

PROYECTO (DRUPAL)

FUNDACIÓN UNIVERSITARIA UNIPANAMERICANA

JESICA TATIANA TIBAQUIRA NUÑEZ

JHON DIEGO LOAIZA ARABA

NATHALY URREGO CASAS

INGENIERIA DE SISTEMAS

BOGOTA D.C

2018

Tabla de Contenido

1. Introducción.....	6
2. Que Es Sistema de Contenido Drupal	7
2.1 Historia	8
2.2 Información Logotipo	8
2.3 Gestión de páginas	9
2.4 Tipo de Contenido	10
3. Requerimientos del Funcionamiento	11
3.1 Requerimientos del Servidor	11
3.2 Servidor Web	11
3.3 Base de Datos	11
3.4 PHP y JSON	11
4. Requerimientos de Ejecución	12
4.1 Consola Drupal	12
4.2 ¿Qué es Symfony?	12
4.3 Herramientas de la Depuración del Navegador	12
5. ETL (MySQL - MariaDB)	13
5.1 MySQL	13
5.2 MariaDB	14
6. Proceso de Instalación para el ETL	15
6.1 Instalar XAMPP	15
6.2 Instalar Drupal	18
6.3 Instalar MariaDB	21
7. Motor Base de Datos	25
8. Diccionario de Datos	36
9. Implementación ETL	39
9.1 Estrategia Bing Bang	39
9.1.1 Ventajas	39
9.1.2 Desventajas	40
10. Planificación de Costos	41
10.1 Planificación de actividades	41

11. Proceso	42
12. Conclusión	49

Tabla de Ilustraciones

Figura 1. Logotipo Drupal	8
Figura 2. MySQL	14
Figura 3 MariaDB	14
Figura 4. Descargador Xamp	15
Figura 5. Xampp Paso 1	15
Figura 6. Xampp Paso 2	16
Figura 7. Xampp Paso 3	16
Figura 8. Xampp Paso 4	17
Figura 9. Xampp Paso 5	17
Figura 10. Xampp Paso 6	17
Figura 11. Finalización Instalación	18
Figura 12. Descargador Drupal	18
Figura 13. Drupal Paso 1	19
Figura 14. Drupal Paso 2	19
Figura 15. Drupal Paso 3	20
Figura 16. Drupal Paso 4	20
Figura 17. Drupal Paso 5	21
Figura 18. Finalización Instalación	21
Figura 19. Instalación MariaDB	22
Figura 20. Paso 1	22
Figura 21. Paso 2	22
Figura 22. Paso 3	23
Figura 23. Paso 4	23
Figura 24. Finalización Instalación	23

Figura 25. Diagrama de Gantt	41
Figura 26. Configuración	43
Figura 27. Acceso de Base de Datos	43
Figura 28. Creación	43
Figura 29. Importación	44
Figura 30. Buscar Archivo	44
Figura 31. Carga	45
Figura 32. Resultado	45
Figura 33. Tabla Batch	46
Figura 34. Tabla Cache_Data	46
Figura 35. Tabla bloc_content_body	47
Figura 36. Tabla history	47
Figura 37. Tabla User_roles	48
Figura 38. Inicio Drupal	48

1. Introducción

El siguiente documento tiene como finalidad dar a conocer un proceso de ETL (Extract, Transform and Load - Extracción, transformación y carga) de MySQL a MariaDB el cual se lleva a cabo con un sistema de gestión de contenido concreto llamado Drupal donde se describe la implementación de una estrategia que permite realizar este proceso de una manera pertinente dentro de las necesidades de los clientes que lo manejan, de igual forma se definen los puntos claves para el desarrollo del mismo como por ejemplo la funcionalidad de Drupal, la estrategia utilizada, los riesgos que se deben tener en cuenta, la base de datos a utilizar, etc. (Propia, 2018).

