

INFORME FINAL DE INVESTIGACIÓN

Desarrollo de una Aplicación de Software para la
Migración de Datos de una Estructura Relacional
a una Estructura de Big Data

DOCENTE INVESTIGADORA PRINCIPAL:
LIC. LILIAN JUDITH SANDOVAL SERRANO

ESCUELA DE INGENIERÍA DE COMPUTACIÓN ITCA-FEPADE SEDE CENTRAL – SANTA TECLA

FEBRERO 2017



ESCUELA ESPECIALIZADA EN INGENIERÍA ITCA-FEPADE
DIRECCIÓN DE INVESTIGACIÓN Y PROYECCIÓN SOCIAL
SANTA TECLA, LA LIBERTAD, EL SALVADOR, CENTRO AMÉRICA



Comisión de Acreditación
de la Calidad Académica
INSTITUCIÓN
ACREDITADA
2015-2020

INFORME FINAL DE INVESTIGACIÓN

Desarrollo de una Aplicación de Software para la
Migración de Datos de una Estructura Relacional
a una Estructura de Big Data

DOCENTE INVESTIGADORA PRINCIPAL:
LIC. LILIAN JUDITH SANDOVAL SERRANO

ESCUELA DE INGENIERÍA DE COMPUTACIÓN ITCA-FEPADE SEDE CENTRAL – SANTA TECLA

FEBRERO 2017



ESCUELA ESPECIALIZADA EN INGENIERÍA ITCA-FEPADE
DIRECCIÓN DE INVESTIGACIÓN Y PROYECCIÓN SOCIAL
SANTA TECLA, LA LIBERTAD, EL SALVADOR, CENTRO AMÉRICA



Rectora

Licda. Elsy Escolar SantoDomingo

Vicerrector Académico

Ing. Carlos Alberto Arriola Martínez

Vicerrectora Técnica Administrativa

Inga. Frineé Violeta Castillo

**Dirección de Investigación
y Proyección Social**

Ing. Mario Wilfredo Montes, Director

Ing. David Emmanuel Ágreda

Inga. Lorena Victoria Ramírez de Contreras

Sra. Edith Aracely Cardoza

Directora Escuela de

Ingeniería de Computación

Ing. Marta Corina Quijano de García

005.74

S218d

SV

Sandoval Serrano, Lilian Judith, 1968-

Desarrollo de una aplicación de software para la migración de datos de una estructura relacional a una estructura Big Data / Lilian Judith Sandoval Serrano. – 1ª ed. - Santa Tecla, La Libertad, El Salv. : ITCA Editores, 2017. 68 p. ; 28 cm.

ISBN : 978-99961-50-52-4

1. Sistemas de almacenamiento y recuperación de la información. 2. Bases de datos. 3. Redes de información. 4. Recuperación de Información. I. Título.

Autor

Lic. Lilian Judith Sandoval Serrano

Docentes de Apoyo

Tec. Giovanni Ariel Tzec Chávez

Tec. Guillermo Adalberto Jandres Escobar

Tiraje: 13 ejemplares

Año 2017

Este documento técnico es una publicación de la Escuela Especializada en Ingeniería ITCA-FEPADE; tiene el propósito de difundir la Ciencia, la Tecnología y la Innovación CTI, entre la comunidad académica y el sector empresarial, como un aporte al desarrollo del país. El contenido de este informe de investigación no puede ser reproducido parcial o totalmente sin previa autorización escrita de la Escuela Especializada en Ingeniería ITCA-FEPADE. Para referirse al contenido debe citar el nombre del autor y el título del documento. El contenido de este Informe es responsabilidad de los autores.

Escuela Especializada en Ingeniería ITCA-FEPADE

Km 11.5 carretera a Santa Tecla, La Libertad, El Salvador, Centro América

Sitio web: www.itca.edu.sv

TEL: (503)2132-7423

FAX: (503)2132-7599

CONTENIDO

1.	INTRODUCCIÓN	4
2.	PLANTEAMIENTO DEL PROBLEMA	4
2.1.	DEFINICIÓN DEL PROBLEMA	4
2.2.	JUSTIFICACIÓN	5
3.	OBJETIVOS	5
3.1.	OBJETIVO GENERAL	5
3.2.	OBJETIVOS ESPECÍFICOS	5
4.	HIPÓTESIS	5
5.	MARCO TEÓRICO	6
5.1.	MONGO DB	8
6.	METODOLOGÍA DE INVESTIGACIÓN	10
7.	RESULTADOS.	10
7.1.	MODELADO DE LA BASE DE DATOS RELACIONAL	10
7.2.	PROCESO DE MIGRACIÓN DE DATOS.	27
7.3.	DISEÑO DE LA APLICACIÓN	32
7.4.	DESARROLLO DE LA APLICACIÓN.	38
7.5.	PRUEBAS DE LA APLICACIÓN	38
8.	CONCLUSIONES	45
9.	RECOMENDACIONES	47
10.	REFERENCIAS BIBLIOGRÁFICAS	47
11.	ANEXOS.	48

1. INTRODUCCIÓN

Con el nacimiento de las redes sociales y las aplicaciones móviles, surgió la necesidad de manejar volúmenes de información gigantescos y variados, que las bases de datos tradicionales ya no pueden soportar, además de lo complejo que se ha vuelto el proceso de búsqueda de información. Por lo anterior, se ha hecho necesario pensar en estructuras de datos completamente distintas, donde la limitación de espacio no fuera más un problema. Por ello se ha creado la nueva tecnología Big Data para manejo de volúmenes de datos e interpretación de ellos para diferentes propósitos.

Big Data maneja conjuntos de datos enormes que crecen tan rápido que se vuelve muy difícil manipular y analizar a una granularidad tal donde los procesos normales colapsan.

Esta nueva tecnología no solo viene a resolver los problemas de almacenamiento y gestión que plantean las redes sociales, también auxilia a otros sectores que presentaban las mismas dificultades como el científico, el médico, el mercadológico, etc.

En este sentido y en vista de la dificultad que se aproxima para la mayor parte de empresas que en la actualidad manejan su información con bases de datos tradicionales (relacionales), se ve la necesidad de diseñar una aplicación e implementar una solución que permita migrar esa información de forma rápida y segura a la nueva tecnología Big Data. De esta manera no es necesario comenzar a cargar los datos desde cero, ya que además de generar la estructura de la base de datos Big Data, también se realizará la extracción de los datos.

2. PLANTEAMIENTO DEL PROBLEMA

2.1. DEFINICIÓN DEL PROBLEMA

Por la creciente cantidad de datos que generan las aplicaciones en la actualidad, como las aplicaciones móviles, las redes sociales, transacciones comerciales en línea, etc.; las bases de datos tradicionales no dan abasto y una nueva tecnología llamada Big Data va comenzando a tomar su lugar.

La problemática radica en que aquella información que está almacenada en bases de datos de tipo relacional (esquemas que se usan actualmente), en algún momento requerirá ser escalada a esta nueva tecnología. Algunas empresas comercializadoras de gestores de bases de datos propietarios como Oracle están actualizándose y ofreciendo esta nueva tecnología. En este caso no hay inconveniente pues ellos se encargarán de la migración de datos a Big Data, pero para la mayoría de gestores no hay una solución que se encuentre documentada al momento de manera formal.

2.2. JUSTIFICACIÓN

En vista de la dificultad que se aproxima para la mayor parte de empresas que en la actualidad manejan su información con bases de datos tradicionales (relacionales), se ve la necesidad de implementar una solución que permita migrar esa información de forma rápida y segura a la nueva tecnología Big Data. De esta manera no es necesario comenzar a cargar los datos desde cero, ya que la aplicación además de generar la estructura de la base de datos, también se encargará de la extracción de los datos.

3. OBJETIVOS

3.1. OBJETIVO GENERAL

Desarrollar una aplicación para la migración de datos de un esquema de base de datos relacional a un esquema Big Data, con el fin de permitir que las empresas conserven sus aplicaciones actuales y su información.

3.2. OBJETIVOS ESPECÍFICOS

1. Modelar una base de datos Big Data teniendo como fuente una base de datos con esquema relacional para determinar las sentencias necesarias a utilizar en la migración.
2. Definir y ejecutar los procesos de migración de una estructura de base de datos relacional a una estructura Big Data, para ser incorporados luego en la aplicación.
3. Verificar la integridad de los datos migrados del esquema relacional al esquema Big Data para confirmar que todo el procedimiento se realizó de una manera óptima.
4. Desarrollar una aplicación que genere el esquema de cualquier base de datos relacional a una base de datos Big Data y la migración de los datos y hacer pruebas de migración y de stress.

4. HIPÓTESIS

Al implementar una base de datos de tecnología Big Data no es necesario comenzar con la carga de los datos desde cero si existe una aplicación que lo haga de forma automática, creando la estructura de la base de datos y migrando la información.

Al migrar la base de datos a la tecnología Big Data, además de conservar los datos en su totalidad, se incrementará notablemente su rendimiento.

5. MARCO TEÓRICO

Big Data es una disciplina que se ocupa de todas las actividades relacionadas con los sistemas que manipulan grandes conjuntos de datos. Las dificultades más habituales vinculadas a la gestión de estas cantidades de datos se centran en la recolección y el almacenamiento, búsqueda, compartición, análisis, y visualización. La tendencia a manipular enormes cantidades de datos se debe a la necesidad en muchos casos de incluir dicha información para la creación de informes estadísticos y modelos predictivos utilizados en diversas materias, como los análisis de negocio, publicitarios, los datos de enfermedades infecciosas, el espionaje y seguimiento a la población o la lucha contra el crimen organizado.

Big Data maneja conjuntos de datos enormes que crecen tan rápido que se vuelve muy difícil manipular y analizar a una granularidad tal donde los procesos colapsan.

Esta nueva tecnología no solo viene a resolver los problemas de almacenamiento y gestión que plantean las redes sociales, sino que también auxilia a otros sectores que también presentaban las mismas dificultades como el científico, el médico, el mercadológico, etc.

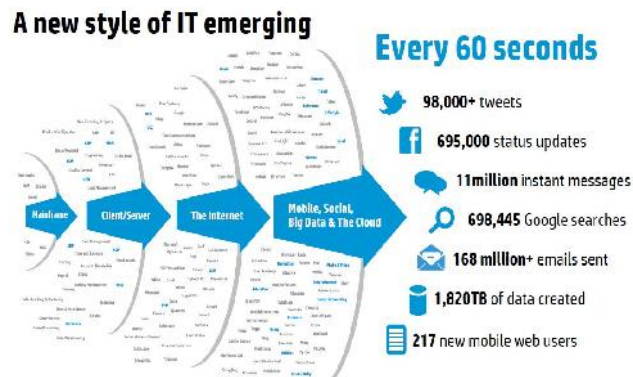


Fig. 1. Velocidad de la información.

A. Tipos de Datos.

Existen tres tipos de datos en el ambiente de Big Data:

-) **Estructurados:** Son datos que tienen bien definidos su tipo, ya sea texto, numérico, fecha, etc. Por lo general estos datos son almacenados en tablas en un sistema de base de datos relacional.
-) **No estructurados:** Son datos que conservan el formato con el que fueron recolectados, carecen de un formato específico. No se pueden almacenar dentro de una tabla ya que no se puede desgranar su información a tipos básicos de datos. Algunos ejemplos son los [PDF](#), documentos multimedia, [e-mails](#), etc.
-) **Semi estructurados:** Son datos que no se limitan a tipos determinados, pero que contiene marcadores para separar los diferentes elementos. Es información poco regular como para ser gestionada de una forma estándar. Estos datos poseen sus propios [metadatos](#) semiestructurados que describen los objetos y las relaciones entre ellos y pueden acabar siendo aceptados por convención. Algunos ejemplos son [HTML](#), [XML](#) y [JSON](#).

B. Almacenamiento NoSQL.

El término [NoSQL](#) significa Not Only SQL y son sistemas de almacenamiento que no cumplen con el esquema entidad-relación. Proveen un sistema de almacenamiento mucho más flexible y concurrente y permiten manipular grandes cantidades de información de manera mucho más rápida que las [bases de datos relacionales](#).

Existen cuatro tipos de almacenamiento [NoSQL](#):

- J) [Almacenamiento Clave-Valor \(Key-Value\)](#): Son sistemas de almacenamiento donde se accede al dato a partir de una clave única. Los valores son aislados e independientes entre ellos y no son interpretados por el sistema. Pueden ser [enteros](#), [caracteres](#) u [objetos](#). Por otro lado, este sistema de almacenamiento carece de una estructura de datos clara y establecida, por lo que no requiere un formateo de los datos muy estricto. Son útiles para operaciones simples basadas en claves. Un ejemplo es el aumento de velocidad de carga de un sitio [web](#) que pueden utilizar diferentes perfiles de [usuario](#), teniendo mapeados los archivos que hay que incluir según el id de usuario y que han sido calculados con anterioridad. [Cassandra](#) es la tecnología de almacenamiento clave-valor más reconocida por los usuarios.
- J) [Almacenamiento Documental](#): Bases de datos con este sistema de almacenamiento guardan un gran parecido con las bases de datos Clave-Valor, diferenciándose en el dato que guardan. Si en la anterior no requería una [estructura de datos](#) concreta, en este caso si se guardan datos semiestructurados. Estos datos pasan a llamarse documentos, y pueden estar formateados en [XML](#), [JSON](#), o en el formato que acepte la misma [base de datos](#). Un ejemplo de este tipo de almacenamiento es un [blog](#): se almacena el autor, la fecha, el título, el resumen y el contenido del post. [CouchDB](#) o [MongoDB](#) son las bases de datos documentales más conocidas.
- J) [Almacenamiento en Grafo](#): Las [bases de datos en grafo](#) rompen con la idea de tablas y se basan en la [teoría de grafos](#), donde se establece que la información son los nodos y las relaciones entre la información son las aristas. Relacionan grandes cantidades de datos que pueden ser muy variables. Por ejemplo, los [nodos](#) pueden contener [objetos](#), [variables](#) y [atributos](#) diferentes unos de los otros. Las uniones se sustituyen por recorridos a través del grafo, y se guarda una [lista de adyacencias](#) entre los nodos. Un ejemplo es el [Facebook](#), donde cada usuario es un nodo que puede tener [aristas de amistad](#) con otros usuarios, o [aristas de publicación](#) con [nodos de contenidos](#). Soluciones como [Neo4J](#) y [GraphDB](#) son las más conocidas dentro de las [bases de datos en grafo](#).
- J) [Almacenamiento Orientado a Columnas](#): Este sistema de almacenamiento es similar al [Documental](#). Su modelo de datos es definido como “un mapa de datos [multidimensional](#) poco denso, distribuido y persistente. Se orienta a almacenar datos con tendencia a escalar horizontalmente, por lo que permite guardar diferentes [atributos](#) y [objetos](#) bajo una misma Clave. A diferencia del [Documental](#) y el Key-Value, en este caso podremos almacenar varios [atributos](#) y [objetos](#), pero no serán interpretables directamente por el sistema. Permite agrupar columnas en familias y guardar la información cronológicamente, mejorando el rendimiento. Esta tecnología se utiliza en casos de contar 100 o más atributos por clave. Su precursor es [BigTable](#) de Google, pero han aparecido nuevas soluciones como HBase o HyperTable.

5.1. MONGO DB

Es un sistema de bases de datos no relacionales, multiplataforma e inspirada en el tipo de bases de datos documental y clave/valor. Su nombre proviene del término en inglés "humongous" que significa "enorme".

Está liberada bajo licencia de software libre, específicamente GNU AGPL 3.0. MongoDB usa el formato BSON (JSON Compilado) para guardar la información, dando la libertad de manejar un esquema libre. Este motor de bases de datos es uno de los más conocidos y usados, pudiéndolo comparar en popularidad con MySQL en el caso de las bases de datos relacionales.

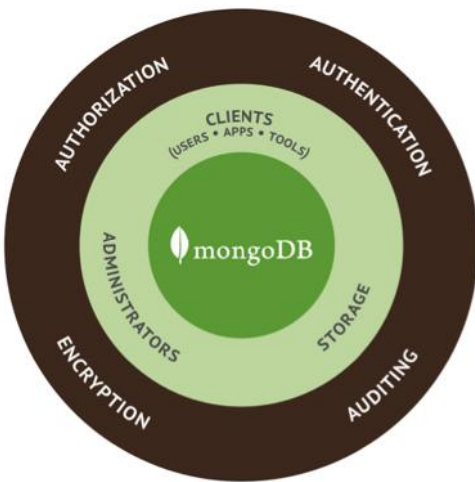


Fig. 2. Seguridad en MongoDB.

JSON - JavaScript Object Notation

JSON es el formato compacto de representación de objetos para intercambio de datos. Las especificaciones las publicó Douglas Crockford en el documento RFC 6274. JSON es un formato independiente del lenguaje, aunque su uso extendido hasta hace poco era en el lenguaje JavaScript. Actualmente se usa JSON en gran cantidad de sistemas para intercambiar información por su simplicidad en comparación con XML. Este formato soporta gran cantidad de tipos de datos, lo que lo hace atractivo para un uso generalizado, y cada vez más lenguajes de programación dan soporte a este formato.

Documento

Un documento es un conjunto de datos estructurados (mas no con un esquema estricto), que contiene pares clave/valor, y se usa BSON (JSON Binario) como formato para almacenar los documentos. Un documento puede ser comparado con una fila o registro en una base de datos relacional.

Colección

Es un conjunto de documentos, similar a una tabla en las bases de datos relacionales.

CRUD - IFUR

El propósito de las bases de datos no es solo almacenar información, sino manipular esa información de distintas maneras con el objetivo de alimentar un conjunto de procesos previamente establecidos en los sistemas informáticos.

En los sistemas de base de datos existen 4 funciones básicas las cuales son: Crear, Leer, Modificar y Eliminar (Create, Read, Update and Delete - CRUD), de estas 4 funciones básicas derivan todo un sin fin de funciones para el tratamiento de la información.

MongoDB provee un grupo de métodos en JavaScript para realizar CRUD y es conocido como IFUR (Insert, Find, Update and Remove).

MIGRACION DE DATOS A MONGODB

Los equipos de proyectos tienen múltiples opciones para importar datos desde una base de datos relacional existente hacia MongoDB. La herramienta a escoger deberá depender de la fase en que se encuentre el proyecto y del entorno existente. Entre las diferentes opciones que se pueden tomar están:

1. Crear los scripts necesarios para transformar los datos de origen, ya sea dentro o fuera del sistema gestor de base de datos relacional en una estructura jerárquica en formato de JSON que pueda ser importada a MongoDB usando la instrucción **MongoImport**, que para efectos de diseño de la aplicación, será el método a utilizar.
2. Utilizar herramientas ETL (Extract, Transform and Load) que también son muy comúnmente usadas para la transformación de datos desde bases de datos relacionales a MongoDB. Pentaho, Informatica y Talend son algunos de los proveedores de ETL que han desarrollado conectores para MongoDB que habilitan el flujo de trabajo en que los datos son extraídos desde la base de datos origen, transformados en un esquema MongoDB, y una vez en esa etapa, son cargados en colecciones de documentos.

Migración Incremental.

En muchos casos se hace necesario ejecutar el motor del sistema gestor de bases de datos relacional en paralelo con la base de datos de MongoDB incrementalmente transfiriendo los datos en producción:

- ⌋ Como registros, que son restaurados desde el RDBMS, la aplicación los escribe de nuevo en MongoDB en el esquema de documento requerido.
- ⌋ Verificando la consistencia, por ejemplo, usando checksums MD5 se puede validar los datos migrados y asegurarse que su integridad.
- ⌋ Todos los datos creados o actualizados nuevamente, solo son escritos a MongoDB.

La migración incremental puede ser usada cuando características de una nueva aplicación son implementadas con MongoDB o cuando múltiples aplicaciones están corriendo contra el RDBMS heredado. Migrar solo aquellas aplicaciones que están siendo modernizadas permite a los equipos dividir los proyectos en otros más manejables y con sprints de desarrollo ágiles.

La migración incremental elimina la interrupción del servicio mientras que también proporciona la disponibilidad de poder realizar un fail-back para resincronizar los datos de nuevo a la base de datos original.

6. METODOLOGÍA DE INVESTIGACIÓN

La metodología aplicada a la investigación es la exploratoria, ya que la temática en investigación, a pesar de tener ya algunos años de existencia, en nuestro país es relativamente nueva. Se investigarán los diferentes procesos para la migración de datos de una tecnología ya ampliamente utilizada, como es el modelo relacional, hacia la nueva tecnología BigData, donde no se conoce si existe un método ya definido o es necesario definir uno, y determinar las herramientas que pudieran ser de utilidad para lograrlo.

Las fases en que se seguirán para el desarrollo del proyecto son las siguientes:

Fase 1: Investigación Técnica.

Consiste en hacer una investigación de los procedimientos necesarios para realizar una migración de la información de un esquema relacional a un esquema Big Data.

Fase 2: Modelado de la Base de Datos.

Se modelará una base de datos Big Data utilizando el sistema gestor MongoDB, tomando como fuente una base de datos relacional creada en SQL Server. Posteriormente se definirán los procesos de migración haciendo uso de sentencias de MongoDB y se verificará la integridad de los datos para confirmar los procedimientos de migración.

