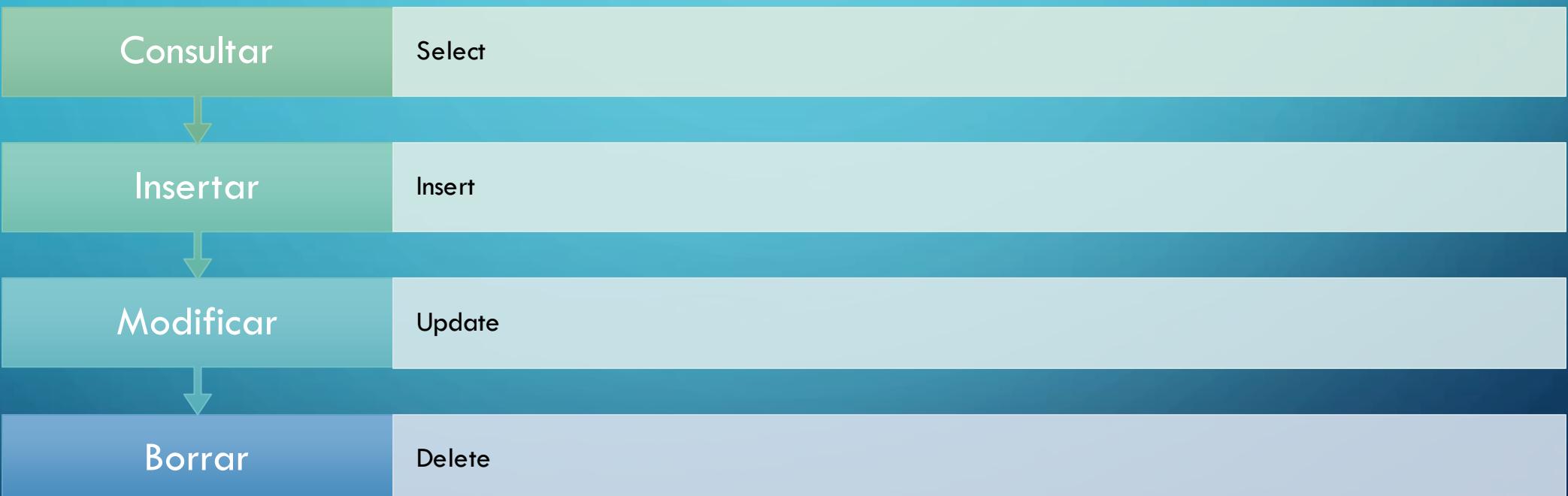


# TEMA 4: CONSULTANDO NUESTROS DATOS

BERNAT COSTA

[BERNAT.COSTA@CESURFORMACION.COM](mailto:BERNAT.COSTA@CESURFORMACION.COM)

# DML DATA MANIPULATION LANGUAGE



# SELECT \* FROM

- Con el select podremos ver el contenido de una tabla.

Listar los datos

Filtrar los datos

Ordenar los datos

Agrupar los datos

**Select \* from TABLA**

**Where condicion**

**Order by fecha**

**Group by  
nombrecolumna**

# ESTRUCTURA DEL SELECT

**SELECT** [\* o lista campos]

**FROM** [NOMBRETABLA]

**WHERE** [CONDICIONES]

**ORDER BY** [NOMBRECAMPO] ASC/DESC

# PARTES DEL SELECT

Comando	Descripción
FROM	Utilizada para especificar la tabla de la cual se van a seleccionar los registros
WHERE	Utilizada para determinar los registros seleccionados en la clausula FROM
ORDER BY	Utilizada para ordenar los registros seleccionados de acuerdo con un orden específico
GROUP BY	Utilizada para separar los registros seleccionados en grupos específicos
HAVING	Utilizada para expresar condición que debe satisfacer cada grupo

# EL WHERE...

PODEMOS PONER CONDICIONES  
USANDO LOS OPERADORES  
COMO MAYOR QUE, MENOR  
QUE, IGUAL....

Operador	Uso
<	Menor que
>	Mayor que
<>	Distinto de
<=	Menor o igual que
>=	Mayor o igual que
BETWEEN	Intervalo
LIKE	Comparación
In	Especificar

# ...Y PODEMOS CONCATENAR VARIAS CONDICIONES

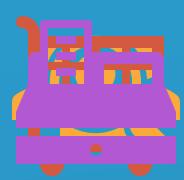
Operador	Uso
AND	Es el “y” lógico. Evalúa dos condiciones y devuelve un valor de verdad sólo si ambas son ciertas.
OR	Es el “o” lógico. Evalúa dos condiciones y devuelve un valor de verdad si alguna de las dos es cierta.
NOT	Negación lógica. Devuelve el valor contrario de la expresión.

Usando operadores lógicos como and, or o not

Podemos hacer uso de parentesis para agrupar condiciones y operadores

# FUNCIONES DE AGREGADO

Comando	Descripción
AVG	Utilizada para calcular el promedio de los valores de un campo determinado
COUNT	Utilizada para devolver el número de registros de la selección
SUM	Utilizada para devolver la suma de todos los valores de un campo determinado
MAX	Utilizada para devolver el valor más alto de un campo especificado
MIN	Utilizada para devolver el valor más bajo de un campo especificado

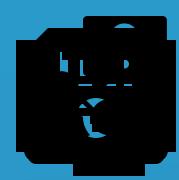


# EJEMPLOS DE FUNCIONES DE AGREGADO

```
SELECT Count(1) FROM CLIENTES
```

```
SELECT Max(total) from PEDIDOS
```

```
SELECT AVG(gastosEnvio) from PEDIDOS
```



# COUNT(1), COUNT(ID) O COUNT(\*)

¿Qué diferencia hay?

