

## Plan de migración

Leidy Katherine Calderón Castaño

Camilo Andres Tiria Corredor

SENA

Análisis y Desarrollo de Sistemas de Información

Bogotá D.C

6/29/2021



Copyright © 2021

Calle 49ª Bis sur # 10d -20

**Absence and Bad Grades Software**  
Todos los Derechos Reservados

320 447 3192  
313 710 3188

## Índice

### Contenido

Índice .....	2
Introducción .....	4
Justificación .....	4
ANÁLISIS PREVIO DEL SISTEMA ACTUAL Y FINAL.....	5
1. Número de registro filas: .....	5
1.1 Tablas:.....	5
2. Numero de atributos (NA).....	5
2.1 Tablas:.....	5
3. Número de claves: .....	6
3.1 Llaves Primarias:.....	6
3.2 Llaves Foráneas: .....	6
4. Numero de Objetos de conversión no trivial: .....	6
5. Otros aspectos a evaluar:.....	6
Análisis de migración .....	7
ESTRATEGIA DE MIGRACIÓN .....	8
1. Elaborar plan de ejecución:.....	8
Plan de Migración .....	8
1. Plan de formación:.....	8
2. Plan de ejecución: .....	8
3. Plan de soporte: .....	8
EJECUCIÓN.....	9
1. Instalación de Strawberry Perl: .....	9
2. Visualización de PostgreSQL: .....	18



FICHA DE CONTROL DE CAMBIOS				
PROYECTO		Absence and Bad Grades Software (ABGS)		
DOCUMENTO		Plan de migración		
VERSION		1.0		
FECHA DE CREACIÓN		06/29/2021		
FECHA CAMBIO		06/29/2021		
RESPONSABLES		Leidy Katherine Calderón Castaño y Camilo Andres Tiria Corredor		
LIDER		Camilo Andres Tiria Corredor		
HISTORIAL				
FECHA	NUMERO DE VERSION	OBSERVACIONES	AUTOR(ES)	VER
06/29/2021	1.0	-Mejora en la GUI -Implementación de capas	Leidy Katherine Calderón Castaño y Camilo Andres Tiria Corredor	



## Introducción

Para entender el proceso realizado en el siguiente documento es importante saber que es migrar:

La migración de BD es un proceso que se lleva a cabo para trasladar datos almacenados previamente de un gestor de base de datos a otro o desde el mismo gestor, pero a versiones diferentes.

Determinar si se va a migrar la estructura de la base de datos, o solo los datos, o ambas cosas varían de acuerdo a la necesidad y objetivo de la migración en la entidad. No dejar de lado el tamaño de la base de datos y la complejidad de la base de datos, la forma en la que se va a controlar que en realidad si se migraron todos los objetos y datos de una base de datos, las restricciones, y como tal que se esté pasando toda la lógica de negocio.

## Justificación

Se requiere realizar una migración desde el gestor de base de datos de MySQL a PostgreSQL. En el mercado se presentan diferentes variables entre grandes y pequeñas empresas, cada una de ellas presenta sus desventajas y ventajas. En este caso, se requiere pasar de MySQL a postgresql, por lo que lo fundamental es la información, los datos contenidos en dicho gestor.

Hay muchos factores que intervienen en un proceso de migración: tiempo, costo, escalabilidad, tamaño de la base de datos, estructura, restricciones, complejidad, soporte técnico, experiencia, planeación, control y seguimiento, compatibilidad, etc. A continuación, se va a realizar un plan de migración de base de datos MySQL a POSTGRESQL. Se espera que este documento sea de gran utilidad y sirva como base para posibles migraciones en el sistema de información de **Absence and Bad Grades Software (ABGS)** para la gestión de las novedades de los aprendices SENA.



## ANÁLISIS PREVIO DEL SISTEMA ACTUAL Y FINAL

### 1. Número de registro filas:

#### 1.1 Tablas:

- Aprendiz: 3 registros
- TipoInstructor: 2 registros
- Instructor: 4 registros
- EstadoInstructor: 2 registros
- Solicitudes: 0 registros
- Estado: 2 registros
- Programa: 2 registros
- Usuario: 5 registros
- Roles: 2 registros
- Inasistencia: 24 registros
- R\_aprendizaje: 5 registros
- Notas: 12 registros

### 2. Numero de atributos (NA)

#### 2.1 Tablas:

- Aprendiz:  
Atributos o campos: Num\_doc, Estado\_idEstado, PROGRAMA\_Ficha\_carac, INSTRUCTOR\_Num\_doc, Tip\_doc, Nombres, Apellidos, Tel\_apre, Correo\_SENA, Correo\_pl.
- TipoInstructor:  
Atributos o campos: idTipoInstructor, TipoInstructor.
- Instructor:  
Atributos o campos: Num\_doc, TipoInstructor\_idTipoInstructor, EstadoInstructor\_idEstadoInstructor, Tip\_doc, NombresI, ApellidosI, Telefono, Dirección, Correo\_corp, Correo\_Pl.
- EstadoInstructor:  
Atributos o campos: idEstadoInstructor, EstadoInstruc.
- Solicitudes:  
Atributos o campos: Cod\_s, APRENDIZ\_Num\_doc\_apre, INSTRUCTOR\_Num\_doc, Mensaje.
- Estado:  
Atributos o campos: idEstado, Nombre.



- Programa:  
Atributos o campos: Ficha\_carac, Nom\_Program, Area, Fecha\_Ingr, Tipo\_Formacion, Modalidad.
  - Usuario:  
Atributos o campos: UsuLogin, UsuPassword.
  - Roles:  
Atributos o campos: idRoles, NomRoles.
  - Inasistencia:  
Atributos o campos: Cod\_Inas, APRENDIZ\_Num\_doc, Fecha\_Inas, Observaciones.
  - R\_Aprendizaje:  
Atributos o campos: Code\_Res, INSTRUCTOR\_Num\_doc, PROGRAMA\_Ficha\_carac, Nom\_Res.
  - Notas:  
Atributos o campos: Cod\_Nota, APRENDIZ\_Num\_doc, R\_Aprendizaje\_Code\_Res, Nota.
3. Número de claves:
- 3.1 Llaves Primarias:  
El número de llaves primarias es 12
- 3.2 Llaves Foráneas:  
El número de llaves foráneas es 12
4. Numero de Objetos de conversión no trivial:  
Los objetos de la base de datos cuya conversión a lenguaje PostgreSQL, de nivel de complejidad bajo son:
- El número total de tablas es 12, el número de índices es 12.
5. Otros aspectos a evaluar:  
Hay una serie de factores comunes a cualquier base de datos que influyen en la complejidad de la migración y que hay que ponderar:
- ✓ Cantidad y tipo de SQL propietario que se use.
  - ✓ Calidad de datos.
  - ✓ Existencia de documentación del sistema.
  - ✓ Requisitos de diseño tales como la alta disponibilidad y replicación.
  - ✓ Software de terceros dependencias.