Fase 3: Desarrollo de la aplicación.

Se desarrollará una aplicación en un entorno web que permitirá la migración de una base de datos relacional a una de tecnología Big Data, mostrando una interfaz amigable para el usuario. Para el desarrollo de la aplicación se utilizarán las herramientas PHP, JSON, Ajax, JQuery, JavaScript, HTML5 y CSS3.

Fase 4: Prueba de la aplicación.

Ejecución de la migración haciendo uso de la aplicación desarrollada en la fase anterior y verificación de la información obtenida como resultado del proceso.

Fase 5: Prueba de stress.

Prueba de velocidad de recuperación de datos usando grandes volúmenes de información para comprobar tiempos de respuesta y verificar la eficiencia de la base de datos Big Data en comparación a la base de datos relacional con el uso de MongoDB.

7. RESULTADOS.

7.1. MODELADO DE LA BASE DE DATOS RELACIONAL.

A continuación se presenta un fragmento del script de la base de datos relacional, creada en el gestor de base de datos SQL Server:

```
DROP DATABASE neptuno;  
CREATE DATABASE neptuno  
USE neptuno
```

```
create table productos(  
    idproducto int primary key,  
    nombreProducto varchar(40),  
    idProveedor int null,  
    idCategoria int null,  
    cantidadPorUnidad varchar(20) null,  
    precioUnidad decimal null,  
    unidadesEnExistencia smallint null,  
    unidadesEnPedido smallint null,  
    nivelNuevoPedido smallint null,  
    suspendido smallint,  
    categoriaProducto varchar(20) null  
);
```

```
-----  
create table proveedores(  
    idProveedor int primary key,  
    nombreCompañia varchar(40) not null,  
    nombrecontacto varchar(30),  
    cargocontacto varchar(30),  
    direccion varchar(60),  
    ciudad varchar(15),  
    region varchar(15),  
    codPostal varchar(10),  
    pais varchar(15),  
    telefono varchar(24),  
    fax varchar(24),  
    paginaprincipal text  
);
```

```
-----  
create table detallesdepedidos(  
    idpedido int,  
    idproducto int,  
    preciounidad decimal not null,  
    cantidad int not null,  
    descuento decimal not null  
);
```

```
-----  
create table companiasdeenvios(  
-----
```

```
idCompañiaEnvios int primary key,  
nombreCompañia varchar(40) not null,  
telefono varchar(24)  
);
```

```
-----  
create table clientes(  
    idCliente varchar(5) primary key,  
    NombreCompañia varchar(100) not null,  
    NombreContacto varchar(100),  
    CargoContacto varchar(100),  
    Direccion varchar(100),  
    Ciudad varchar(100),  
    Region varchar(100),  
    CodPostal varchar(100),  
    Pais varchar(100),  
    Telefono varchar(30),  
    Fax varchar(30)  
);
```

```
-----  
create table Empleados(  
    IdEmpleado int primary key,  
    Apellidos varchar(20),  
    Nombre varchar(20),  
    cargo varchar(40),  
    Tratamiento varchar(40),  
    FechaNacimiento date,  
    FechaContratacion date,  
    direccion varchar(60),  
    ciudad varchar(15),  
    region varchar(15),  
    codPostal varchar(10),  
    pais varchar(15),  
    TelDomicilio varchar(24),  
    Extension varchar(4),  
    notas text,  
    Jefe int,  
    sueldoBasico decimal  
);
```

```
-----  
create table Pedidos(  
    IdPedido int primary key,  
    IdCliente varchar(5) not null,  
    IdEmpleado int not null,
```

```

FechaPedido date,
FechaEntrega date,
FechaEnvio date,
FormaEnvio int,
Cargo decimal,
Destinatario varchar(60),
DireccionDestinatario varchar(60),
CiudadDestinatario varchar(60),
RegionDestinatario varchar(60),
CodPostalDestinatario varchar(60),
PaisDestinatario varchar(60)
);
alter table detallesdepedidos
add constraint fk_1
foreign key (idpedido)
references pedidos (idpedido);

alter table detallesdepedidos
add constraint fk_2
foreign key (idproducto)
references productos (idproducto);

-----
create table categorias(
    idcategoria int primary key,
    nombrecategoria varchar(100) not null,
    descripcion text
);
-----
alter table productos
add constraint fk_3
foreign key (idproveedor)
references proveedores (idproveedor);
-----
alter table pedidos
add constraint fk_4
foreign key (formaenvio)
references companiasdeenvios (idcompaniaenvios);
-----
alter table pedidos
add constraint fk_5
foreign key (idcliente)
references clientes (idcliente);

```

```

-----
alter table pedidos
add constraint fk_6
foreign key (idempleado)
references empleados (idempleado);
-----

alter table productos
add constraint fk_7
foreign key (idcategoria)
references categorias (idcategoria);
/*
    INSERCIONES DE EMPLEADOS
*/
INSERT INTO
empleados(IdEmpleado,apellidos,nombre,cargo,tratamiento,fechaNacimiento,fechaContratacion,direcc
ion,ciudad,region,codpostal,pais,teldomicilio,extension,notas,jefe,sueldoBasico)
values(1,'Davolio','Nancy','Representante de ventas','Srta.','1948-12-08 00:00:00','1992-05-01
00:00:00','507 - 20th Ave. E.Apt. 2A','Seattle','WA','98122','EE.UU.','(206) 555-9857','5467','Su
formación incluye una licenciatura en Psicología por la Universidad del Estado de Colorado. en 1970.
También completó "El arte de la llamada fría". Nancy es miembro de Toastmasters
Internacional.','2,1000.00);
INSERT INTO
empleados(IdEmpleado,apellidos,nombre,cargo,tratamiento,fechaNacimiento,fechaContratacion,direcc
ion,ciudad,region,codpostal,pais,teldomicilio,extension,notas,jefe,sueldoBasico)
values(2,'Fuller','Andrew','Vicepresidente comercial','Dr.','1952-02-19 00:00:00','1992-08-14
00:00:00','908 W. Capital Way','Tacoma','WA','98401','EE.UU.','(206) 555-9482','3457','Andrew
completó su licenciatura en Comercio en 1974 y un doctorado en Marketing Internacional de la
Universidad de Dallas en 1981. Habla con fluidez en francés e italiano y lee el alemán. Ingresó en la
empresa como representante de ventas. fue ascendido en Enero de 1992 y a vicepresidente comercial
en Marzo de 1993. Andrew es miembro de la Mesa redonda de administración de Ventas. la Cámara de
comercio de Seattle y de la Asociación de importadores Rim del Pacífico.','NULL,NULL);
INSERT INTO
empleados(IdEmpleado,apellidos,nombre,cargo,tratamiento,fechaNacimiento,fechaContratacion,direcc
ion,ciudad,region,codpostal,pais,teldomicilio,extension,notas,jefe,sueldoBasico)
values(3,'Leverling','Janet','Representante de ventas','Srta.','1963-08-30 00:00:00','1992-04-01
00:00:00','722 Moss Bay Blvd.','Kirkland','WA','98033','EE.UU.','(206) 555-3412','3355','Janet es
licenciada en Química por la Universidad de Boston (1984). También ha completado un programa de
formación en Gestión de minoristas de alimentación. Janet fue contratada como vendedora asociada en
1991 y fue ascendida a representante de ventas en Febrero de 1992','2,1000.00);
INSERT INTO
empleados(IdEmpleado,apellidos,nombre,cargo,tratamiento,fechaNacimiento,fechaContratacion,direcc
ion,ciudad,region,codpostal,pais,teldomicilio,extension,notas,jefe,sueldoBasico)
values(4,'Peacock','Margaret','Representante de ventas','Sra.','1937-09-19 00:00:00','1993-05-03

```

00:00:00','4110 Old Redmond Rd.','Redmond','WA','98052','EE.UU.','(206) 555-8122','5176','Margaret es licenciada en Literatura inglesa por el Colegio Universitario Concordia (1958). y tiene un master del Instituto Americano de Artes Culinarias (1966). Estuvo asignada temporalmente a la oficina de Londres desde julio hasta Noviembre de 1992.','2,1000.00);

INSERT INTO

empleados(IdEmpleado,apellidos,nombre,cargo,tratamiento,fechaNacimiento,fechaContratacion,direccion,ciudad,region,codpostal,pais,teldomicilio,extension,notas,jefe,sueldoBasico)

values(5,'Buchanan','Steven','Gerente de ventas','Sr.','1955-03-04 00:00:00','1993-10-17 00:00:00','14 Garrett Hill','Londres',NULL,'SW1 8JR','Reino Unido','(71) 555-4848','3453','Steven Buchanan se graduó en la Universidad de St. Andrews. Escocia. en 1976. Tras su ingreso en la empresa. en 1992. dedicó 6 meses a un programa de orientación en la oficina de Seattle y luego volvió a su puesto permanente en Londres.Fue ascendido a gestor de ventas en Marzo de 1993. Ha completado con éxito los cursos de Telemarketing y Gestión de ventas internacional. Habla francés.','2,1000.00);

INSERT INTO

empleados(IdEmpleado,apellidos,nombre,cargo,tratamiento,fechaNacimiento,fechaContratacion,direccion,ciudad,region,codpostal,pais,teldomicilio,extension,notas,jefe,sueldoBasico)

values(6,'Suyama','Michael','Representante de ventas','Sr.','1963-07-02 00:00:00','1993-10-17 00:00:00','Coventry HouseMiner Rd.','Londres',NULL,'EC2 7JR','Reino Unido','(71) 555-7773','428','Michael se graduó en la Universidad de Sussex (MA. economía. 1983) y la Universidad de California en Los Angeles (MBA. marketing. 1986). También ha asistido a cursos de Ventas multiculturales y Administración del tiempo para profesionales de ventas. Habla japonés y lee y escribe en francés. portugués y español.','5,NULL);

INSERT INTO

empleados(IdEmpleado,apellidos,nombre,cargo,tratamiento,fechaNacimiento,fechaContratacion,direccion,ciudad,region,codpostal,pais,teldomicilio,extension,notas,jefe,sueldoBasico)

values(7,'King','Robert','Representante de ventas','Sr.','1960-05-29 00:00:00','1994-01-02 00:00:00','Edgeham HollowWinchester Way','Londres',NULL,'RG1 9SP','Reino Unido','(71) 555-5598','465','Robert King sirvió en la organización Peace Corps y viajó mucho. antes de completar su licenciatura en inglés en la Universidad de Michigan en 1992. el año en que ingresó en la empresa. Después de completar un curso denominado "Ventas en Europa". fue transferido a la oficina de Londres en Marzo de 1993.','5,NULL);

INSERT INTO

empleados(IdEmpleado,apellidos,nombre,cargo,tratamiento,fechaNacimiento,fechaContratacion,direccion,ciudad,region,codpostal,pais,teldomicilio,extension,notas,jefe,sueldoBasico)

values(8,'Callahan','Laura','Coordinador ventas interno','Srta.','1958-01-09 00:00:00','1994-03-05 00:00:00','4726 - 11th Ave. N.E.','Seattle','WA','98105','EE.UU.','(206) 555-1189','2344','Laura se graduó en Psicología por la Universidad de Washington. También completó un curso de francés de negocios. Lee y escribe en francés.','2,1000.00);

INSERT INTO

empleados(IdEmpleado,apellidos,nombre,cargo,tratamiento,fechaNacimiento,fechaContratacion,direccion,ciudad,region,codpostal,pais,teldomicilio,extension,notas,jefe,sueldoBasico)

values(9,'Dodsworth','Anne','Representante de ventas','Srta.','1966-01-27 00:00:00','1994-11-15 00:00:00','7 Houndstooth Rd.','Londres',NULL,'WG2 7LT','Reino Unido','(71) 555-4444','452','Anne tiene

una licenciatura en inglés por el St. Lawrence College. Habla con fluidez el francés y el alemán.',5
,NULL);
/*

INSERCIONES DE CLIENTES

*/

INSERT INTO

clientes(idCliente,NombreCompañia,NombreContacto,CargoContacto,Direccion,Ciudad,region,codpostal,pais,telefono,fax) values('ALFKI','Alfreds Futterkiste','Maria Anders','Representante de ventas','Obere Str. 57','Berlín',NULL,'12209','Alemania','030-0074321','030-0076545');

INSERT INTO

clientes(idCliente,NombreCompañia,NombreContacto,CargoContacto,Direccion,Ciudad,region,codpostal,pais,telefono,fax) values('ANATR','Ana Trujillo Emparedados y helados','Ana Trujillo','Propietario','Avda. de la Constitución 2222','México D.F.',NULL,'05021','México','(5) 555-4729','(5) 555-3745');

INSERT INTO

clientes(idCliente,NombreCompañia,NombreContacto,CargoContacto,Direccion,Ciudad,region,codpostal,pais,telefono,fax) values('ANTON','Antonio Moreno Taquería','Antonio Moreno','Propietario','Mataderos 2312','México D.F.',NULL,'05023','México','(5) 555-3932',NULL);

INSERT INTO

clientes(idCliente,NombreCompañia,NombreContacto,CargoContacto,Direccion,Ciudad,region,codpostal,pais,telefono,fax) values('AROUT','Around the Horn','Thomas Hardy','Representante de ventas','120 Hanover Sq.','Londres',NULL,'WA1 1DP','Reino Unido','(71) 555-7788','(71) 555-6750');

INSERT INTO

clientes(idCliente,NombreCompañia,NombreContacto,CargoContacto,Direccion,Ciudad,region,codpostal,pais,telefono,fax) values('BERGS','Berglunds snabbköp','Christina Berglund','Administrador de pedidos','Berguvsvägen 8','Luleå',NULL,'S-958 22','Suecia','0921-12 34 65','0921-12 34 67');

INSERT INTO

clientes(idCliente,NombreCompañia,NombreContacto,CargoContacto,Direccion,Ciudad,region,codpostal,pais,telefono,fax) values('BLAUS','Blauer See Delikatessen','Hanna Moos','Representante de ventas','Forsterstr. 57','Mannheim',NULL,'68306','Alemania','0621-08460','0621-08924');

INSERT INTO

clientes(idCliente,NombreCompañia,NombreContacto,CargoContacto,Direccion,Ciudad,region,codpostal,pais,telefono,fax) values('BLONP','Blondel père et fils','Frédérique Citeaux','Gerente de marketing','24, place Kléber','Estrasburgo',NULL,'67000','Francia','88.60.15.31','88.60.15.32');

INSERT INTO

clientes(idCliente,NombreCompañia,NombreContacto,CargoContacto,Direccion,Ciudad,region,codpostal,pais,telefono,fax) values('BOLID','Bólido Comidas preparadas','Martín Sommer','Propietario','C/ Araquil, 67','Madrid',NULL,'28023','España','(91) 555 22 82','(91) 555 91 99');

INSERT INTO

clientes(idCliente,NombreCompañia,NombreContacto,CargoContacto,Direccion,Ciudad,region,codpostal,pais,telefono,fax) values('BONAP','Bon app''','Laurence Lebihan','Propietario','12, rue des Bouchers','Marsella',NULL,'13008','Francia','91.24.45.40','91.24.45.41');

INSERT INTO

```
clientes(idCliente,NombreCompañia,NombreContacto,CargoContacto,Direccion,Ciudad,region,codpostal,pais,telefono,fax) values('BOTTM','Bottom-Dollar Markets','Elizabeth Lincoln','Gerente de contabilidad','23 Tsawassen Blvd.','Tsawassen','BC','T2F 8M4','Canadá','(604) 555-4729','(604) 555-3745');
```

```
INSERT INTO
```

```
clientes(idCliente,NombreCompañia,NombreContacto,CargoContacto,Direccion,Ciudad,region,codpostal,pais,telefono,fax) values('BSBEV','B's Beverages','Victoria Ashworth','Representante de ventas','Fauntleroy Circus','Londres',NULL,'EC2 5NT','Reino Unido','(71) 555-1212',NULL);
```

```
INSERT INTO
```

```
clientes(idCliente,NombreCompañia,NombreContacto,CargoContacto,Direccion,Ciudad,region,codpostal,pais,telefono,fax) values('CACTU','Cactus Comidas para llevar','Patricio Simpson','Agente de ventas','Cerrito 333','Buenos Aires',NULL,'1010','Argentina','(1) 135-5555','(1) 135-4892');
```

```
INSERT INTO
```

```
clientes(idCliente,NombreCompañia,NombreContacto,CargoContacto,Direccion,Ciudad,region,codpostal,pais,telefono,fax) values('CENTC','Centro comercial Moctezuma','Francisco Chang','Gerente de marketing','Sierras de Granada 9993','México D.F.',NULL,'05022','México','(5) 555-3392','(5) 555-7293');
```

```
INSERT INTO
```

```
clientes(idCliente,NombreCompañia,NombreContacto,CargoContacto,Direccion,Ciudad,region,codpostal,pais,telefono,fax) values('CHOPS','Chop-suey Chinese','Yang Wang','Propietario','Hauptstr. 29','Berna',NULL,'3012','Suiza','0452-076545',NULL);
```

```
INSERT INTO
```

```
clientes(idCliente,NombreCompañia,NombreContacto,CargoContacto,Direccion,Ciudad,region,codpostal,pais,telefono,fax) values('COMMI','Comércio Mineiro','Pedro Afonso','Asistente de ventas','Av. dos Lusíadas, 23','São Paulo','SP','05432-043','Brasil','(11) 555-7647',NULL);
```

```
INSERT INTO
```

```
clientes(idCliente,NombreCompañia,NombreContacto,CargoContacto,Direccion,Ciudad,region,codpostal,pais,telefono,fax) values('CONSH','Consolidated Holdings','Elizabeth Brown','Representante de ventas','Berkeley Gardens12 Brewery','Londres',NULL,'WX1 6LT','Reino Unido','(71) 555-2282','(71) 555-9199');
```

```
INSERT INTO
```

```
clientes(idCliente,NombreCompañia,NombreContacto,CargoContacto,Direccion,Ciudad,region,codpostal,pais,telefono,fax) values('DRACD','Drachenblut Delikatessen','Sven Ottlieb','Administrador de pedidos','Walserweg 21','Aachen',NULL,'52066','Alemania','0241-039123','0241-059428');
```

```
INSERT INTO
```

```
clientes(idCliente,NombreCompañia,NombreContacto,CargoContacto,Direccion,Ciudad,region,codpostal,pais,telefono,fax) values('DUMON','Du monde entier','Janine Labrune','Propietario','67, rue des Cinquante Otages','Nantes',NULL,'44000','Francia','40.67.88.88','40.67.89.89');
```

```
INSERT INTO
```

```
clientes(idCliente,NombreCompañia,NombreContacto,CargoContacto,Direccion,Ciudad,region,codpostal,pais,telefono,fax) values('EASTC','Eastern Connection','Ann Devon','Agente de ventas','35 King George','Londres',NULL,'WX3 6FW','Reino Unido','(71) 555-0297','(71) 555-3373');
```

```
INSERT INTO
```

```
clientes(idCliente,NombreCompañia,NombreContacto,CargoContacto,Direccion,Ciudad,region,codpostal
```

```

l,pais,telefono,fax) values('ERNSH','Ernst Handel','Roland Mendel','Gerente de ventas','Kirchgasse
6','Graz',NULL,'8010','Austria','7675-3425','7675-3426');
INSERT INTO
clientes(idCliente,NombreCompañia,NombreContacto,CargoContacto,Direccion,Ciudad,region,codposta
l,pais,telefono,fax) values('FAMIA','Familia Arquibaldo','Aria Cruz','Asistente de marketing','Rua Orós,
92','Sao Paulo','SP','05442-030','Brasil','(11) 555-9857',NULL);
/*
    INSERCIONES COMPAÑIAS DE ENVIO
*/
INSERT INTO companiasdeenvios(idCompañiaEnvios,nombreCompañia,telefono) values(1,'Speedy
Express','(503) 555-9831');
INSERT INTO companiasdeenvios(idCompañiaEnvios,nombreCompañia,telefono) values(2,'United
Package','(503) 555-3199');
INSERT INTO companiasdeenvios(idCompañiaEnvios,nombreCompañia,telefono) values(3,'Federal
Shipping','(503) 555-9931');
/*
    INSERCIONES CATEGORIAS
*/
INSERT INTO categorias(idcategoria,nombrecategoria,descripcion) values(1,'Bebidas','Gaseosas, café, té,
cervezas y maltas');
INSERT INTO categorias(idcategoria,nombrecategoria,descripcion) values(2,'Condimentos','Salsas dulces
y picantes, delicias, comida para untar y aderezos');
INSERT INTO categorias(idcategoria,nombrecategoria,descripcion) values(3,'Repostería','Postres, dulces
y pan dulce');
INSERT INTO categorias(idcategoria,nombrecategoria,descripcion) values(4,'Lácteos','Quesos');
INSERT INTO categorias(idcategoria,nombrecategoria,descripcion) values(5,'Granos/Cereales','Pan,
galletas, pasta y cereales');
INSERT INTO categorias(idcategoria,nombrecategoria,descripcion) values(6,'Carnes','Carnes
preparadas');
INSERT INTO categorias(idcategoria,nombrecategoria,descripcion) values(7,'Frutas/Verduras','Frutas
secas y queso de soja');
INSERT INTO categorias(idcategoria,nombrecategoria,descripcion)
values(8,'Pescado/Marisco','Pescados, mariscos y algas');
/*
    INSERCIONES PROVEEDORES
*/
INSERT INTO
proveedores(idProveedor,nombreCompañia,nombrecontacto,cargocontacto,direccion,ciudad,region,co
dpostal,pais,telefono,fax,paginaprincipal) values(1,'Exotic Liquids','Charlotte Cooper','Gerente de
compras','49 Gilbert St.','Londres',NULL,'EC1 4SD','Reino Unido','(171) 555-2222',NULL,NULL);
INSERT INTO
proveedores(idProveedor,nombreCompañia,nombrecontacto,cargocontacto,direccion,ciudad,region,co
dpostal,pais,telefono,fax,paginaprincipal) values(2,'New Orleans Cajun Delights','Shelley