2. Que es Sistema de Contenido Drupal

Drupal es un sistema de gestión de contenido (CMS “Content Management System”) el cual le permite a los usuarios crear un espacio único de soluciones innovadoras donde día tras día se puede agregar, editar o eliminar contenido de aquel sitio WEB en cualquier dispositivo como lo sería un teléfono inteligente, tableta o PC, normalmente CMS está escrito en lenguaje scripting, los cuales se ejecutan en una computadora donde está instalado el servidor web y su base de datos, la configuración y el contenido suelen almacenarse en la BD donde cada solicitud realizada llega al servidor, los scripts combinan información y los activos (archivos, Java Scripts, CSS, imágenes, etc.) forman parte del CMS para construir las páginas del sitio WEB. (Drupal, 2018)

Drupal es un CMS flexible basado en la pila LAMP que es una combinación del sistema operativo Linux, un servidor WEB apache, base de datos MySQL y lenguaje de scripting PHP, pero esto no quiere decir que siempre se tenga que ejecutar en la pila LAMP también se puede desde otras pilas de la tecnología como lo son:

- Sistema operativo Windows o Mac OS.
- Servidor Web Nginx o IIS.
- Base de datos PostgreSQL o SQLite o un remplazo compatible con MySQL como MariaDB o Percona.

Sin embargo, la secuencia de comandos que utiliza el software está en PHP, por lo que no puede modificarse. (Drupal, 2018)

2.1 Historia

En el año 2000, las conexiones permanentes de Internet eran un premio para los estudiantes de la Universidad de Antwerp. Dries Buytaert y Hans Snijder configuraron un puente inalámbrico entre los dormitorios de estudiantes para compartir la conexión ADSL (acrónimo en inglés de Asymmetric Digital Subscriber Line) es un tipo de tecnología de línea de abonado digital DSL por módem de Hans entre ocho estudiantes. Mientras esto era una situación extremadamente lujuriosa en aquel tiempo, algo faltaba: no existía un medio para discutir o compartir cosas simples. (EcuRed, 2018)

Esto inspiró a Dries a trabajar en un pequeño sitio de noticias, permitiendo al grupo de amigos dejarse notas acerca del estado de la red, anuncios sobre donde iban a cenar o para compartir algunas cosas. (EcuRed, 2018)

Una vez establecido drop.org en la web, su audiencia cambia y los miembros comenzaron a hablar sobre nuevas tecnologías web, como moderación, sindicación y autenticación distribuida. En enero del 2001, Dries decidió liberar el software tras drop.org como Drupal el cual es derivado de la pronunciación en inglés de la palabra holandesa “druppel”, que significa “gota”. Su propósito era permitir a otros usar y extender la plataforma de experimentación para que más personas pudieran explorar nuevos caminos para el desarrollo. (EcuRed, 2018)

2.2 Información Logotipo



Figura 1. Logotipo Drupal, Recuperado de <https://www.drupal.org/about/media-kit/logos>

Inicialmente se incluyó un círculo dentro de la gota, para significar el círculo de una comunidad, más tarde Steven Wittens pensó en unir dos gotas enfrentadas dibujando el signo del infinito y a la vez aportar un significado que en sí mismo aporta un “por siempre”.

(Drupal, s.f).

2.3 Gestión de Páginas

Drupal maneja unos conceptos básicos para localizar los distintos sitios web o aplicaciones creadas, por ejemplo el Localizador uniforme de recursos “URL” que se define como una dirección que se va determinando para que el navegador identifique la página que se quiere mostrar pero para que esto sea más específico es decir muestre un contenido determinado se puede utilizar una ruta que será el acompañamiento de esa URL principal que ira al final de esta como por ejemplo “[http://miPaginaWeb.com /diseños/mundial](http://miPaginaWeb.com/diseños/mundial)” donde permite una búsqueda más eficiente. (Drupal, s.f)

Menús

Son una colección de enlaces que ayudan a navegar en el sitio web proporcionado una interfaz para controlar y personalizar dicho menú, Drupal (s.f) define cinco diferentes tipos de menús que se pueden utilizar para la eficacia y eficiencia del sitio.

Navegación Principal: Enlaces que se crean para direccionar a los visitantes a un contenido explícito.

Administración: Enlaces para los creadores del sitio donde manejan las tareas que deben ir cumpliendo a medida que pase el tiempo.

Menú de Cuenta de Usuario: Enlaces asociados con la cuenta de cada usuario.

Pie de Página: Enlaces que direccionan a páginas relacionadas.

Herramientas: Enlaces de las tareas necesarias de cada visitante.

Cada persona administradora tiene el poder de manejar y manipular el diseño, lugar y demás cosas que se crean convenientes y es el único rol que puede dar permisos para que los visitantes y demás puedan ver lo necesario. (Drupal, s.f)

2.4 Tipos de Contenido

Drupal (s.f) introduce el concepto de nodo como sinónimo de tipos de contenido, cualquier recurso que se ingrese al sistema pasa a ser un nodo que puede ser variable e incluir artículos, historias, comentarios, encuestas, imágenes, libros colaborativos, reseñas, recetas, etcétera. Este nuevo concepto nos permite estandarizar la información asignándoles las mismas características a distintos tipos de objetos y la posibilidad de tener toda la información centralizada y a la vez catalogada.