Todas dan el mismo resultado, cuentan filas.

Cuando hacemos un COUNT(1), contamos solo los registros de la primera columna. Que es lo mismo que contar los de todas las filas.

... Pero, gasta menos memoria, ya que solo se trae la primera columna

# ORDENAR UNA CONSULTA

Ponemos despues del where ORDER BY nombrecolumna

Podemos agregarle ASC o DESC para indicar el orden que queremos

Podemos añadir tantas columnas como queramos en el Order by

Ejemplo: Select \* from clientes ORDER BY apellido,nombre ASC

# SELECCIONAR LOS PRIMEROS ELEMENTOS

---

Con la instrucción TOP podemos seleccionar los primeros elementos de nuestra tabla

---

Si tenemos muchos datos, nos puede ser útil si solo queremos una muestra.

---

También es útil cuando queremos el primero o el último, hacer un top 1 ordenado de forma correcta.

---

```
SELECT TOP 10 * from Clientes Order by Nombre ASC
```

---

```
SELECT TOP 1 * FROM Pedidos order by Total
```

# OPERACIONES ARITMETICAS

En el SELECT o el WHERE, podemos hacer operaciones aritmeticas con las columnas.

Podemos Sumar dos campos y mostrar el resultado en una columna adicional

Podemos restar dos campos y compararlo con un valor fijo en el where

Tb podemos "sumar" dos varchars, los concatenará.

# EJEMPLOS CON EL AREPEZO

---

```
SELECT subtotal + gastosenvio, *  
from pedidos
```

---

```
SELECT * from pedidos WHERE  
(total - gastosenvio-subtotal) <>0
```

---

```
SELECT direccion + ',' + ciudad,  
nombre from clientes
```

## FUNCION REPLACE

REPLACE(cadena1,  
cadena2,cadena3)

Nos cambiará en la cadena1, la  
cadena2 por la cadena3

Ejemplo: select REPLACE(Descripcion,  
'Arepa',' Kebab') from menu

# FUNCION SUBSTRING

- La función substring nos recorta un STRING (VARCHAR) por los caracteres que le indiquemos.
- Se le pasan 3 parametros.
  - Cadena de texto que queremos recortar
  - Posición inicial
  - Numero de caracteres que queremos recortar.
- Ejemplo select LoginID,SUBSTRING(LoginID,17,100) from HumanResources.Employee

# BÚSQUEDAS DE TEXTO

Para buscar una cadena de texto libre en un varchar, usaremos el comparador Like.

Usaremos el comodín % para indicar que queremos buscar una cadena cualquiera.

Si queremos buscar una la palabra arepa en la descripción, pondremos %arepa%

Esto nos buscará cualquier varchar que empiece con algo, luego contenga arepa, y luego siga con más texto.

Select \* from menu where Descripcion like '%arepa%'

## BUSQUEDA DE NULLS

Para buscar NULLS no podemos igualar a NULL

Debemos poner la condición "is null" o " is not null"

Select \* from CLIENTES where EDAD IS NOT NULL

# QUITAR REPETIDOS

Hay veces que en una consulta, puede darnos valores repetidos.

Podemos quitarlos con la palabra DISTINCT

Select DISTINCT ciudad from clientes

# AGRUPAR RESULTADOS

- Podemos agrupar los resultados por campos repetidos
- El group by especifica por que campos vamos a "agrupar".
- En el Select, SOLO podremos poner los campos del group by. Solo podremos añadir funciones de agregación como sum, count, max,min, avg...
- `Select nombre, ciudad, count(1) from personas  
Group by nombre,ciudad`

# GROUP BY

- IMPORTANTE recordar que no podremos mostrar NINGUN campo de la tabla que no estemos agrupando.
- Se puede obtener valores medios, sumas, max, min de los otros campos usando las funciones de agregado.
- Nos sirve para contar personas por ciudades.

## HAVING

---

Nos sirve para poner un filtro sobre un group by y usar una función de agregado.

---

Select count(1),ciudad from clientes group by ciudad having count(1)>2

# RENOMBRAR COLUMNAS

- Nuestras consultas devuelven tablas, podemos renombrar las columnas para su visualización o para quitar los 'no name' que aparecen cuando en el select le ponemos una operación aritmética o una función de agregado
- `select count(1) as NUMERODECLIENTES,ciudad from clientes group by ciudad having count(1)>2`

# FUNCIONES CON LAS FECHAS

GETDATE()	YEAR(fecha)	MONTH(fecha)	DAY(fecha)
<ul style="list-style-type: none"><li>Nos devuelve la fecha actual</li></ul>	<ul style="list-style-type: none"><li>Nos devuelve el INT del año de la fecha que le pasemos</li></ul>	<ul style="list-style-type: none"><li>Nos devuelve el mes</li></ul>	<ul style="list-style-type: none"><li>Nos devuelve el dia</li></ul>

```
SELECT fecha_pedido,year(fecha_pedido),MONTH(fecha_pedido),day(Fecha_pedido)  
FROM PEDIDOS
```

# UNIR DOS TABLAS

Podemos unir el resultado de dos  
Selects siempre que tengan el mismo numero de campos.