```

Burke','Administrador de pedidos','P.O. Box 78934','New Orleans','LA','70117','Estados Unidos','(100) 555-4822',NULL,'CAJUN.HTM#CAJUN.HTM#');

INSERT INTO

proveedores(idProveedor,nombreCompañia,nombrecontacto,cargocontacto,direccion,ciudad,region,co
dpostal,pais,telefono,fax,paginaprincipal) values(3,'Grandma Kelly's Homestead','Regina
Murphy','Representante de ventas','707 Oxford Rd.','Ann Arbor','MI','48104','Estados Unidos','(313)
555-5735','(313) 555-3349',NULL);

INSERT INTO

proveedores(idProveedor,nombreCompañia,nombrecontacto,cargocontacto,direccion,ciudad,region,co
dpostal,pais,telefono,fax,paginaprincipal) values(4,'Tokyo Traders','Yoshi Nagase','Gerente de
marketing','9-8 SekimaiMusashino-shi','Tokyo',NULL,'100','Japón','(03) 3555-5011',NULL,NULL);

INSERT INTO

proveedores(idProveedor,nombreCompañia,nombrecontacto,cargocontacto,direccion,ciudad,region,co
dpostal,pais,telefono,fax,paginaprincipal) values(5,'Cooperativa de Quesos "Las Cabras"',Antonio del
Valle Saavedra ','Administrador de exportaciones','Calle del Rosal
4','Oviedo','Asturias','33007','España','(98) 598 76 54',NULL,NULL);

INSERT INTO

proveedores(idProveedor,nombreCompañia,nombrecontacto,cargocontacto,direccion,ciudad,region,co
dpostal,pais,telefono,fax,paginaprincipal) values(6,'Mayumi's','Mayumi Ohno','Representante de
marketing','92 SetsukoChuo-ku','Osaka',NULL,'545','Japón','(06) 431-7877',NULL,'Mayumi's (en el World
Wide Web)#http://www.microsoft.com/accessdev/sampleapps/mayumi.htm#');

INSERT INTO

proveedores(idProveedor,nombreCompañia,nombrecontacto,cargocontacto,direccion,ciudad,region,co
dpostal,pais,telefono,fax,paginaprincipal) values(7,'Pavlova, Ltd.','Ian Devling','Gerente de
marketing','74 Rose St.Moonie Ponds','Melbourne','Victoria','3058','Australia','(03) 444-2343','(03) 444-
6588',NULL);

INSERT INTO

proveedores(idProveedor,nombreCompañia,nombrecontacto,cargocontacto,direccion,ciudad,region,co
dpostal,pais,telefono,fax,paginaprincipal) values(8,'Specialty Biscuits, Ltd.','Peter
Wilson','Representante de ventas','29 King's Way','Manchester',NULL,'M14 GSD','Reino Unido','(161)
555-4448',NULL,NULL);

INSERT INTO

proveedores(idProveedor,nombreCompañia,nombrecontacto,cargocontacto,direccion,ciudad,region,co
dpostal,pais,telefono,fax,paginaprincipal) values(9,'PB Knäckebröd AB','Lars Peterson','Agente de
ventas','Kaloadagatan 13','Göteborg',NULL,'S-345 67','Suecia','031-987 65 43','031-987 65 91',NULL);

INSERT INTO

proveedores(idProveedor,nombreCompañia,nombrecontacto,cargocontacto,direccion,ciudad,region,co
dpostal,pais,telefono,fax,paginaprincipal) values(10,'Refrescos Americanas LTDA','Carlos Diaz','Gerente
de marketing','Av. das Americanas 12.890','São Paulo',NULL,'5442','Brasil','(11) 555 4640',NULL,NULL);

INSERT INTO

proveedores(idProveedor,nombreCompañia,nombrecontacto,cargocontacto,direccion,ciudad,region,co
dpostal,pais,telefono,fax,paginaprincipal) values(11,'Heli Süßwaren GmbH & Co. KG','Petra
Winkler','Gerente de ventas','Tiergartenstraße 5','Berlín',NULL,'10785','Alemania','(010)

```

9984510',NULL,NULL);
INSERT INTO
proveedores(idProveedor,nombreCompañia,nombrecontacto,cargocontacto,direccion,ciudad,region,co
dpostal,pais,telefono,fax,paginaprincipal) values(12,'Plutzer Lebensmittelgroßmärkte AG','Martin
Bein','Ger. marketing internacional','Bogenallee 51','Frankfurt',NULL,'60439','Alemania','(069)
992755',NULL,'Plutzer (en el World Wide
Web)#http://www.microsoft.com/accessdev/sampleapps/plutzer.htm#');
INSERT INTO
proveedores(idProveedor,nombreCompañia,nombrecontacto,cargocontacto,direccion,ciudad,region,co
dpostal,pais,telefono,fax,paginaprincipal) values(13,'Nord-Ost-Fisch Handelsgesellschaft mbH','Sven
Petersen','Coordinador de mercados','Frahmredder 112a','Cuxhaven',NULL,'27478','Alemania','(04721)
8713','(04721) 8714',NULL);
INSERT INTO
proveedores(idProveedor,nombreCompañia,nombrecontacto,cargocontacto,direccion,ciudad,region,co
dpostal,pais,telefono,fax,paginaprincipal) values(14,'Formaggi Fortini s.r.l.','Elio Rossi','Representante
de ventas','Viale Dante, 75','Ravenna',NULL,'48100','Italia','(0544) 60323','(0544)
60603','FORMAGGI.HTM#FORMAGGI.HTM#');
INSERT INTO
proveedores(idProveedor,nombreCompañia,nombrecontacto,cargocontacto,direccion,ciudad,region,co
dpostal,pais,telefono,fax,paginaprincipal) values(15,'Norske Meierier','Beate Vileid','Gerente de
marketing','Hatlevegen 5','Sandvika',NULL,'1320','Noruega','(0)2-953010',NULL,NULL);
INSERT INTO
proveedores(idProveedor,nombreCompañia,nombrecontacto,cargocontacto,direccion,ciudad,region,co
dpostal,pais,telefono,fax,paginaprincipal) values(16,'Bigfoot Breweries','Cheryl Saylor','Repr. de cuentas
regional','3400 - 8th AvenueSuite 210','Bend','OR','97101','Estados Unidos','(503) 555-
9931',NULL,NULL);
INSERT INTO
proveedores(idProveedor,nombreCompañia,nombrecontacto,cargocontacto,direccion,ciudad,region,co
dpostal,pais,telefono,fax,paginaprincipal) values(17,'Svensk Sjöföda AB','Michael Björn','Representante
de ventas','Brovallavägen 231','Stockholm',NULL,'S-123 45','Suecia','08-123 45 67',NULL,NULL);
/*
    INSERCIONES PRODUCTOS
*/
INSERT INTO
productos(idproducto,nombreProducto,idProveedor,idCategoria,cantidadPorUnidad,precioUnidad,unidadesEnExistencia,unidadesEnPedido,nivelNuevoPedido,suspendido,categoriaProducto) values(1,'Té
Dharamsala',1,1,'10 cajas x 20 bolsas',18.00,39,0,10,0,'Categoria D');
INSERT INTO
productos(idproducto,nombreProducto,idProveedor,idCategoria,cantidadPorUnidad,precioUnidad,unidadesEnExistencia,unidadesEnPedido,nivelNuevoPedido,suspendido,categoriaProducto)
values(2,'Cerveza tibetana Barley',1,1,'24 - bot. 12 l',19.00,17,40,25,0,'Categoria D');
INSERT INTO
productos(idproducto,nombreProducto,idProveedor,idCategoria,cantidadPorUnidad,precioUnidad,unidadesEnExistencia,unidadesEnPedido,nivelNuevoPedido,suspendido,categoriaProducto)

```

```

adesEnExistencia,unidadesEnPedido,nivelNuevoPedido,suspendido,categoriaProducto) values(3,'Sirope
de regaliz',1,2,'12 - bot. 550 ml',10.00,13,70,25,0,'Categoria D');
INSERT INTO
productos(idproducto,nombreProducto,idProveedor,idCategoria,cantidadPorUnidad,precioUnidad,unid
adesEnExistencia,unidadesEnPedido,nivelNuevoPedido,suspendido,categoriaProducto)
values(4,'Especias Cajun del chef Anton',2,2,'48 - frascos 6 l',22.00,53,0,0,0,'Categoria D');
INSERT INTO
productos(idproducto,nombreProducto,idProveedor,idCategoria,cantidadPorUnidad,precioUnidad,unid
adesEnExistencia,unidadesEnPedido,nivelNuevoPedido,suspendido,categoriaProducto) values(5,'Mezcla
Gumbo del chef Anton',2,2,'36 cajas',21.35,0,0,0,1,'Categoria D');
INSERT INTO
productos(idproducto,nombreProducto,idProveedor,idCategoria,cantidadPorUnidad,precioUnidad,unid
adesEnExistencia,unidadesEnPedido,nivelNuevoPedido,suspendido,categoriaProducto)
values(6,'Mermelada de grosellas de la abuela',3,2,'12 - frascos 8 l',25.00,120,0,25,0,'Categoria D');
INSERT INTO
productos(idproducto,nombreProducto,idProveedor,idCategoria,cantidadPorUnidad,precioUnidad,unid
adesEnExistencia,unidadesEnPedido,nivelNuevoPedido,suspendido,categoriaProducto) values(7,'Peras
secas orgánicas del tío Bob',3,7,'12 - paq. 1 kg',30.00,15,0,10,0,'Categoria D');
INSERT INTO
productos(idproducto,nombreProducto,idProveedor,idCategoria,cantidadPorUnidad,precioUnidad,unid
adesEnExistencia,unidadesEnPedido,nivelNuevoPedido,suspendido,categoriaProducto) values(8,'Salsa
de arándanos Northwoods',3,2,'12 - frascos 12 l',40.00,6,0,0,0,'Categoria D');
INSERT INTO
productos(idproducto,nombreProducto,idProveedor,idCategoria,cantidadPorUnidad,precioUnidad,unid
adesEnExistencia,unidadesEnPedido,nivelNuevoPedido,suspendido,categoriaProducto) values(9,'Buey
Mishi Kobe',4,6,'18 - paq. 500 g',97.00,29,0,0,1,'Categoria D');
INSERT INTO
productos(idproducto,nombreProducto,idProveedor,idCategoria,cantidadPorUnidad,precioUnidad,unid
adesEnExistencia,unidadesEnPedido,nivelNuevoPedido,suspendido,categoriaProducto) values(10,'Pez
espada',4,8,'12 - frascos 200 ml',31.00,31,0,0,0,'Categoria D');
INSERT INTO
productos(idproducto,nombreProducto,idProveedor,idCategoria,cantidadPorUnidad,precioUnidad,unid
adesEnExistencia,unidadesEnPedido,nivelNuevoPedido,suspendido,categoriaProducto)
values(11,'Queso Cabrales',5,4,'paq. 1 kg',21.00,22,30,30,0,'Categoria D');
INSERT INTO
productos(idproducto,nombreProducto,idProveedor,idCategoria,cantidadPorUnidad,precioUnidad,unid
adesEnExistencia,unidadesEnPedido,nivelNuevoPedido,suspendido,categoriaProducto)
values(12,'Queso Manchego La Pastora',5,4,'10 - paq. 500 g',38.00,86,0,0,0,'Categoria D');
INSERT INTO
productos(idproducto,nombreProducto,idProveedor,idCategoria,cantidadPorUnidad,precioUnidad,unid
adesEnExistencia,unidadesEnPedido,nivelNuevoPedido,suspendido,categoriaProducto) values(13,'Algas
Konbu',6,8,'caja 2 kg',6.00,24,0,5,0,'Categoria D');
INSERT INTO

```

```

productos(idproducto,nombreProducto,idProveedor,idCategoria,cantidadPorUnidad,precioUnidad,unidadesEnExistencia,unidadesEnPedido,nivelNuevoPedido,suspendido,categoriaProducto)
values(14,'Cuajada de judías',6,7,'40 - paq. 100 g',23.25,35,0,0,0,'Categoria D');
INSERT INTO
productos(idproducto,nombreProducto,idProveedor,idCategoria,cantidadPorUnidad,precioUnidad,unidadesEnExistencia,unidadesEnPedido,nivelNuevoPedido,suspendido,categoriaProducto) values(15,'Salsa
de soja baja en sodio',6,2,'24 - bot. 250 ml',15.50,39,0,5,0,NULL);
INSERT INTO
productos(idproducto,nombreProducto,idProveedor,idCategoria,cantidadPorUnidad,precioUnidad,unidadesEnExistencia,unidadesEnPedido,nivelNuevoPedido,suspendido,categoriaProducto)
values(16,'Postre de merengue Pavlova',7,3,'32 - cajas 500 g',17.45,29,0,10,0,'Categoria D');
INSERT INTO
productos(idproducto,nombreProducto,idProveedor,idCategoria,cantidadPorUnidad,precioUnidad,unidadesEnExistencia,unidadesEnPedido,nivelNuevoPedido,suspendido,categoriaProducto)
values(17,'Cordero Alice Springs',7,6,'20 - latas 1 kg',39.00,0,0,0,1,'Categoria D');
INSERT INTO
productos(idproducto,nombreProducto,idProveedor,idCategoria,cantidadPorUnidad,precioUnidad,unidadesEnExistencia,unidadesEnPedido,nivelNuevoPedido,suspendido,categoriaProducto)
values(18,'Langostinos tigre Carnarvon',7,8,'paq. 16 kg',62.50,42,0,0,0,'Categoria D');
INSERT INTO
productos(idproducto,nombreProducto,idProveedor,idCategoria,cantidadPorUnidad,precioUnidad,unidadesEnExistencia,unidadesEnPedido,nivelNuevoPedido,suspendido,categoriaProducto)
values(19,'Pastas de té de chocolate',8,3,'10 cajas x 12 piezas',9.20,25,0,5,0,'Categoria D');
INSERT INTO
productos(idproducto,nombreProducto,idProveedor,idCategoria,cantidadPorUnidad,precioUnidad,unidadesEnExistencia,unidadesEnPedido,nivelNuevoPedido,suspendido,categoriaProducto)
values(20,'Mermelada de Sir Rodney's',8,3,'30 cajas regalo',81.00,40,0,0,0,'Categoria D');
INSERT INTO
productos(idproducto,nombreProducto,idProveedor,idCategoria,cantidadPorUnidad,precioUnidad,unidadesEnExistencia,unidadesEnPedido,nivelNuevoPedido,suspendido,categoriaProducto) values(21,'Bolos
de Sir Rodney's',8,3,'24 paq. x 4 piezas',10.00,3,40,5,0,'Categoria D');
INSERT INTO
productos(idproducto,nombreProducto,idProveedor,idCategoria,cantidadPorUnidad,precioUnidad,unidadesEnExistencia,unidadesEnPedido,nivelNuevoPedido,suspendido,categoriaProducto) values(22,'Pan
de centeno crujiente estilo Gustaf's',9,5,'24 - paq. 500 g',21.00,104,0,25,0,'Categoria D');
INSERT INTO
productos(idproducto,nombreProducto,idProveedor,idCategoria,cantidadPorUnidad,precioUnidad,unidadesEnExistencia,unidadesEnPedido,nivelNuevoPedido,suspendido,categoriaProducto) values(23,'Pan
fino',9,5,'12 - paq. 250 g',9.00,61,0,25,0,'Categoria D');
INSERT INTO
productos(idproducto,nombreProducto,idProveedor,idCategoria,cantidadPorUnidad,precioUnidad,unidadesEnExistencia,unidadesEnPedido,nivelNuevoPedido,suspendido,categoriaProducto)
values(24,'Refresco Guaraná Fantástica',10,1,'12 - latas 355 ml',4.50,20,0,0,1,'Categoria D');

```

/*

INSERCIONES PEDIDOS

*/

INSERT INTO

pedidos(IdPedido,IdCliente,IdEmpleado,Fechapedido,fechaentrega,fechaenvio,FormaEnvio,cargo,destinatario,DireccionDestinatario,CiudadDestinatario,RegionDestinatario,CodPostalDestinatario,PaisDestinatario) values(10248,'WILMK',5,'1994-08-04 00:00:00','1994-09-01 00:00:00','1994-08-16 00:00:00',3,32.38,'Wilman Kala','Keskuskatu 45','Helsinki',NULL,'21240','Finlandia');

INSERT INTO

pedidos(IdPedido,IdCliente,IdEmpleado,Fechapedido,fechaentrega,fechaenvio,FormaEnvio,cargo,destinatario,DireccionDestinatario,CiudadDestinatario,RegionDestinatario,CodPostalDestinatario,PaisDestinatario) values(10249,'TOMSP',6,'1994-08-05 00:00:00','1994-09-16 00:00:00','1994-08-10 00:00:00',1,11.61,'Toms Spezialitäten','Luisenstr. 48','Münster',NULL,'44087','Alemania');

INSERT INTO

pedidos(IdPedido,IdCliente,IdEmpleado,Fechapedido,fechaentrega,fechaenvio,FormaEnvio,cargo,destinatario,DireccionDestinatario,CiudadDestinatario,RegionDestinatario,CodPostalDestinatario,PaisDestinatario) values(10250,'HANAR',4,'1994-08-08 00:00:00','1994-09-05 00:00:00','1994-08-12 00:00:00',2,65.83,'Hanari Carnes','Rua do Paço. 67','Rio de Janeiro','RJ','05454-876','Brasil');

INSERT INTO

pedidos(IdPedido,IdCliente,IdEmpleado,Fechapedido,fechaentrega,fechaenvio,FormaEnvio,cargo,destinatario,DireccionDestinatario,CiudadDestinatario,RegionDestinatario,CodPostalDestinatario,PaisDestinatario) values(10251,'VICTE',3,'1994-08-08 00:00:00','1994-09-05 00:00:00','1994-08-15 00:00:00',1,41.34,'Victuailles en stock','2. rue du Commerce','Lyon',NULL,'69004','Francia');

INSERT INTO

pedidos(IdPedido,IdCliente,IdEmpleado,Fechapedido,fechaentrega,fechaenvio,FormaEnvio,cargo,destinatario,DireccionDestinatario,CiudadDestinatario,RegionDestinatario,CodPostalDestinatario,PaisDestinatario) values(10252,'SUPRD',4,'1994-08-09 00:00:00','1994-09-06 00:00:00','1994-08-11 00:00:00',2,51.30,'Suprêmes délices','Boulevard Tirou. 255','Charleroi',NULL,'B-6000','Bélgica');

INSERT INTO

pedidos(IdPedido,IdCliente,IdEmpleado,Fechapedido,fechaentrega,fechaenvio,FormaEnvio,cargo,destinatario,DireccionDestinatario,CiudadDestinatario,RegionDestinatario,CodPostalDestinatario,PaisDestinatario) values(10253,'HANAR',3,'1994-08-10 00:00:00','1994-08-24 00:00:00','1994-08-16 00:00:00',2,58.17,'Hanari Carnes','Rua do Paço. 67','Rio de Janeiro','RJ','05454-876','Brasil');

INSERT INTO

pedidos(IdPedido,IdCliente,IdEmpleado,Fechapedido,fechaentrega,fechaenvio,FormaEnvio,cargo,destinatario,DireccionDestinatario,CiudadDestinatario,RegionDestinatario,CodPostalDestinatario,PaisDestinatario) values(10254,'CHOPS',5,'1994-08-11 00:00:00','1994-09-08 00:00:00','1994-08-23 00:00:00',2,22.98,'Chop-suey Chinese','Hauptstr. 31','Bern',NULL,'3012','Suiza');