3. Requerimientos del Funcionamiento

3.1 Requisitos del servidor:

Se requiere cantidad total de espacio en el disco duro, ya que los archivos bases para el software ocupan aproximadamente unos 100 MB en el servidor web. La base de datos también usa espacio en el disco, aunque normalmente no está en la misma área (y en algunos casos, ni siquiera en el mismo servidor) que la utilizada por los archivos del sitio. (Drupal, 2018)

3.2 Servidor web:

El software central funcionará en Apache 2.x alojado en UNIX / Linux, OS X o Windows que tengan el módulo mod_rewrite de Apache instalado y habilitado. (Drupal, 2018)

3.3 Base de datos:

La base de datos que maneja el sistema de gestión de contenido Drupal en la actualidad es MySQL, aunque cabe resaltar que no es en la única que se pueda utilizar ya que también pueden ser:

- MariaDB 5.5.20
- Percona 5.5.8
- PostgreSQL - 9.1.2 o superior
- SQLite - 3.4.2 o superior

3.4 PHP Y JSON:

PHP 5.5.9 o superior compilado con JSON, el soporte de JSON normalmente se compila como parte del núcleo PHP.

4. Requerimientos de Ejecución

4.1 Consola Drupal

Drupal Console es una herramienta de línea de comandos que genera código repetitivo e interactúa con un sitio de Drupal. Puede generar, bloquear, formar código, instalar módulos o temas y crear contenido ficticio también hace uso de la consola de Symfony. (Drupal.2018)

4.2 ¿Qué es Symfony?

Symfony es un conjunto de componentes PHP reutilizables y un marco PHP para construir aplicaciones web, API, microservicios y servicios web. (Drupal.2018)

4.3 Herramientas de la depuración del navegador

Los navegadores web como Firefox y Chrome incluyen herramientas que permiten ver, editar, depurar y monitorear CSS, HTML y JavaScript. Puede abrir el panel o la ventana de depuración haciendo clic con el botón derecho del mouse en un área de su ventana y seleccionando "Inspeccionar" o "Inspeccionar elemento". (Drupal.2018)

5. ETL (MySQL – MariaDB)

La migración del sistema de Gestión de contenido (DRUPAL) será de MySQL a MariaDB por lo tanto a continuación se describen estas herramientas:



Figura 2. MySQL, Recuperado de <https://www.definicionabc.com/tecnologia/mysql.php>

5.1 MySQL: es un sistema de gestión de base de datos relacional o SGBD. Este gestor de base de datos en multihilo y multiusuario, lo que le permite ser utilizado por varias personas al mismo tiempo, e incluso, realizar varias consultas a la vez, lo que lo hace sumamente versátil. (culturación, s.f)

Nació como una iniciativa de **Software Libre**. La mayor parte del código se encuentra escrito en **lenguaje C/C++** y la sintaxis de su uso es bastante simple, lo que permite crear bases de datos simples o complejas con mucha facilidad. Además, es compatible con múltiples plataformas informáticas y ofrece una infinidad de aplicaciones que permiten acceder rápidamente a las sentencias del gestor de base de datos. (culturación, s.f)



Figura 3. MariaDB, Recuperado de <https://fututel.com/es/tutoriales-guias-manuales-videotutoriales/2623-instalar-mariadb-en-centos-7>

5.2 MariaDB: Es un sistema gestor de bases de datos (**SGBD**), es decir, un conjunto de programas que permiten modificar, almacenar, y extraer información de una base de datos. Disponiendo de otro tipo de funcionalidades como la administración de usuarios, y recuperación de la información si el sistema se corrompe, entre otras. (nerion, s.f)

MariaDB surge a raíz de la compra, de la compañía desarrolladora de otro (SGBD) llamado MySQL, por la empresa Sun Microsystems. El desarrollador original, decide tomar el código fuente original de MySQL y genera un derivado con mejoras y cambios a los que llama MariaDB. Permitiendo así la existencia de una versión de este producto con licencia GPL (General Public License). (nerion, s.f)

6. Proceso de Instalación para el ETL

6.1 Instalar XAMPP

Descarga XAMPP versión 7.2.9 para Windows de Apache Friends y ejecuta el instalador.