CONCATENAMOS los resultados de dos selects en una sola tabla, unos tras otros.

```
SELECT salesorderid, TotalDue, 'Sales' as tipo  
FROM sales.Salesorderheader  
UNION  
SELECT PurchaseORDERID, TotalDue,'Purchase'  
FROM PURCHASING.PurchaseOrderHeader
```

# CRUZAR DOS TABLAS

- Para cruzar dos tablas siguiendo una FK, usamos los INNER JOINS
- Le indicamos después del FROM, con que tabla queremos unirlo y que campo es el que nos une cada tabla con un ON.
- En el ON pondremos una condición.
- `SELECT * FROM tabla1  
INNER JOIN tabla2 ON tabla1.id = tabla2.id`

- Se usan de la misma forma XXX JOIN tabla on CONDICION
- Con el Inner Join, no mostramos los datos no coincidentes en alguna de las dos tablas
- Con el Left Join, nos deja los valores de la tabla 1 aunque no tenga coincidencias en la tabla2. Nos pondrá a null los datos de la tabla 2 cuando no haya coincidencias

```
SELECT p.idPersona, p.nombre, p.apellido, p.email,c.id,c.direccion, c.cod_postal
```

## OTROS JOINS

```
FROM Personas p  
LEFT JOIN Clientes c ON p.email = c.email
```

- Esta select nos dará los datos de la tabla personas y clientes, cuando coincida el email, los pondrá en la misma fila, cuando el email de Personas no lo encuentre en la tabla CLIENTES, pondrá los campos de la tabla CLIENTES a null.

1 Paco Gonzalez paco@gmail.com 1 calle 41002

2 Bernat Costa bernat@gmail.com NULL NULL NULL

(Con un Inner Join, la linea 2 con Bernat Costa, no se muestra ya que no

# SURFEANDO LAS FKS

---

Podemos concatenar tantos INNER JOINS COMO NECESITEMOS.

---

Con el diagrama de tablas en la mano, podemos mostrar datos distantes si hay líneas entre esas tablas.

---

Una vez encontrado el camino que nos une los campos, ponemos un INNER JOIN por cada tabla por la que debemos pasar.



# EJERCICIO

En la BBDD de AdventureWorks2017:

1. Dame el nombre y apellidos de los empleados
2. Dame las Personas que son de CALIFORNIA

# TUTORIAL WEB INTERESANTE

- <https://www.sql-easy.com/es/>
  - Aunque es de MYSQL, por ejemplo la opción LIMIT NO EXISTE en SQL SERVER
- Videos de Aprende SQL en 1 Minuto con piezas de lego de **@dev\_gabo** subidos al teams

# RESTAURAR BBDD

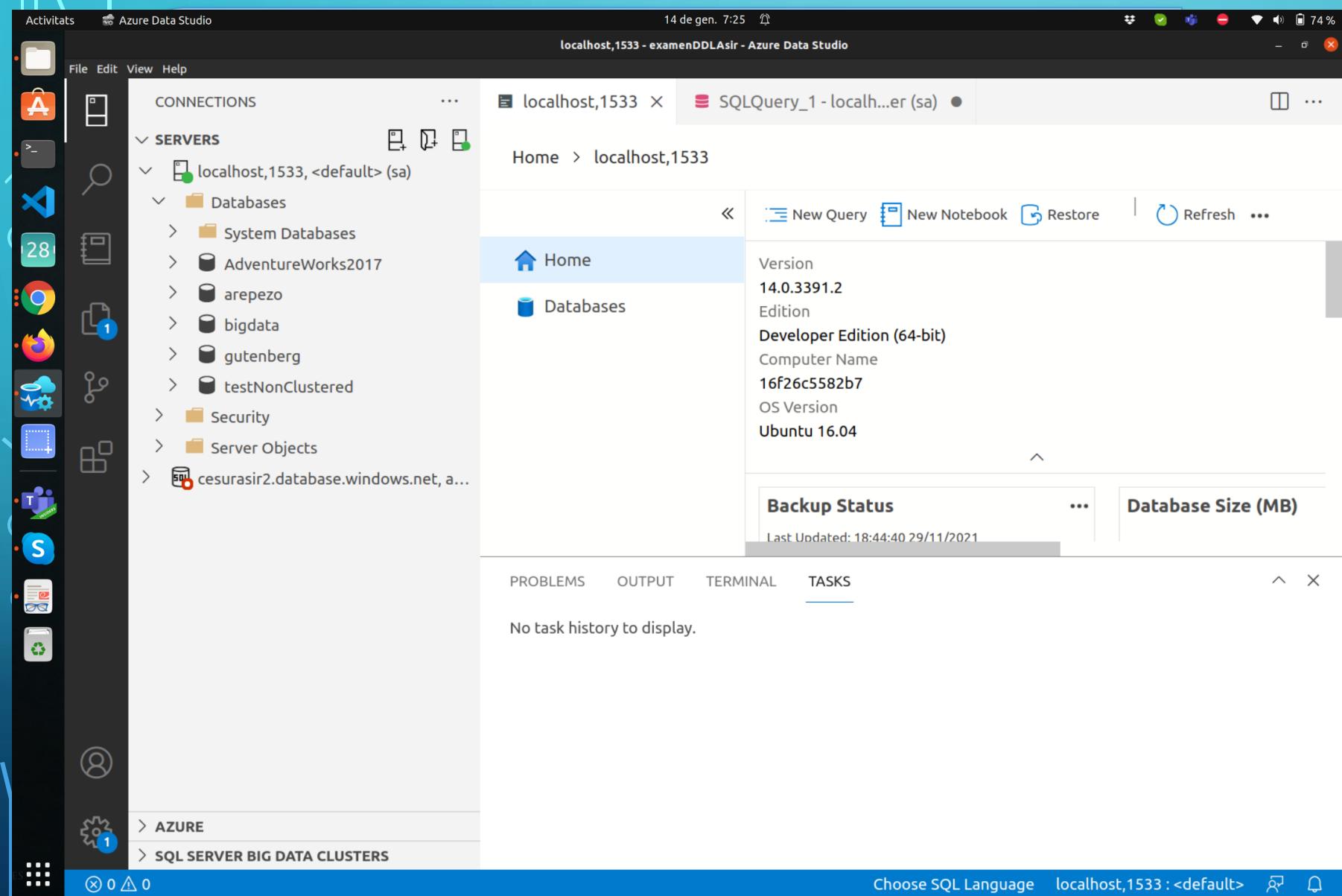
- Con Azure Data Studio o SSMS. Debemos seguir estos pasos.
  1. Crear la BBDD con el mismo nombre que la que queremos restaurar
  2. Copiar el fichero .bak dentro del contenedor
  3. Seleccionar la tarea de RESTAURAR
  4. Seleccionar nuestro fichero
  5. En Files, marcar la opción de **relocate files**
  6. En Options, marcar la primera opción (**Overwrite**) y desmarcar la que pone "**Take...**"

