resource allocation of '.25 ↔ 2 CU'.

PROJECT

- Dashboard
- Branches
- Integrations
- Settings

BRANCH

production

- Overview
- Monitoring
- SQL Editor
- Tables
- Backup & Restore
- Data Masking (BETA)

APP BACKEND

- Data API
- Auth

Collapse menu

Project dashboard

Connect Import data Share

Get connected to your new database

- Connection string: Copy your project's connection string and add it to your app config.
- Neon init: Set up your local dev environment with a single command.
- IDE extension: Set up your app with the Neon extension in VS Code or Cursor.

Dismiss

Branches 1 / 10 Compute 0.28 / 100 CU-hrs Storage 0.03 / 0.5 GB Network transfer 0 / 5 GB

Usage since May 1, 2026. Metrics may be delayed by an hour and are not updated for inactive projects. [Learn more.](#)

Monitoring

View all metrics

Branch Compute

p.. Default F Idle Refresh

1 Branch

View all

Name Primary compute

production Default .25 ↔ 2 CU Idle

Preview Workflow

You don't have any preview branches yet. Improve

Neon Serverless Postgres — Ship faster

Utilizar el Dashboard de Neon:

The screenshot displays the Neon console dashboard in a web browser. The URL bar shows the path to the SQL editor for a specific project and branch. The dashboard includes a sidebar with navigation options like Dashboard, Branches, Integrations, and Settings. The main area is titled 'SQL Editor' and shows a 'production' branch. A red box highlights the 'Tables' option in the sidebar. Another red box highlights the 'Run' button in the bottom right corner of the SQL editor. The SQL editor itself contains a message about an archived branch and a large red text area labeled 'Codigo SQL'. Below the editor, a status bar indicates 'Connected (no queries)' and shows a 'Run' button. A final red box highlights the 'Run' button and the resulting output area, which displays 'Resultado de Ejecución Código SQL'.

console.neon.tech/app/projects/square-water-75864072/branches/br-broad-block-ann3i5ki/sql-editor?database=neondb

Sebastian Free / FIO-UNICEN

All OK Ask AI Upgrade

PROJECT

- Dashboard
- Branches
- Integrations
- Settings

BRANCH

production

- Overview
- Monitoring
- SQL Editor
- Tables
- Backup & Restore
- Data Masking BETA

APP BACKEND

- Data API
- Auth

Collapse menu

SQL Editor

production

Saved History

Untitled Save Primary Idle neondb

Archived branch
This branch is archived. Querying will unarchive it.

1 | Type SQL or Generate with AI...

Codigo SQL

Connected (no queries)

Run Explain Analyze 201ms No result

Statement executed successfully

Resultado de Ejecución
Código SQL



TP-Creación de Tablas

spotify		
id	serial	PK
artist	varchar(250)	
track	varchar(250)	
album	varchar(250)	
album_type	varchar(250)	
danceability	double precision	
energy	double precision	
loudness	double precision	
speechiness	double precision	
acousticness	double precision	
instrumentalness	double precision	
liveness	double precision	
valence	double precision	
tiempo	double precision	
title	varchar(250)	
channel	varchar(250)	
views	int	
likes	int	
comments	int	
licensed	boolean	
oficial_video	boolean	
stream	int	
most_playedon	varchar(250)	

- Cada columna debe tener un determinado tipo de dato.
- El tipo de dato limita el conjunto de valores posibles que se pueden asignar a una columna

```
CREATE TABLE spotify (  
  id serial,  
  artist VARCHAR(250),  
  track VARCHAR(250),  
  album VARCHAR(250),  
  album_type VARCHAR(250),  
  danceability DOUBLE PRECISION,  
  energy DOUBLE PRECISION,  
  loudness DOUBLE PRECISION,  
  speechiness DOUBLE PRECISION,  
  acousticness DOUBLE PRECISION,  
  instrumentalness DOUBLE PRECISION,  
  liveness DOUBLE PRECISION,  
  valence DOUBLE PRECISION,  
  tiempo DOUBLE PRECISION,  
  title VARCHAR(250),  
  channel VARCHAR(250),  
  views int,  
  likes int,  
  comments int,  
  licensed boolean,  
  oficial_video boolean,  
  stream int,  
  most_playedon VARCHAR(250),  
  CONSTRAINT PK_SPOTIFY PRIMARY KEY (id)  
);
```

Neon Serverless Postgres — Ship faster

Conexion con desde Python:

The screenshot displays the Neon console interface for a project named 'square-water-75864072'. The left sidebar shows the 'PROJECT' menu with options like Dashboard, Branches, Integrations, and Settings. The 'BRANCH' section is highlighted with a red box, showing the 'production' branch selected. The main content area shows the 'Branch overview' for the 'production' branch. It includes a 'Compute' section with 0.28 CU-hrs, a 'Storage' section with 30.95 MB, a 'Storage delta' section with 29.5 MB, and a 'Network transfer' section with 155.67 kB. Below these, there is a table with columns for ID, Created on, and Created by. The 'Compute' section is highlighted with a red box. The 'Compute' section shows a 'Primary' instance with a 'SUSPENDED' status. The 'Connect' button is highlighted with a red box.

console.neon.tech/app/projects/square-water-75864072/branches/br-broad-block-ann3i5ki

Sebastian Free FIO-UNICEN All OK Ask AI Upgrade

PROJECT

- Dashboard
- Branches
- Integrations
- Settings

BRANCH

- production
- Overview
- Monitoring
- SQL Editor
- Tables
- Backup & Restore
- Data Masking BETA

APP BACKEND

- Data API
- Auth

Branch overview

Create child branch Protect More

production Default

Compute 0.28 CU-hrs Storage 30.95 MB Storage delta 29.5 MB Network transfer 155.67 kB

Usage since May 1, 2026. Metrics may be delayed by an hour and are not updated for inactive projects. [Learn more.](#)

ID br-broad-block-ann3i5ki Created on 2026-04-16 19:49:38 -03:00 Created by Sebastian

Computes Roles & Databases Child branches

Primary SUSPENDED Suspended Size .25 ↔ 2 CU

ep-quiet-rain-angtzukh May 4, 2026 9:35 am

Connect Edit

Read Replicas: Scale your application by offloading your read workload to a read-only instance of your database. Add Read Replica

Neon Serverless Postgres — Ship faster

Path de connexion a Python:

Connect to your database

Branch

production Default

Compute

Primary Idle

Database

neondb

Role

neondb_owner

[Reset password](#)

Connection string

☒ Connection pooling

```
postgresql://neondb_owner:*****@ep-quiet-rain-angtzukh-pooler.c-6.us-east-1.aws
.neon.tech/neondb?sslmode=require&channel_binding=require
```

Copy snippet

Show password

Your password is saved in a secure storage vault. [Learn more about connecting](#)

Base de Datos en Python



Objetivo:

El objetivo es desarrollar una herramienta que permita a una empresa de ingeniería evaluar la consistencia en la calidad de cables de acero a partir de ensayos de resistencia a la tracción.

Base de Datos en Python



Análisis del Problema:

Se necesita procesar datos provenientes desde una **base de datos** de entrada que contiene las mediciones de resistencia de múltiples muestras, organizándolos en una estructura matricial de dos columnas.

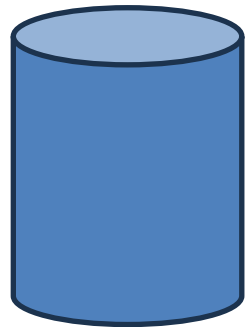
Con estos datos, se necesita calcular las métricas de media, moda y rango.

Finalmente, los resultados **deberán ser mostrados por pantalla**.

Base de Datos en Python



Análisis del Problema:



Base de Datos



Leer Tabla BD

Calcular Media

Calcular Moda

Calcular Rango

Escribir por
Pantalla

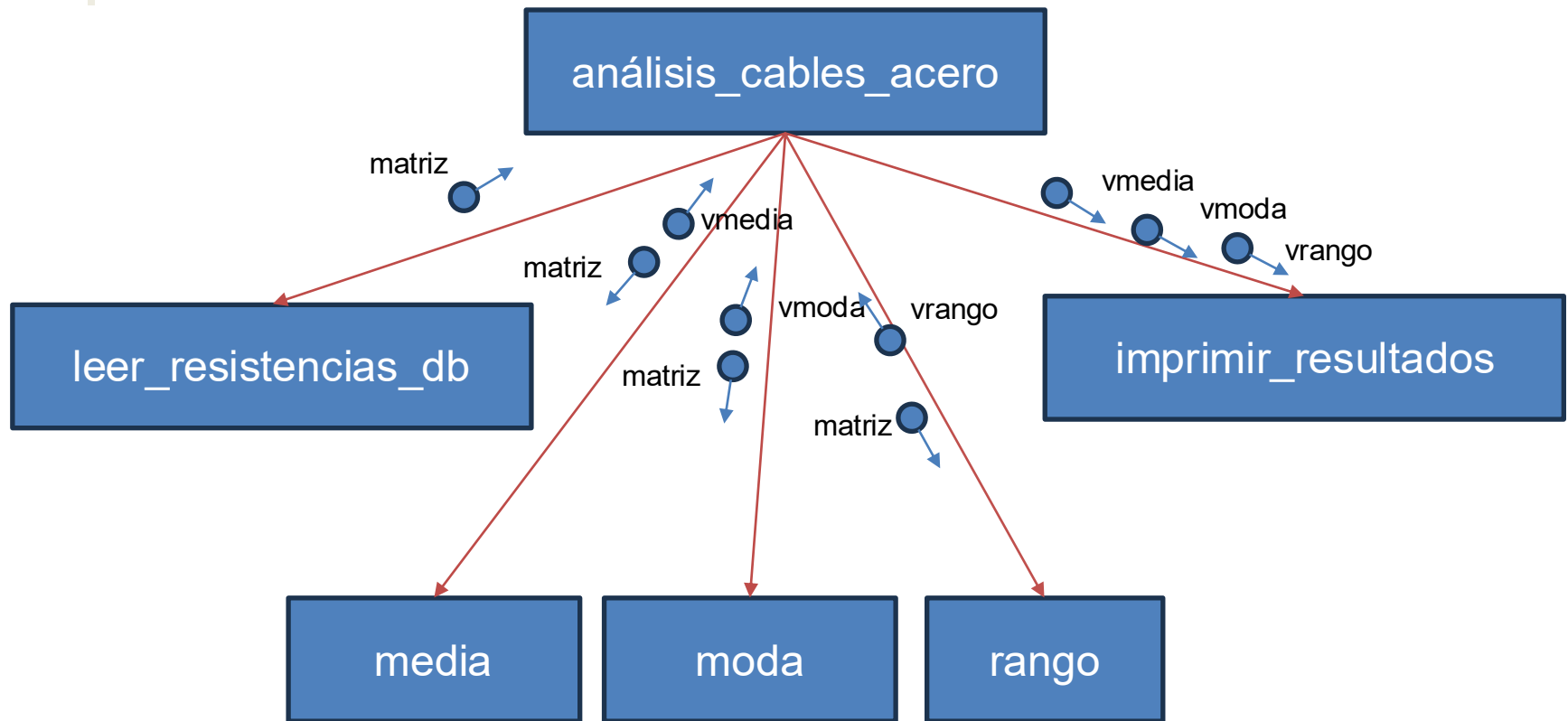


Salida por
Pantalla

Base de Datos en Python



Diseño Problema: Diagrama de Estructura



Base de Datos en Python



Algoritmo Leer BD:

Funcion leer_resistencias_db() : **Entero**

Definir matriz **Como** Real

Definir i, n **Como** Entero

// Simulamos la conexión a la BD

bd = Neon(PATH)

// Simulamos la operación de lectura a la BD

filas = **Leer**(bd, "SELECT id, resistencia FROM DATOS")

Dimension matriz[n,2]

Para i <- 0 **Hasta** n-1 **Hacer**

Leer matriz[i,0] = filas[0]

Leer matriz[i,1] = filas[1]

FinPara

Retornar matriz

FinFuncion

Neon Serverless Postgres — Ship faster

Código Python Requerimiento:

```
pip install psycopg2-binary
```



Base de Datos en Python

```
import psycopg2
import os

DATABASE_URL = "PATH_BASE_DE_DATOS"

def leer_resistencias_db():
    matriz = []

    try:
        conexion_bd = psycopg2.connect(DATABASE_URL)
        bd = conexion_bd.cursor()

        bd.execute("SELECT id, resistencia FROM DATOS")
        filas = bd.fetchall()

        for fila in filas:
            matriz.append([int(fila[0]), float(fila[1])])

        bd.close()
        conexion_bd.close()

    except Exception as e:
        print("Error al leer desde la base de datos:", e)

    return matriz
```

Base de Datos en Python



Resultado:

- **Media:** 464.6666666666667
- **Moda:** 460.0
- **Rango** (max - min): 480.0

Contextualizados:

- **Media:** 464.6 KN
- **Moda:** 460.0 KN
- **Rango** (max - min): 480.0 KN