- ✓ Cambio en el sistema operativo y/o cambio de la plataforma de hardware como consecuencia de la migración.
- ✓ Cualificación y experiencia del personal involucrado en la migración.
- ✓ La disponibilidad de un equipo dedicado para el desarrollo de la migración.
- ✓ Imposibilidad de detener los cambios y los nuevos desarrollos sobre el código a migrar.
- ✓ Tiempo máximo que se permite para la realización de la migración

## Análisis de migración

En este punto deben identificarse los cambios específicos que hay que realizar para transformar cada elemento de la base de datos MySQL en un elemento de PostgreSQL que funcione de la misma manera.

Identificando tipos de datos, funciones, triggers, elementos DML, procedimientos, etc., no soportados en PostgreSQL.

Por otro lado, se va a mencionar algunos tipos de datos MySQL y su equivalente en PostgreSQL:

TIPOS DE DATOS	
MySQL	PostgreSQL
<b>bigint</b>	<b>bigint</b>
<b>double,decimal,float</b>	<b>numeric</b>
<b>int,mediumint</b>	<b>integer</b>
<b>smallint,tinyint</b>	<b>smallint</b>
<b>char</b>	<b>character</b>
<b>varchar,enum,set</b>	<b>character varying</b>
<b>longtext,mediumtext,text,tinytext</b>	<b>text</b>
<b>tinyblob,blob,mediumblob,longblob</b>	<b>bytes</b>
<b>datetime,timestamp</b>	<b>timestamp</b>
<b>time,date,year</b>	<b>date</b>



## ESTRATEGIA DE MIGRACIÓN

### 1. Elaborar plan de ejecución:

Si se presenta algún inconveniente respecto a la migración, es importante realizar un back up (copia de seguridad) para así dar una solución rápida al evento imprevisto.

Es importante no modificar el ambiente donde estas instalada MySql para que si se cancela la migración solo sea iniciar el servidor en la base de datos MySql.

### Plan de Migración

Una vez identificadas las actividades necesarias para realizar la transformación y por ende la migración a Postgresql, se identifican las necesidades de la migración y se planifican las diferentes tareas:

#### 1. Plan de formación:

Los desarrolladores deben conocer cómo hacer las modificaciones pertinentes al código ya finalizada la migración de la base de datos.

#### 2. Plan de ejecución:

- ✓ Reemplazar los tipos de datos de MySQL con los de postgresQL.
- ✓ Cambiar los campos de autoincremento de mySQL (Auto\_increment) con los de PostgreSQL (serial).
- ✓ Revisar que todas las tablas hayan sido correctamente migradas.
- ✓ Cambiar el código en las consultas a la base de datos por el del nuevo gestor.

#### 3. Plan de soporte:

Incluye lo necesario para detectar incidencias, además de los siguientes pasos a seguir para completar la migración:

Paso 1. Generar un back up de la estructura de la base de datos en MySQL.

Paso 2. Traducir SQL de MySQL a SQL a PostgreSQL usando mysql2pgsql.perl.

Paso 3. Enviar los datos generados desde MySQL a PostgreSQL.

A partir del paso 3, se migran los datos y se debe sacar un respaldo de los datos sin la estructura de las tablas.





## EJECUCIÓN

Para iniciar el proceso de migración es necesario instalar “Strawberry”, un archivo perl llamado “mysql2pgsql.pl” y el aplicativo de PostgreSQL de las siguientes direcciones:

- <https://drive.google.com/file/d/11EG1> (mysql2pgsql.pl)
- <https://code.google.com/archive/p/pad> (Strawberry)
- <https://www.postgresql.org/download/> (PostgreSQL)

### 1. Instalación de Strawberry Perl:

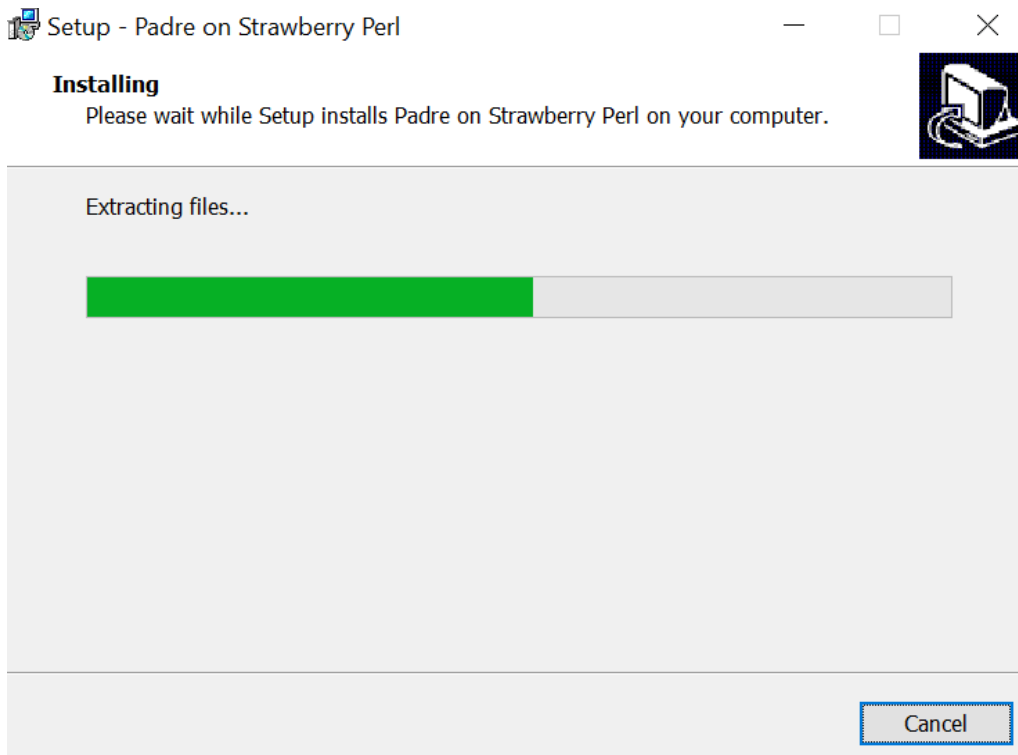


Figure 1. Instalación de Strawberry

Se siguen las instrucciones del instalador y se guarda en la carpeta por defecto en el Disco C:

- Nos dirigimos al Disco C: e ingresamos en la carpeta llamada “Strawberry”