# TUTORIAL RESTAURAR BACKUP

A screenshot of the Azure Data Studio interface. The left sidebar shows various icons for file management, databases, and connectivity. The main window has a title bar "SQLQuery\_1 - localhost,1533.master (sa) - examenDDLAsir - Azure Data Studio". The top menu bar includes File, Edit, View, Help, CONNECTIONS, and a context menu. The left pane under SERVERS shows connections to "localhost,1533, <default> (sa)" and lists several databases: Databases, System Databases, AdventureWorks2017, arepezo, bigdata, gutenberg, testNonClustered, Security, Server Objects, and cesurasir2.database.windows.net, a... A connection to "localhost,1533" is selected. The center pane contains a query editor with the command "create database AdventureWorks2017" and a messages section showing the execution log: "Started executing query at Line 1", "Commands completed successfully.", and "Total execution time: 00:00:00.423". Below the message log are tabs for PROBLEMS, OUTPUT, TERMINAL, and TASKS, with the latter being active. The bottom status bar shows "Ln 1, Col 35", "Spaces: 4", "UTF-8", "LF", "SQL", "0 rows", "MSSQL", "00:00:00", "localhost,1533:master", and icons for search, refresh, and notifications.

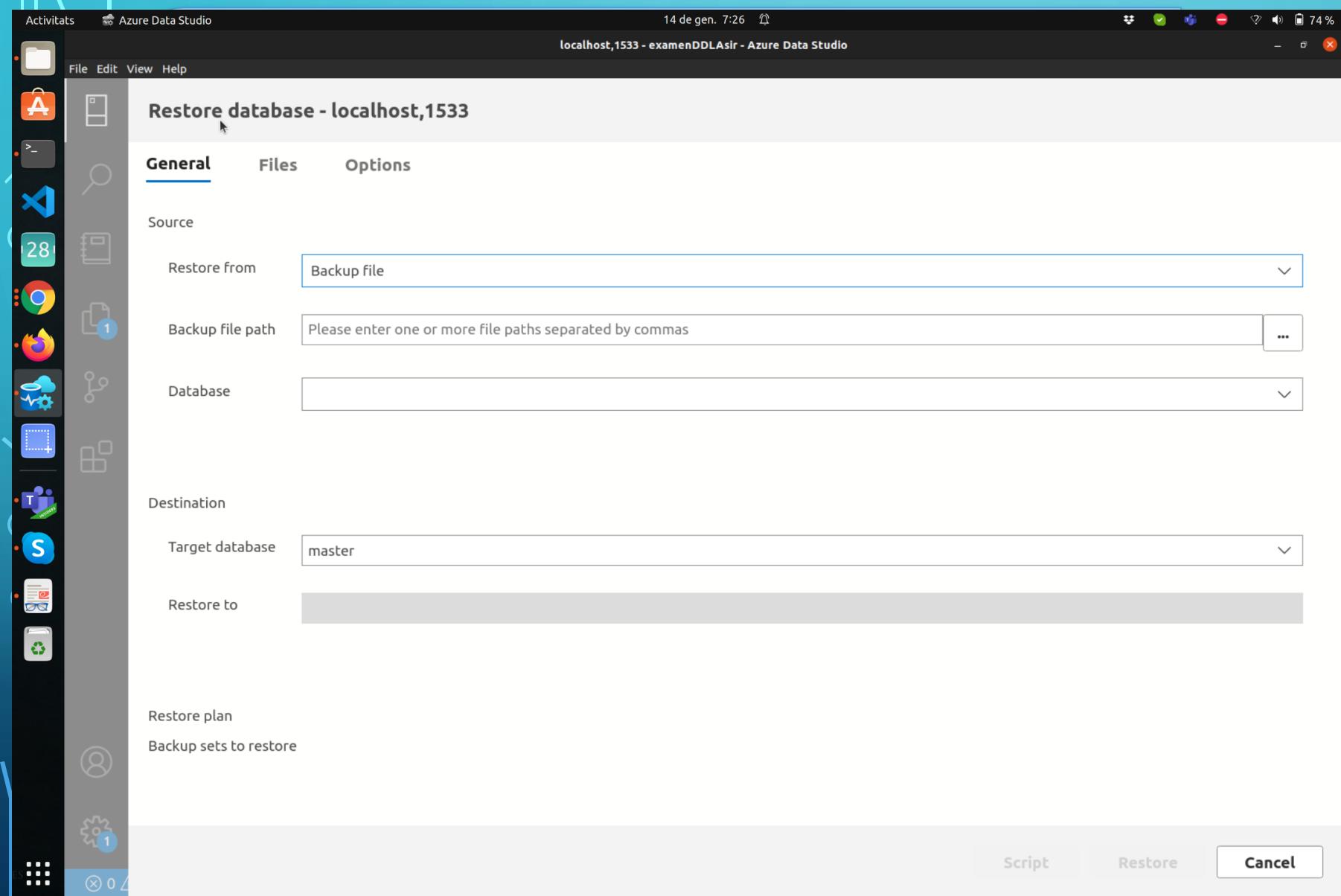
- Crear la BBDD con el mismo nombre que la que queremos restaurar

