

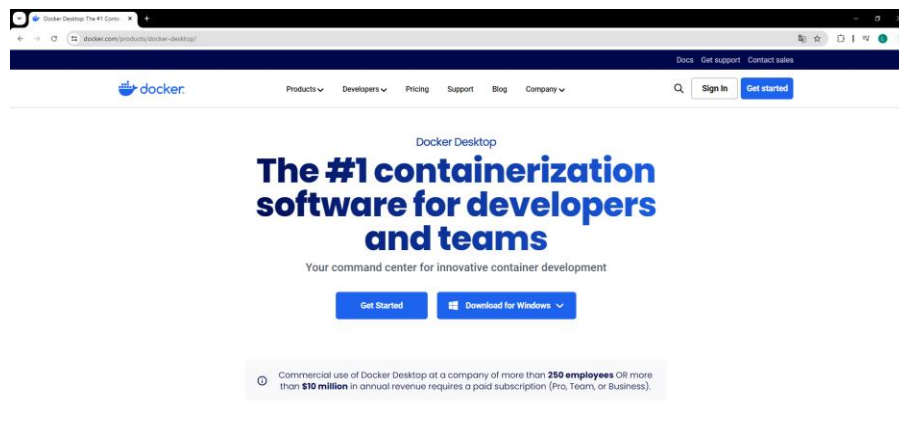
Nombre: Luisa Reyna Beatriz Ayon Muñoz.

Prueba Técnica

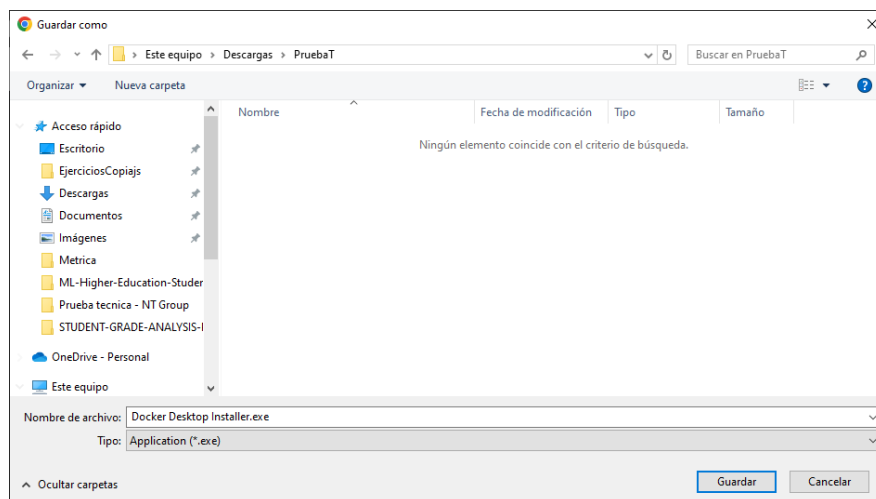
Para llevar a cabo la prueba se instalaron algunas herramientas como Dockers y PostgreSQL. A continuación se observan los pasos para su instalación.

Dockers

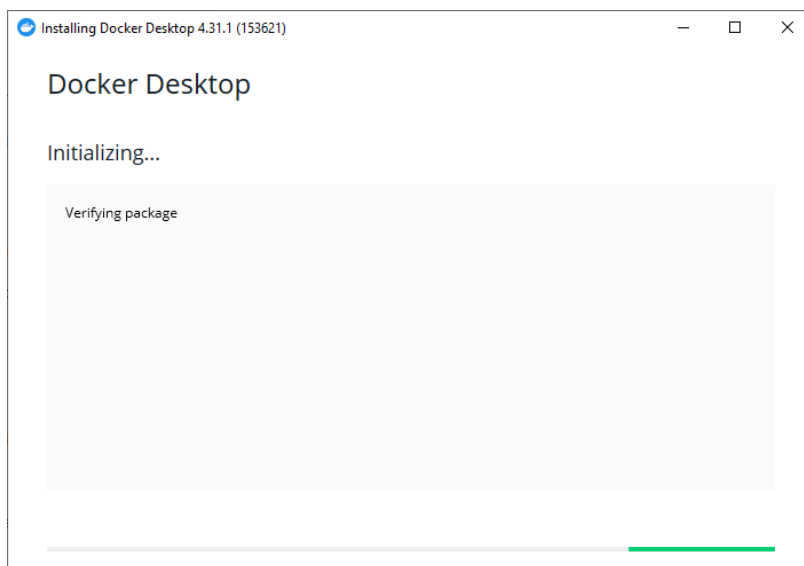
En primera instancia se ingresa a su sitio oficial y se selecciona la opción de descarga acorde al sistema operativo, en este caso Windows 10.



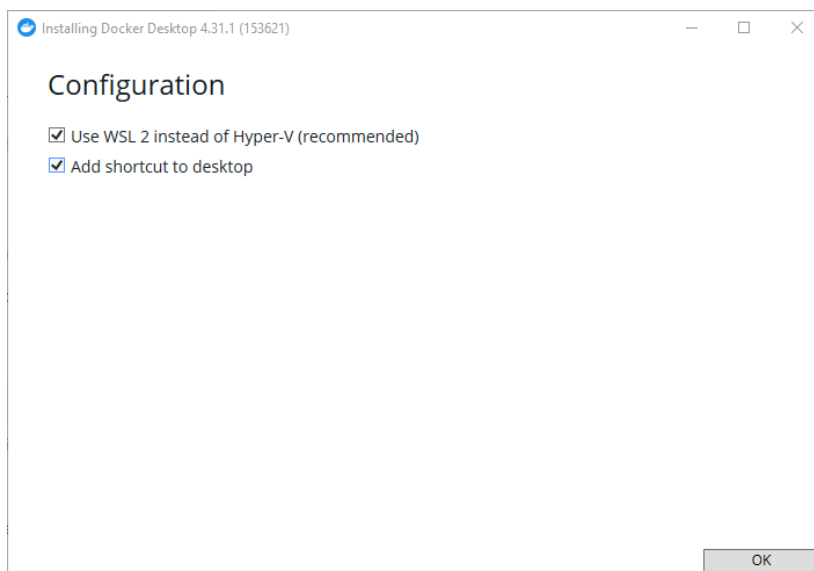
Se guarda el instalador en la carpeta de preferencia.



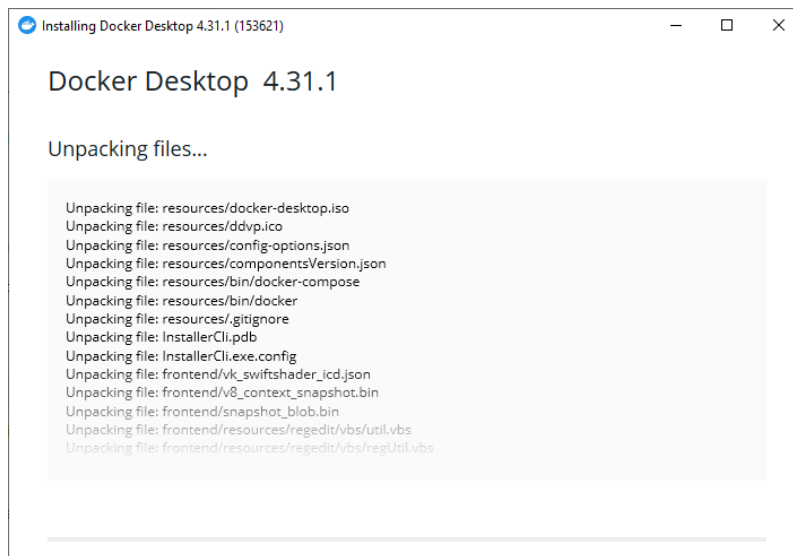
Una vez almacenado, se ejecuta y se inicia la instalación.



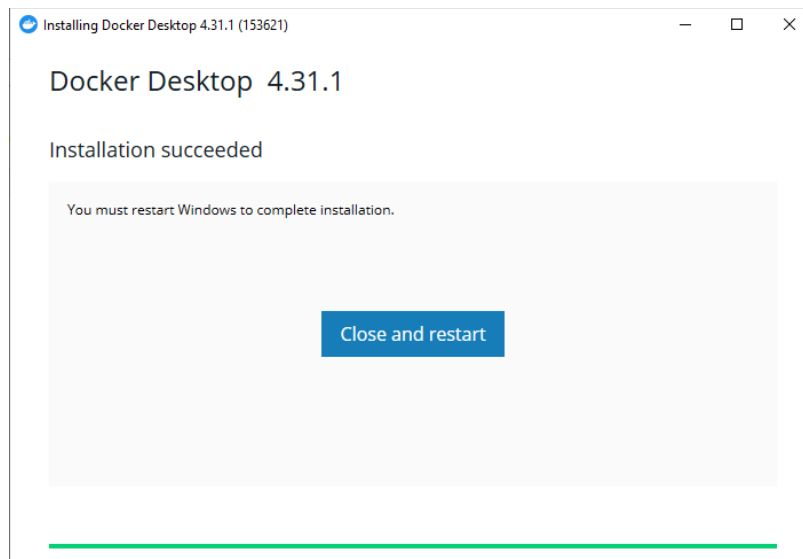
Se acepta la configuración por defecto para la virtualización.