INSERT INTO

pedidos(IdPedido,IdCliente,IdEmpleado,Fechapedido,fechaentrega,fechaenvio,FormaEnvio,cargo,destinatario,DireccionDestinatario,CiudadDestinatario,RegionDestinatario,CodPostalDestinatario,PaisDestinatario) values(10255,'RICSU',9,'1994-08-12 00:00:00','1994-09-09 00:00:00','1994-08-15

```

00:00:00',3,148.33,'Richter Supermarkt','Starenweg 5','Genève',NULL,'1204','Suiza');
INSERT INTO
pedidos(IdPedido,IdCliente,IdEmpleado,Fechapedido,fechaentrega,fechaenvio,FormaEnvio,cargo,destin
atario,DireccionDestinatario,CiudadDestinatario,RegionDestinatario,CodPostalDestinatario,PaisDestinat
ario) values(10256,'WELL',3,'1994-08-15 00:00:00','1994-09-12 00:00:00','1994-08-17
00:00:00',2,13.97,'Wellington Importadora','Rua do Mercado. 12','Resende','SP','08737-363','Brasil');
INSERT INTO
pedidos(IdPedido,IdCliente,IdEmpleado,Fechapedido,fechaentrega,fechaenvio,FormaEnvio,cargo,destin
atario,DireccionDestinatario,CiudadDestinatario,RegionDestinatario,CodPostalDestinatario,PaisDestinat
ario) values(10257,'HILAA',4,'1994-08-16 00:00:00','1994-09-13 00:00:00','1994-08-22
00:00:00',3,81.91,'HILARIÓN-Abastos','Carrera 22 con Ave. Carlos Soublette #8-35','San
Cristóbal','Táchira','5022','Venezuela');
INSERT INTO
pedidos(IdPedido,IdCliente,IdEmpleado,Fechapedido,fechaentrega,fechaenvio,FormaEnvio,cargo,destin
atario,DireccionDestinatario,CiudadDestinatario,RegionDestinatario,CodPostalDestinatario,PaisDestinat
ario) values(10258,'ERNSH',1,'1994-08-17 00:00:00','1994-09-14 00:00:00','1994-08-23
00:00:00',1,140.51,'Ernst Handel','Kirchgasse 6','Graz',NULL,'8010','Austria');
INSERT INTO
pedidos(IdPedido,IdCliente,IdEmpleado,Fechapedido,fechaentrega,fechaenvio,FormaEnvio,cargo,destin
atario,DireccionDestinatario,CiudadDestinatario,RegionDestinatario,CodPostalDestinatario,PaisDestinat
ario) values(10259,'CENTC',4,'1994-08-18 00:00:00','1994-09-15 00:00:00','1994-08-25
00:00:00',3,3.25,'Centro comercial Moctezuma','Sierras de Granada 9993','México
D.F.',NULL,'05022','México');
INSERT INTO
pedidos(IdPedido,IdCliente,IdEmpleado,Fechapedido,fechaentrega,fechaenvio,FormaEnvio,cargo,destin
atario,DireccionDestinatario,CiudadDestinatario,RegionDestinatario,CodPostalDestinatario,PaisDestinat
ario) values(10260,'OTTIK',4,'1994-08-19 00:00:00','1994-09-16 00:00:00','1994-08-29
00:00:00',1,55.09,'Otilies Käseladen','Mehrheimerstr. 369','Köln',NULL,'50739','Alemania');
INSERT INTO
pedidos(IdPedido,IdCliente,IdEmpleado,Fechapedido,fechaentrega,fechaenvio,FormaEnvio,cargo,destin
atario,DireccionDestinatario,CiudadDestinatario,RegionDestinatario,CodPostalDestinatario,PaisDestinat
ario) values(10261,'QUEDE',4,'1994-08-19 00:00:00','1994-09-16 00:00:00','1994-08-30
00:00:00',2,3.05,'Que Delícia','Rua da Panificadora. 12','Río de Janeiro','RJ','02389-673','Brasil');
INSERT INTO
pedidos(IdPedido,IdCliente,IdEmpleado,Fechapedido,fechaentrega,fechaenvio,FormaEnvio,cargo,destin
atario,DireccionDestinatario,CiudadDestinatario,RegionDestinatario,CodPostalDestinatario,PaisDestinat
ario) values(10262,'RATTC',8,'1994-08-22 00:00:00','1994-09-19 00:00:00','1994-08-25
00:00:00',3,48.29,'Rattlesnake Canyon Grocery','2817 Milton Dr.','Albuquerque','NM','87110','Estados
Unidos');
INSERT INTO
pedidos(IdPedido,IdCliente,IdEmpleado,Fechapedido,fechaentrega,fechaenvio,FormaEnvio,cargo,destin
atario,DireccionDestinatario,CiudadDestinatario,RegionDestinatario,CodPostalDestinatario,PaisDestinat
ario) values(10263,'ERNSH',9,'1994-08-23 00:00:00','1994-09-20 00:00:00','1994-08-31

```

00:00:00',3,146.06,'Ernst Handel','Kirchgasse 6','Graz',NULL,'8010','Austria');

INSERT INTO

pedidos(IdPedido,IdCliente,IdEmpleado,Fechapedido,fechaentrega,fechaenvio,FormaEnvio,cargo,destinatario,DireccionDestinatario,CiudadDestinatario,RegionDestinatario,CodPostalDestinatario,PaisDestinatario) values(10264,'FOLKO',6,'1994-08-24 00:00:00','1994-09-21 00:00:00','1994-09-23 00:00:00',3,3.67,'Folk och fä HB','Åkergatan 24','Bräcke',NULL,'S-844 67','Suecia');

INSERT INTO

pedidos(IdPedido,IdCliente,IdEmpleado,Fechapedido,fechaentrega,fechaenvio,FormaEnvio,cargo,destinatario,DireccionDestinatario,CiudadDestinatario,RegionDestinatario,CodPostalDestinatario,PaisDestinatario) values(10265,'BLONP',2,'1994-08-25 00:00:00','1994-09-22 00:00:00','1994-09-12 00:00:00',1,55.28,'Blondel père et fils','24. place Kléber','Strasbourg',NULL,'67000','Francia');

INSERT INTO

pedidos(IdPedido,IdCliente,IdEmpleado,Fechapedido,fechaentrega,fechaenvio,FormaEnvio,cargo,destinatario,DireccionDestinatario,CiudadDestinatario,RegionDestinatario,CodPostalDestinatario,PaisDestinatario) values(10266,'WARTH',3,'1994-08-26 00:00:00','1994-10-07 00:00:00','1994-08-31 00:00:00',3,25.73,'Wartian Herkku','Torikatu 38','Oulu',NULL,'90110','Finlandia');

INSERT INTO

pedidos(IdPedido,IdCliente,IdEmpleado,Fechapedido,fechaentrega,fechaenvio,FormaEnvio,cargo,destinatario,DireccionDestinatario,CiudadDestinatario,RegionDestinatario,CodPostalDestinatario,PaisDestinatario) values(10267,'FRANK',4,'1994-08-29 00:00:00','1994-09-26 00:00:00','1994-09-06 00:00:00',1,208.58,'Frankenversand','Berliner Platz 43','München',NULL,'80805','Alemania');

INSERT INTO

pedidos(IdPedido,IdCliente,IdEmpleado,Fechapedido,fechaentrega,fechaenvio,FormaEnvio,cargo,destinatario,DireccionDestinatario,CiudadDestinatario,RegionDestinatario,CodPostalDestinatario,PaisDestinatario) values(10268,'GROSR',8,'1994-08-30 00:00:00','1994-09-27 00:00:00','1994-09-02 00:00:00',3,66.29,'GROSELLA-Restaurante','5ª Ave. Los Palos Grandes','Caracas','DF','1081','Venezuela');

/*

INSERCIONES DE DETALLES DE PEDIDOS

*/

INSERT INTO detallesdepedidos(idpedido,idproducto,preciounidad,cantidad,descuento) values(10248,11,14.00,12,0);

INSERT INTO detallesdepedidos(idpedido,idproducto,preciounidad,cantidad,descuento) values(10248,42,9.80,10,0);

INSERT INTO detallesdepedidos(idpedido,idproducto,preciounidad,cantidad,descuento) values(10248,72,34.80,5,0);

INSERT INTO detallesdepedidos(idpedido,idproducto,preciounidad,cantidad,descuento) values(10249,14,18.60,9,0);

INSERT INTO detallesdepedidos(idpedido,idproducto,preciounidad,cantidad,descuento) values(10249,51,42.40,40,0);

INSERT INTO detallesdepedidos(idpedido,idproducto,preciounidad,cantidad,descuento) values(10250,41,7.70,10,0);

INSERT INTO detallesdepedidos(idpedido,idproducto,preciounidad,cantidad,descuento)

```
values(10250,51,42.40,35,0.15);
INSERT INTO detallesdepedidos(idpedido,idproducto,preciounidad,cantidad,descuento)
values(10250,65,16.80,15,0.15);
INSERT INTO detallesdepedidos(idpedido,idproducto,preciounidad,cantidad,descuento)
values(10251,22,16.80,6,0.05);
INSERT INTO detallesdepedidos(idpedido,idproducto,preciounidad,cantidad,descuento)
values(10251,57,15.60,15,0.05);
INSERT INTO detallesdepedidos(idpedido,idproducto,preciounidad,cantidad,descuento)
values(10251,65,16.80,20,0);
INSERT INTO detallesdepedidos(idpedido,idproducto,preciounidad,cantidad,descuento)
values(10252,20,64.80,40,0.05);
INSERT INTO detallesdepedidos(idpedido,idproducto,preciounidad,cantidad,descuento)
values(10252,33,2.00,25,0.05);
INSERT INTO detallesdepedidos(idpedido,idproducto,preciounidad,cantidad,descuento)
values(10252,60,27.20,40,0);
INSERT INTO detallesdepedidos(idpedido,idproducto,preciounidad,cantidad,descuento)
values(10253,31,10.00,20,0);
INSERT INTO detallesdepedidos(idpedido,idproducto,preciounidad,cantidad,descuento)
values(10253,39,14.40,42,0);
INSERT INTO detallesdepedidos(idpedido,idproducto,preciounidad,cantidad,descuento)
values(10253,49,16.00,40,0);
INSERT INTO detallesdepedidos(idpedido,idproducto,preciounidad,cantidad,descuento)
values(10254,24,3.60,15,0.15);
```

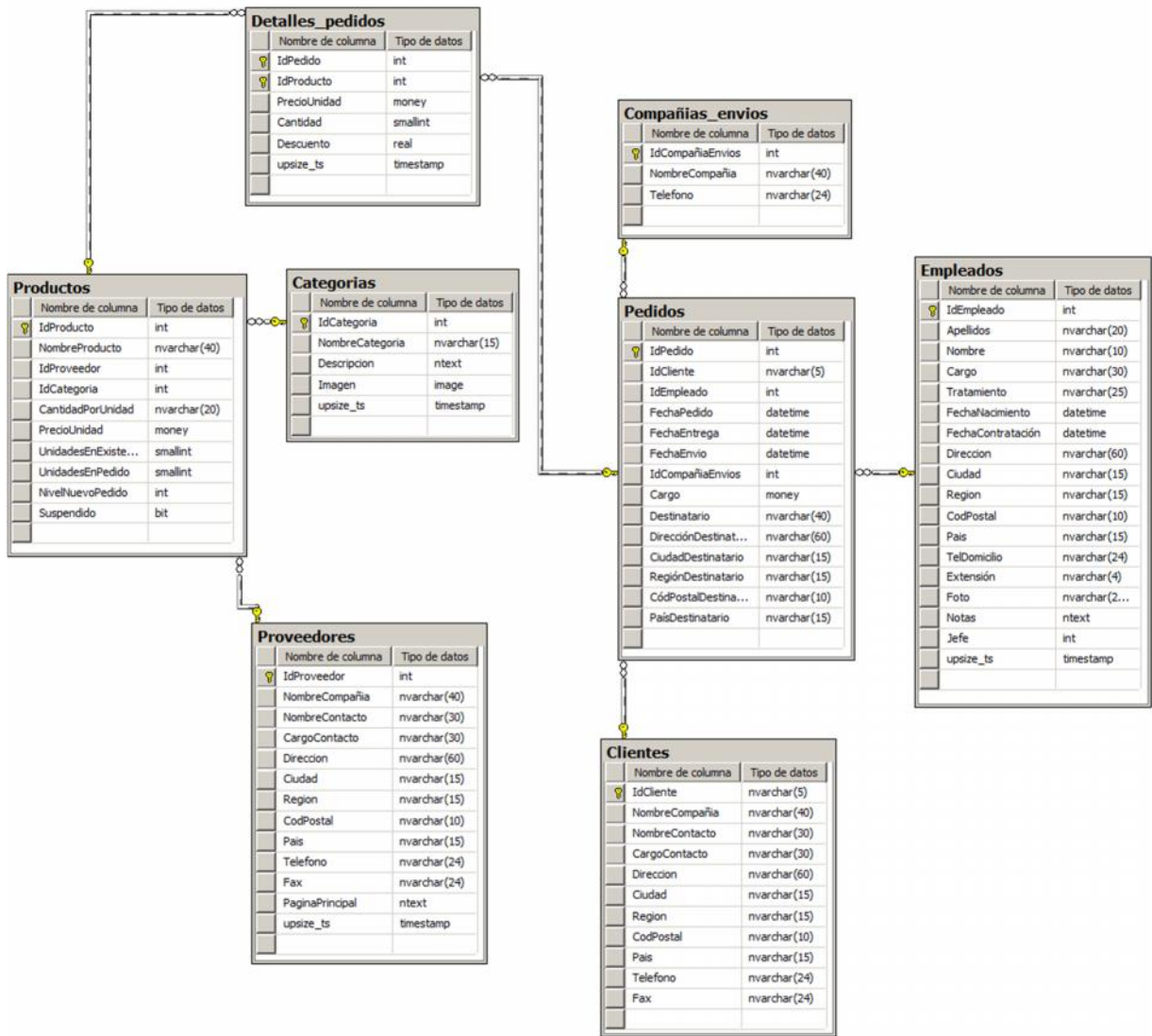


Fig. 3. Diagrama Entidad Relación de la base de datos relacional.

7.2. PROCESO DE MIGRACIÓN DE DATOS.

El proceso para migrar la información de un esquema relacional a BigData, utilizando los gestores de bases de datos, consiste en crear scripts para transformar datos fuentes a una estructura jerárquica en formato JSON que pueda ser importada a MongoDB. Este proceso utiliza subdocumentos y arreglos, que es la forma en como JSON se ajusta a la estructura de objetos a nivel de aplicación.

Esto hace más fácil a los programadores el proceso de mapear los datos usados en la aplicación a sus documentos asociados en la base de datos. En contraste, tratando de mapear la representación de objetos de los datos a la representación tabular de un RDBMS hace más lento el desarrollo. Agregar Mapeadores de Objetos Relacionales (ORMs) permite crear mayor complejidad reduciendo la flexibilidad para evolucionar esquemas y optimizar consultas para cumplir con los requerimientos de las aplicaciones.

El equipo del proyecto debe comenzar el proceso de diseño del esquema considerando los requerimientos de la aplicación. Deben modelar los datos de manera que aprovechen la flexibilidad del modelo de documentos. En migraciones de esquemas, debe ser fácil reflejar el esquema plano de la base de datos relacional al modelo de documentos. Sin embargo, este método relacional niega las ventajas habilitadas por el modelo de documentos enriquecidos, que son las estructuras de datos incrustadas. Por ejemplo, datos que pertenecen a una relación padre-hijo en dos tablas relacionales, comúnmente deberían estar incrustadas dentro de un solo documento en MongoDB.

Pasos para la migración:

1. Agregar una nueva columna dentro del sistema relacional para el UUID (Identificador Único Universal) creándola en la nueva tabla y poblándola usando la función NEWID().

```
SELECT *, UUID = NEWID() INTO NEWTABLE FROM OLDTABLE
```

2. Agregar los índices a la tabla.
3. Renombrar la nueva tabla y eliminar la anterior.
4. En el sistema relacional crear el script con el formato JSON en un procedimiento almacenado conteniendo las consultas necesarias para recuperar todos los datos de la base de datos y agregar la cláusula FOR JSON PATH al final. Ver figura 4.
5. Como en el esquema de BigData no existen relaciones de muchos a muchos, es necesario agregar índices para hacer coincidir la información de una colección a una colección padre. Todos los motores de almacenamiento de MongoDB soportan todos los tipos de índices y los índices pueden ser creados en cualquier parte de documentos JSON, incluyendo subdocumentos internos y elementos de arreglos.
6. En MongoDB ejecutar los procedimientos almacenados usando el comando BCP para generar el archivo JSON.
7. Una vez teniendo generado el archivo JSON, dentro de MongoDB ejecutar el comando **mongoimports**. Se debe usar este comando mientras estamos conectados a una instancia de **mongod**.

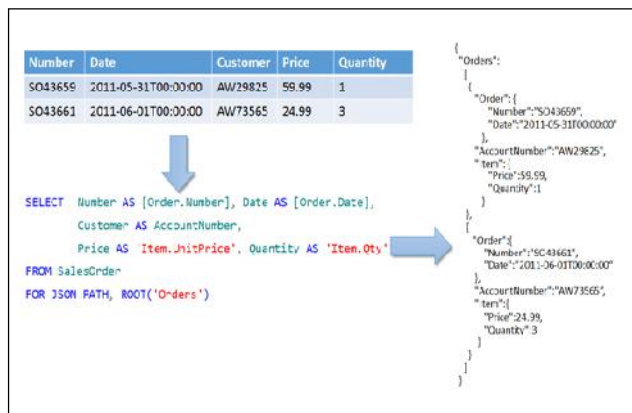


Fig. 4. Transformación de datos de SQL Server a JSON.

En el siguiente ejemplo, el comando `mongoimport` importa los datos del archivo con formato JSON **contacts.json** dentro de la base de datos **users**, en la colección **contacts**, en la instancia de MongoDB corriendo en el localhost en el puerto número 27017.

```
mongoimport --db users --collection contacts --file contacts.json
```

1.1 Configuración del Servidor.

Antes de comenzar con el desarrollo de la herramienta de migración es necesario llevar a cabo la siguiente configuración en el servidor web.

- 1- Instalar en motor de base de datos Xampp para controlar la base de datos de los usuarios de la aplicación, la que está en MySQL.
- 2- Ejecutar Xampp y Levantar los servicios del servidor Apache y del motor de bases de datos MySQL.

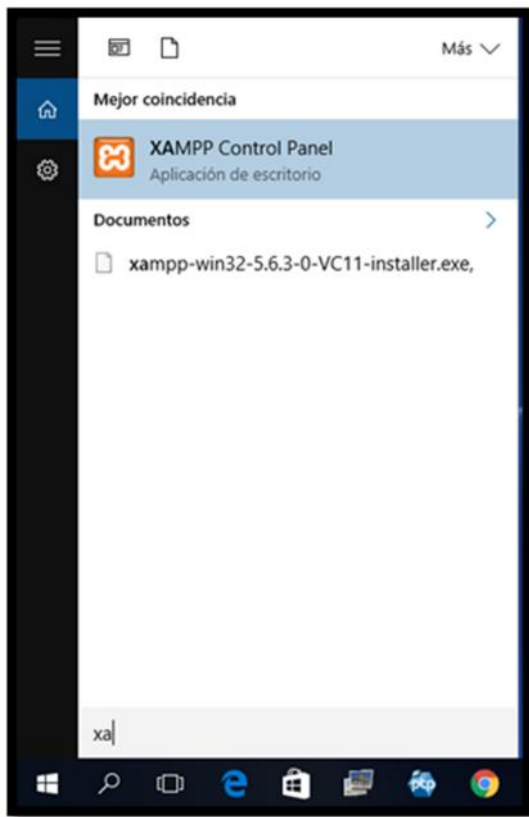


Fig. 5. Ejecución de la herramienta Xampp.

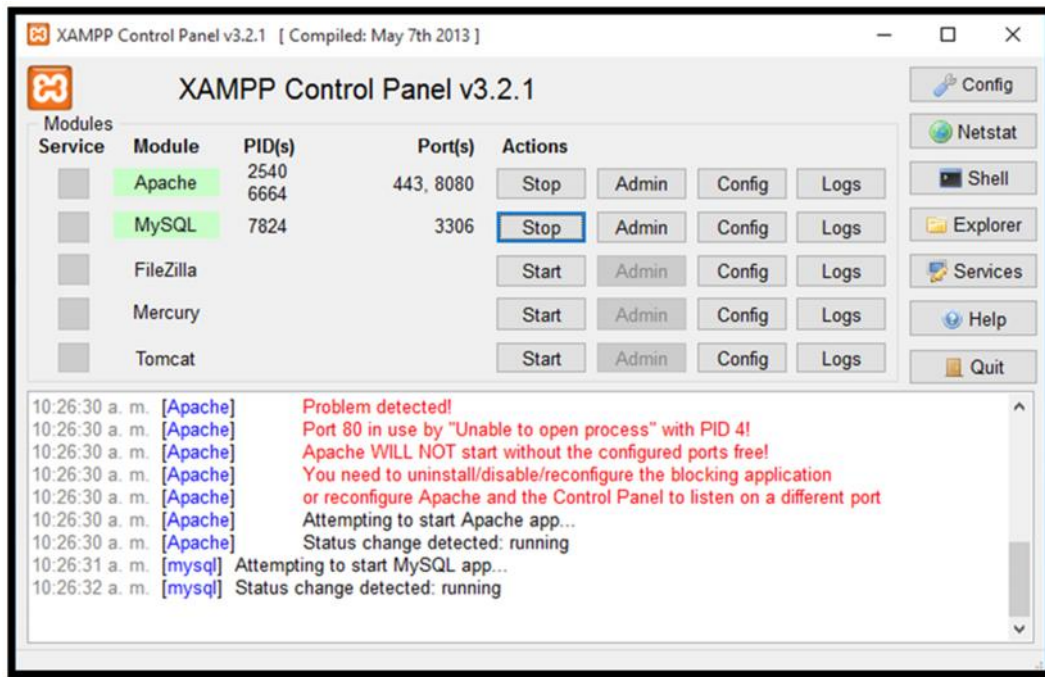


Fig. 6. Panel de control para levantar los servicios de Apache y MySQL.