# TUTORIAL RESTAURAR BACKUP



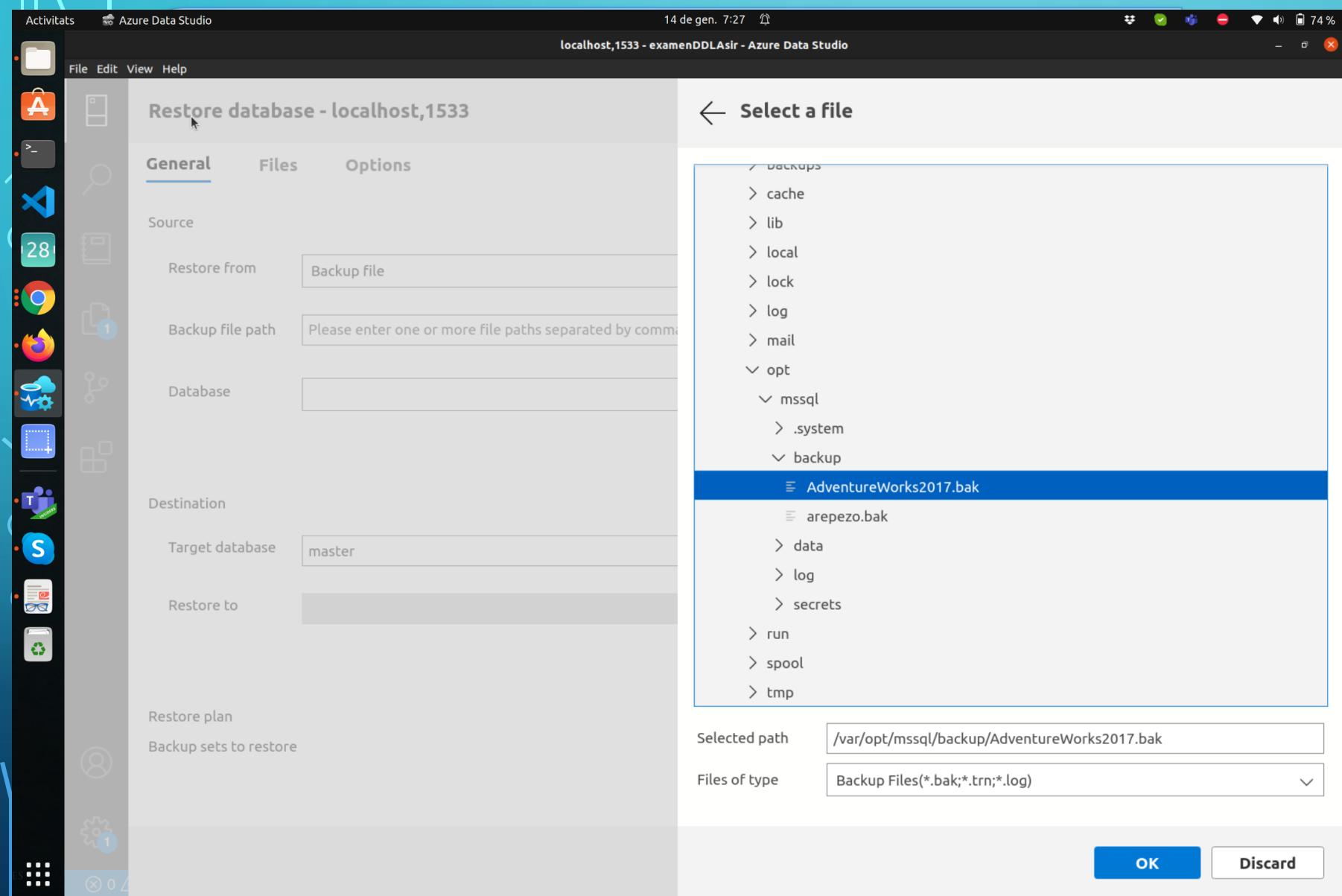
- Seleccionar Opcion Restore sobre la bbdd recien creada o sobre la opcion Manage del localhost.