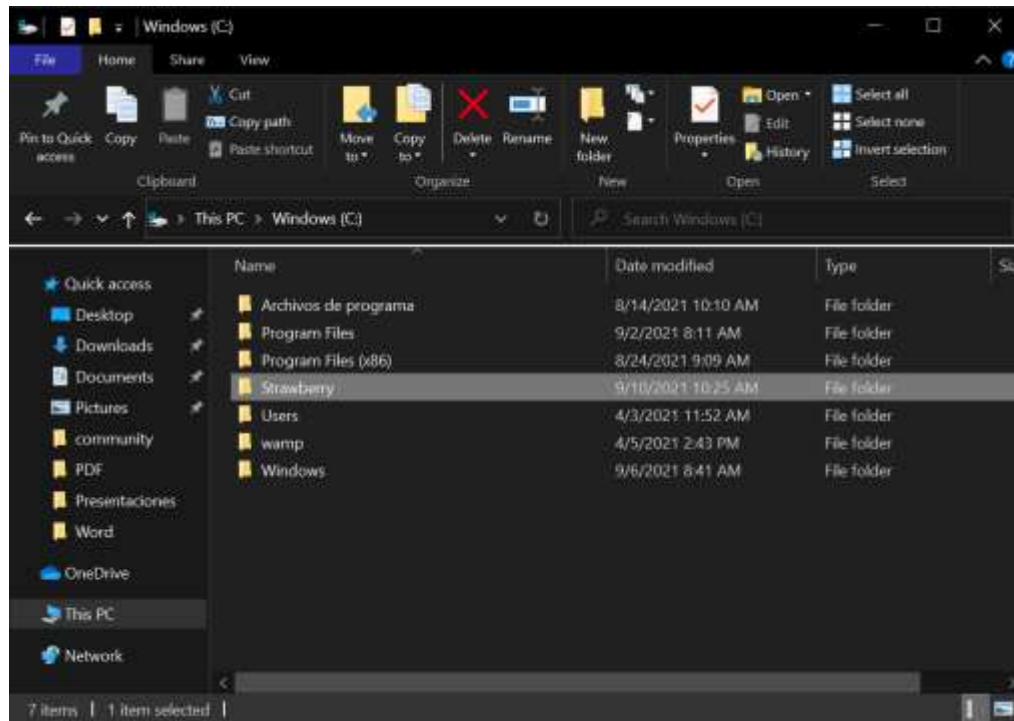


Figure 2. Instalación de Strawberry

- Luego, ingresamos en la carpeta “Perl” seguido de la carpeta “bin” y finalmente copiamos el documento descargado previamente “mysql2pgsql.pl”

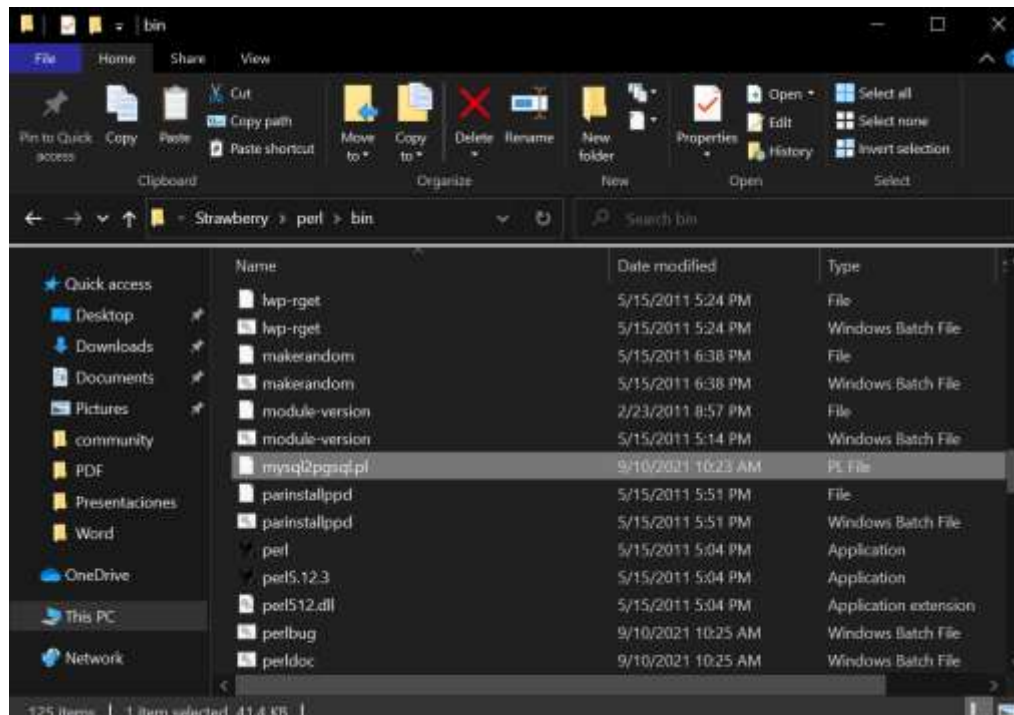


Figure 3. Instalación de Strawberry

- Por consiguiente, buscamos la dirección donde se encuentre MySQL y abrimos el CMD como administradores, ingresamos la dirección a la carpeta, luego un primer código (Mysqldump -u root -p abgs), y la contraseña que tenga nuestra base de datos.