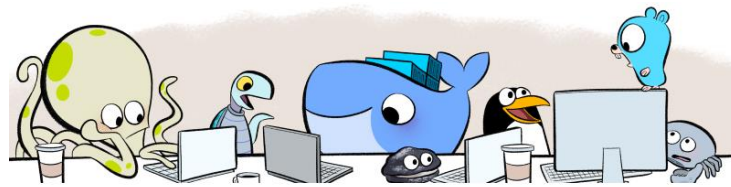
Se instalan los paquetes correspondientes



Se finaliza la instalación y se reinicia el equipo.



Se abre la aplicación y se aceptan los términos y condiciones.



Docker Subscription Service Agreement

By selecting **accept**, you agree to the [Subscription Service Agreement](#), the [Docker Data Processing Agreement](#), and the [Data Privacy Policy](#).

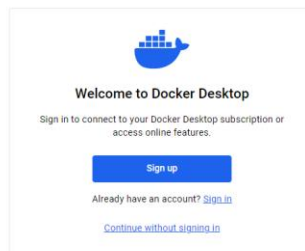
Commercial use of Docker Desktop at a company of more than 250 employees OR more than \$10 million in annual revenue requires a paid subscription (Pro, Team, or Business). [See subscription details](#)

[View Full Terms](#)

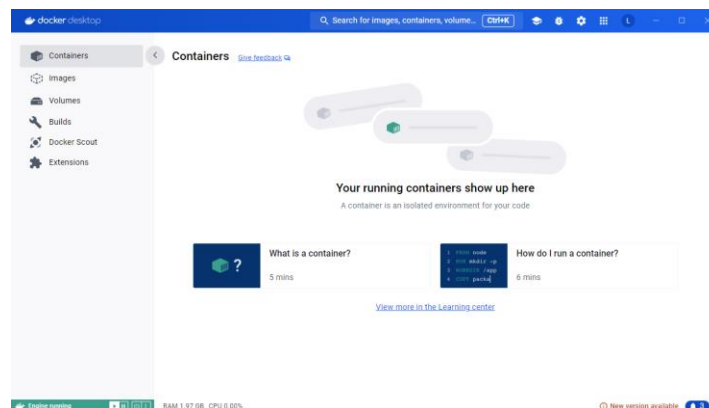
[Accept](#)

[Close](#)

Posteriormente se solicita el inicio de sesión o la indicación de acceder como invitado. Esto se efectúa según las preferencias de cada uno.



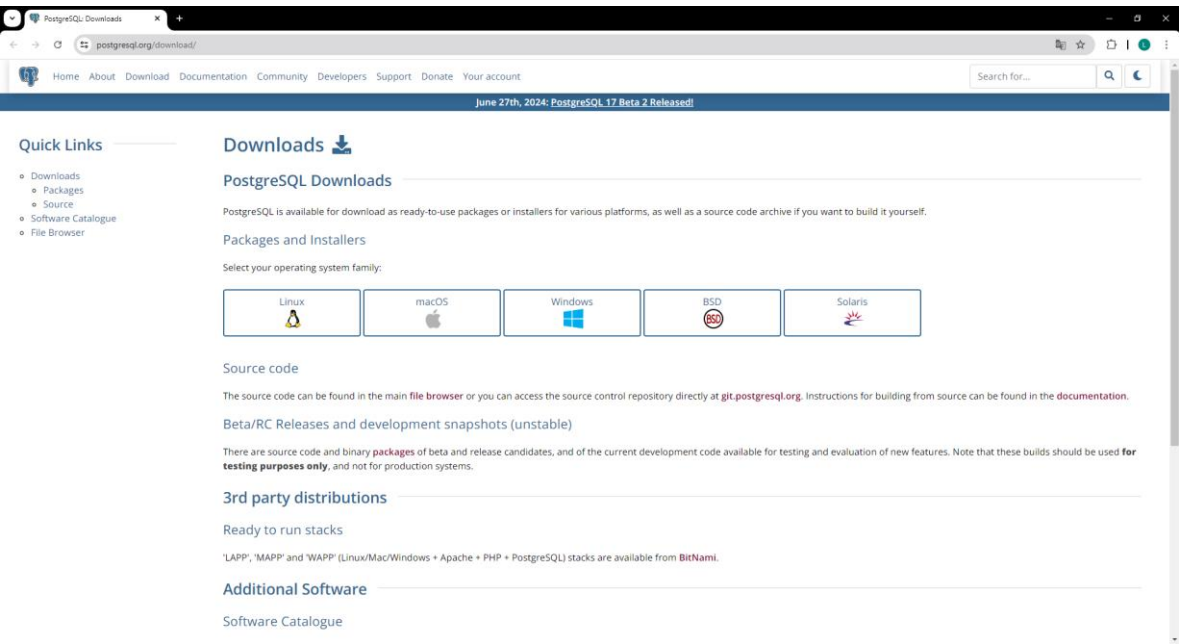
Finalmente se aprecia la pantalla de inicio.



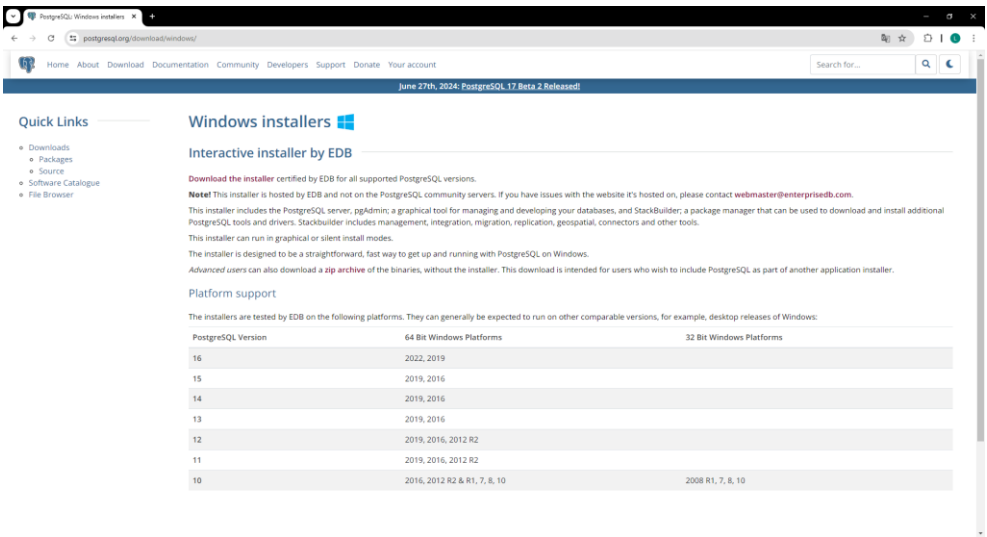
PostgreSQL.

Esta base de datos fue seleccionada debido a su capacidad para tratar con consultas complejas y grandes cantidades de datos, considerando que se trata de datos pertenecientes a una compañía grande que guarda relación con otras más; a continuación se muestran los pasos realizados para su instalación.

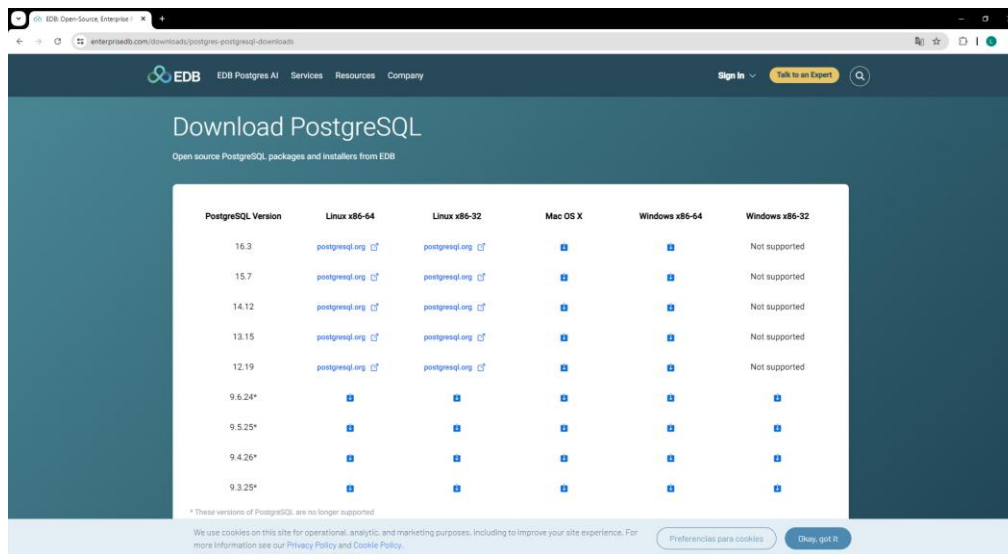
Se accede al sitio web oficial para la descarga acorde al sistema operativo.