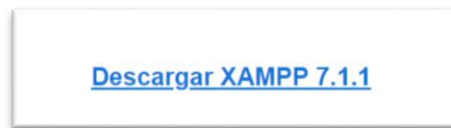


Figura 4. Descargador Xampp Recuperado de <https://www.filehorse.com/es/descargar-xampp/29148/>

Los componentes mínimos que instala Xampp y requiere Drupal son: el servidor de apache, lenguaje php, MySQL y phpMyAdmin.

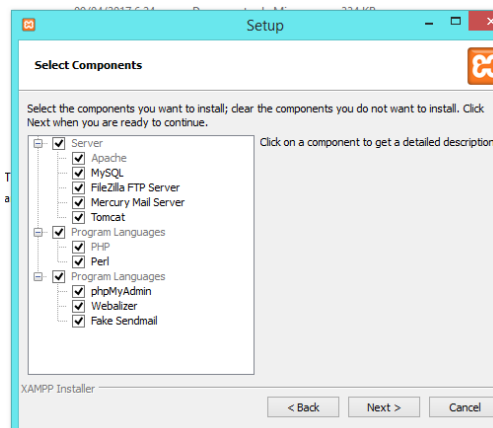


Figura 5. Xampp Paso 1 Recuperado de Propia

En la siguiente pantalla se elige la carpeta de instalación de XAMPP, por lo general la carpeta ya es predeterminada es **C:\xampp**.

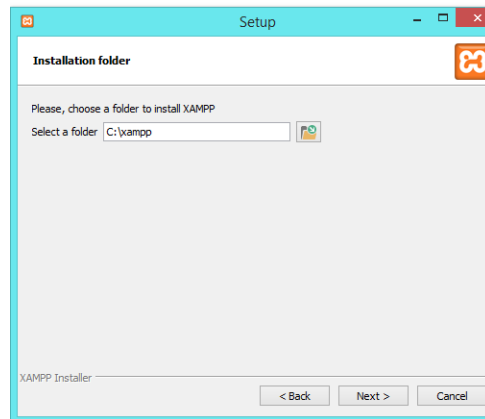


Figura 6. Xampp Paso 2 Recuperado de Propia

Hacer click en el botón “Next”



Figura 7. Xampp Paso 3 Recuperado de Propia

Cuando se termina la copia de archivos, la pantalla muestra la finalización de la instalación de XAMPP.

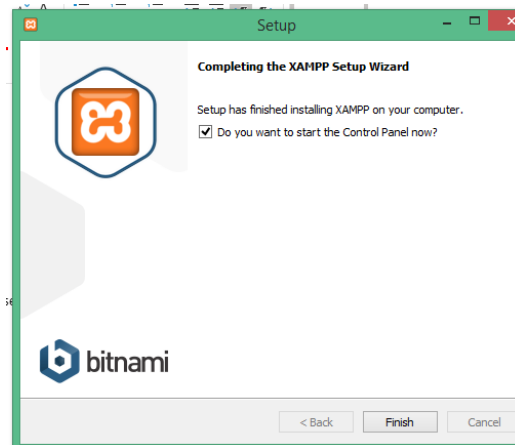


Figura 8. Xampp Paso 4 Recuperado de Propia

Al dar click en el botón “Finish” nos dirige al panel del control de XAMPP.

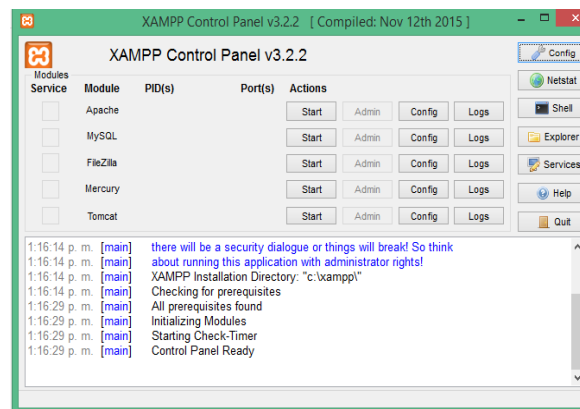


Figura 9. Xampp Paso 5 Recuperado de Propia

Para poner en funcionamiento Apache y MySQL, hay que dar click en el botón “Start”.