# TUTORIAL RESTAURAR BACKUP



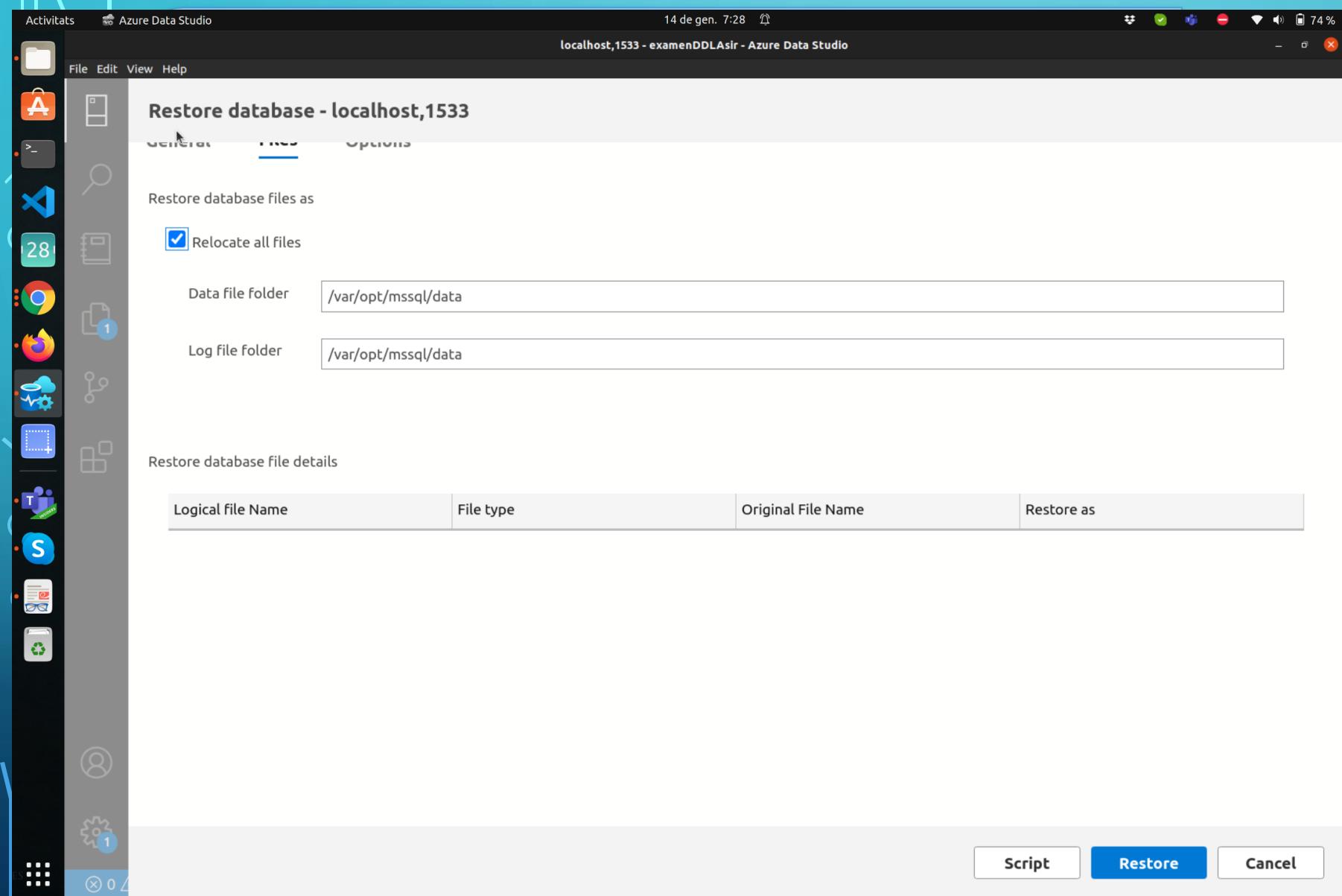
- Seleccionar Backup File y seleccionar los 3 ...