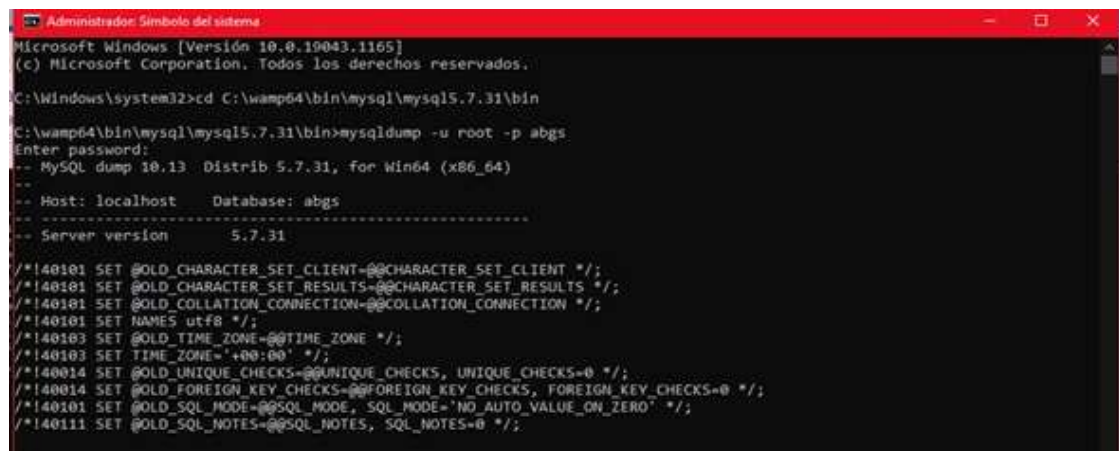
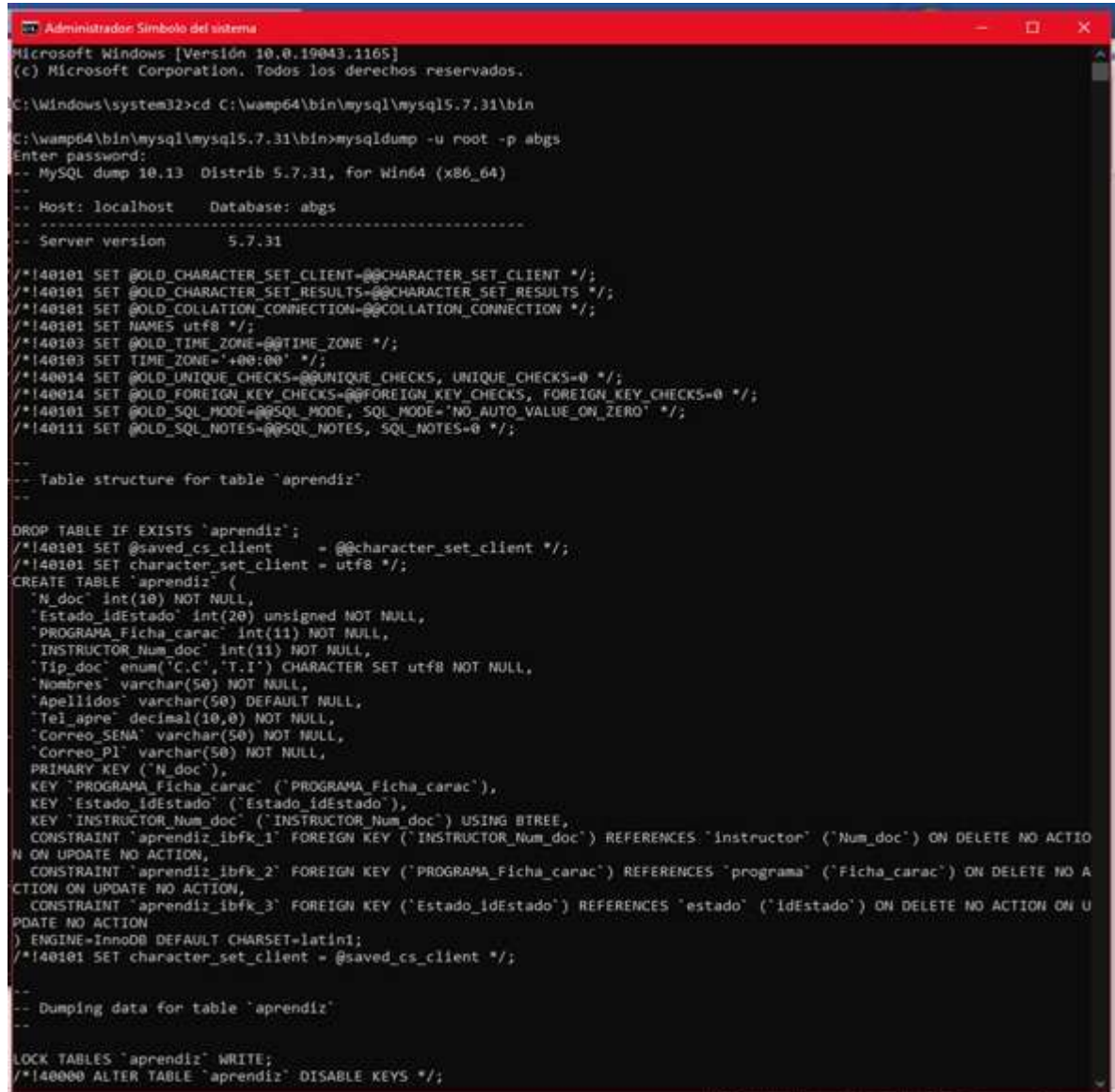


Figure 4. Creación de Back Up

- De esta manera visualizaremos el **Back up** de nuestra base de datos en MySQL:



```

Administrador Símbolo del sistema
Microsoft Windows [Versión 10.0.19043.1165]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Windows\system32>cd C:\wamp64\bin\mysql\mysql5.7.31\bin

C:\wamp64\bin\mysql\mysql5.7.31\bin>mysqldump -u root -p abgs
Enter password:
-- MySQL dump 10.13 Distrib 5.7.31, for Win64 (x86_64)
--
-- Host: localhost    Database: abgs
--
-- Server version
--      5.7.31
--
/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8 */;
/*!40103 SET @OLD_TIME_ZONE=@@TIME_ZONE */;
/*!40103 SET TIME_ZONE='+00:00' */;
/*!40014 SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0 */;
/*!40014 SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0 */;
/*!40101 SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='NO_AUTO_VALUE_ON_ZERO' */;
/*!40111 SET @OLD_SQL_NOTES=@@SQL_NOTES, SQL_NOTES=0 */;

--
-- Table structure for table `aprendiz`
--
DROP TABLE IF EXISTS `aprendiz`;
/*!40101 SET @saved_cs_client      = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `aprendiz` (
  `N_doc` int(10) NOT NULL,
  `Estado_IdEstado` int(20) unsigned NOT NULL,
  `PROGRAMA_Ficha_carac` int(11) NOT NULL,
  `INSTRUCTOR_Num_doc` int(11) NOT NULL,
  `Tip_doc` enum('C','T','I') CHARACTER SET utf8 NOT NULL,
  `Nombres` varchar(50) NOT NULL,
  `Apellidos` varchar(50) DEFAULT NULL,
  `Tel_apre` decimal(10,0) NOT NULL,
  `Correo_SENA` varchar(50) NOT NULL,
  `Correo_PL` varchar(50) NOT NULL,
  PRIMARY KEY (`N_doc`),
  KEY `PROGRAMA_Ficha_carac` (`PROGRAMA_Ficha_carac`),
  KEY `Estado_IdEstado` (`Estado_IdEstado`),
  KEY `INSTRUCTOR_Num_doc` (`INSTRUCTOR_Num_doc`) USING BTREE,
  CONSTRAINT `aprendiz_ibfk_1` FOREIGN KEY (`INSTRUCTOR_Num_doc`) REFERENCES `instructor` (`Num_doc`) ON DELETE NO ACTION ON UPDATE NO ACTION,
  CONSTRAINT `aprendiz_ibfk_2` FOREIGN KEY (`PROGRAMA_Ficha_carac`) REFERENCES `programa` (`Ficha_carac`) ON DELETE NO ACTION ON UPDATE NO ACTION,
  CONSTRAINT `aprendiz_ibfk_3` FOREIGN KEY (`Estado_IdEstado`) REFERENCES `estado` (`IdEstado`) ON DELETE NO ACTION ON UPDATE NO ACTION
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `aprendiz`
--
LOCK TABLES `aprendiz` WRITE;
/*!40000 ALTER TABLE `aprendiz` DISABLE KEYS */;