- 3- Tener instalado el driver ODBC 11 para SQL Server. El instalador va dentro de la carpeta **Recursos /Odbc Driver**.

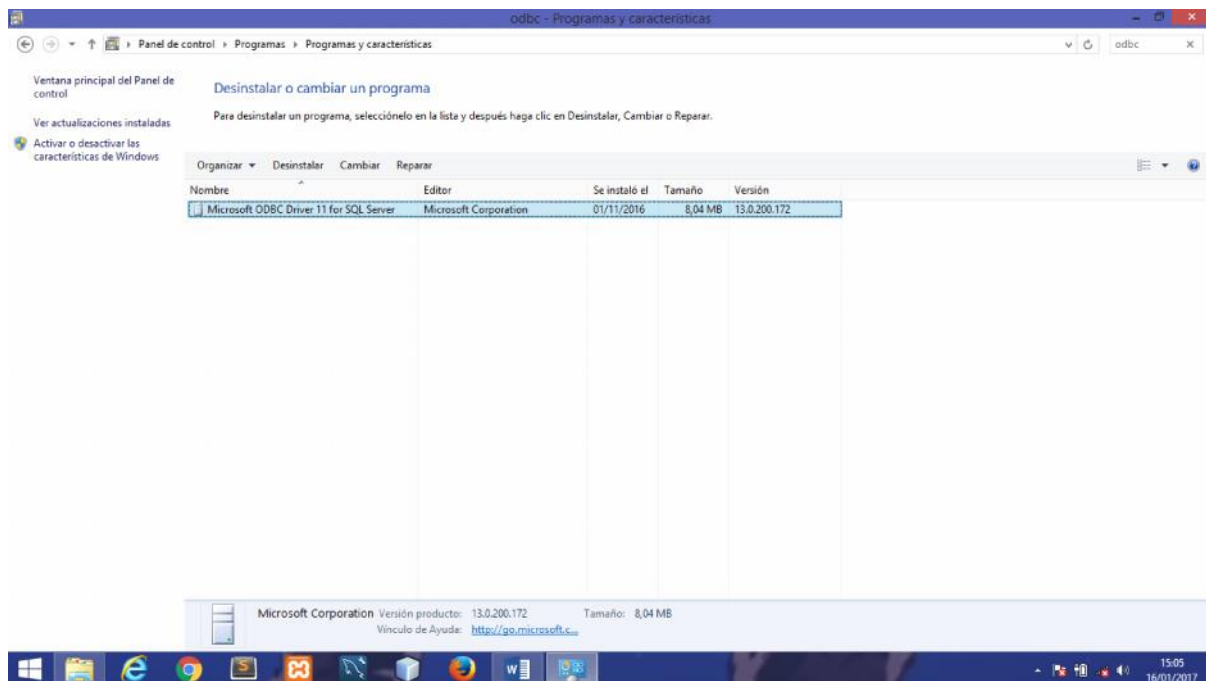


Fig. 7. Microsoft ODBC Driver 11 for SQL Server.

- 4- Luego configurar las librerías dll dentro del servidor apache (escribirlas dentro del archivo php.ini). El archivo php.ini es independiente del servidor que está moviendo PHP, es por ello que la escritura de las .dll del odbc driver siempre se especifican en el php.ini
- 5- Las librerías .dll tienen que estar dentro de la carpeta **ext** de la carpeta **php** del servidor. Las dll utilizadas dentro del proyecto solamente son 2. Están dentro de Recursos/dll/odbc dll:
 1. php_pdo_sqlsrv_56_nts.dll
 2. php_sqlsrv_56_ts.dll

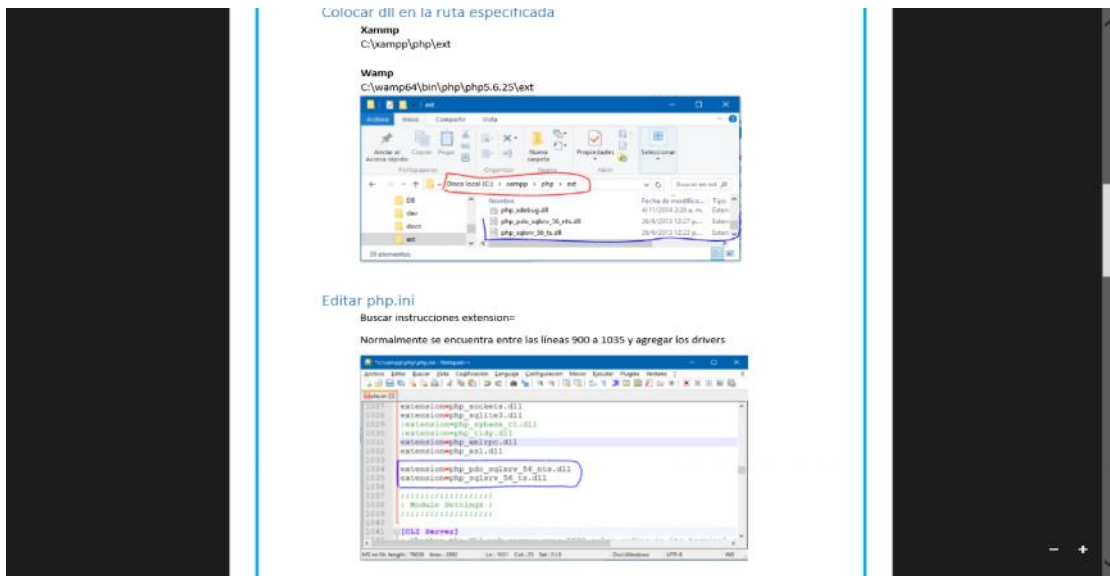


Fig. 8. Librerías de PHP para SQL Server.

- 6- A nivel de depuración también se ha utilizado el xDebug y el XCache, que son debugueadores propios de PHP, pero para poder utilizarlos en un servidor con Windows, tienen que anexarse las siguientes librerías dll dentro de la carpeta php del servidor. También tienen que reescribirse en el archivo php.ini así como se hizo con las dll del ODBC driver.
- 7- Las dll de los debugueadores están dentro de la carpeta: **Recursos/dll/xDebug-xCache**.

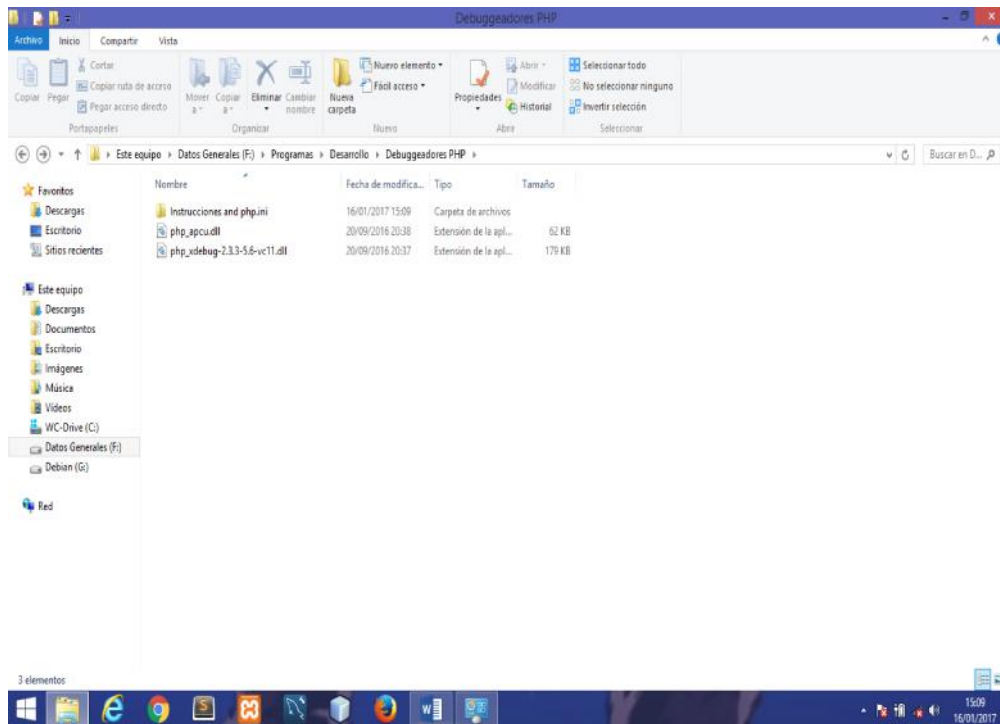


Fig. 9. Debugueadores de PHP.

7.3. DISEÑO DE LA APLICACIÓN

Escribir la siguiente dirección en cualquier navegador.



Fig. 10. URL para ingresar a la aplicación BigData Converter.

Se presentará la pantalla de inicio, donde se ingresarán las siguientes credenciales por defecto: Usuario: **administracion**, y contraseña: **123**.

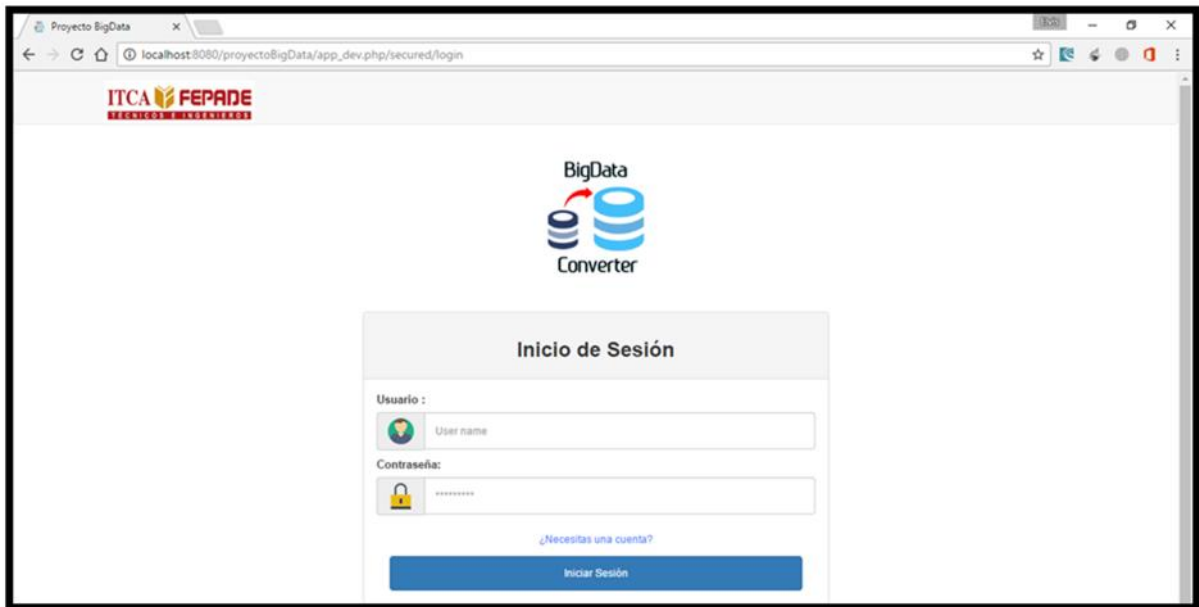


Fig. 11. Pantalla de Inicio de Sesión

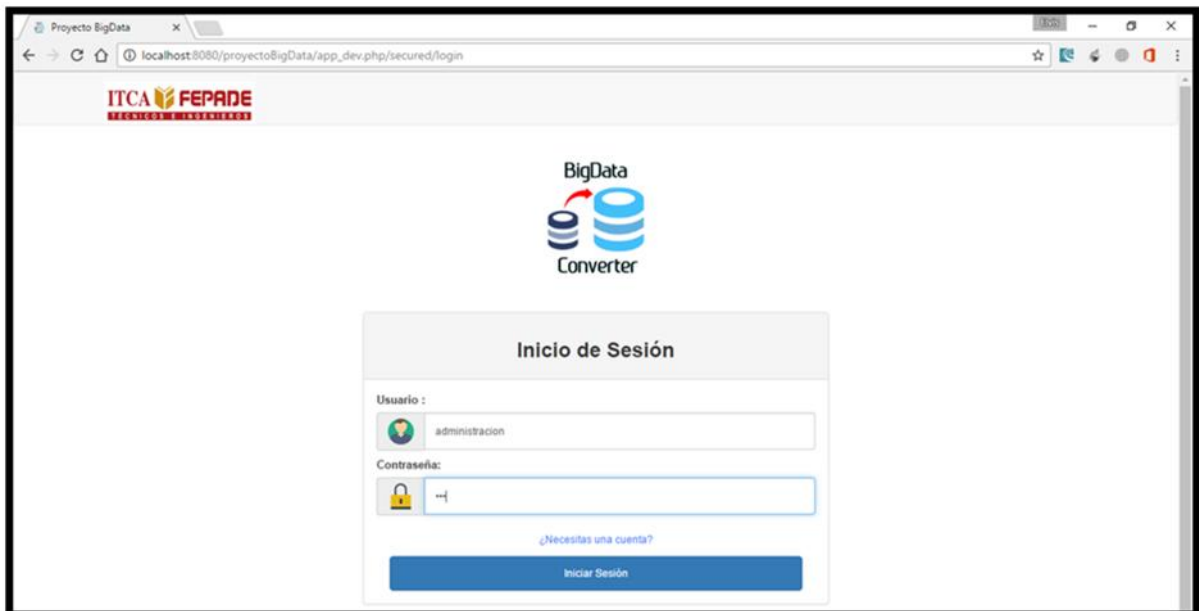


Fig. 12. Credenciales para el inicio de sesión.

Si el usuario y contraseña son correctas, se mostrará la siguiente pantalla de inicio.

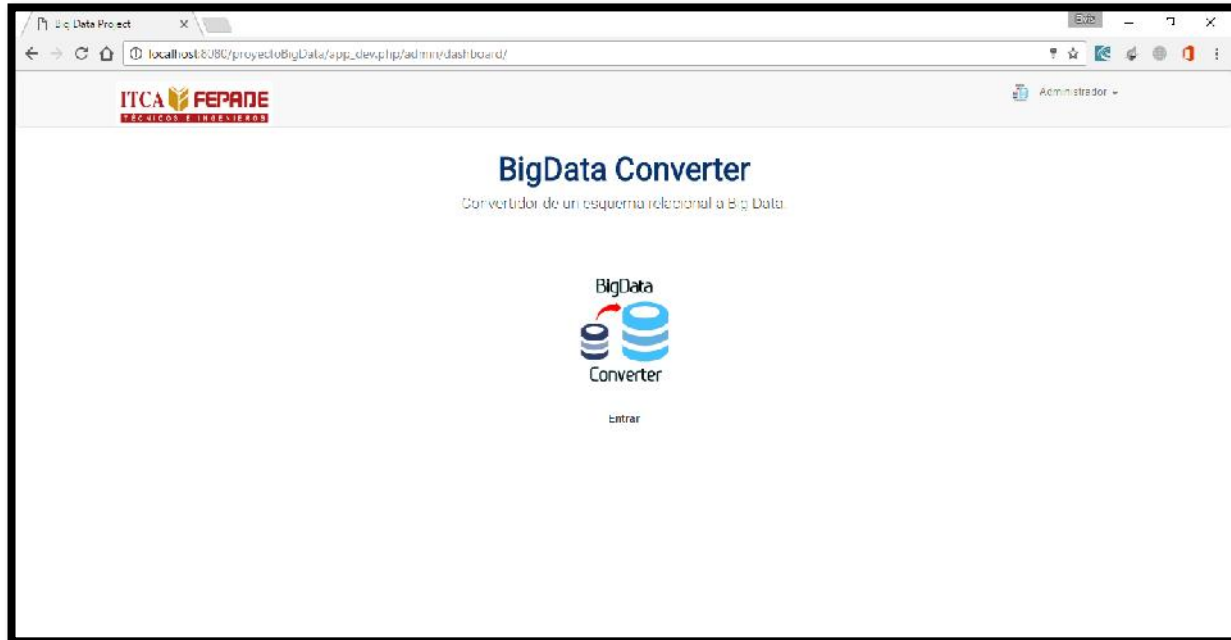


Fig. 13. Pantalla de inicio de BigData Converter.

Luego de dar click en el link **Entrar**, se mostrará la siguiente pantalla, donde se encuentra el panel principal. Dar click en el botón **Iniciar** para comenzar con la migración.

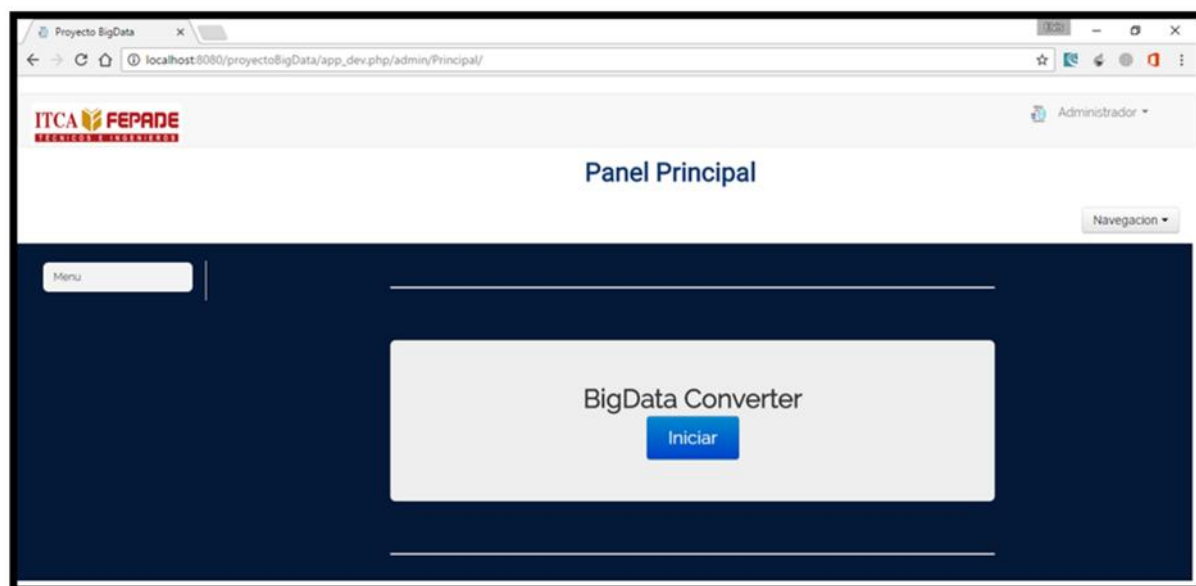


Fig. 14. Panel Principal de BigData Converter.

Se mostrará el siguiente asistente para iniciar el proceso de la migración.

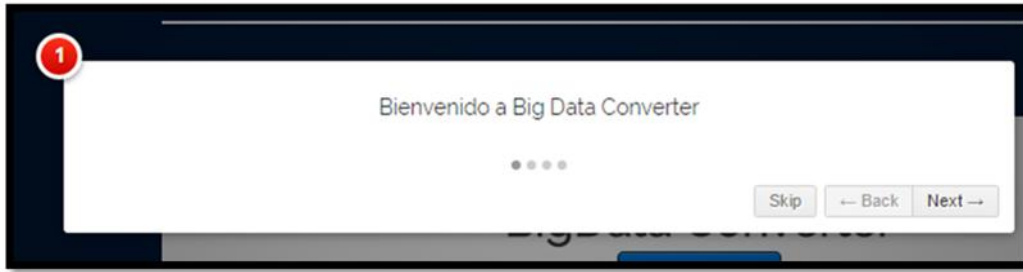


Fig. 15. Bienvenida a la herramienta BigData Converter.

Seleccionar el gestor de base de datos relacional. Por el momento el único gestor es SQL Server.

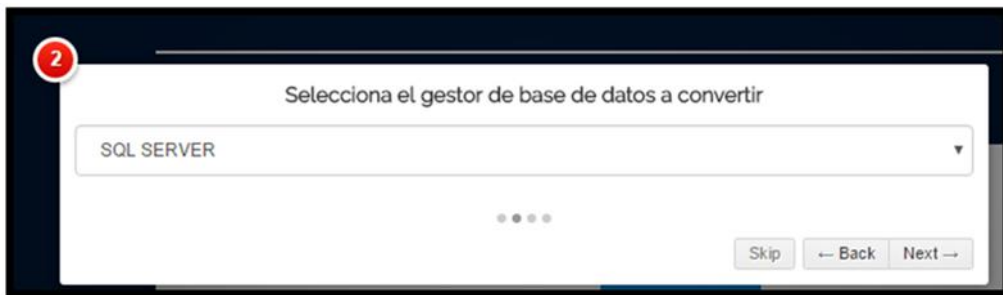


Fig. 16. Selección del gestor de base de datos relacional.

Ingresar el nombre del servidor de base de datos SQL Server, el puerto, el nombre del usuario del SGBD y su contraseña.