# TUTORIAL RESTAURAR BACKUP



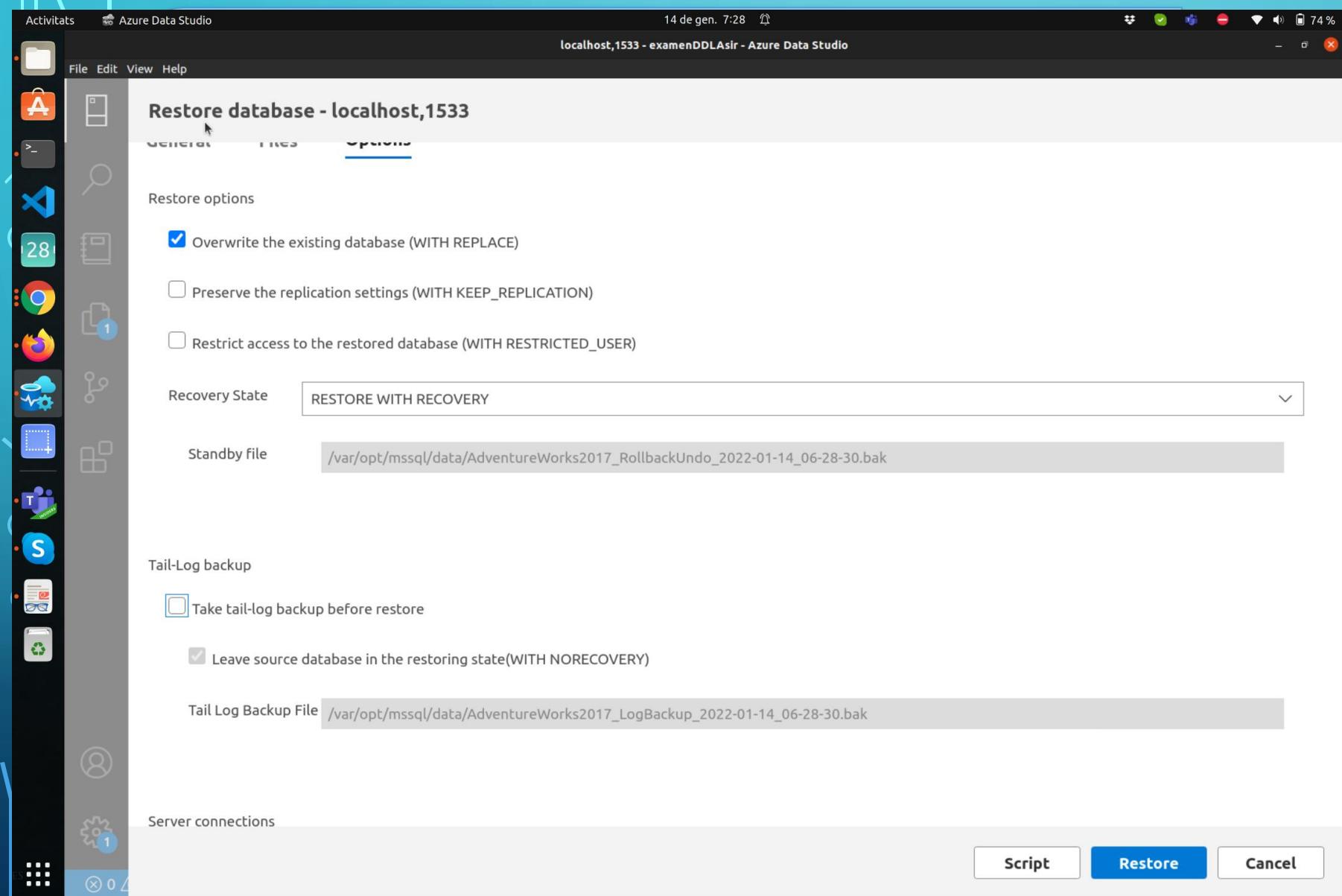
- Copiar el .bak a la carpeta backup del contenedor ( desde el explorador de windows)
- Seleccionar este fichero