```

Figure 5.Creación de Back Up



```

Administrador Símbolo del sistema
--
LOCK TABLES `aprendiz` WRITE;
/*I40000 ALTER TABLE `aprendiz` DISABLE KEYS */;
INSERT INTO `aprendiz` VALUES (2067470,2,2067472,1021123589,'C.C','Alison yulieth','Calderon sanchez',3201547892,'galen@
misena.edu.co','brayant@gmail.com'),(2067476,2,2067472,10005890,'C.C','Gustavo andres','Calderon fernandez',32015478
92,'uhb@misena.edu.co','brayant@gmail.com'),(90586247,1,102145,10005890,'C.C','karen','casas',3201457854,'kkkh@misena.ed
u.co','wiyfy@gmail.com'),(1000154521,2,2067472,1021123589,'C.C','Santiago alexander','Trigos casas',3201547892,'trigos@
misena.edu.co','trigos@gmail.com'),(1000214566,2,2067472,10005890,'C.C','Wilmer Edisson','Mendez',3201457854,'w@misena.
edu.co','willa@gmail.com'),(1000215864,1,2067472,1000215896,'C.C','Leidy Katherine','Calderon castaño',3204473192,'lk
calderon46@misena.edu.co','leidy.khaterinecc2002@gmail.com'),(1000255852,1,2067472,1000215896,'C.C','Camilo','Cruz soler
',300725494,'cruz@misena.edu.co','cami89@gmail.com'),(1002215478,1,2067472,1000215896,'C.C','Brayan','Cabrera hurtado',
3201547892,'brayan@misena.edu.co','brayan@gmail.com'),(1034343434,2,102145,10005890,'C.C','Ronald','Casas',3201457854,'
ronald@misena.edu.co','ronald@gmail.com'),(1214578963,1,102145,10005890,'C.C','Augusto','Solaniña',3201457854,'augusto@
misena.edu.co','augusto@gmail.com');
/*I40000 ALTER TABLE `aprendiz` ENABLE KEYS */;
UNLOCK TABLES;

--
-- Table structure for table `estado`
--
DROP TABLE IF EXISTS `estado`;
/*I40101 SET @saved_cs_client = @@character_set_client */;
/*I40101 SET character_set_client = utf8 */;
CREATE TABLE `estado` (
  `idEstado` int(20) unsigned NOT NULL,
  `Nombre` varchar(20) NOT NULL,
  PRIMARY KEY (`idEstado`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
/*I40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `estado`
--

LOCK TABLES `estado` WRITE;
/*I40000 ALTER TABLE `estado` DISABLE KEYS */;
INSERT INTO `estado` VALUES (1,'Activo'),(2,'Inactivo');
/*I40000 ALTER TABLE `estado` ENABLE KEYS */;
UNLOCK TABLES;

--
-- Table structure for table `estadoinstructor`
--
DROP TABLE IF EXISTS `estadoinstructor`;
/*I40101 SET @saved_cs_client = @@character_set_client */;
/*I40101 SET character_set_client = utf8 */;
CREATE TABLE `estadoinstructor` (
  `idEstadoInstructor` int(11) NOT NULL,
  `EstadoInstruc` varchar(50) NOT NULL,
  PRIMARY KEY (`idEstadoInstructor`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
/*I40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `estadoinstructor`
--

LOCK TABLES `estadoinstructor` WRITE;
/*I40000 ALTER TABLE `estadoinstructor` DISABLE KEYS */;

```

Figure 6.Creación de Back Up





```
Administrador: Símbolo del sistema
-- Table structure for table `usuario`
--
DROP TABLE IF EXISTS `usuario`;
/*!40101 SET @saved_cs_client      = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `usuario` (
  `usuPassword` varchar(255) NOT NULL,
  `Correo_SENA` varchar(50) NOT NULL,
  `ROL_id_rol` int(11) NOT NULL,
  PRIMARY KEY (`Correo_SENA`),
  KEY `ROL_id_rol` (`ROL_id_rol`) USING BTREE
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
/*!40101 SET character_set_client = @saved_cs_client */;

-- Dumping data for table `usuario`
--
LOCK TABLES `usuario` WRITE;
/*!40000 ALTER TABLE `usuario` DISABLE KEYS */;
INSERT INTO `usuario` VALUES ('09e6762dd1c8eaf6d61b3c6192fc408d4d6d5f1176d0c29169bc24e71c3f274ad27fcd5811b313d681f7e55ec02d73d499c95455b6b5bb503acf574fba8ffe85','lkcalderon46@misena.edu.co',2),('d9e6762dd1c8eaf6d61b3c6192fc408d4d6d5f1176d0c29169bc24e71c3f274ad27fcd5811b313d681f7e55ec02d73d499c95455b6b5bb503acf574fba8ffe85','mw@misena.edu.co',1);
/*!40000 ALTER TABLE `usuario` ENABLE KEYS */;
UNLOCK TABLES;
/*!40103 SET TIME_ZONE=@OLD_TIME_ZONE */;

/*!40101 SET SQL_MODE=@OLD_SQL_MODE */;
/*!40014 SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS */;
/*!40014 SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS */;
/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
/*!40111 SET SQL_NOTES=@OLD_SQL_NOTES */;

-- Dump completed on 2021-09-10 16:27:52
```

Figure 7.Creación de Back Up





- Luego de que se carguen nuestras tablas, creamos una carpeta llamada “BD”, ingresamos el siguiente código dirigiéndonos a la carpeta creada (Mysqldump -u root -p abgs > C:\wamp\BD\abgs.sql), con un archivo .sql que se creará automáticamente con el nombre de nuestra base de datos (abgs.sql) e ingresamos nuevamente la contraseña:

```
C:\wamp64\bin\mysql\mysql5.7.31\bin>mysqldump -u root -p abgs > C:\wamp64\BD\abgs.sql
Enter password:
```

Figure 8. Procedimiento Migración

- Nos dirigimos a la carpeta de “Strawberry” en el Disco C:, luego a la carpeta “perl” y finalmente en “bin” y agregamos el siguiente código: perl mysql2pgsql.pl --nodrop C:\wamp\BD\abgs.sql C:\wamp\BD\abgsPostgreSQL.sql. De esta manera, la información de las tablas empezará a trasladarse de MySQL a PostgreSQL:

```
C:\wamp64\bin\mysql\mysql5.7.31\bin>mysqldump -u root -p abgs > C:\wamp64\BD\abgs.sql
Enter password:

C:\wamp64\bin\mysql\mysql5.7.31\bin>cd C:\Strawberry\perl\bin

C:\Strawberry\perl\bin>perl mysql2pgsql.pl --nodrop C:\wamp64\BD\abgs.sql C:\wamp64\BD\abgsPostgreSQL.sql
Dumping aprendiz...
Dumping estado...
Dumping estadoinstructor...
Dumping inasistencia...
Dumping instructor...
Dumping login...
Dumping notas...
Dumping programa...
Dumping r_aprendizaje...
Dumping roles...
Dumping solicitudes...
Dumping tipoinstructor...
Dumping usuario...

C:\Strawberry\perl\bin>_
```

Figure 9. Procedimiento Migración



1. Instalación de PostgreSQL:

- Se crea una contraseña para la nueva base de datos.