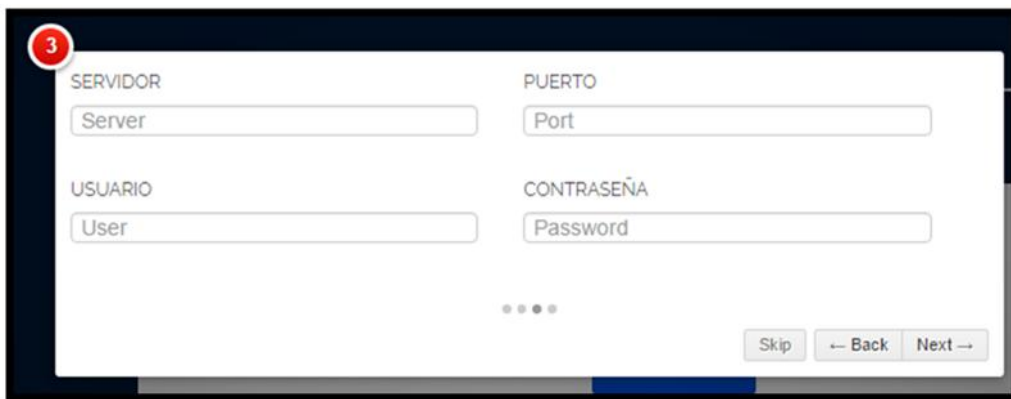


Fig. 17. Ingreso de datos de conexión para SQL Server.

3

SERVIDOR
DESKTOP-89E77LQ\SQLEXPRESS

PUERTO
Port

USUARIO
DESKTOP-89E77LQ\ITCA-COMPUTAC

CONTRASEÑA
Password

.....

Skip ← Back Next →

Fig. 18. Datos de ejemplo.

La aplicación mostrará todas las bases de datos que están creadas en el SGBD.



Fig. 19. Bases de datos encontradas en el gestor.

Se mostrará la siguiente pantalla de confirmación, antes de proceder a convertir la base de datos.



Fig. 20. Pantalla de confirmación de la base de datos.

Luego de confirmar la selección de la base de datos, se mostrarán todas las tablas que contiene la base de datos seleccionada.



Fig. 21. Lista de tablas de la base de datos seleccionada.

Al dar click en el botón **Convertir a BIGDATA**, se mostrará la siguiente pantalla donde se observa el proceso de conversión:

-)] Se extraen los datos de la base de datos en SQL server a través de la generación de una consulta SQL que incluye todas las tablas.
-)] Se genera un archivo intermedio con la información, en formato JSON.
-)] Se ejecuta desde el gestor MongoDB la instrucción para importar los datos que se encuentran en el archivo JSON a BigData.

Si la migración ha sido satisfactoria, se presentará el mensaje respectivo.

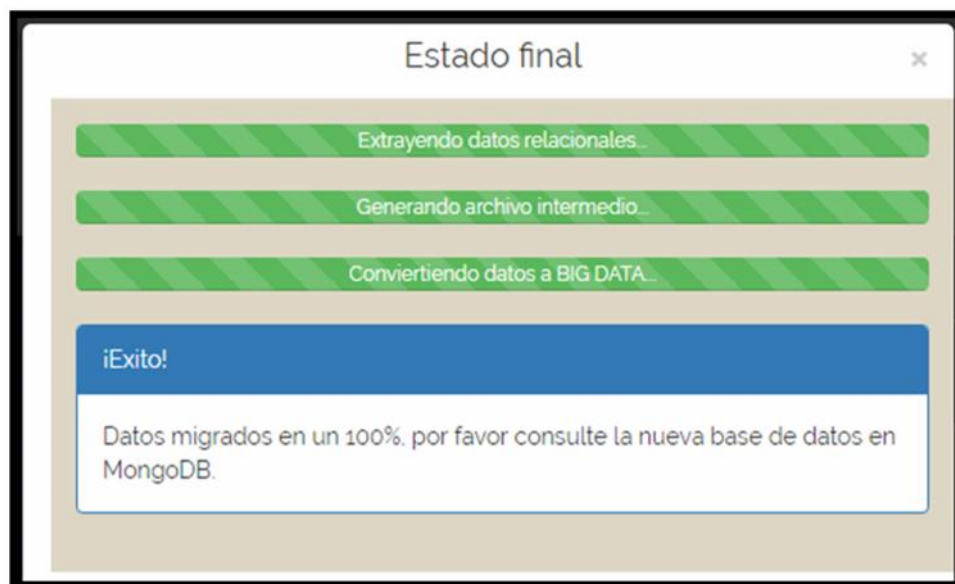


Fig. 22. Procesos realizados durante la migración.

7.4. DESARROLLO DE LA APLICACIÓN.

Para realizar el desarrollo de la aplicación, se utilizaron las siguientes herramientas de software:

-) Xampp: Motor de la base de datos MySQL,
-) Lenguaje de programación PHP ver. 5.6.3.
-) Servidor Apache,
-) SQL Server
-) Symfony: framework de PHP.
-) MongoDB: motor de base de datos BigData.
-) HTML5, JavaScript, JQuery, CSS3.
-) NetBeans: Editor de código PHP.

7.5. PRUEBAS DE LA APLICACIÓN.

El proceso que lleva a cabo la aplicación **BigData Converter** consiste en ejecutar un asistente que permite seleccionar la base de datos en un esquema relacional para ser transformada a un esquema BigData.

El proceso de transformación consiste en generar una consulta SQL que incluya los datos de todas las tablas en un solo comando. Para el ejemplo, se utilizó la base de datos NEPTUNO.

La consulta que se genera es la siguiente:


```

select c.*,pd.*,pe.*,dp.*,ca.*,pv.*,e.*,ca.*
from Pedidos pe
left outer join clientes c on c.idCliente=pe.IdCliente
left outer join Empleados e on e.IdEmpleado=pe.IdEmpleado
left outer join detallesdepedidos dp on dp.idpedido=pe.IdPedido
left outer join productos pd on pd.idproducto=dp.idproducto
left outer join categorias ca on ca.idcategoria=pd.idCategoria
left outer join proveedores pv on pv.idProveedor=pd.idProveedor
for json auto, root('neptuno')

```

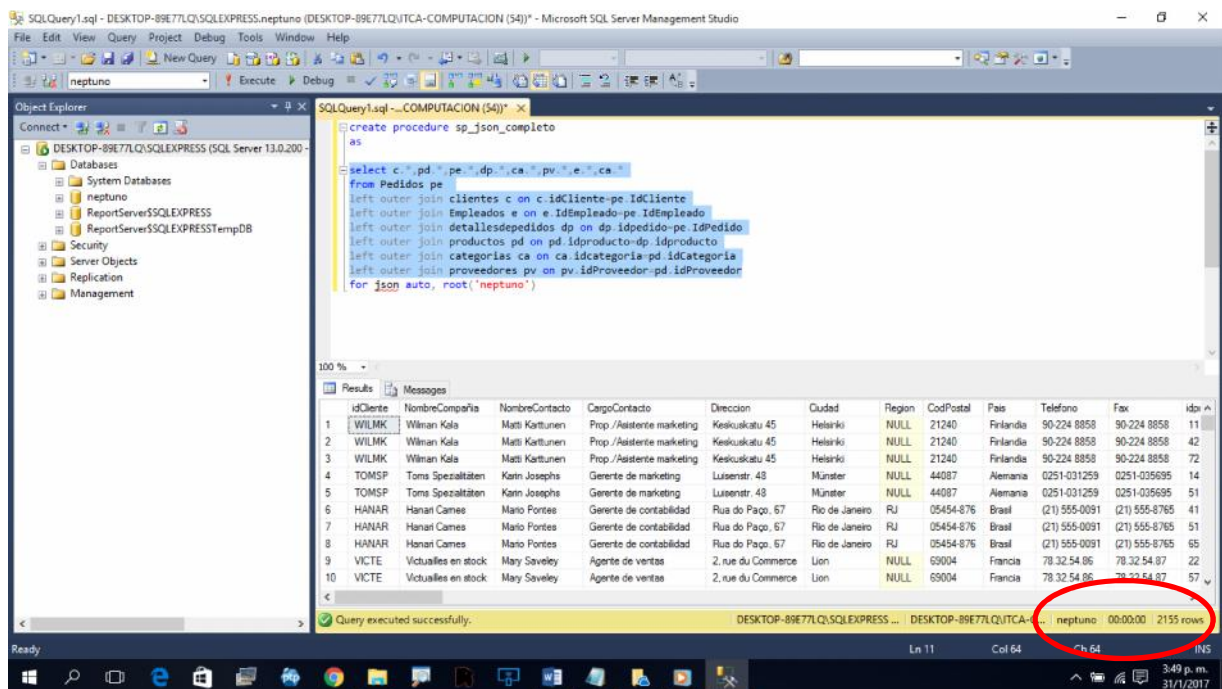


Fig. 23. Consulta SQL que recupera todos los datos de la base de datos relacional.

La consulta del ejemplo genera 2155 registros, que es el número de registros que contiene la tabla **detallesdepedidos**. Esta es la tabla transaccional ya que es la que tiene el mayor volumen de datos y es alrededor de la cual giran los datos de las demás tablas.

Con esta premisa se generará el archivo intermedio en formato JSON.

El archivo queda almacenado en la carpeta **jsonFiles**.

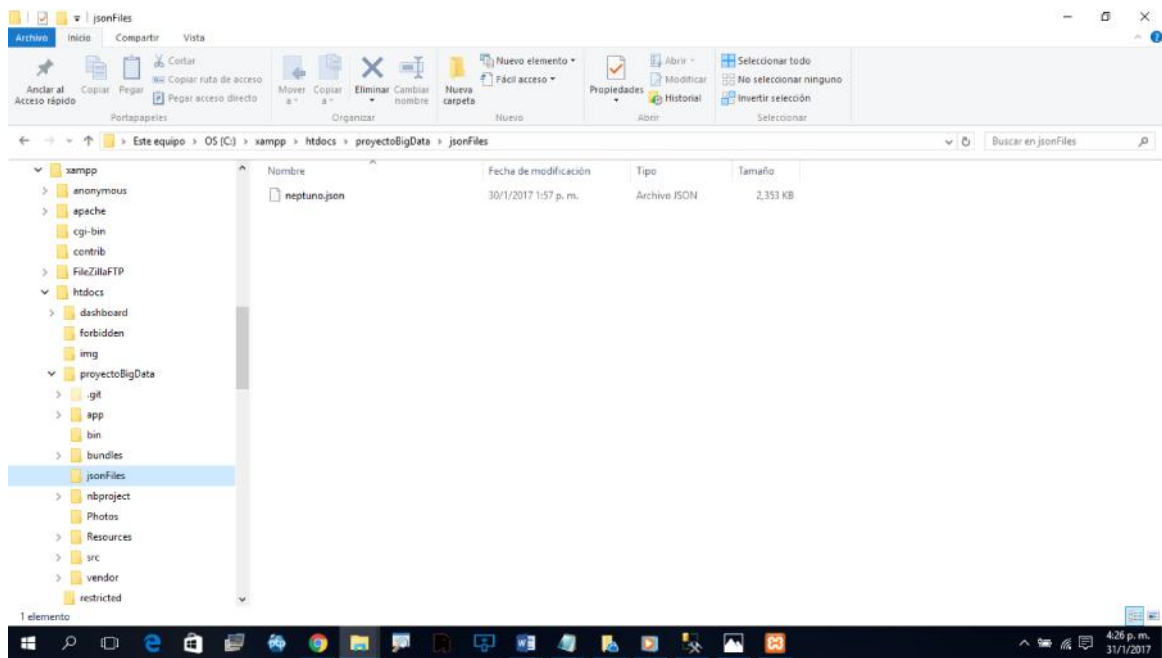


Fig. 24. Archivo JSON generado.

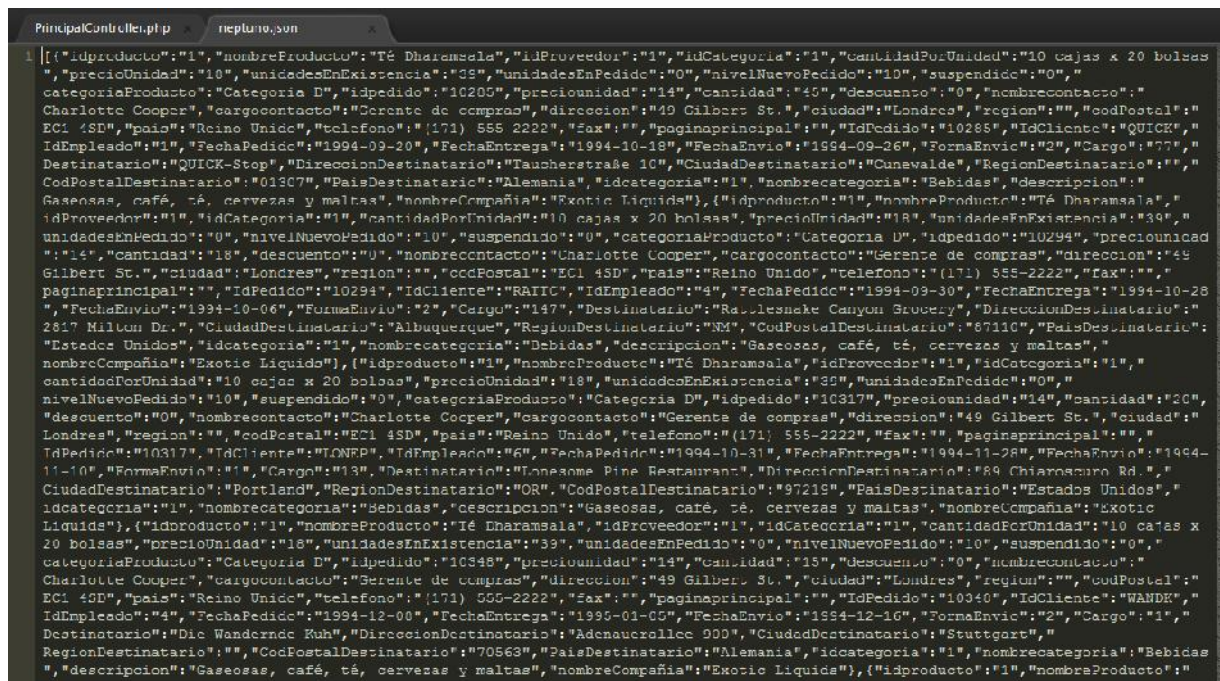


Fig. 25. Contenido del archivo JSON.

Luego que la herramienta crea el archivo JSON, ejecuta la importación desde el gestor de datos de BigData en MongoDB tomando como fuente la información del archivo JSON. Para verificar que los datos han sido migrados es necesario levantar el servidor de MongoDB, desde una ventana de símbolo del sistema con el comando **mongod**:

```
Simbolo del sistema - mongod
4 archivos      90,328 bytes
3 dirs  907,124,269,056 bytes libres

C:\Program Files\MongoDB\Server\3.2>cd bin

C:\Program Files\MongoDB\Server\3.2\bin>mongod
2017-01-30T11:25:20.271-0600 I CONTROL [initandlisten] MongoDB starting : pid=6076 port=27017 dbpath=C:\data\db\ 64-bit host=DESKTOP-89E77LQ
2017-01-30T11:25:20.273-0600 I CONTROL [initandlisten] targetMinOS: Windows 7/Windows Server 2008 R2
2017-01-30T11:25:20.273-0600 I CONTROL [initandlisten] db version v3.2.10
2017-01-30T11:25:20.275-0600 I CONTROL [initandlisten] git version: 79d9b3ab5ce20f51c272b4411202710a082d0317
2017-01-30T11:25:20.277-0600 I CONTROL [initandlisten] OpenSSL version: OpenSSL 1.0.1t-fips  3 May 2016
2017-01-30T11:25:20.280-0600 I CONTROL [initandlisten] allocator: tcmalloc
2017-01-30T11:25:20.282-0600 I CONTROL [initandlisten] modules: none
2017-01-30T11:25:20.284-0600 I CONTROL [initandlisten] build environment:
2017-01-30T11:25:20.287-0600 I CONTROL [initandlisten]     distmod: 2008plus-ssl
2017-01-30T11:25:20.289-0600 I CONTROL [initandlisten]     distarch: x86_64
2017-01-30T11:25:20.294-0600 I CONTROL [initandlisten]     target_arch: x86_64
2017-01-30T11:25:20.294-0600 I CONTROL [initandlisten] options: {}
2017-01-30T11:25:20.298-0600 I - [initandlisten] Detected data files in C:\data\db\ created by the 'wiredTiger' storage engine, so setting the active storage engine to 'wiredTiger'.
2017-01-30T11:25:20.301-0600 I STORAGE [initandlisten] wiredtiger_open config: create,cache_size=4G,session_max=20000,eviction=(threads_max=4),config_base=false,statistics=(fast),log=(enabled=true,archive=true,path=journal,compressor=snappy),file_manager=(close_idle_time=100000),checkpoint=(wait=60,log_size=2GB),statistics_log=(wait=0),
2017-01-30T11:25:21.573-0600 I NETWORK [HostnameCanonicalizationWorker] Starting hostname canonicalization worker
2017-01-30T11:25:21.573-0600 I FTDC [initandlisten] Initializing full-time diagnostic data capture with directory 'C:\data\db\diagnostic.data'
2017-01-30T11:25:21.593-0600 I NETWORK [initandlisten] waiting for connections on port 27017
```

Fig. 26. Ejecución del servidor de MongoDB.

En otra ventana de símbolo del sistema, iniciar MongoDB con el comando **mongo**:

```
Simbolo del sistema - mongo
Microsoft Windows [Versión 10.0.10240]
(c) 2015 Microsoft Corporation. Todos los derechos reservados.

C:\Users\ITCA-COMPUTACION>cd \
C:\>cd "program files"
C:\Program Files>cd mongodb
C:\Program Files\MongoDB>cd server
C:\Program Files\MongoDB\Server>cd 3.2
C:\Program Files\MongoDB\Server\3.2>cd bin
C:\Program Files\MongoDB\Server\3.2\bin>mongo
MongoDB shell version: 3.2.10
connecting to: test
>
```

Fig. 27. Ejecución del gestor MongoDB.

Con el comando **show dbs** verificar que se haya creado la base de datos. En nuestro ejemplo, la base de datos se llama Neptuno.

```
Símbolo del sistema - mongo
Microsoft Windows [Versión 10.0.10240]
(c) 2015 Microsoft Corporation. Todos los derechos reservados.

C:\Users\ITCA-COMPUTACION>cd \

C:\>cd "program files"

C:\Program Files>cd mongodb

C:\Program Files\MongoDB>cd server

C:\Program Files\MongoDB\Server>cd 3.2

C:\Program Files\MongoDB\Server\3.2>cd bin

C:\Program Files\MongoDB\Server\3.2\bin>mongo
MongoDB shell version: 3.2.10
connecting to: test
> show dbs
local    0.000GB
neptuno  0.002GB
>
```

Fig. 28. Listado de bases de datos de MongoDB.

Para acceder la base de datos, se emite la sentencia **use neptuno**. Luego de abrir la base de datos, ya es posible verificar la cantidad de registros con la función **db.neptuno.count()**. Debe coincidir con la cantidad de registros que se obtuvo en el gestor SQL server con la consulta emitida. En nuestro caso, podemos confirmar que, en efecto, son 2155 documentos. Ver figura 23.

```
Símbolo del sistema - mongo
C:\Program Files\MongoDB>cd server
C:\Program Files\MongoDB\Server>cd 3.2
C:\Program Files\MongoDB\Server\3.2>cd bin
C:\Program Files\MongoDB\Server\3.2\bin>mongo
MongoDB shell version: 3.2.10
connecting to: test
> show dbs
local    0.000GB
neptuno  0.003GB
> use neptuno
switched to db neptuno
> db.dropdatabase();
2017-01-30T13:52:41.812-0600 E QUERY [Lthread1] TypeError: db.dropdatabase is not a function :
@(shell):1:1
> db.dropDatabase();
{ "dropped" : "neptuno", "ok" : 1 }
> show dbs
local    0.000GB
neptuno  0.000GB
> use neptuno
switched to db neptuno
> db.neptuno.count();
2155
>
```

Fig. 29.

Conteo de registros.