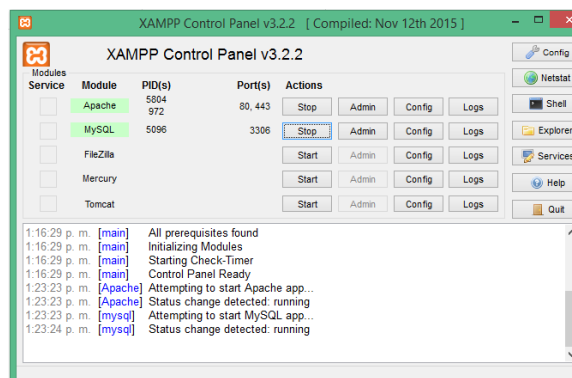


Figura 10. Xampp Paso 6 Recuperado de Propia

Para comprobar el funcionamiento correcto, hay que escribir en el navegador la dirección **http://localhost** dirigiéndonos a la página del panel de administración web.

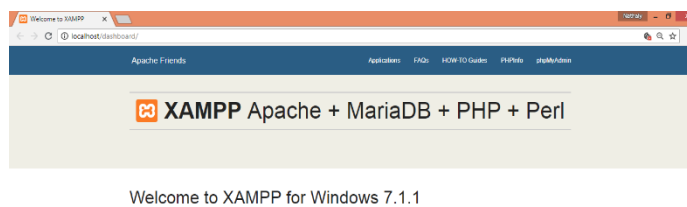


Figura 11. Finalización Instalación Recuperado de Propia

6.2 Instalar Drupal

Se descarga Drupal versión 8.5.6 para Windows y se da click en el botón “Download zip”

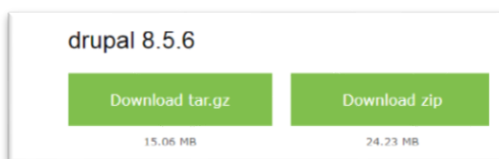


Figura 12. Descargador Drupal Recuperado de

<https://www.drupal.org/project/drupal/releases/8.5.6>

Se descomprime la carpeta Drupal 8.5.6 y se ubica en la siguiente dirección dentro del explorador de archivos de su computador `C:\xampp\htdocs\drupal`.

Una vez copiado la carpeta, Se ingresa al navegador y se digita la siguiente url `http://localhost/drupal/` y directamente aparecerá en la pantalla de instalación de Drupal 8.



Figura 13. Drupal Paso 1 Recuperado de Propia

Configuración de lenguaje

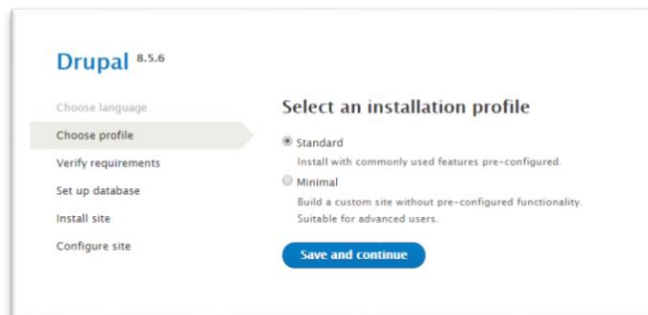


Figura 14. Drupal Paso 2 Recuperado de Propia

Drupal cuenta de 2 perfiles de instalación: mínima y standard. La diferencia entre ambos es que el perfil de instalación standard nos instalará Drupal con una serie de configuraciones predeterminadas como tipos de contenido, menús, módulos, etc. mientras que el perfil de instalación mínima se instala con lo mínimo imprescindible para que Drupal funcione. (Propia, 2018)

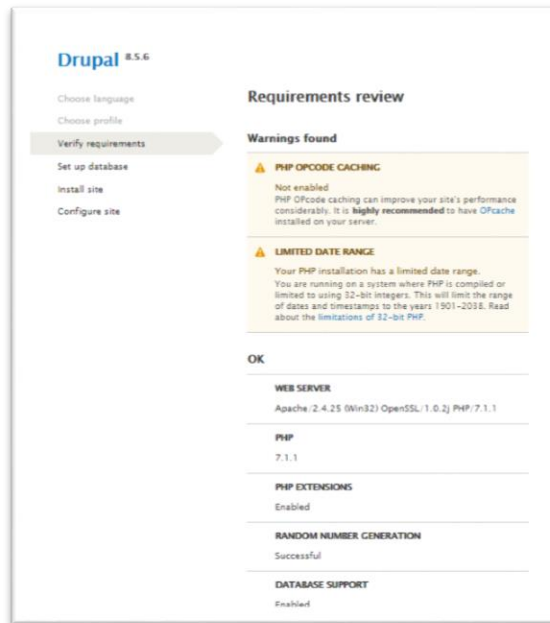


Figura 15. Drupal Paso 3 Recuperado de Propia

Verificación de requisitos



Figura 16. Drupal Paso 4 Recuperado de Propia

Para la configuración de la base de datos se debe ingresar el nombre de la BD con sus privilegios correspondientes.