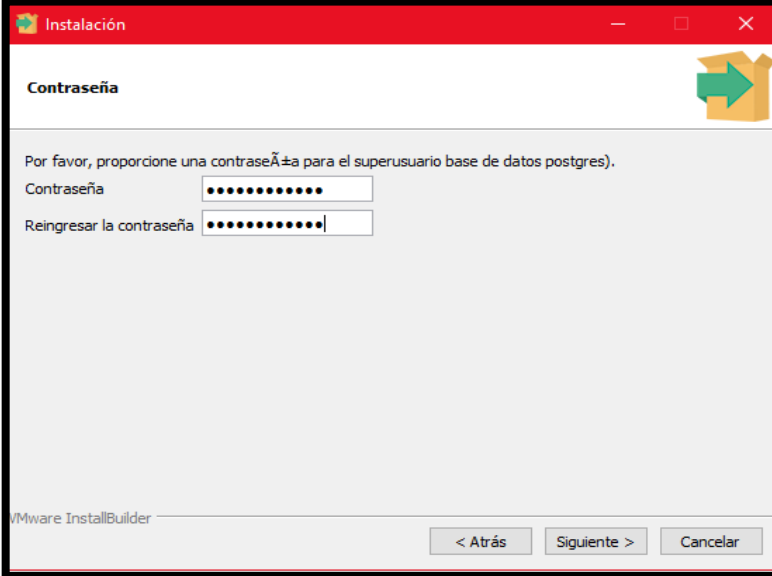


Figure 10. Instalación PostgreSQL

el Puerto donde se conectará.

- Seleccionamos

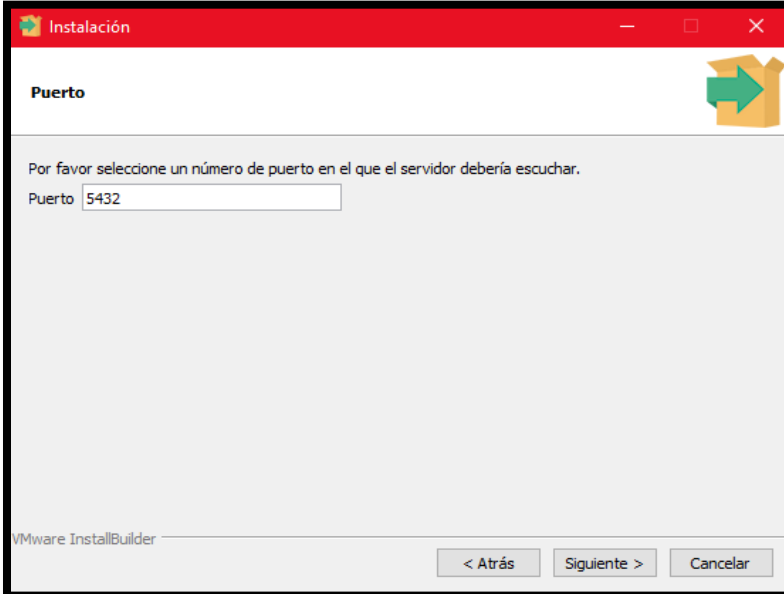


Figure 11. Instalación PostgreSQL



- Esperamos a que se cargue la información de instalación y seleccionamos “Finalizar”

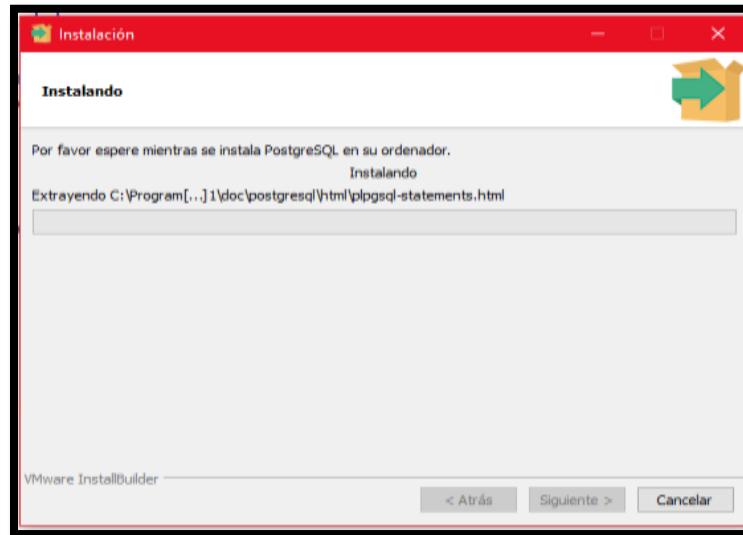


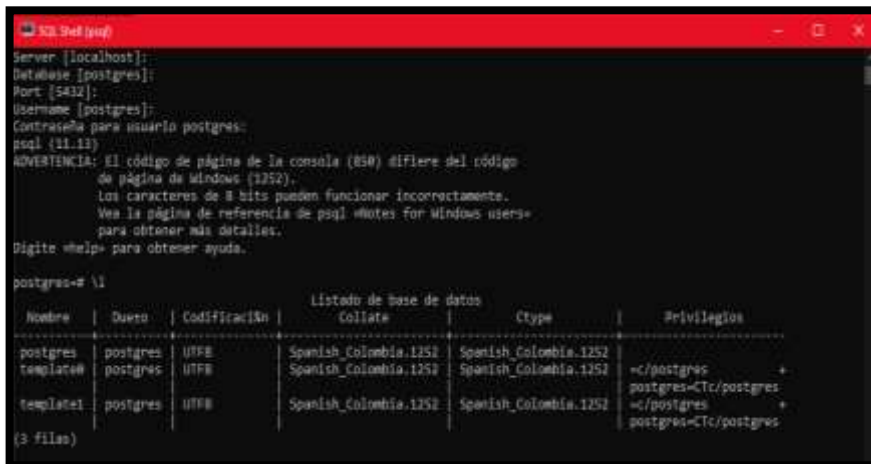
Figure 12.Instalación PostreSQL



Figure 13.Instalación PostreSQL

## 2. Visualización de PostgreSQL:

- Abrimos “SQL Shell”, para visualizar el listado de base de datos ingresamos \l y de esta manera notamos que nuestra base de datos no existe.



```

SQL Shell (psql)
Server [localhost]:
Database [postgres]:
Port [5432]:
Username [postgres]:
Contraseña para usuario postgres:
psql (11.13)
ADVERTENCIA: El código de página de la consola (850) difiere del código
de página de Windows (1252).
Los caracteres de 8 bits pueden funcionar incorrectamente.
Vea la página de referencia de psql «Notes for Windows users»
para obtener más detalles.
Digite «help» para obtener ayuda.

postgres=# \l
          Listado de base de datos

```

Nombre	Dueño	Codificación	Collate	Ctype	Privilegios
postgres	postgres	UTF8	Spanish_Colombia.1252	Spanish_Colombia.1252	
template0	postgres	UTF8	Spanish_Colombia.1252	Spanish_Colombia.1252	+</postgres
template1	postgres	UTF8	Spanish_Colombia.1252	Spanish_Colombia.1252	+</postgres

(3 filas)

Figure 14. Listado de base de datos PostgreSQL

- Por ende,

procedemos a crearla con \c abgs; 'CREATE DATABASE' y verificamos con \l



```

postgres=# create database abgs;
CREATE DATABASE
postgres=# \l
          Listado de base de datos