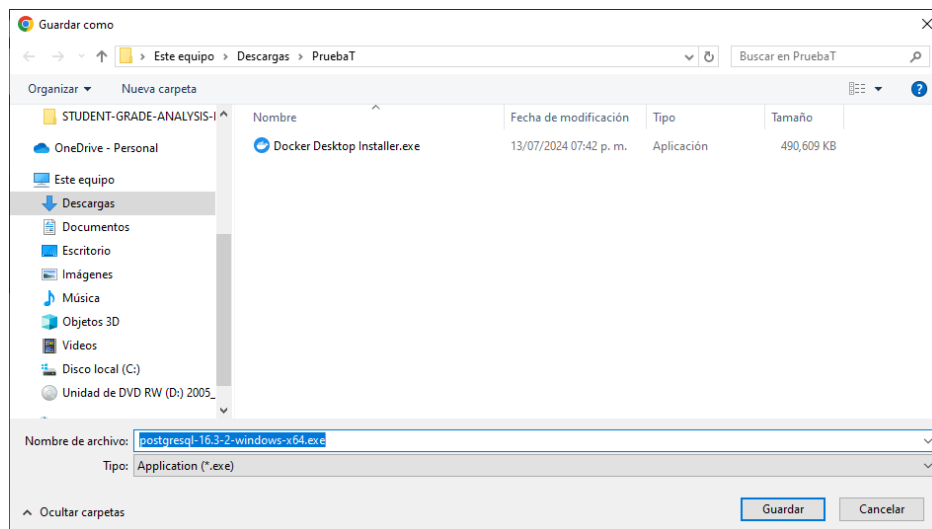
Se selecciona la versión más reciente.



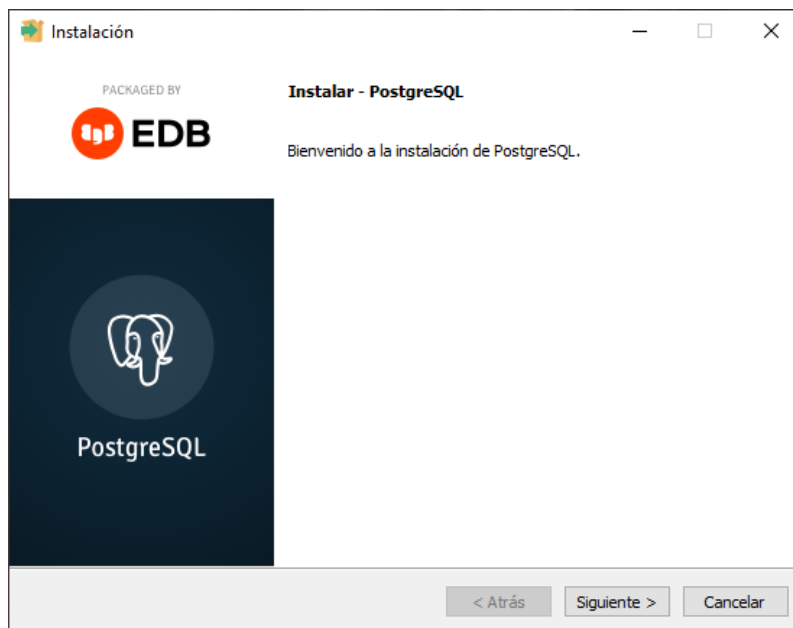
Se inicia el proceso de descarga.



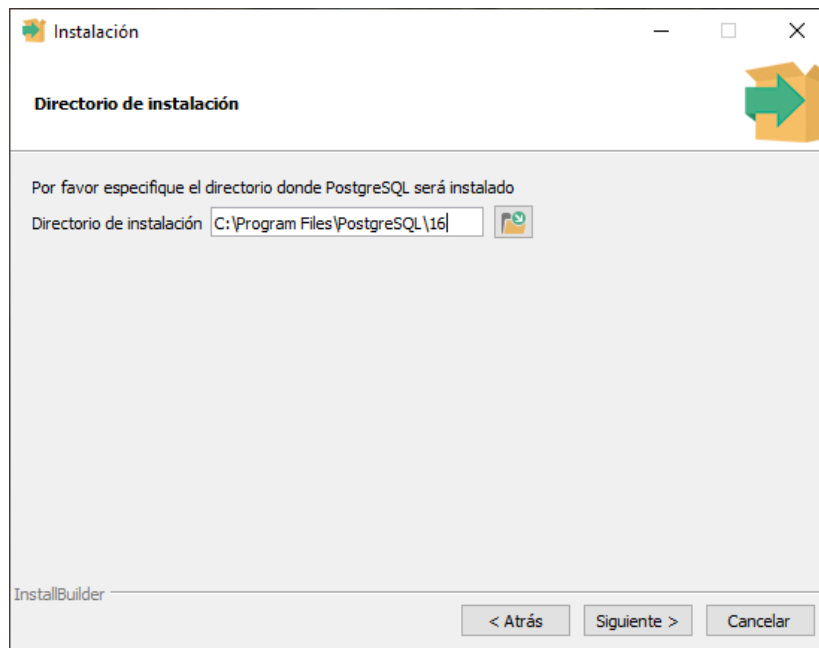
Se almacena el ejecutable en la carpeta de preferencia.



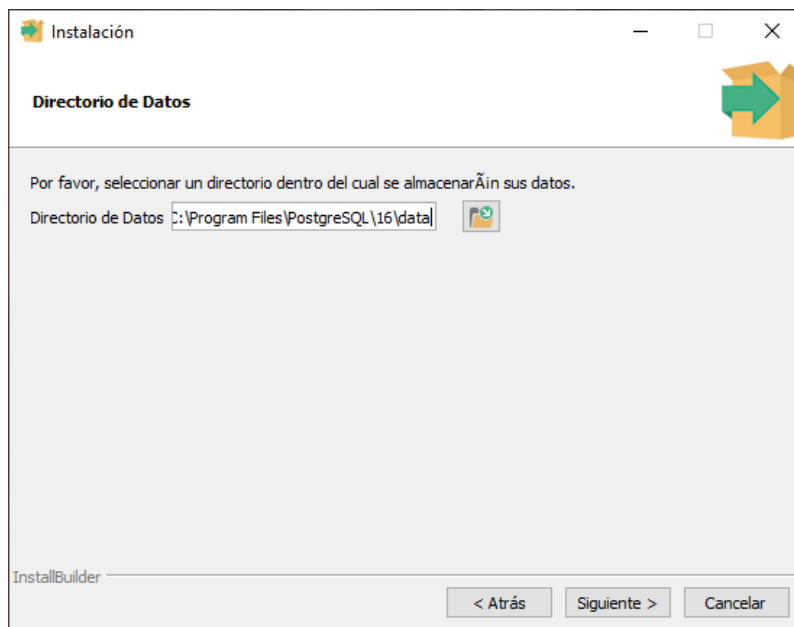
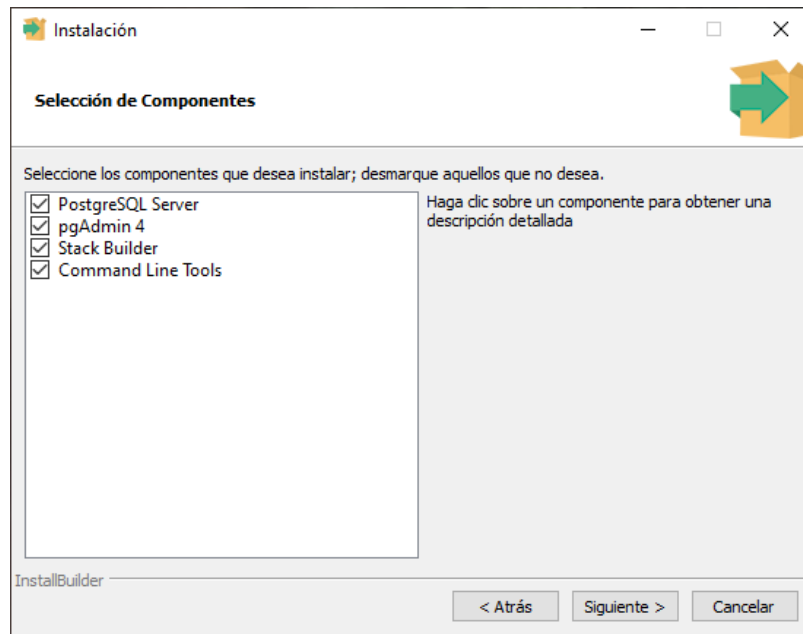
Una vez descargado, ejecuta y se comienza el proceso de instalación.



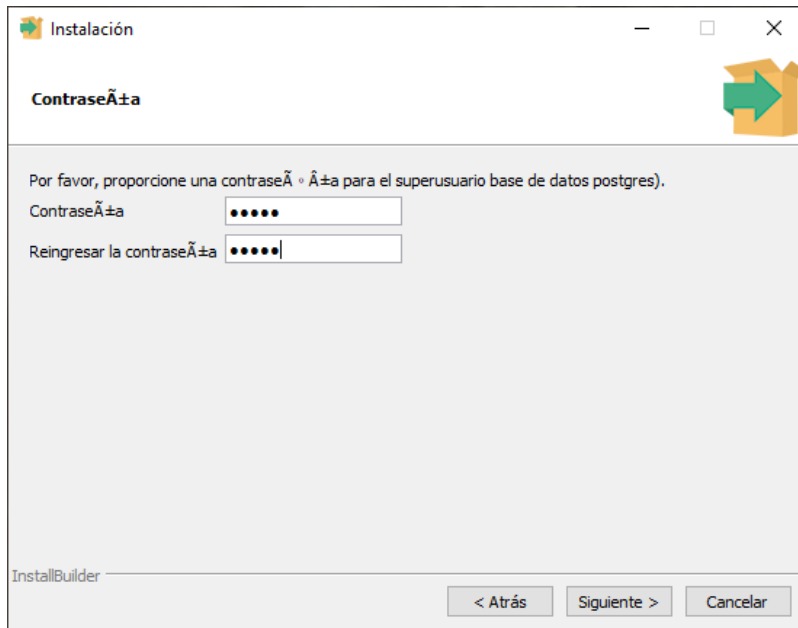
Se selecciona la carpeta donde se almacenarán los archivos.