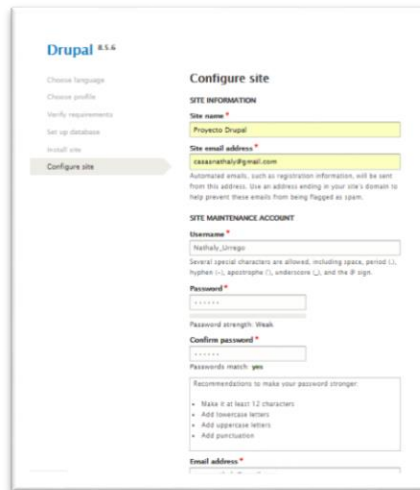


Figura 17. Drupal Paso 5 Recuperado de Propia

Configuración de sitio

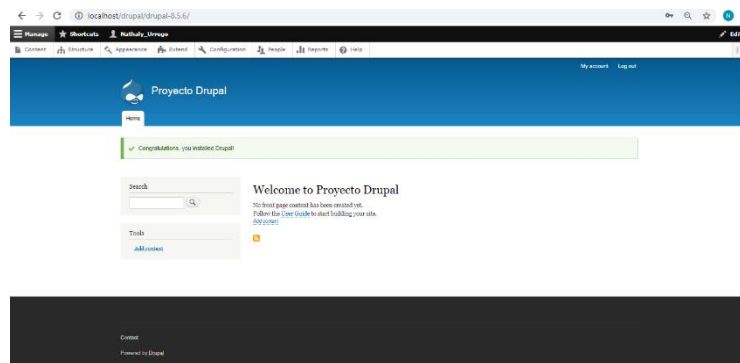


Figura 18. Finalización Instalación Recuperado de Propia

Instalación Finalizada, se recomienda instalar XAMPP"C: \ Archivos de programa \ xampp \ " o similar) (2). El resto de esta guía asume que usaste "C: \ xampp \".

Pruebe la instalación de XAMPP abriendo un navegador web e ingresando `http://localhost` / en el campo de dirección. La página de bienvenida de XAMPP debería aparecer.

6.3 Instalar MariaDB

Se descarga la versión de mariaDB 10.3.10

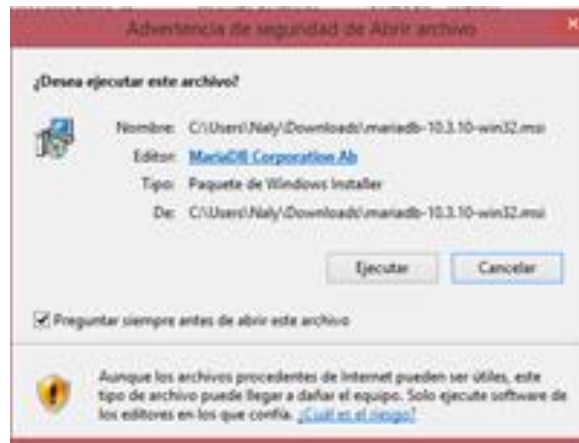


Figura 19. Instalación MariaDB Recuperado de Propia

Damos clic al botón Next.



Figura 20. Paso 1 Recuperado de Propia

Aceptamos en reglamento y damos clic en Next.



Figura 21. Paso 2 Recuperado de Propia

Se ingresa contraseña ya que MariaDB la requiere y luego damos clic en Next.

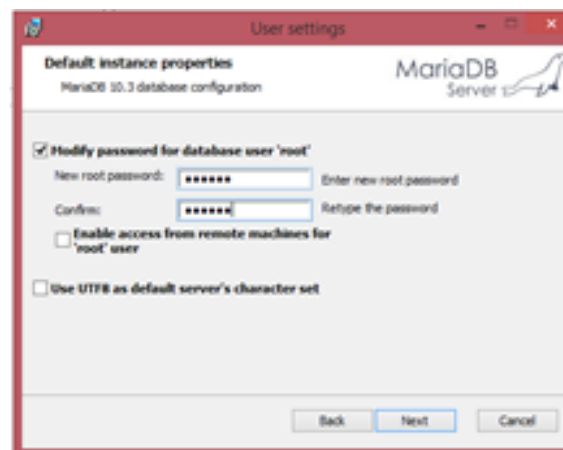


Figura 22. Paso 3 Recuperado de Propia

La pantalla nos muestra, el nombre del servicio, el puerto y el tamaño de agrupación de almacenamiento intermedio, que por lo general ya vienen por defecto.