```

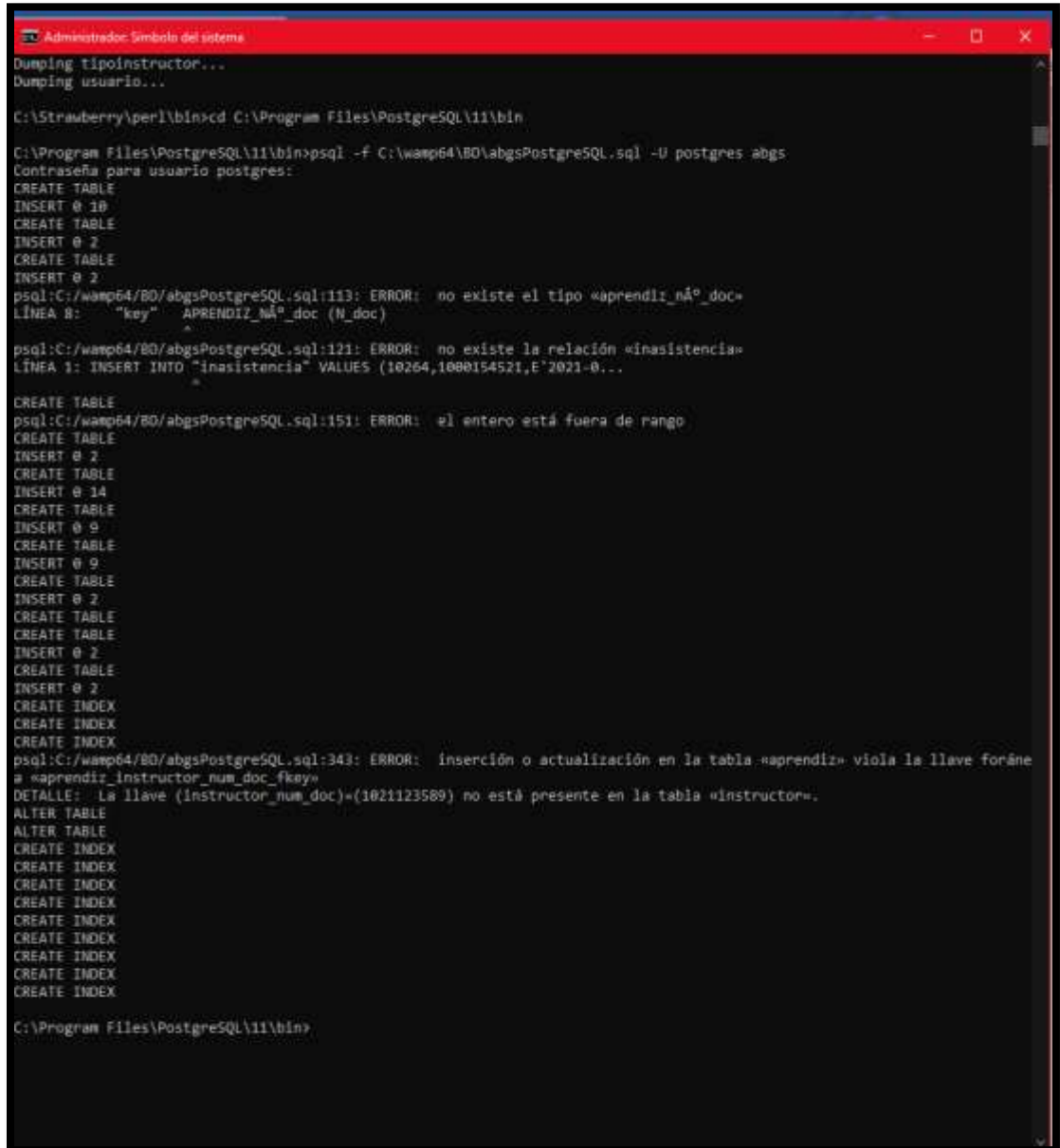
Nombre	Dueño	Codificación	Collate	Ctype	Privilegios
abgs	postgres	UTF8	Spanish_Colombia.1252	Spanish_Colombia.1252	
postgres	postgres	UTF8	Spanish_Colombia.1252	Spanish_Colombia.1252	
template0	postgres	UTF8	Spanish_Colombia.1252	Spanish_Colombia.1252	+</postgres
template1	postgres	UTF8	Spanish_Colombia.1252	Spanish_Colombia.1252	+</postgres

(4 filas)

Figure 15. Actualización listado de bases de datos



- Luego de esto nos dirigimos al CMD e ingresamos la dirección en el Disco C:/Program Files/ PostgreSQL/11/bin, enter, y luego ingresamos el siguiente código: `psql -f C:\wamp\BD\abgsPostgreSQL.sql -U postgres abgs`. De esta manera estamos permitiendo a PostgreSQL que transfiera la información.



```

Administrador: Símbolo del sistema
Dumping tipoInstructor...
Dumping usuario...

C:\Strawberry\perl\bin>cd C:\Program Files\PostgreSQL\11\bin

C:\Program Files\PostgreSQL\11\bin>psql -f C:\wamp64\BD\abgsPostgreSQL.sql -U postgres abgs
Contraseña para usuario postgres:
CREATE TABLE
INSERT 0 10
CREATE TABLE
INSERT 0 2
CREATE TABLE
INSERT 0 2
psql:C:\wamp64\BD\abgsPostgreSQL.sql:113: ERROR: no existe el tipo «aprendiz_ná°_doc»
LÍNEA 8:      "key"      APRENDIZ_NA°_doc (N_doc)
                ^
psql:C:\wamp64\BD\abgsPostgreSQL.sql:121: ERROR: no existe la relación «inasistencia»
LÍNEA 1: INSERT INTO "inasistencia" VALUES (10264,1000154521,E'2021-0...
                ^
CREATE TABLE
psql:C:\wamp64\BD\abgsPostgreSQL.sql:151: ERROR: el entero está fuera de rango
CREATE TABLE
INSERT 0 2
CREATE TABLE
INSERT 0 14
CREATE TABLE
INSERT 0 9
CREATE TABLE
INSERT 0 9
CREATE TABLE
INSERT 0 2
CREATE TABLE
INSERT 0 2
CREATE INDEX
CREATE INDEX
psql:C:\wamp64\BD\abgsPostgreSQL.sql:343: ERROR: inserción o actualización en la tabla «aprendiz» viola la llave foránea
«aprendiz_instructor_num_doc_fkkey»
DETALLE: La llave (instructor_num_doc)=(1021123589) no está presente en la tabla «instructor».
ALTER TABLE
ALTER TABLE
CREATE INDEX
CREATE INDEX
CREATE INDEX
CREATE INDEX
CREATE INDEX
CREATE INDEX
CREATE INDEX
CREATE INDEX
CREATE INDEX
C:\Program Files\PostgreSQL\11\bin>

```

Figure 16. Transferencia de información

- De esta manera, al volver a SQL Shell, ya podemos ingresar a la base de datos.

```
postgres=# \c abgs
Ahora está conectado a la base de datos «abgs» con el usuario «postgres».
abgs=# \dt
          Listado de relaciones
Esquema |      Nombre      | Tipo | Dueño
-----+-----+-----+-----
public  | aprender         | tabla | postgres
public  | estado           | tabla | postgres
public  | estadoinstructor | tabla | postgres
public  | instructor       | tabla | postgres
public  | login            | tabla | postgres
public  | notas            | tabla | postgres
public  | programa         | tabla | postgres
public  | r_aprendizaje    | tabla | postgres
public  | roles            | tabla | postgres
public  | solicitudes      | tabla | postgres
public  | tipoinstructor   | tabla | postgres
public  | usuario          | tabla | postgres
(12 filas)
```

Figure 17. Ingreso a base de datos

- Finalmente realizamos una consulta para verificar el funcionamiento de la Base de Datos.