# TUTORIAL RESTAURAR BACKUP

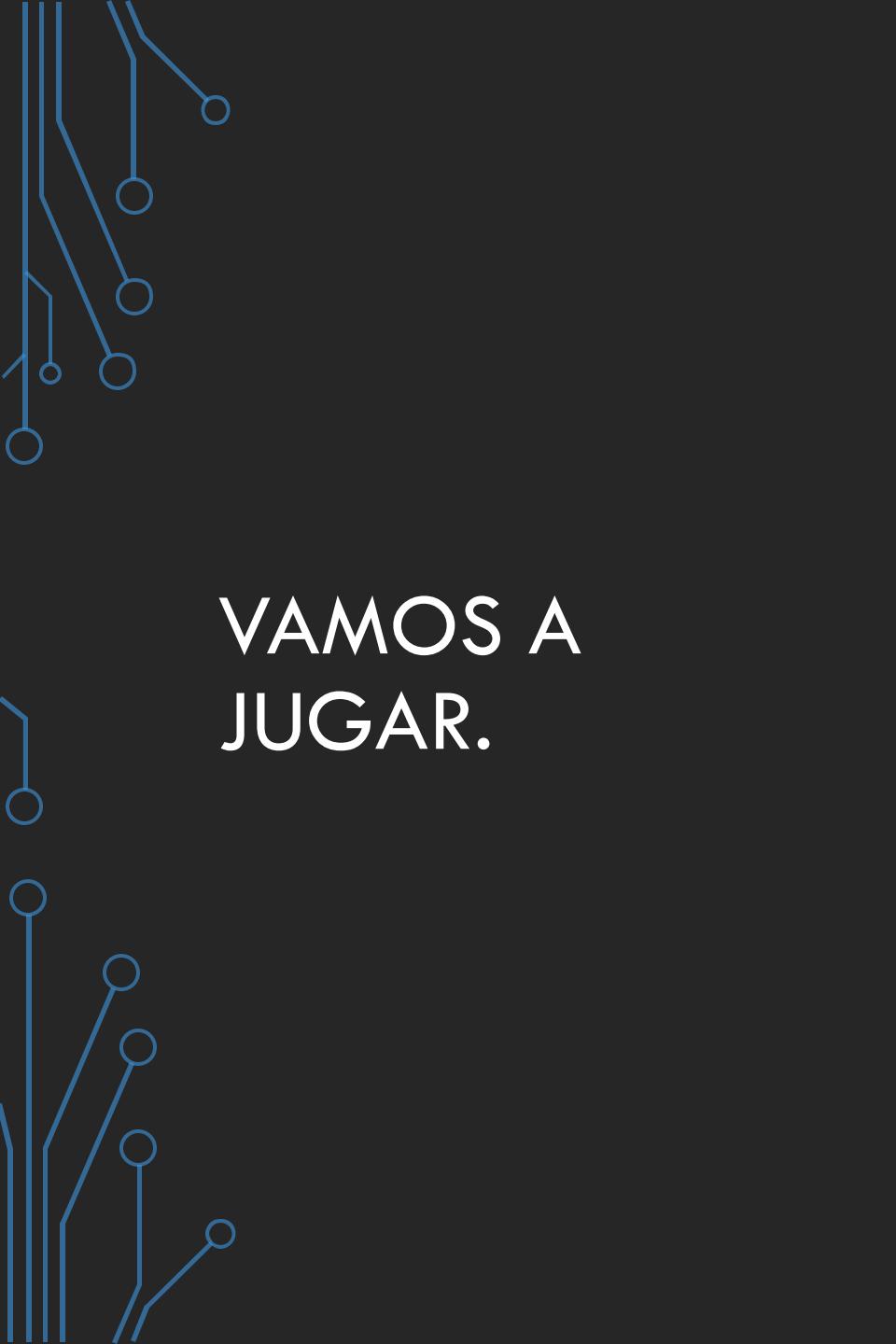


- En la opcion Files Marcar el chec Relocate all files

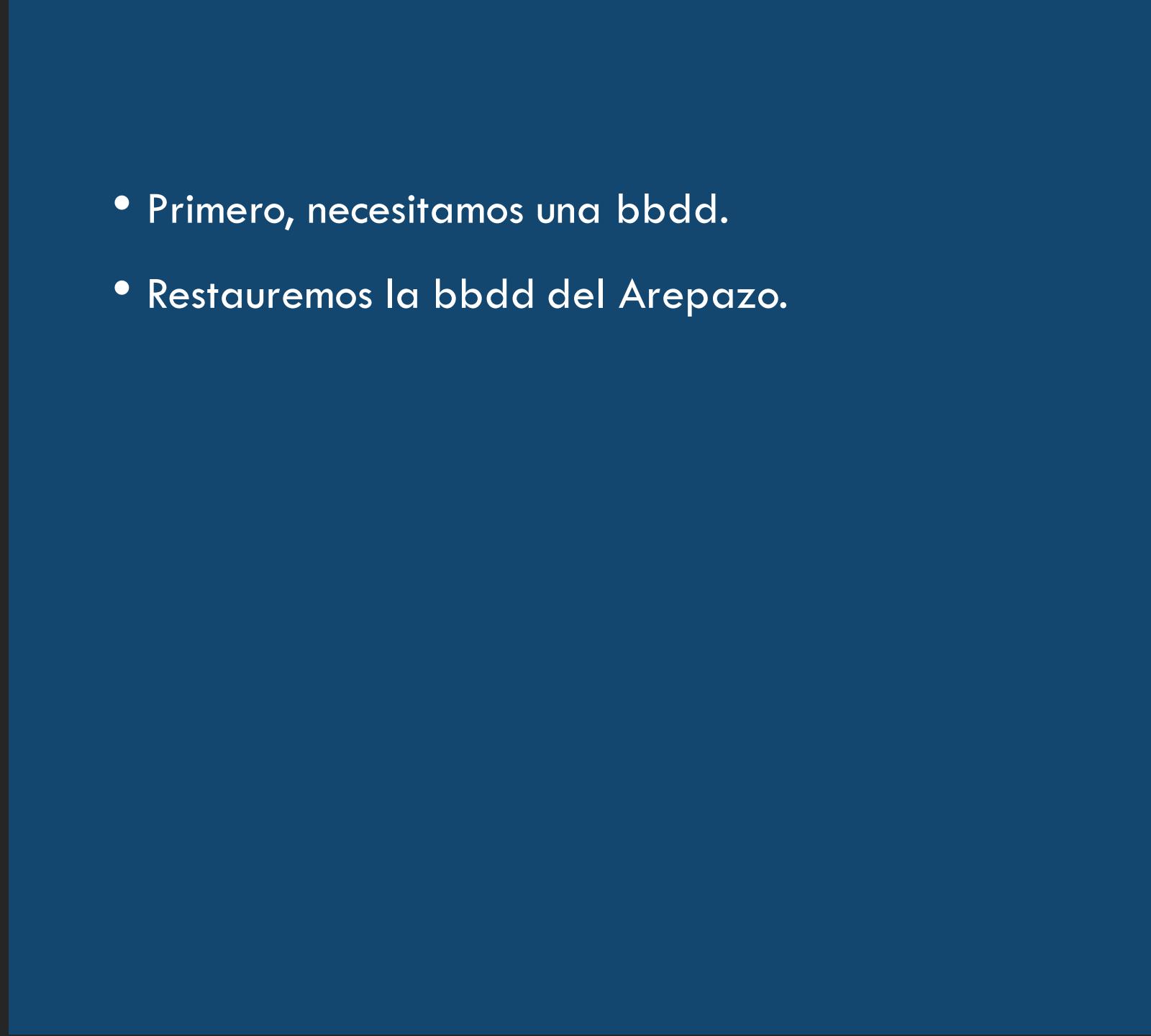
# TUTORIAL RESTAURAR BACKUP



- En Options, marcar primera opción y desmarcar la ultima.
- Apretar el botón Restaurar



# VAMOS A JUGAR.

- 
- Primero, necesitamos una bbdd.
  - Restauremos la bbdd del Arepazo.

That's all Folks!