Se instalan el resto de los componentes.

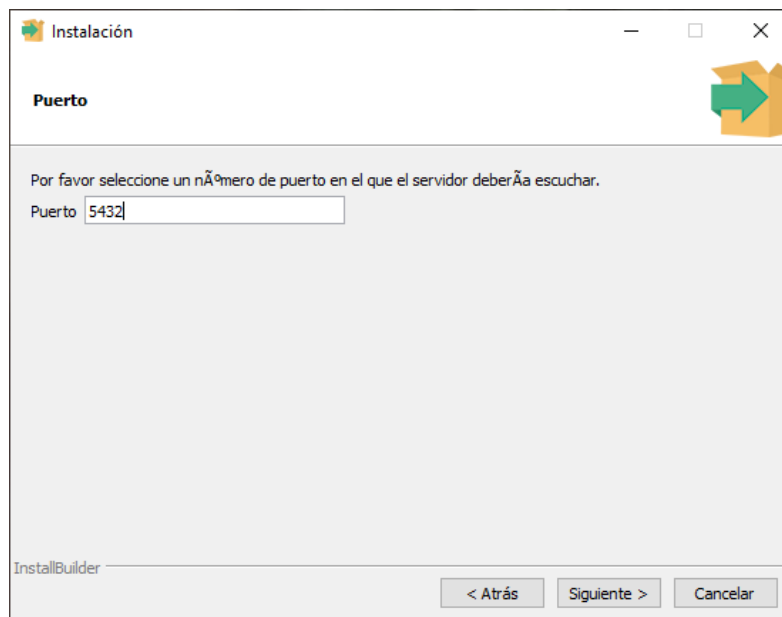


Se establece una contraseña para el usuario root.



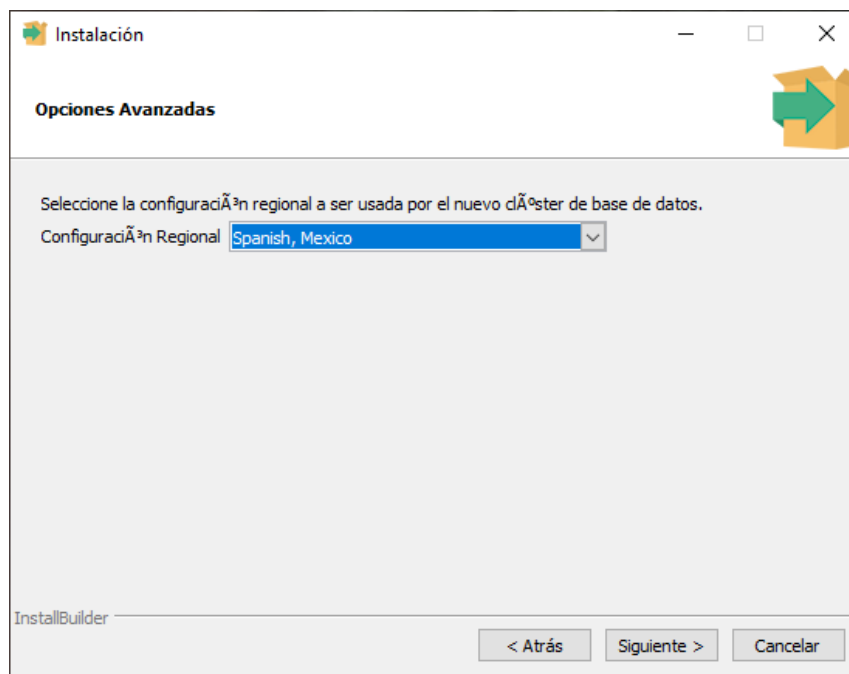
The screenshot shows a window titled 'Instalación' (Installation) with a green arrow icon in the top right corner. The main heading is 'Contraseña' (Password). Below it, a message reads: 'Por favor, proporcione una contraseña para el superusuario base de datos postgres.' (Please provide a password for the postgres database superuser). There are two input fields: 'Contraseña' (Password) and 'Reingresar la contraseña' (Re-enter password), both masked with dots. At the bottom, there are three buttons: '< Atrás' (Back), 'Siguiente >' (Next), and 'Cancelar' (Cancel). The 'InstallBuilder' logo is visible in the bottom left corner.

Se selecciona el puerto en el servidor. Por defecto es el 5432

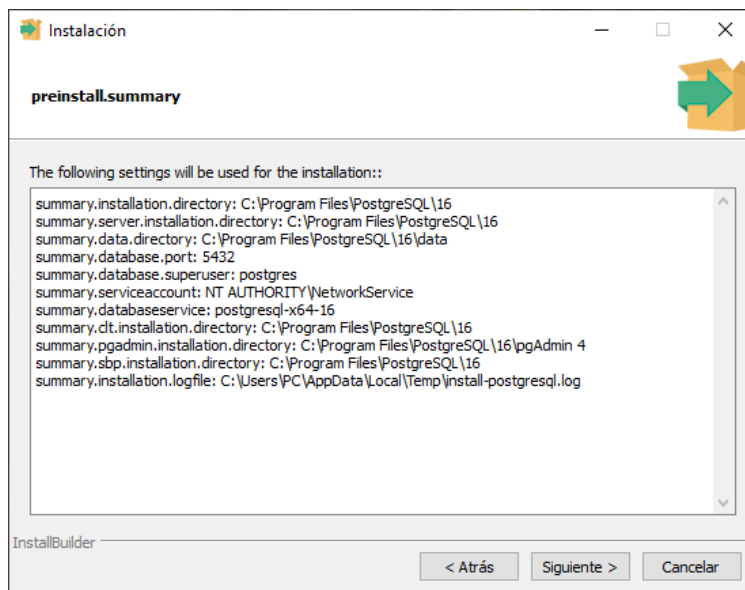


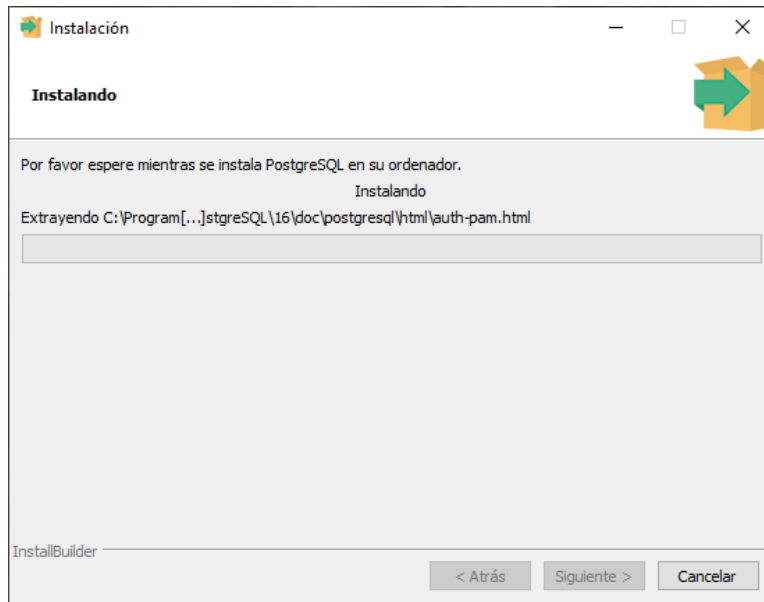
The screenshot shows a window titled 'Instalación' (Installation) with a green arrow icon in the top right corner. The main heading is 'Puerto' (Port). Below it, a message reads: 'Por favor seleccione un número de puerto en el que el servidor deberá escuchar.' (Please select a port number on which the server should listen). There is a single input field labeled 'Puerto' (Port) containing the value '5432'. At the bottom, there are three buttons: '< Atrás' (Back), 'Siguiente >' (Next), and 'Cancelar' (Cancel). The 'InstallBuilder' logo is visible in the bottom left corner.

Se selecciona el idioma de preferencia.



Se observa la instalaci n de los componentes.