Luego de contar los registros y comprobar que no se ha perdido ninguno, se puede proseguir a ver el contenido de la base de datos en MongoDB usando la sentencia **db.neptuno.find().pretty()**.

```
Selecionar Símbolo del sistema - mongo
Microsoft Windows [Versión 10.0.10240]
(c) 2015 Microsoft Corporation. Todos los derechos reservados.

C:\Users\ITCA-COMPUTACION>cd \

C:\>cd "program files"

C:\Program Files>cd mongodb

C:\Program Files\MongoDB>cd server

C:\Program Files\MongoDB\Server>cd 3.2

C:\Program Files\MongoDB\Server\3.2>cd bin

C:\Program Files\MongoDB\Server\3.2\bin>mongo
MongoDB shell version: 3.2.10
connecting to: test
> show dbs
local    0.000GB
neptuno  0.002GB
> use neptuno
switched to db neptuno
> db.find().pretty();
2017-01-30T11:30:45.936-0600 E QUERY [thread1] TypeError: db.find is not a function :
@(shell):1:1
> db.neptuno.find().pretty();
{
  "id" : ObjectId("588bb9cc7d7227012ba0ec44"),
```

Fig. 30. Sentencia find() para ver los registros.

```
Selecionar Símbolo del sistema - mongo
> db.neptuno.find().pretty();
{
  "_id" : ObjectId("588bb9cc7d7227012ba0ec44"),
  "idproducto" : "1",
  "nombreProducto" : "Té Dharamsala",
  "idProveedor" : "1",
  "idCategoria" : "1",
  "cantidadPorUnidad" : "10 cajas x 20 bolsas",
  "precioUnidad" : "18",
  "unidadesEnExistencia" : "39",
  "unidadesEnPedido" : "0",
  "nivelNuevoPedido" : "10",
  "suspendido" : "0",
  "categoriaProducto" : "Categoria D",
  "idpedido" : "10348",
  "preciounidad" : "14",
  "cantidad" : "15",
  "descuento" : "0",
  "nombrecontacto" : "Charlotte Cooper",
  "cargocontacto" : "Gerente de compras",
  "direccion" : "49 Gilbert St.",
  "ciudad" : "Londres",
  "region" : "",
  "codPostal" : "E1 4SD",
  "pais" : "Reino Unido",
  "telefono" : "(171) 555-2222",
  "fax" : "",
  "paginaPrincipal" : "",
  "idPedido" : "10348",
  "idCliente" : "WANDK",
```

Fig. 31. Información devuelta.

Si se desea hacer búsquedas específicas de algún documento en especial o grupo de documentos para validar la información, se puede emitir la sentencia find() especificando alguna condición de búsqueda. Por ejemplo, db.neptuno.find("IdCliente":"CHOPS").pretty() y luego db.neptuno.count("IdCliente":"CHOPS") para confirmar el número de documentos devueltos y comparar que coincidan con los registros devueltos por la consulta SQL.

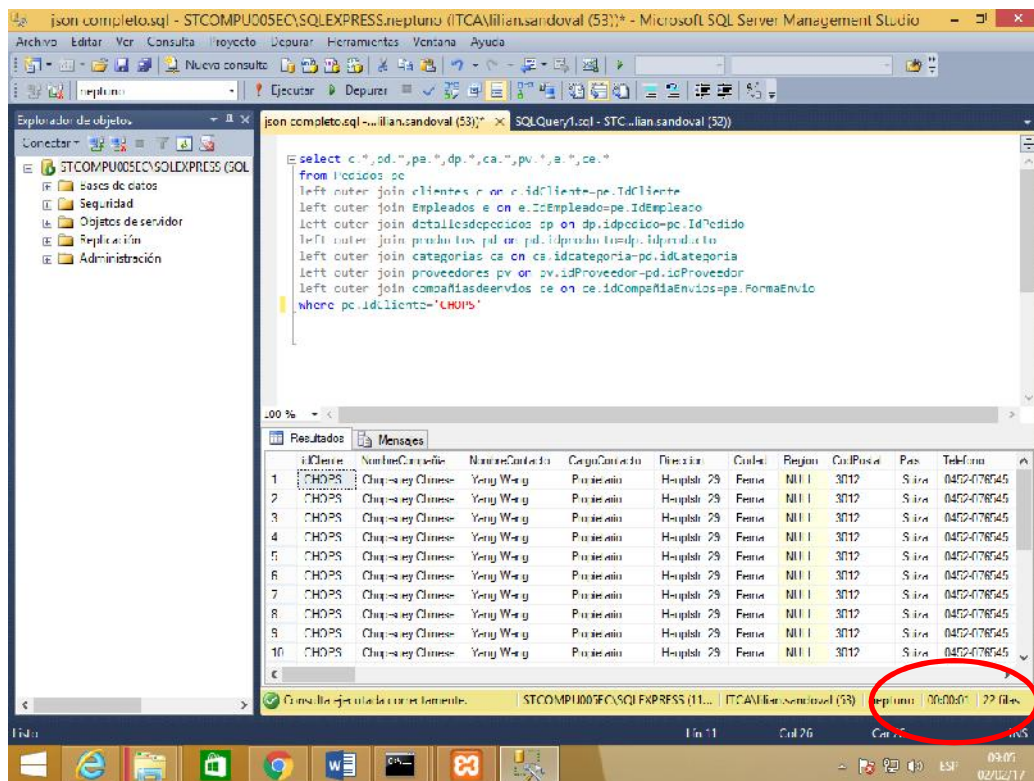


Fig. 32. Consulta en SQL para realizar una búsqueda específica.

```
Símbolo del sistema - mongo
{"unidadesEnExistencia": "24",
"unidadesEnPedido": "0",
"nivelNuevoPedido": "5",
"suspendido": "0",
"categoriaProducto": "Categoria D",
"idpedido": "11029",
"preciounidad": "44",
"cantidad": "12",
"descuento": "0",
"nombrecontacto": "Ian Devling",
"cargcontacto": "Gerente de marketing",
"direccion": "74 Rose St.Moonie Ponds",
"ciudad": "Melbourne",
"region": "Victoria",
"codPostal": "3058",
"pais": "Australia",
"telefono": "(03) 444-2343",
"fax": "(03) 444-6588",
"paginaPrincipal": "",
"IdPedido": "11029",
"IdCliente": "CHOPS",
"IdEmpleado": "4",
"FechaPedido": "1996-05-16",
"FechaEntrega": "1996-06-13",
"FechaEnvio": "1996-05-27",
"FormaEnvio": "1",
"Cargo": "48",
"Destinatario": "Chop-suey Chinese",
"DireccionDestinatario": "Hauptstr. 31",
"CiudadDestinatario": "Bern",
"RegionDestinatario": "",
"CodPostalDestinatario": "3012",
"PaisDestinatario": "Suiza",
"idcategoria": "2",
"nombrecategoria": "Condimentos",
"descripcion": "Salsas dulces y picantes, delicias, comida para untar y aderezos",
"nombreCompañia": "Pavlova, Ltd."
}
Type "it" for more
> db.neptuno.count(<<"IdCliente":"CHOPS">>);
22
>
```

Fig. 33. Búsqueda en MongoDB para comparar resultados.

8. CONCLUSIONES

-) Con el desarrollo de la investigación se está brindando a nivel de país, primeramente, el conocimiento básico de la tecnología BigData, la cual aún no ha sido ampliamente difundida; y luego se está aportando de una herramienta que, aunque se encuentra en su primera versión, puede auxiliar a las empresas que tomen la decisión de migrar sus sistemas a BigData sin necesidad de comenzar la alimentación de sus bases de datos desde cero.
-) Se hizo una recopilación de terminología importante sobre el nuevo esquema Big Data, profundizando en los diferentes métodos de migración de datos desde un sistema relacional hacia un sistema de base de datos Big Data, que para nuestro estudio fue MongoDB. Se expusieron las diferentes técnicas de migración y validación de los datos migrados.

-
-) Se seleccionó una metodología donde las instrucciones se emiten desde los gestores de bases de datos, tanto relacional, como BigData para realizar la migración.
 -) Generalmente, a nivel académico, se enseña los métodos para conectarse a una base de datos específica, pero en nuestro caso, teníamos que encontrar las herramientas necesarias para poder recuperar todas las bases de datos existentes en un servidor, sin saber el nombre de estas ni la ubicación. Lo que se pudo realizar exitosamente.
 -) Se generó un asistente para pedir la información sobre la base de datos que se desea migrar, mostrando todas las bases de datos encontradas en el sistema haciendo uso de las herramientas anteriores.
 -) En el gestor de base de datos SQL Server 2016 ya trae incorporadas las funciones necesarias para enviar la información recuperada desde la base de datos relacional hacia un archivo intermedio en formato JSON, el que luego se usa como insumo para convertir la información a BigData. Pero se tuvo el inconveniente que el lenguaje PHP aún no posee la capacidad de ejecutar estas funciones, por lo que no se pudo realizar esta acción desde el gestor SQL Server. La solución fue buscar otras herramientas de PHP que realizaran la acción de crear el archivo JSON a partir de la información recolectada con consultas SQL ejecutadas dentro de PHP.
 -) Nos encontramos con el problema de la configuración de caracteres regionales como la ñ y vocales tildadas en los datos, pues no eran reconocidos al crear el archivo JSON, siendo sustituidos por otros caracteres desconocidos. Se investigó sobre las funciones de PHP resuelven este tipo de problema y se logró corregir. Pero luego se dió el problema que también los nombres de las tablas y los campos pueden contener este tipo de caracteres, y para lo cual, si no se encontraron funciones que lo corrigieran, por lo que se tuvieron que crear estas funciones para resolver el problema, y que el archivo JSON reflejara exactamente la información contenida en la base de datos relacional.
 -) Se investigó sobre los comandos necesarios para ejecutar el archivo JSON desde MongoDB y poder importar los datos al nuevo esquema.
 -) Se realizaron las pruebas necesarias para comprobar que los datos fueron migrados correctamente y se concluyó que, efectivamente, los datos se habían convertido de manera exitosa.
 -) No se logró realizar una prueba de stress pues no se tuvo acceso a una base de datos lo suficientemente grande para poder verificar tiempos de respuesta y poder constatar la eficiencia de tener una base de datos en un esquema BigData. Aunque por los estudios realizados a la fecha por organizaciones competentes, se tiene el dato que entre mas grande es la base de datos, es mas eficiente la tecnología BigData.

9. RECOMENDACIONES

- J Tomar en cuenta que antes de comenzar la migración, se haga la configuración recomendada en el servidor y la instalación de las librerías necesarias para poder ejecutar las funciones utilizadas en el código de la aplicación.
- J Tener los permisos necesarios como administrador o dueño de las bases de datos para poder tener acceso a ellas.
- J Depurar la base de datos del esquema relacional, que se cumplan todas las reglas de validación y de integridad referencial.
- J Que todas las tablas de la base de datos estén relacionadas, de preferencia, pues es a través de las relacionales que se va integrando la información en las consultas para recolectar los datos en forma integral.

10. REFERENCIAS BIBLIOGRÁFICAS

Libros:

- [1] V. Mayer Schonberger, K. Cukier. Big Data: La Revolución de los Datos Masivos. Editorial Turner. 2013.
- [2] O'Really Media Inc. Big Data Now. Kindle Edition. 2012.
- 2 [3] José López Quijano. Domine PHP y MySQL: Programación Dinámica en el Lado del Servidor. Editorial Alfaomega. Mexico. 2007.
- 3 [4] Abraham Gutiérrez Rodríguez. PHP 5: a Través de Ejemplos. Editorial: Alfaomega.
- 4 Mexico. 2005.

Artículos presentados en conferencias:

- [5] L. J. Sandoval "Design of Business Intelligence Applications Using Big Data Technology" en Proc. "Proceedings of the 2015 IEEE Central America and Panama Convention (CONCAPAN XXXV)". 2015. IEEE.

Documentos en Internet:

- [6] MongoDB en Español. Tomo I – El Principio. Yohan Graterol.
Disponible: <https://github.com/yograterol/ebook-mongodb-basico>
- [7] RDBMS to MongoDB. Migration Guide. A MongoDB White Paper.
Disponible: <https://webassets.mongodb.com/RDBMStoMongoDBMigration.pdf>
- [8] IBM Developer Works. Que es Big Data?

Disponible: <https://www.ibm.com/developerworks/ssa/local/im/que-es-big-data/>

[9] Brandchats. Tipos de datos que comprende el Big Data.

Disponible: <http://www.brandchats.com/7-tipos-de-datos-que-comprende-el-big-data/>

[10] Manual de PHP.

Disponible: <http://php.net/manual/es/index.php>

[11] IBM developerWorks. ¿Qué es Big Data?

Disponible: <https://www.ibm.com/developerworks/ssa/local/im/que-es-big-data/>

Videos:

[12] Alexander Copquin and Vlad Carballo, The Knot. Migration from SQL to MongoDB – A Case Study at Theknot.com.

Disponible: <https://www.mongodb.com/presentations/migration-from-sql-to-mongodb-a-case-study-at-theknot-com>.

11. ANEXOS.

A continuación, se muestran los programas principales que integran la aplicación.

ARCHIVO: layout2.html.twig. Programa que muestra el panel principal y el asistente para la selección de la base de datos relacional.

```
{% extends ':Layout:layout2.html.twig' %}
{% block css %}
    {{parent()}}
    <link href="{{ asset('Resources/intro/assets/css/bootstrap.min.css')}}" rel="stylesheet">
    <link href="{{ asset('Resources/intro/assets/css/demo.css')}}" rel="stylesheet">
    <!-- Add IntroJs styles -->
    <link href="{{ asset('Resources/intro/introjs.css')}}" rel="stylesheet">
    <link href="{{ asset('Resources/intro/assets/css/bootstrap-responsive.min.css')}}" rel="stylesheet">
    <link rel="stylesheet" href="{{ asset('Resources/sweetalert-master/dist/sweetalert.css')}}">
{% endblock css%}
{% block js %}
    {{parent()}}
    <script type="text/javascript" src="{{ asset('Resources/intro/intro.js')}}"></script>
    <script type="text/javascript" src="{{ asset('Resources/js/dbc/dbc.js')}}"></script>
    <script src="{{ asset('Resources/sweetalert-master/dist/sweetalert.min.js') }}"></script>
```

```
{% endblock %}
```

```
{% block descripcion_y_acciones %}
```

```
<div class="row">
```

```
{% block descripcion %}
```

```
<div class="col-md-8">
```

```
<p class="robo" style="font-weight: 300; margin-bottom: 0px; font-size: 30px;text-align:center;margin-left:70%;color:07326B;"><b>Panel Principal</b></p>
```

```
</div>
```

```
{% endblock%}
```

```
{% block acciones %}
```

```
<div class="col-md-4" style="margin-top: 50px;">
```

```
{{parent()}}
```

```
</div>
```

```
{% endblock %}
```

```
</div>
```

```
{% endblock %}
```

```
{% block navegacion_Y_body %}
```

```
<div class="row" style='background-color:031937;'>
```

```
{% block navegacion %}
```

```
<hr style="margin-top: 0px !important;">
```

```
<div class="col-md-2 colorAnclas" style="border-right: 1px solid #f3f3f3;">
```

```
<!-- Left Navigation -->
```

```
<ul style="list-style: none; padding: 0;">
```

```
<a href="{{ path('dashboard_index') }}" class="leftnava colorAnclas"><li class="leftnav left active">Menu<span class="pull-right right"></span></li></a>
```

```
</ul>
```

```
</div>
```

```
{% endblock %}
```

```
{% block cuerpo %}
```

```
<div class="col-md-10">
```

```
<div class="row">
```

```
<div class="container-narrow">
```

```
<div class="masthead">
```

```
</div>
```

```
<hr>
```

```
<div class="jumbotron">
```

```
<h4 id="step1" style='font-size: 30px !important ;'>BigData Converter</h4>
```

```
<a class="btn btn-primary" href="javascript:void(0);" onclick="startIntro();">Iniciar</a>
```

```

</div>
<hr>
</div>

<script type="text/javascript">
function startIntro(){
var intro = introJs();
intro.setOptions({
  steps: [
    {
      intro: "<p style='text-align:justify;'><center>Bienvenido a Big Data Converter</center></p>"
    },
    {
      intro: "<center><p style='text-align:justify;'><b>Selecciona el gestor de base de datos a
convertir</b></p></center>\n\
      <select class='form-control' name='dbMotor' id='dbMotor'>\n\
        <option value='1'>SQL SERVER</option>\n\
        <option value='2'>Mysql</option>\n\
      </select>"
    },
    {
      intro: '<div class="container">\n\
        <div class="row">\n\
          <div class="form-column col-sm-3">\n\
            <div class="form-group" >\n\
              <label for="host" class="control-label">SERVIDOR</label>\n\
              <input type="text" class="form-control" id="host" placeholder="Server" name="host"
>\n\
            </div>\n\
          </div>\n\
          <div class="form-column col-sm-3">\n\
            <div class="form-group" >\n\
              <label for="host" class="control-label">PUERTO</label>\n\
              <input type="text" class="form-control" id="port" placeholder="Port" name="port"
>\n\
            </div>\n\
          </div>\n\
        </div>\n\
      <div class="col-sm-6"></div>\n\
      <div class="clearfix"></div>\n\
      <div class="form-column col-sm-3">\n\
        <div class="form-group" >\n\
          <label for="user" class="control-label">USUARIO</label>\n\
          <input type="text" class="form-control" id="user" placeholder="User" name="user"

```

```

>\n\
        </div>\n\
    </div>\n\
    <div class="form-column col-sm-3"><div class="form-group" >\n\
        <label for="pass" class="control-label">CONTRASEÑA</label>\n\
        <input type="password" class="form-control" id="pass" placeholder="Password"
name="pass" >\n\
        </div>\n\
    </div>\n\
    <div class="col-sm-6"></div>\n\
    </div></div>'
    },
    {
        element: '#step5',
        intro: 'Bases de datos encontradas'
    }
]
});
intro.start();
}
</script>
</div>
</div>

```

{#Modal donde se listan todas las bases de datos encontradas#}

```

    <div class="modal-content modal fade modal" id="modal" tabindex="-1" role="dialog" aria-
labelledby="myModalLabel" aria-hidden="true">
        <div class="modal-header " Style="height:45px;">
            <button type="button" class="close" data-dismiss="modal"><span aria-
hidden="true">&times;</span><span class="sr-only">Cerrar</span></button>
            <p id="encabezado" style="text-align: center;font-size: 20px;">Bases de datos
encontradas en el servidor SQL</p>

        </div>
        <div class="modal-body" id="bdExistentes" style='background-color:DFD7C5;'>
        </div>
    </div>

{% endblock %}
</div>
{% endblock %}

```

```
{% block js_extend %}
```

```
{% endblock %}
```

ARCHIVO: PrincipalController.php. Programa que controla el procesamiento de toda la información y lleva a cabo las reglas de la lógica del negocio.

```
<?php
```

```
namespace DG\AdminBundle\Controller;
use Symfony\Component\HttpFoundation\Request;
use Symfony\Component\HttpFoundation\JsonResponse;
use Symfony\Component\HttpFoundation\Response;
use Symfony\Bundle\FrameworkBundle\Controller\Controller;
use Sensio\Bundle\FrameworkExtraBundle\Configuration\Method;
use Sensio\Bundle\FrameworkExtraBundle\Configuration\Route;
use Symfony\Component\HttpKernel\Exception;

/**
 * Principal controller.
 *
 * @Route("admin/Principal")
 */
class PrincipalController extends Controller
{
    /**
     * Lists all Principal entities.
     *
     * @Route("/", name="admin_index_principal",options={"expose"=true})
     * @Method("GET")
     */
    public function indexAction()
    {
        return $this->render('principal/dbc.html.twig', array(
        ));
    }

    /**
     * @Route("/mostrarBd/data", name="mostrarBd", options={"expose"=true})
     * @Method("POST")
     */
}
```

```

public function MostrarBdAction(Request $request) {
    $isAjax = $this->get('Request')->isXMLHttpRequest();
    if($isAjax){

        $em = $this->getDoctrine()->getManager();
        $tipoBD = $request->get('dbMotor');
        $user = $request->get('user');
        $host = $request->get('host');
        $port = $request->get('port');
        $pass = $request->get('pass');
        switch ($tipoBD) {

            case 1:
                $dsn = "Driver={ODBC Driver 11 for SQL
Server};Server={".$host."};Database=tempdb;Trusted_Connection=yes;ColumnEncryption=Enabled;";
                $usuario = $user;
                $clave=$pass;

                try {
                    $cid=odbc_connect($dsn, $usuario, $clave);
                } catch (\Exception $exc) {
                    $datas["estado"]=false;
                    return new JsonResponse($datas);
                }
                if (!$cid){
                    var_dump('Error en la conexion :(');
                }
                else{
                    $PARAMETROS[0]=$host;
                    $PARAMETROS[1]=$user;
                    $PARAMETROS[2]=$pass;
                    $PARAMETROS[3]=$port;

                    $_SESSION["parametros"]=$PARAMETROS;

                    $valores = array();
                    $sql="SELECT name FROM master.dbo.sysdatabases";
                    $result=odbc_exec($cid,$sql);
                    while(odbc_fetch_row($result)){
                        for($i=1;$i<=odbc_num_fields($result);$i++){
                            $valores[]=odbc_result($result,$i);
                        }
                    }
                }
            }
        }
    }
}

```

```

        }
    }
}
break;
default:
var_dump("Valores invalidos");
break;
}

return new Response(json_encode($valores));
}
}

/**
 * @Route("/mostrarTablas/data/", name="mostrarTablas", options={"expose"=true})
 * @Method("POST")
 */

public function MostrarTablasAction(Request $request) {
    $isAjax = $this->get('Request')->isXMLHttpRequest();
    if($isAjax){
        $em = $this->getDoctrine()->getManager();
        $nombreBase = $request->get('nombreBase');
        $_SESSION["estado"]=$nombreBase;
        $valor = $this->retornarTablas($nombreBase);
        $var= array_map('utf8_encode', $valor);
        return new JsonResponse($var);
    }
}

/**
 * @Route("/conversion/data/", name="conversion", options={"expose"=true})
 * @Method("POST")
 */

public function ConversionAction(Request $request) {
    header("Content-Type:text/html;charset=utf-8");
    $isAjax = $this->get('Request')->isXMLHttpRequest();

    if($isAjax){
        $em = $this->getDoctrine()->getManager();
        $nombreBase = $_SESSION["estado"];
    }
}