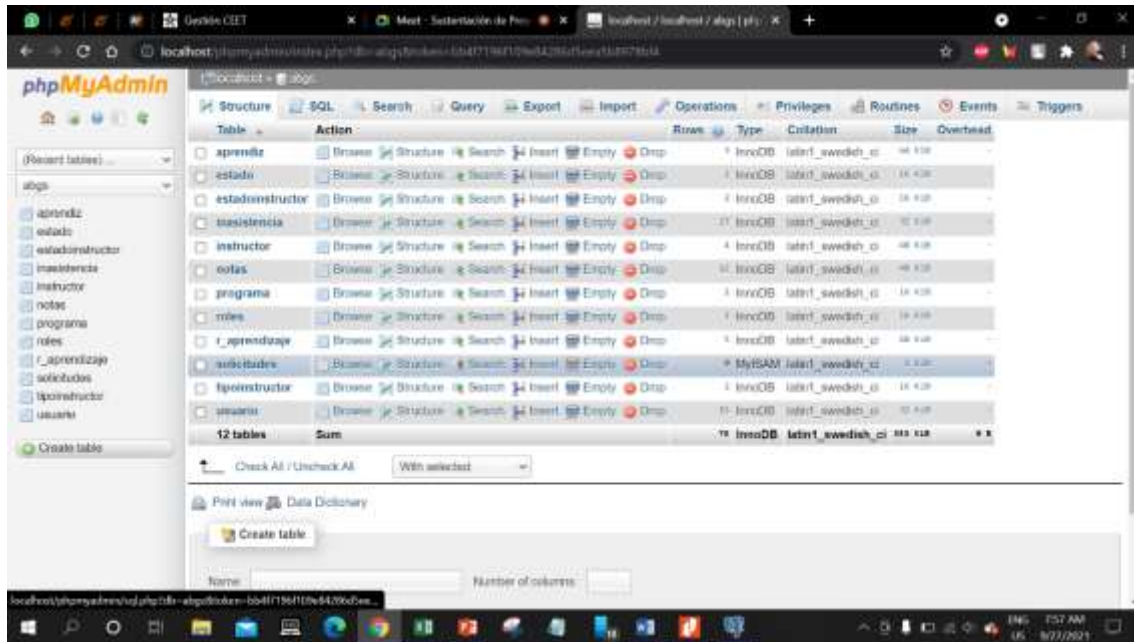
```
SQL Shell (psql)
abgs=# select * from estado;
 idestado | nombre
-----+-----
        1 | Activo
        2 | Inactivo
(2 filas)

abgs=# select * from aprender;
 n_doc | estado_idestado | programa_ficha_carac | instructor_num_doc | tip_doc | nombres | apellid
-----+-----+-----+-----+-----+-----+-----
2067470 | 2 | 2067472 | 1021123589 | C.C | Alison yulieth | Calderon sanc
2067476 | 2 | 2067472 | 10005890 | C.C | Gustavo andres | Calderon Fern
2067476 | 2 | 2067472 | 10005890 | C.C | Karen | Casas
90508247 | 1 | 102145 | 10005890 | C.C | Santiago alexander | Trigos casas
3201457854 | 2 | 2067472 | 1021123589 | C.C | Wilmer Edison | Mendez
3201457892 | 2 | 2067472 | 10005890 | C.C | Willa | Willa
3201457854 | 2 | 2067472 | 1000215896 | C.C | Leidy Katherine | Calderon cast
3204473192 | 1 | leidy.khaterinecc2002@gmail.com | 1000215896 | C.C | Camilo | Cruz soler
300725494 | 1 | cami89@gmail.com | 1000215896 | C.C | Brayan | Cabrera hurta
3201457892 | 1 | brayan@gmail.com | 10005890 | C.C | Ronald | Casas
3201457854 | 2 | ronald@gmail.com | 10005890 | C.C | Augusto | Solanilla
3201457854 | 1 | august@gmail.com | 10005890 | C.C | Augusto | Solanilla
(10 filas)
```

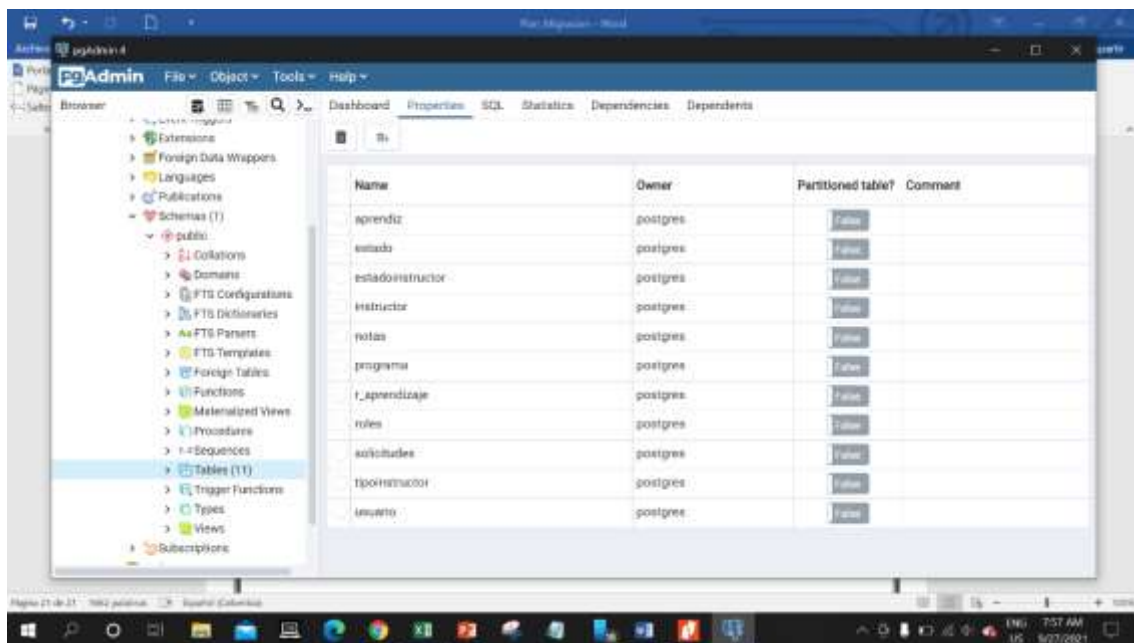
Figure 18. Comprobación de la Migración



Base de datos en MySQL :



Base de datos en PostgreSQL:



Copyright © 2021

Calle 49ª Bis sur # 10d -20

**Absence and Bad Grades Software**  
 Todos los Derechos Reservados

320 447 3192

313 710 3188