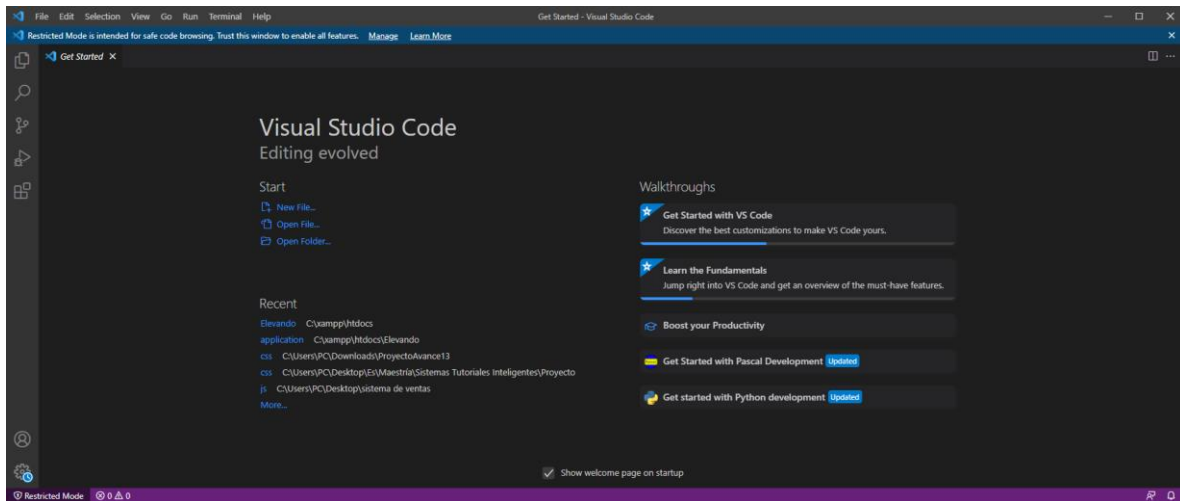


Se concluye la instalación y se procede a cerrar las ventanas.



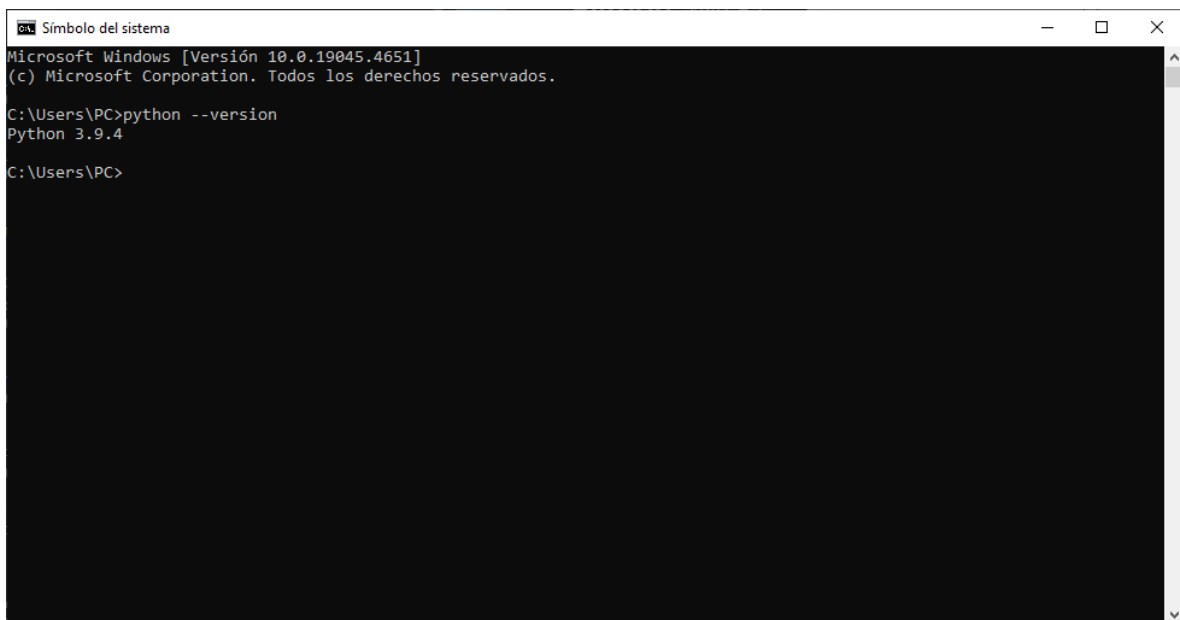
Visual Studio Code

Este editor de texto ya estaba instalado en el equipo de cómputo correspondiente, por lo cual solo se muestra su ventana principal.



Python

Así también se observa que ya estaba instalada la versión 3.9.4 de Python



Una vez instaladas las herramientas, se inició la resolución del problema.

1.1 Carga de información.

El procedimiento inició con la creación de un contenedor de PostgreSQL con Docker.

```
Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

Prueba la nueva tecnología PowerShell multiplataforma https://aka.ms/pscore6

PS C:\Users\PC> docker run --name compras_c -e POSTGRES_PASSWORD=12345 -d postgres
Unable to find image 'postgres:latest' locally
latest: Pulling from library/postgres
f11c1adaa26e:
76ce212b9153: Download complete
919ca406a058: Download complete
6b7a1245fe71: Download complete
8064ffe06c65: Downloading 4.918MB/8.066MB
4b5c59f2d82c: Download complete
fe72764b9070: Download complete
6ef8e2c0f4d9: Verifying Checksum
```

Posteriormente se eliminaron las filas en blanco contenidas dentro del archivo .CSV

id	name	company_id	amount	status	created_at	paid_at
1	48b4b0d96c	MIPassajery	cbf1db09cd	3	voided	19/03/2019
2	09d685c0d2	MIPassajery	cbf1db09cd	3	pending_pay	06/05/2019
3	2cdc0231c1	MIPassajery	cbf1db09cd	3	voided	22/02/2019
4	91833ba319	MIPassajery	cbf1db09cd	102.61	paid	27/02/2019 27/02/2019
5	6cc0b2447f	MIPassajery	cbf1db09cd	184.49	paid	05/02/2019 05/02/2019
6	b250e015c4	MIPassajery	cbf1db09cd	85.27	pending_pay	04/01/2019
7	4f5db7096d	MIPassajery	cbf1db09cd	41	pre_authoriz	04/05/2019
8	7e06211c7d	MIPassajery	cbf1db09cd	3	pending_pay	23/01/2019
9	6ed51d9b9a	MIPassajery	cbf1db09cd	41.61	voided	03/05/2019
10	cd1a7a6fcd	MIPassajery	cbf1db09cd	115.84	paid	22/01/2019 22/01/2019
11	508302c2b0	MIPassajery	cbf1db09cd	3	voided	02/03/2019
12	90820a39a6	MIPassajery	cbf1db09cd	81.33	paid	12/01/2019 12/01/2019
13	3159b708b3	MIPassajery	cbf1db09cd	230.27	paid	29/03/2019 29/03/2019
14	796bc10670	MIPassajery	cbf1db09cd	117.88	paid	18/05/2019 18/05/2019
15	221b7a7b37	MIPassajery	cbf1db09cd	73.76	voided	10/01/2019
16	0804a2a96c	MIPassajery	cbf1db09cd	364.46	pending_pay	11/01/2019
17	78bc783831	MIPassajery	cbf1db09cd	3	pending_pay	16/02/2019
18	32149975c2	MIPassajery	cbf1db09cd	64.36	paid	16/04/2019 16/04/2019
19	796166a4cd	MIPassajery	cbf1db09cd	196.65	pending_pay	10/05/2019
20	1a261c012d	MIPassajery	cbf1db09cd	42.2	voided	03/01/2019
21	977828bbac	MIPassajery	cbf1db09cd	41.82	paid	04/03/2019 04/03/2019
22	0dc0040d08	MIPassajery	cbf1db09cd	163.49	paid	03/04/2019 03/04/2019
23						
24						
25	6a529908c5	MIPassajery	cbf1db09cd	186.7	paid	10/05/2019 10/05/2019
26						
27	ab2beca2b1	MIPassajery	cbf1db09cd	59.75	voided	15/01/2019
28						
29	2908a5c0d1	MIPassajery	cbf1db09cd	102.98	paid	02/01/2019 02/01/2019
30						
31	ba79778857c	MIPassajery	cbf1db09cd	158.33	paid	11/04/2019 11/04/2019
32						
33	721572047a	MIPassajery	cbf1db09cd	152.34	pending_pay	01/03/2019
34						
35	8a30aaeb7b	MIPassajery	cbf1db09cd	51.44	paid	01/03/2019 01/03/2019
36						
37	fd4c8b317e	MIPassajery	cbf1db09cd	49.47	paid	02/01/2019 02/01/2019

Como siguiente paso, se copio el archivo que se encontraba en el equipo local, hacia el contenedor.

```
PS C:\Users\PC> docker cp "C:\Users\PC\Downloads\Prueba tecnica - NT Group\data_prueba_tecnica.csv" compras_c:/data_prueba_tecnica.csv
Successfully copied 1.27MB to compras_c:/data_prueba_tecnica.csv
PS C:\Users\PC> 
```

Después se accedió al contenedor de PostgreSQL

```
PS C:\Users\PC> docker cp "C:\Users\PC\Downloads\Prueba tecnica - NT Group\data_prueba_tecnica.csv" compras_c:/data_prueba_tecnica.csv
Successfully copied 1.27MB to compras_c:/data_prueba_tecnica.csv
PS C:\Users\PC> docker exec -it compras_c bash
root@564189f2dfaa:/# 
```

```
root@564189f2dfaa:/# psql -U postgres
psql (16.3 (Debian 16.3-1.pgdg120+1))
Type "help" for help.

postgres=# 
```