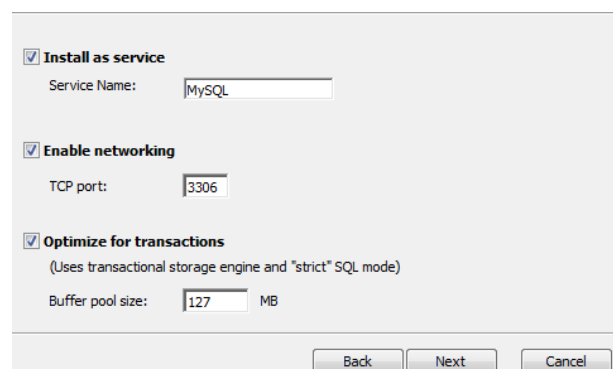


Figura 23. Paso 4 Recuperado de Propia

Se finaliza instalación.



Figura 24. Finalización Instalación Recuperado de Propia

7. Motor Base De Datos

Drupal maneja MySQL que es un Sistema de Gestión de Base de Datos Relacionales (RDBMS) de código abierto, basado en lenguaje de consulta estructurado (SQL) donde toman la forma de un lenguaje de comandos que permiten seleccionar, insertar, actualizar, averiguar la ubicación de los datos, entre muchas cosas más. Nace como una iniciativa de software libre creado por Michael “Mondy” Widenius en el año 1995. Se reconoce por su multihilo y multiusuario, lo que le permite ser utilizado por varias personas al mismo tiempo, e incluso, realizar varias consultas a la vez, por tal motivo es sumamente versátil; es compatible con múltiples plataformas informáticas y ofrece una infinidad de aplicaciones que permiten acceder rápidamente a las sentencias del gestor de base de datos, La mayor parte de su código se encuentra escrito en **lenguaje C/C++** y tiene una facilidad de uso en su sintaxis. (Aculturación, s.f).

8. Diccionario de Datos

Batch

Almacena detalles sobre lotes (procesos que se ejecutan en...

Columna	Tipo	Comentarios
bid (<i>Primaria</i>)	int(10)	Primary Key: Unique batch ID.
token	varchar(64)	Un token de cadena generado contra el ID de sesión del usuario actual y el ID de lote, utilizado para garantizar que solo el usuario que envió el lote pueda acceder a él de manera efectiva.
timestamp	int(11)	Una marca de tiempo Unix que indica cuándo se envió este lote para su procesamiento. Los lotes antiguos se purgan en el momento cron.
batch	longblob	Una matriz serializada que contiene los datos de procesamiento para el lote.

block_content

Comentarios de la tabla: La tabla base para las entidades de bloque contenido:

Columna	Tipo	Comentarios
id (<i>Primaria</i>)	int(10)	
revision_id	int(10)	
type	varchar(32)	El ID de la entidad objetivo
uuid	varchar(128)	
langcode	varchar(12)	

block_content_field_data

La tabla de datos para las entidades de contenido de bloque:

Columna	Tipo	Comentarios
id (<i>Primaria</i>)	int(10)	
revision_id	int(10)	
type	varchar(32)	El ID de la entidad objetivo.
langcode (<i>Primaria</i>)	varchar(12)	
status	tinyint(4)	
info	varchar(255)	
changed	int(11)	
default_langcode	tinyint(4)	

block_content_field_revision

Los datos de revisión

Columna	Tipo	Nulo	Comentarios
id	int(10)	No	
revision_id (Primaria)	int(10)	No	
langcode (Primaria)	varchar(12)	No	
status	tinyint(4)	No	
info	varchar(255)	Sí	
changed	int(11)	Sí	
default_langcode	tinyint(4)	No	
revision_translation_affected	tinyint(4)	Sí	

block_content_revision**Bloquear_contenido_revisión**

La tabla de revisión para las entidades block_content.