```

```

/*objeto de conexion*/
$cid = $this->crearConexion($nombreBase);

/*Tablas de la base de datos*/
$tablasBase = $this->retornarTablas($nombreBase);
$numero = count($tablasBase);

$metaDataTablas = $this->retornarTablasSegunGerarquia($nombreBase);
$_SESSION["metadata"]=$metaDataTablas;

//Crear Json File
$metaTabla = $_SESSION["metadata"];

$SUPERSQL="";
$superSQL1="SELECT ";
$superSQL2=" from ";
$superSQL3="\n";

//Generación de la consulta SQL que extrae todos los datos.
$tablasOrdenadas = $_SESSION["banderas"];

for ($l=0;$l<count($tablasOrdenadas);$l++) {

    if ($l==0){
//        $alias = $this->retornarAlias($tablasOrdenadas[$l]);
        $superSQL2.=$tablasOrdenadas[$l];
    }
    if($l<count($tablasOrdenadas)-1){
        $setx=".*,";
    }else{
        $setx=".*";
    }
//    $alias = $this->retornarAlias($tablasOrdenadas[$l]);
    $superSQL1.=" ".$tablasOrdenadas[$l].$setx;

}

//

// Se generan todos los left outter joins para generar la consulta SQL.

foreach ($metaTabla as $row){

```

```

    $superSQL3.=" left outer join ".$row["tablaHija"]." on ".$row["tablaPadre"]." ".$row["pk"]."
= " ".$row["tablaHija"]." ".$row["llaveExtranjera"]." \n";

    }
    $v= array();
    $SUPERSQL=$superSQL1.$superSQL2.$superSQL3;

    $result=odbc_exec($cid,$SUPERSQL);

    while ($info = odbc_fetch_array($result)) {
        $nuevoArray = $this->utf8_converter2($info);
        $v[] = array_map('utf8_encode', $nuevoArray);
    }

    $near = json_encode($v, JSON_UNESCAPED_UNICODE);
//    print(json_last_error_msg());

    $fp = fopen('jsonFiles/'.$nombreBase.'.json', 'w+');
    fwrite($fp, print_r($near,true));
    fclose($fp);

    $ultima_linea = system('"C:\Program Files\MongoDB\Server\3.2\bin\mongoimport" --db
'.$nombreBase.' --collection '.$nombreBase.' --file
C:\xampp\htdocs\proyectoBigData\jsonFiles\\'.$nombreBase.'.json --jsonArray', $retval);

    $data["estado"]=true;
    return new JsonResponse($data);
    }
}

function utf8_converter2(&$array) {
    foreach ($array as $key =>$row) {
        if(!mb_detect_encoding($key, 'utf-8', true)){
            $key2 = utf8_encode($key);
            $array[$key2]=$array[$key];
            unset($array[$key]);
        }
    }
    return $array;
}

function retornarAlias($tabla) {

```

```

$valor = str_split($tabla);
$dimension = count($valor);
$retorno = $valor[0].$valor[$dimension-3].$valor[$dimension-2].$valor[$dimension-1];
return $retorno;
}
function crearConexion($nombreBase) {
    if ($_SESSION["parametros"]){
        $parametros = $_SESSION["parametros"];
        $dsn = "Driver={ODBC Driver 11 for SQL
Server};Server={".$parametros[0]."};Database=".$nombreBase.";Trusted_Connection=yes;ColumnEncryption=Enabled;";
        $usuario = $parametros[1];
        $clave = $parametros[2];
        $cid = odbc_connect($dsn, $usuario, $clave);
        return $cid;
    }
}

function retornarTablasSegunGerarquia($nombreBase) {
    $cid = $this->crearConexion($nombreBase);
    $tablasNormales = $this->retornarTablas($nombreBase);
    if (!$cid){
        var_dump('Error en la conexion :(');
    }
    else{
        $valor = array();
        $tablas = array();
        for ($i=0;$i<count($tablasNormales);$i++) {
            $sql = " EXEC sp_fkeys ".$tablasNormales[$i]."";
            $result = odbc_exec($cid,$sql);
            while(odbc_fetch_row($result)){
                for($k=1;$k<=odbc_num_fields($result);$k++){
                    $valor[] = odbc_result($result,$k);
                }
            }

            $tablas[$tablasNormales[$i]] = ($valor);
            unset($valor);
            $valor = array();
        }

        //Empezamos el ordenamiento de las llaves foraneas de los registros
        $cantidadForaneas = 0;

```

```

$nArreglo = array();
$banderas = array();
$p=0;
foreach ($tablas as $registro){
    $p=$p+1;
    for ($m=0;$m<count($registro);$m=$m+14){
        if($p==1){
            $nombrePrimeraTabla=$registro[2];
        }
    }
}

```

```

$banderas[]=$nombrePrimeraTabla;

```

```

$existenciaHijo = false;
$existenciaPadre = false;
foreach ($tablas as $registro){

    for ($m=0;$m<count($registro);$m=$m+14){
        $numeroBanderas=count($banderas);
        for($j=0;$j<$numeroBanderas;$j++){

            if($banderas[$j]==$registro[$m+2]){
                $existenciaPadre = true;
                for ($k=0;$k<$numeroBanderas;$k++){
                    if ($banderas[$k]==$registro[$m+6]){
                        $existenciaHijo = true;
                        exit();
                    }
                }
            }
        }

        if ($existenciaPadre == false){
            for ($g=0;$g<$numeroBanderas;$g++){
                if ($banderas[$g]==$registro[$m+6]){
                    $existenciaHijo = true;
                }
            }
        }
    }
}
if($existenciaHijo==false && $existenciaPadre== true){

```

```

$nArreglo=["tablaPadre"=>$registro[$m+2],"pk"=>$registro[$m+3],"tablaHija"=>$registro[$m+6],"laveExtranjera"=>$registro[$m+7]];
        $banderas[]=$registro[$m+6];
        }else if($existenciaHijo==true && $existenciaPadre== false){

$nArreglo=["tablaPadre"=>$registro[$m+6],"pk"=>$registro[$m+7],"tablaHija"=>$registro[$m+2],"laveExtranjera"=>$registro[$m+3]];
        $banderas[]=$registro[$m+2];
        }
        }
        $existenciaHijo = false;
        $existenciaPadre = false;

        }
        $_SESSION["banderas"]=$banderas;
//        var_dump($banderas);
//        die();
        return $nArreglo;

    }
}

function retornarTablas($nombreBase) {
    $cid= $this->crearConexion($nombreBase);
    if (!$cid){
        var_dump('Error en la conexion');
    }
    else{
        $valor = array();
        $sql="SELECT name FROM sys.tables WHERE type = 'U' AND sys.tables.name != 'sysdiagrams'
ORDER BY create_date asc;";
        $result=odbc_exec($cid,$sql);
        while(odbc_fetch_row($result)){
            for($i=1;$i<=odbc_num_fields($result);$i++){
                $valor[]=odbc_result($result,$i);
            }
        }
    }
    return $valor;
}

```

DBC.JS (Funcionamiento, dinamismo y validación de la aplicación)

```

$(document).ready(function(){
    $("#salirConversion").hide();
    var contadorNext = 0;
    var dbMotor;
    $(document).on("click",".introjs-nextbutton",function() {
        contadorNext= contadorNext+1;
        if (contadorNext==2){
            dbMotor = $("#dbMotor").val();
            if (dbMotor==1){
//                Aqui tiene que ir el valor de la variable del append
                var textoFormulario =";
                $("#.introjs-tooltiptext").append(textoFormulario);
            }

        }else if (contadorNext==3){
            var user=$("#user").val();
            var pass=$("#pass").val();
            var port=$("#port").val();
            var host=$("#host").val();

            if (user!="" && host !=""){
                $.ajax({
                    type: 'POST',
                    async: false,
                    dataType: 'json',
                    data: {user:user,pass:pass,port:port,host:host,dbMotor:dbMotor},
                    url: Routing.generate('mostrarBd'),
                    success: function (data)
                    {
                        if (data.estado==false){
                            $("#.introjs-prevbutton").click();
                        }else{
                            var form ="";
                            //data = jQuery.parseJSON(data);
                            $.each(data, function( k, v ) {

                                var form="<div class='row' style='margin-top:20px;'><center>\n\
                                <div class='col-md-12' style='font-size:16px;'><b>"+v+"</b></div>\n\
                                <div class='clearfix'></div>\n\
                                <div class='col-md-4'></div>\n\
                                <div class='col-md-4 bdSeleccionada' value='"+v+"' id='"+v+"'><button class='btn-
                                success'>Seleccionar</button></div>\n\
                                <div class='col-md-4'></div><div class='clearfix'></div></center></div>";

```

```

        $("#bdExistentes").append(form);
    });

    $("#modal").modal();
    $(".introjs-skipbutton").click();
    }
    },
    error: function (xhr, status)
    {
    }
    });

    //Para el modal
    }else{
        //Si el valor del usuario es igual a nada entonces volvemos atras
        $(".introjs-prevbutton").click();
    }

    }else if(contadorNext==4){
        //Si el valor del usuario es igual a nada entonces volvemos atras
        $(".introjs-skipbutton").click();
        $(".modal").modal();
    }

    });
    $(document).on("click", ".introjs-prevbutton", function() {
        contadorNext= contadorNext-1;
    });

    $(document).on("click", ".bdSeleccionada", function() {
        var nombreBase = $(this).attr("id");
        swal({
            title: "Advertencia",
            text: "¿Seguro que quieres seleccionar la base de datos:
<b>"+nombreBase+"</b> ?",
            type: "warning",
            html:true,
            showCancelButton: true,
            cancelButtonText: "No",
            confirmButtonText: "Si",
            confirmButtonColor: "#00A59D",
            closeOnConfirm: true,
            closeOnCancel: false

```

```

    },
    function (isConfirm) {
        if (isConfirm) {

            $.ajax({
                type: 'POST',
                async: false,
                dataType: 'json',
                data: {nombreBase:nombreBase},
                url:Routing.generate('mostrarTablas') ,
                success: function (data)
                {
                    $("#bdExistentes").html("");
                    $.each(data, function( k, v ) {
                        var form="<div class='row' style='margin-top:20px;'><div class='col-md-12' style='text-align:center;font-size:16px;'>"+v+"</div>\n\
                            <div class='clearfix'></div></div>";
                        $("#bdExistentes").append(form);
                        $("#encabezado").text("Tablas encontradas en la Base");
                    });
                    var btn = "<div class='col-md-4'></div>\n\
                        <div class='col-md-4 btn btn-primary' id='convertirBd'
style='margin-top:20px;'>Convertir a BIG DATA</div>\n\
                        <div class='col-md-4'></div>";
                    $("#bdExistentes").append(btn);
                },
                error: function (xhr, status)
                {
                }
            });
        } else {
        }
    });

});
$(document).on("click","#convertirBd",function() {

    $.ajax({
        type: 'POST',
        async: false,
        dataType: 'json',

```



```

data: {},
url: Routing.generate('conversion'),
success: function (data)
{
    var agregado1 = "";
    var agregado2 = "";
    var agregado3 = "";
    var agregado4 = "";
    if (data.estado==true){
        agregado1 ='<div class="progress">\n\
            <div      class="progress-bar      progress-bar-success      progress-bar-striped"
role="progressbar"\n\
                aria-valuenow="100" aria-valuemin="0" aria-valuemax="100" style="width:100%"
id="">\n\
                Extrayendo datos relacionales...\n\
            </div>\n\
        </div>';
        agregado2='<div class="progress">\n\
            <div      class="progress-bar      progress-bar-success      progress-bar-striped"
role="progressbar"\n\
                aria-valuenow="100" aria-valuemin="0" aria-valuemax="100" style="width:100%"
id="">\n\
                Generando archivo intermedio...\n\
            </div>\n\
        </div>';
        agregado3='<div class="progress">\n\
            <div      class="progress-bar      progress-bar-success      progress-bar-striped"
role="progressbar"\n\
                aria-valuenow="100" aria-valuemin="0" aria-valuemax="100" style="width:100%"
id="">\n\
                Convirtiendo datos a BIG DATA...\n\
            </div>\n\
        </div>';
        agregado4='<div class="panel panel-primary">\n\
            <div class="panel-heading">¡Exito!</div>\n\
            <div class="panel-body">Datos migrados en un 100%, por favor consulte la nueva
base de datos en MongoDB.</div>\n\
        </div>';

    $("#encabezado").text("Estado final");
    $("#bdExistentes").html("");
}

```

```

var delay=3000; //3 second

setTimeout(function() {
    $("#bdExistentes").append(agregado1);
}, delay);
delay = delay+1000;
setTimeout(function() {
    $("#bdExistentes").append(agregado2);
}, delay);
delay = delay+1000;
setTimeout(function() {
    $("#bdExistentes").append(agregado3);
}, delay);
delay = delay+1000;
setTimeout(function() {
    $("#bdExistentes").append(agregado4);
}, delay);
delay = delay+500;
setTimeout(function() {
    var salirButton = "<center><button id='salirConversion' class='btn btn-primary'
style='margin-top:10px;'>Salir</button><center>"
    $("#bdExistentes").append(salirButton);
    $("#salirConversion").show();
}, delay);

}
},
error: function (xhr, status)
{

}

});
$(document).on("click", "#salirConversion",function() {
    $("#modal").modal("toggle");
    location.reload();
});
});

```

ARCHIVO: dbc.html.twig. Vista principal de la aplicación.

```
{% extends ':Layout:layout2.html.twig' %}

{% block css %}
    {{parent()}}
    <link href="{{ asset('Resources/intro/assets/css/bootstrap.min.css')}}" rel="stylesheet">
    <link href="{{ asset('Resources/intro/assets/css/demo.css')}}" rel="stylesheet">
    <!-- Add IntroJs styles -->
    <link href="{{ asset('Resources/intro/introjs.css')}}" rel="stylesheet">
    <link href="{{ asset('Resources/intro/assets/css/bootstrap-responsive.min.css')}}" rel="stylesheet">

    <link rel="stylesheet" href="{{ asset('Resources/sweetalert-master/dist/sweetalert.css')}}">
{% endblock css%}

{% block js %}
    {{parent()}}
    <script type="text/javascript" src="{{ asset('Resources/intro/intro.js')}}"></script>
    <script type="text/javascript" src="{{ asset('Resources/js/dbc/dbc.js')}}"></script>
    <script src="{{ asset('Resources/sweetalert-master/dist/sweetalert.min.js') }}"></script>

{% endblock %}

{% block descripcion_y_acciones %}

<div class="row">
    {% block descripcion %}
        <div class="col-md-8">
            <p class="robo" style="font-weight: 300; margin-bottom: 0px; font-size: 30px;text-align:center;margin-left:70%;color:07326B;"><b>Panel Principal</b></p>
        </div>
    {% endblock%}
    {% block acciones %}
        <div class="col-md-4" style="margin-top: 50px;">
            {{parent()}}
        </div>
    {% endblock %}
</div>

{% endblock %}

{% block navegacion_Y_body %}
    <div class="row" style='background-color:031937;'>
        {% block navegacion %}
```

```

<hr style="margin-top: 0px !important;">
  <div class="col-md-2 colorAnclas" style="border-right: 1px solid #f3f3f3;">
    <!-- Left Navigation -->
    <ul style="list-style: none; padding: 0;">
      <a href="{{ path('dashboard_index') }}" class="leftnava colorAnclas"><li class="leftnav left
active">Menu<span class="pull-right right"></span></li></a>

      </ul>
    </div>
  {% endblock %}

```

```
{% block cuerpo %}
```

```

<div class="col-md-10">
  <div class="row">
    <div class="container-narrow">
      <div class="masthead">
        </div>

        <hr>
        <div class="jumbotron">
          <h4 id="step1" style='font-size: 30px !important ;'>BigData Converter</h4>
          <a class="btn btn-primary" href="javascript:void(0);" onclick="startIntro();">Iniciar</a>
        </div>
        <hr>

      </div>

```

```

<script type="text/javascript">
function startIntro(){
  var intro = introJs();
  intro.setOptions({
    steps: [
      {
        intro: "<p style='text-align:justify;'><center>Bienvenido a Big Data Converter</center></p>"
      },
      {
        intro: "<center><p style='text-align:justify;'><b>Selecciona el gestor de base de datos a
convertir</b></p></center>\n\
        <select class='form-control' name='dbMotor' id='dbMotor'>\n\
          <option value='1'>SQL SERVER</option>\n\
          <option value='2'>Mysql</option>\n\
        </select>"
      },
    ],

```

```

    {
      intro: '<div class="container">\n\
        <div class="row">\n\
          <div class="form-column col-sm-3">\n\
            <div class="form-group" >\n\
              <label for="host" class="control-label">SERVIDOR</label>\n\
              <input type="text" class="form-control" id="host" placeholder="Server" name="host"
>\n\
            </div>\n\
          </div>\n\
        <div class="form-column col-sm-3">\n\
          <div class="form-group" >\n\
            <label for="host" class="control-label">PUERTO</label>\n\
            <input type="text" class="form-control" id="port" placeholder="Port" name="port"
>\n\
          </div>\n\
        </div>\n\
      <div class="col-sm-6"></div>\n\
    <div class="clearfix"></div>\n\
    <div class="form-column col-sm-3">\n\
      <div class="form-group" >\n\
        <label for="user" class="control-label">USUARIO</label>\n\
        <input type="text" class="form-control" id="user" placeholder="User" name="user"
>\n\
      </div>\n\
    </div>\n\
    <div class="form-column col-sm-3"><div class="form-group" >\n\
      <label for="pass" class="control-label">CONTRASEÑA</label>\n\
      <input type="password" class="form-control" id="pass" placeholder="Password"
name="pass" >\n\
    </div>\n\
    </div>\n\
    <div class="col-sm-6"></div>\n\
  </div></div>'
    },
    {
      element: '#step5',
      intro: 'Bases de datos encontradas'
    }
  ]
});

```

```
        intro.start();
    }
</script>
</div>
</div>
```

{#Modal donde se listan todas las bases de datos encontradas#}

```
    <div class="modal-content modal fade modal" id="modal" tabindex="-1" role="dialog" aria-
labelledby="myModalLabel" aria-hidden="true">
        <div class="modal-header " Style="height:45px;">
            <button type="button" class="close" data-dismiss="modal"><span aria-
hidden="true">&times;</span><span class="sr-only">Cerrar</span></button>
            <p id="encabezado" style="text-align: center;font-size: 20px;">Bases de datos
encontradas en el servidor SQL</p>

        </div>
        <div class="modal-body" id="bdExistentes" style='background-color:DFD7C5;'>
        </div>
    </div>
```

```
{% endblock %}
```

```
</div>
```

```
{% endblock %}
```

```
{% block js_extend %}
```

```
{% endblock %}
```

VISIÓN

Ser una institución educativa líder en educación tecnológica a nivel nacional y regional, comprometida con la calidad, la empresarialidad y la pertinencia de nuestra oferta educativa.

MISIÓN

Formar profesionales integrales y competentes en áreas tecnológicas que tengan demanda y oportunidad en el mercado local, regional y mundial, tanto como trabajadores y como empresarios.

VALORES

EXCELENCIA: *Nuestro diario quehacer está fundamentado en hacer bien las cosas desde la primera vez.*

INTEGRIDAD: *Actuamos congruentemente con los principios de la verdad en todas las acciones que realizamos.*

ESPIRITUALIDAD: *Desarrollamos todas nuestras actividades en la filosofía de servicio, alegría, compromiso, confianza y respeto mutuo.*

COOPERACIÓN: *Actuamos basados en el buen trabajo en equipo, la buena disposición a ayudar a todas las personas.*

COMUNICACIÓN: *Respetamos las diferentes ideologías y opiniones, manteniendo y propiciando un acercamiento con todo el personal.*

La Escuela Especializada en Ingeniería ITCA - FEPADE, fundada en 1969, es una institución estatal con administración privada, conformada actualmente por 5 campus: Sede Central Santa Tecla y cuatro centros regionales ubicados en Santa Ana, San Miguel, Zacatecoluca y La Unión.



**SEDE CENTRAL
SANTA TECLA**

Km. 11.5 Carretera a Santa Tecla, La Libertad.
Tel. (503) 2132-7400
Fax. (503) 2132-7599



**CENTRO REGIONAL
SANTA ANA**

Final 10a. Av. Sur,
Finca Procavia
Tels. (503) 2440-4348
y (503) 2440-2007
Tel./Fax. (503) 2440-3183



**CENTRO REGIONAL
MEGATEC ZACATECOLUCA**

Km. 64 1/2, desvío Hacienda El Nilo, sobre autopista a Zacatecoluca y Usulután.
Tels. (503) 2334-0763
y (503) 2334-0768



**CENTRO REGIONAL
SAN MIGUEL**

Km. 140, Carretera a Santa Rosa de Lima.
Tels. (503) 2669-2292
y (503) 2669-2298
Fax. (503) 2669-0061



**CENTRO REGIONAL
MEGATEC LA UNIÓN**

Calle Santa María, Col. Belén, atrás del Instituto Nacional de La Unión.
Tel. (503) 2668-4700