Se creó la tabla para cargar los datos según lo observable en el archivo .csv, siguiendo el esquema siguiente:

	Com pany _buy
id	varchar(40) NOT NULL
company_name	varchar(20) NULL

company_id varchar(40) NOT NULL

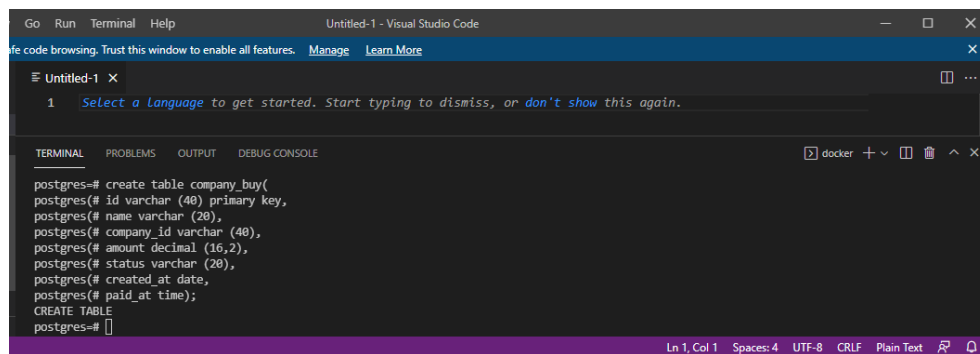
amount varchar (40) NOT NULL

status varchar(20) NOT NULL

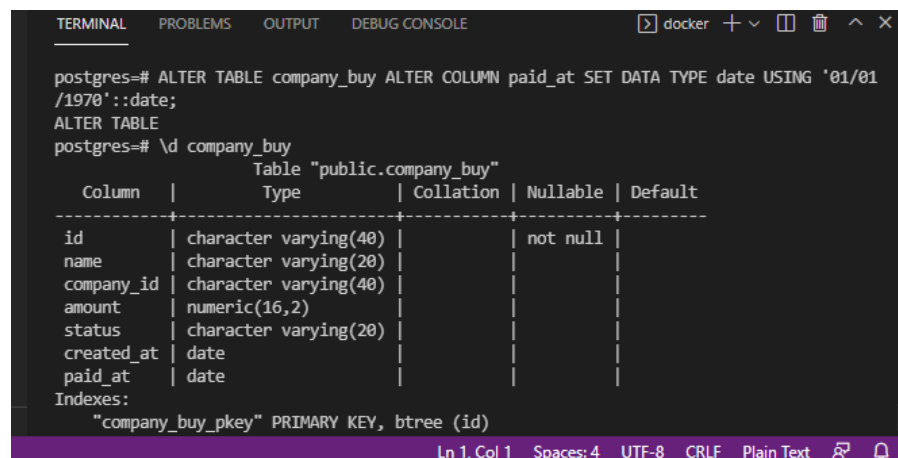
created_at date NOT NULL

updated_at date NULL

Durante la creación de la tabla hubo atributos que cambiaron hasta llegar a definirse como en el esquema antes expuesto. Estas modificaciones se aprecian en las capturas de pantalla.

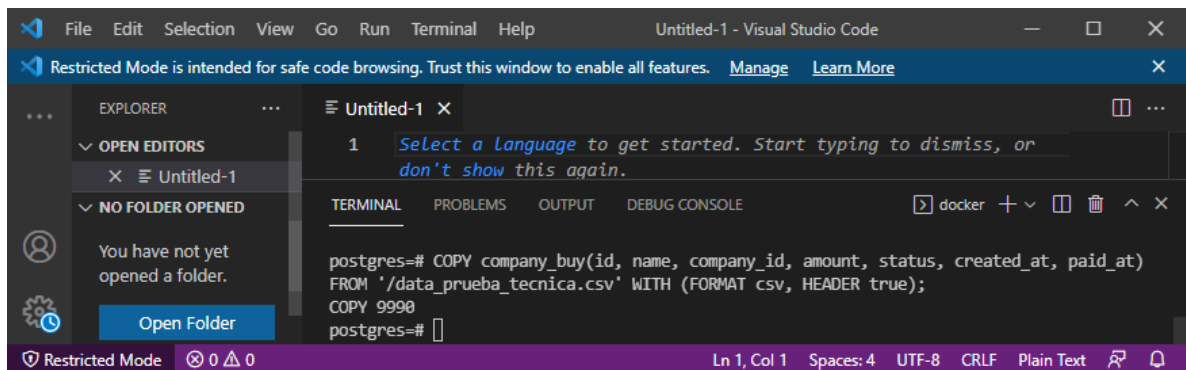


```
Go Run Terminal Help
Untitled-1 - Visual Studio Code
Life code browsing. Trust this window to enable all features. Manage Learn More
Untitled-1 x
1 Select a Language to get started. Start typing to dismiss, or don't show this again.
TERMINAL PROBLEMS OUTPUT DEBUG CONSOLE
postgres=# create table company_buy(
postgres# id varchar (40) primary key,
postgres# name varchar (20),
postgres# company_id varchar (40),
postgres# amount decimal (16,2),
postgres# status varchar (20),
postgres# created_at date,
postgres# paid_at time);
CREATE TABLE
postgres=#
```



```
TERMINAL PROBLEMS OUTPUT DEBUG CONSOLE
postgres=# ALTER TABLE company_buy ALTER COLUMN paid_at SET DATA TYPE date USING '01/01
/1970'::date;
ALTER TABLE
postgres=# \d company_buy
Table "public.company_buy"
Column | Type | Collation | Nullable | Default
-----+-----+-----+-----+-----
id | character varying(40) | | not null |
name | character varying(20) | | |
company_id | character varying(40) | | |
amount | numeric(16,2) | | |
status | character varying(20) | | |
created_at | date | | |
paid_at | date | | |
Indexes:
"company_buy_pkey" PRIMARY KEY, btree (id)
```

Finalmente, se copio lo contenido en el archivo a la tabla de PostgreSQL diseñada, con el comando que se observa a continuación.



Durante esta fase, aparecieron algunos problemas respecto a los tipos de dato en PostgreSQL y los contenidos en el archivo, no obstante, se trató de subsanar buscando que se adaptara a las necesidades del problema.

1.2 Extracción

Para ello se utilizaron dos Dockers, uno que se emplea para la conexión a la base de datos y fue creado en la fase anterior, y el segundo que alberga el script diseñado. Además, para lograr una ejecución conjunta, se elaboró una red a la que se conectaban ambos. En las capturas de pantalla se observan los dos procedimientos:

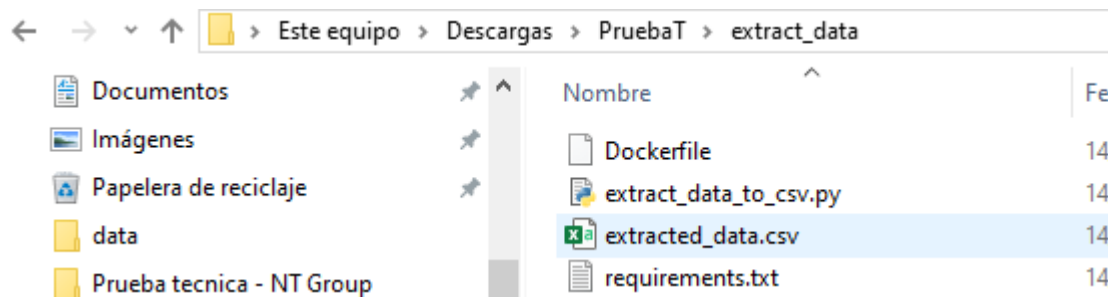
```
PS C:\Users\PC\Downloads\PruebaI\extract_data> docker build -t python_extract .
-> [internal] load build definition from Dockerfile                                0.5s
-> => transferring dockerfile: 327B                                              0.1s
-> [internal] load metadata for docker.io/library/python:3.9-slim                1.4s
-> [auth] library/python:pull token for registry-1.docker.io                    0.0s
-> [internal] load .dockerignore                                                  0.3s
-> => transferring context: 2B                                                    0.0s
-> [1/5] FROM docker.io/library/python:3.9-slim@sha256:a6c12ec09f13df9d4b8b4e4d08678c1b212d89885be14b6c72b63bee2a520f4 0.0s
-> [internal] load build context                                                  0.3s
-> => transferring context: 1.16kB                                                0.1s
-> CACHED [2/5] WORKDIR /app                                                      0.0s
-> CACHED [3/5] COPY requirements.txt .                                           0.0s
-> [4/5] COPY extract_data_to_csv.py .                                           1.2s
-> [5/5] RUN pip install pandas psycpg2-binary                                  29.2s
-> exporting to image                                                            3.7s
-> => exporting layers                                                            2.9s
-> => writing image sha256:a9f8391e88bdf986ddd43aab23c96d8555c37a0f96b683b7aa7f4efba9f11b7 0.1s
-> => naming to docker.io/library/python_extract                                0.2s

PS C:\Users\PC\Downloads\PruebaI\extract_data> docker network create network_prueba
78ae3d8875fa36e61531d1c1116d7679c98beeb182e71db52375a3326025f133
```



```
PS C:\Users\PC\Downloads\PruebaT\extract_data> docker network connect network_prueba_compras_c
docker build -t python_extract .
=> [internal] load build definition from Dockerfile                                0.3s
=> => transferring dockerfile: 327B                                              0.0s
=> [auth] library/python:pull token for registry-1.docker.io                    0.0s
=> => transferring context: 2B                                                  0.0s
=> [internal] load build context                                                0.2s
=> => transferring context: 1.16kB                                             0.0s
=> CACHED [3/5] COPY requirements.txt .                                         0.0s
=> [4/5] COPY extract_data_to_csv.py .                                         1.3s
=> [5/5] RUN pip install pandas psycpg2-binary                               29.5s
=> exporting to image                                                         4.1s
=> => exporting layers                                                         3.2s
=> => writing image sha256:9718c90c051d815ce103e0a4dfecf30a4b7701ef2b89fb237b344d8fd6e91233 0.1s
=> => naming to docker.io/library/python_extract                             0.1s
```

En esta etapa, los problemas surgieron principalmente con el manejo de las rutas, el uso de contenedores por separado y la conexión a la base de datos, no obstante, se logró superar los obstáculos y continuar con el procedimiento. Al correr el Docker, se obtuvo el archivo con los datos extraídos.



1.3 Transformación

En esta fase se solicitaba trasladar los datos al esquema propuesto:

Carg o
id varchar(24) NOT NULL
company_name varchar(130) NULL
company_id varchar(24) NOT NULL
amount decimal(16,2) NOT NULL
status marchar(30) NOT NULL
created_at timestamp NOT NULL

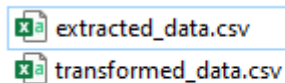
updated_at timestamp NULL

Para ello, se construyó un Docker que almacenaría el script correspondiente.

```
TERMINAL  PROBLEMS 1  OUTPUT  DEBUG CONSOLE  powershell + -  [ ] [ ] [ ]
PS C:\Users\PC\Downloads\PruebaT\transform_data> docker build -t python_transform .
[+] Building 43.7s (11/11) FINISHED
=> [internal] load build definition from Dockerfile                                0.5s
=> => transferring dockerfile: 310B                                              0.1s
=> [internal] load metadata for docker.io/library/python:3.9-slim                2.6s
=> [auth] library/python:pull token for registry-1.docker.io                    0.0s
=> [internal] load .dockerignore                                                  0.5s
=> => transferring context: 2B                                                    0.0s
=> [1/5] FROM docker.io/library/python:3.9-slim@sha256:a6c12ec09f13df9d4b8b4e4d08678c1b212d89885be14b6c72b633bee2a520f4  0.0s
=> [internal] load build context                                                  0.5s
=> => transferring context: 1.12kB                                                0.1s
=> CACHED [2/5] WORKDIR /app                                                    0.0s
=> [3/5] COPY requirements.txt .                                                  1.6s
=> [5/5] RUN pip install pandas                                                  29.2s
=> => exporting layers                                                            3.0s
=> => writing image sha256:ff246afe08d77ae483e883159374fe56f328bd35cdb8833abd6cdb82124cec1c  0.1s
=> => naming to docker.io/library/python_transform                             0.3s

TERMINAL  PROBLEMS 1  OUTPUT  DEBUG CONSOLE  powershell + -  [ ] [ ] [ ]
PS C:\Users\PC\Downloads\PruebaT\transform_data> docker run --rm -v C:/Users/PC/Downloads/PruebaT/transform_data/data:/data python_transform
Los datos se han transformado y guardado en '/data/transformed_data.csv'.
PS C:\Users\PC\Downloads\PruebaT\transform_data> [ ]
```

Como se observa a continuación, el nuevo archivo.csv se almacenó en el equipo local dentro de la misma carpeta que el anterior.



Durante esta fase, hubo algunos inconvenientes con el manejo de las rutas, ya que cada Docker requería la especificación para montar un disco externo (el equipo local). Sin embargo, fueron corregidos y se logró el objetivo.

También se observaron problemas al limitar a un número mucho menor de caracteres cada id, pues éstos podían ser truncados. Para mantener la integridad de los datos, se optó por dejar en un valor de 40, la longitud para cada identificador; así también, en el caso del campo amount, se mantuvo como una cadena en caso de exceder valores que no podían ser gestionados por PostgreSQL

1.4 Dispersión de la información

Para esta fase se creó un nuevo Docker para la base de datos en PostgreSQL, la cual, almacenaría las dos tablas propuestas. En las capturas de pantalla se aprecia su

```
TERMINAL  PROBLEMS 1 OUTPUT  DEBUG CONSOLE  powershell + v [ ] [ ]
PS C:\Users\PC\Downloads\PruebaT\extract_data> cd ..
PS C:\Users\PC\Downloads\PruebaT> docker run --name postgres_db -e POSTGRES_USER=user1 -e POSTGRES_PASSWORD=2468 -e POSTGRES_DB=transactions_d
b -p 5432:5432 -d postgres
22680e91a5bc399d5615a81fa374f992f246f48342f99c02fb8be33beec6b654
PS C:\Users\PC\Downloads\PruebaT> [ ]
```

```
TERMINAL  PROBLEMS 1 OUTPUT  DEBUG CONSOLE  docker + v [ ] [ ]
PS C:\Users\PC\Downloads\PruebaT\extract_data> cd ..
PS C:\Users\PC\Downloads\PruebaT> docker run --name postgres_db -e POSTGRES_USER=user1 -e POSTGRES_PASSWORD=2468 -e POSTGRES_DB=transactions_d
b -p 5432:5432 -d postgres
22680e91a5bc399d5615a81fa374f992f246f48342f99c02fb8be33beec6b654
PS C:\Users\PC\Downloads\PruebaT> docker exec -it postgres_db psql -U user1 -d transactions_db
psql (16.3 (Debian 16.3-1.pgdg120+1))
Type "help" for help.

transactions_db=# [ ]
```

Como siguiente paso se elaboraron las tablas con las consideraciones adaptadas para salvaguardar la integridad de los datos.

```
transactions_db=# CREATE TABLE companies (
transactions_db(#      company_id VARCHAR(40) NOT NULL,
transactions_db(#      company_name VARCHAR(130) NULL,
transactions_db(#      PRIMARY KEY (company_id)
transactions_db(# )
transactions_db-# ;
CREATE TABLE
```

```
transactions_db=# CREATE TABLE charges (
transactions_db(#      id VARCHAR(40) NOT NULL,
transactions_db(#      company_id VARCHAR(40) NOT NULL,
transactions_db(#      amount VARCHAR(255) NOT NULL,
transactions_db(#      status VARCHAR(30) NOT NULL,
transactions_db(#      created_at TIMESTAMP NOT NULL,
transactions_db(#      updated_at TIMESTAMP NULL,
transactions_db(#      PRIMARY KEY (id),
transactions_db(#      FOREIGN KEY (company_id) REFERENCES companies(company_id)
transactions_db(# );
CREATE TABLE
transactions_db=# [ ]
```

Posteriormente se creo un Docker para almacenar el script en Python que permitiría trasladar los datos del archivo a la base de datos.

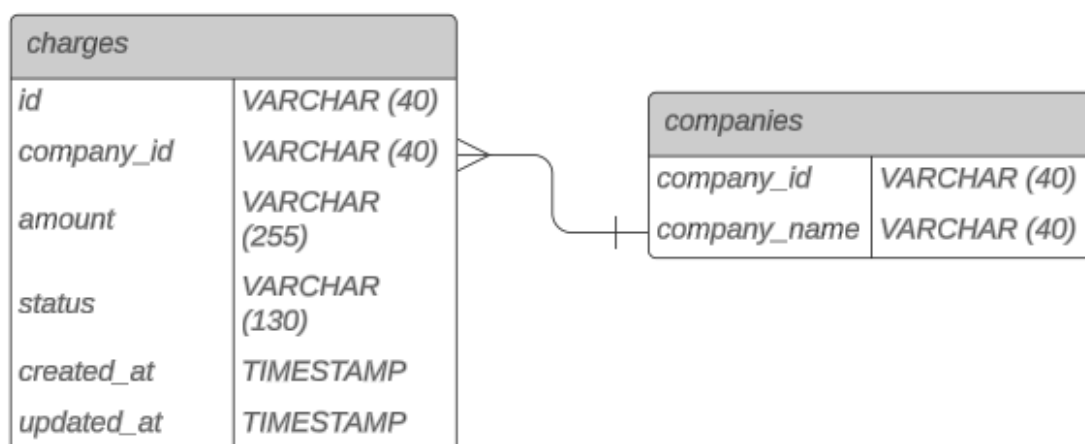
```
PS C:\Users\PC\Downloads\PruebaT\transform_data> docker build -t python_transform_two .  
[+] Building 4.6s (2/3)  
=> [internal] load build definition from Dockerfile  
=> => transferring dockerfile: 310B  
=> [internal] load metadata for docker.io/library/python:3.9-slim  
=> [auth] library/python:pull token for registry-1.docker.io
```

Ambos fueron añadidos a la red creada en fases anteriores.

```
PS C:\Users\PC\Downloads\PruebaT\transform_data> docker network connect network_prueba postgres_db  
PS C:\Users\PC\Downloads\PruebaT\transform_data>
```

Este apartado no fue posible de completar debido a errores por los formatos correspondientes al campo `update_at` en los campos vacíos, ya que aunque en el código se indicaba que podían ser nulos, los errores no se solucionaron.

Para finalizar esta fase se añade el diagrama de base de datos.



1.5 SQL

Debido a los problemas presentados en la fase anterior, la base de datos no pudo ser cargada, y por lo tanto la consulta no se ejecutó, no obstante, la sentencia debería haber quedado de la siguiente forma:

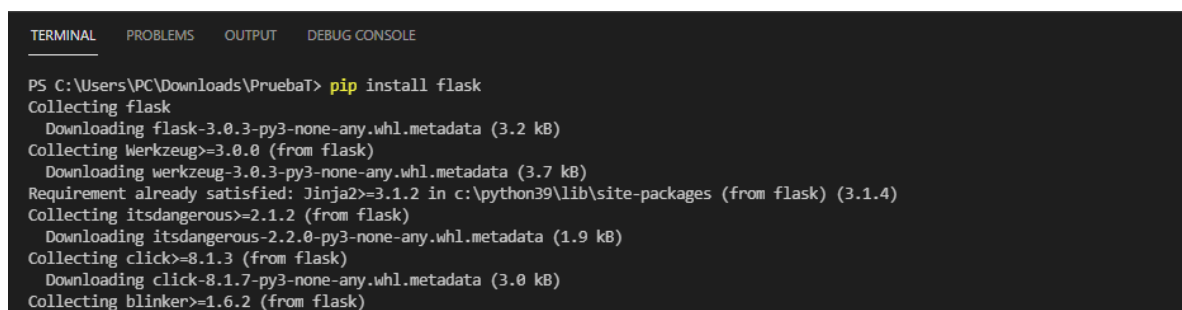
```
CREATE VIEW transactions_d AS
SELECT      c.company_id,    c.company_name,  DATE(ch.created_at) AS
transaction_date, SUM(ch.amount) AS total_amount FROM charges ch JOIN
companies c ON ch.company_id = c.company_id GROUP BY    c.company_id,
c.company_name, DATE(ch.created_at) ORDER BY
transaction_date, c.company_id;
```

Sección 2. Creación de una API

Problema planteado: Calcular el numero faltante de un conjunto de los primeros 100 números naturales del cual se extrajo uno.

Para ello, fue necesario instalar la herramienta flask, que permite crear un programa para ejecutarse en el servidor. Posteriormente se crearon los scripts donde se diseñaron los métodos que permitieron extraer el número y calcularlo. En las siguientes imágenes se aprecian las evidencias.

Instalar flask



```
TERMINAL  PROBLEMS  OUTPUT  DEBUG CONSOLE

PS C:\Users\PC\Downloads\PruebaT> pip install flask
Collecting flask
  Downloading flask-3.0.3-py3-none-any.whl.metadata (3.2 kB)
Collecting Werkzeug>=3.0.0 (from flask)
  Downloading werkzeug-3.0.3-py3-none-any.whl.metadata (3.7 kB)
Requirement already satisfied: Jinja2>=3.1.2 in c:\python39\lib\site-packages (from flask) (3.1.4)
Collecting itsdangerous>=2.1.2 (from flask)
  Downloading itsdangerous-2.2.0-py3-none-any.whl.metadata (1.9 kB)
Collecting click>=8.1.3 (from flask)
  Downloading click-8.1.7-py3-none-any.whl.metadata (3.0 kB)
Collecting blinker>=1.6.2 (from flask)
```

```

PS C:\Users\PC\Downloads\PruebaT\SeccionDos> python api.py
* Serving Flask app 'api'
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 322-336-983

```

Para observar el correcto funcionamiento de la aplicación, se realizaron pruebas con el comando curl.

```

TERMINAL  PROBLEMS 1  OUTPUT  DEBUG CONSOLE
powershell + -  [icon] [icon]

PS C:\Users\PC\Downloads\PruebaT> curl http://127.0.0.1:5000/extract -Method POST -Headers @{"Content-Type":"application/json"} -Body '{"numbe
r": 8}'

StatusCode      : 200
StatusDescription : OK
Content          : {
                    "message": "Numero 8 extraido con exito."
                  }

RawContent       : HTTP/1.1 200 OK
                  Connection: close
                  Content-Length: 48
                  Content-Type: application/json
                  Date: Sun, 14 Jul 2024 18:18:59 GMT
                  Server: Werkzeug/3.0.3 Python/3.9.4

                  {
                    "message": "Numero 8 extraido co...
Forms            : {}
Headers          : {[Connection, close], [Content-Length, 48], [Content-Type, application/json], [Date, Sun, 14 Jul 2024 18:18:59 GMT]...}
Images           : {}
InputFields      : {}
Links            : {}
ParsedHtml       : mshtml.HTMLDocumentClass
RawContentLength : 48

```

```

sers\PC\Downloads\PruebaT> curl http://127.0.0.1:5000/calculate

StatusCode      : 200
StatusDescription : OK
Content          : {
                    "el numero extraido es: ": 8
                  }

RawContent       : HTTP/1.1 200 OK
                  Connection: close
                  Content-Length: 35
                  Content-Type: application/json
                  Date: Sun, 14 Jul 2024 18:19:39 GMT
                  Server: Werkzeug/3.0.3 Python/3.9.4

                  {
                    "el numero extraido es: ": 8
                  }

Forms            : {}
Headers          : {[Connection, close], [Content-Length, 35], [Content-Type, application/json], [Date, Sun, 14 Jul 2024 18:19:39 GMT]...}
Images           : {}
InputFields      : {}
Links            : {}
ParsedHtml       : mshtml.HTMLDocumentClass
RawContentLength : 35

```

```
PS C:\Users\PC\Downloads\PruebaT> curl http://127.0.0.1:5000/extract -Method POST -Headers @{"Content-Type"="application/json"} -Body '{"numbe
r": 112}'
curl : { "error": "El numero debe ser un entero entre 1 y 100." }
En línea: 1 Carácter: 1
+ curl http://127.0.0.1:5000/extract -Method POST -Headers @{"Content-T ...
+ ~~~~~
+ CategoryInfo          : InvalidOperation: (System.Net.HttpWebRequest:HttpWebRequest) [Invoke-WebRequest], WebException
+ FullyQualifiedErrorId : WebCmdletWebResponseException,Microsoft.PowerShell.Commands.InvokeWebRequestCommand
PS C:\Users\PC\Downloads\PruebaT>
```

```
PS C:\Users\PC\Downloads\PruebaT> curl http://127.0.0.1:5000/extract -Method POST -Headers @{"Content-Type"="application/json"} -Body '{"numbe
r": 8}'
curl : { "error": "Numero no valido o ya extraido." }
En línea: 1 Carácter: 1
+ curl http://127.0.0.1:5000/extract -Method POST -Headers @{"Content-T ...
+ ~~~~~
+ CategoryInfo          : InvalidOperation: (System.Net.HttpWebRequest:HttpWebRequest) [Invoke-WebRequest], WebException
+ FullyQualifiedErrorId : WebCmdletWebResponseException,Microsoft.PowerShell.Commands.InvokeWebRequestCommand
```

```
PS C:\Users\PC\Downloads\PruebaT> curl http://127.0.0.1:5000/extract -Method POST -Headers @{"Content-Type"="application/json"} -Body '{"numbe
r": 24}'

StatusCode      : 200
StatusDescription : OK
Content         : {
                  "message": "Numero 24 extraido con exito."
                }

RawContent      : HTTP/1.1 200 OK
                  Connection: close
                  Content-Length: 49
                  Content-Type: application/json
                  Date: Sun, 14 Jul 2024 18:20:59 GMT
                  Server: Werkzeug/3.0.3 Python/3.9.4

                  {
                    "message": "Numero 24 extraido c...
Forms           : {}
Headers         : {[Connection, close], [Content-Length, 49], [Content-Type, application/json], [Date, Sun, 14 Jul 2024 18:20:59 GMT]...}
Images          : {}
InputFields     : {}
Links           : {}
ParsedHtml      : mshtml.HTMLDocumentClass
RawContentLength : 49
```

```
PS C:\Users\PC\Downloads\PruebaT> curl http://127.0.0.1:5000/calculate

StatusCode      : 200
StatusDescription : OK
Content         : {
                  "el numero extraido es: ": 24
                }

RawContent      : HTTP/1.1 200 OK
                  Connection: close
                  Content-Length: 36
                  Content-Type: application/json
                  Date: Sun, 14 Jul 2024 18:21:13 GMT
                  Server: Werkzeug/3.0.3 Python/3.9.4

                  {
                    "el numero extraido es: ": 24
                  }
...
Forms           : {}
Headers         : {[Connection, close], [Content-Length, 36], [Content-Type, application/json], [Date, Sun, 14 Jul 2024 18:21:13 GMT]...}
Images          : {}
InputFields     : {}
Links           : {}
ParsedHtml      : mshtml.HTMLDocumentClass
RawContentLength : 36
```

Los problemas encontrados en esta sección se relacionaron con los argumentos para las pruebas con el comando curl, ya que envió errores de sintaxis en varias ocasiones y se tuvo que ajustar.