Columna	Tipo	Comentarios
id	int(10)	
revision_id (Primaria)	int(10)	
langcode)	varchar(12)	
revision_user	int(10)	El ID de la entidad objetivo.
revision_created	int(11)	
revision_log	longtext	
revision_default	tinyint(4)	

block_content_revision_body**Bloquear_contenido_revisión_cuerpo**

Almacenamiento de archivos de revisión para el cuerpo del campo block_content.

Columna	Tipo	Comentarios
bundle	varchar(128)	El paquete de instancia de campo al que pertenece esta fila, utilizado al eliminar una instancia de campo.
deleted (Primaria)	tinyint(4)	Un valor booleano que indica si este elemento de datos se ha eliminado.
entity_id (Primaria)	int(10)	La identificación de la entidad a la que se adjunta esta información
revision_id (Primaria)	int(10)	La identificación de la revisión de la entidad a la que se adjunta este.
langcode (Primaria)	varchar(32)	El código de idioma para este elemento de datos.
delta (Primaria)	int(10)	El número de secuencia para este elemento de datos, utilizado para campos de valor múltiple

body_value	longtext	
body_summary	longtext	
body_format	varchar(255)	

block_content__body**Bloquear_contenido_cuerpo**

Almacenamiento de datos para el cuerpo del campo block_content.

Columna	Tipo	Comentarios
bundle	varchar(128)	El paquete de instancia de campo al que pertenece esta fila, utilizado al eliminar una instancia de campo.
deleted (Primaria)	tinyint(4)	Un valor booleano que indica si este elemento de datos se ha eliminado.
entity_id (Primaria)	int(10)	La identificación de la entidad a la que se adjunta estos datos.
revision_id	int(10)	La ID de revisión de la entidad a la que se adjuntan estos datos.
langcode (Primaria)	varchar(32)	El código de idioma para este elemento de datos.
delta (Primaria)	int(10)	El número de secuencia para este elemento de datos, utilizado para campos de valor múltiple.
body_value	longtext	
body_summary	longtext	
body_format	varchar(255)	

Cachetags**Etiquetas de identificación**

Tabla de caché para el seguimiento de invalidaciones de etiquetas de caché.

Columna	Tipo	Comentarios
tag (Primaria)	varchar(255)	Cadena de etiquetas con prefijo de espacio de nombres.
invalidations	int(11)	Número incrementado cuando la etiqueta se invalida.

cache_bootstrap

Almacenamiento para la API de caché.

Columna	Tipo	Comentarios
cid (Primaria)	varchar(255)	Clave principal: ID de caché único.
data	longblob	Una colección de datos a caché.
expire	int(11)	Una marca de tiempo de Unix que indica cuándo debe caducar la entrada de caché, o -1 para nunca.
created	decimal (14,3)	Una marca de tiempo con una precisión de milisegundos que indica cuando la entrada de caché fue creado.
serialized	smallint(6)	Una bandera para indicar si el contenido está serializado (1) o no (0).
tags	longtext	Lista separada por espacios de etiquetas de caché para esta entrada.
checksum	varchar(255)	La suma de comprobación de invalidación de la etiqueta cuando se guardó esta entrada.

cache_config**cache_configuración**

Almacenamiento para la API de caché.

Columna	Tipo	Comentarios
cid (Primaria)	varchar(255)	Clave principal: ID de caché único.
data	longblob	Una colección de datos a caché.
expire	int(11)	Una marca de tiempo de Unix que indica cuándo debe caducar la entrada de caché, o -1 para nunca.
created	decimal (14,3)	Una marca de tiempo con una precisión de milisegundos que indica cuando la entrada de caché fue creado.
serialized	smallint(6)	Una bandera para indicar si el contenido está serializado (1) o no (0).
tags	longtext	Lista separada por espacios de etiquetas de caché para esta entrada.
checksum	varchar(255)	La suma de comprobación de invalidación de la etiqueta cuando se guardó esta entrada.

cache_container

Almacenamiento para la API de caché.

Columna	Tipo	Comentarios
cid (Primaria)	varchar(255)	Clave principal: ID de caché único.
data	longblob	Una colección de datos a caché.
expire	int(11)	Una marca de tiempo de Unix que indica cuándo debe caducar la entrada de caché, o -1 para nunca.
created	decimal (14,3)	Una marca de tiempo con una precisión de milisegundos que indica cuando la entrada de caché fue creado.
serialized	smallint(6)	Una bandera para indicar si el contenido está serializado (1) o no (0).
tags	longtext	Lista separada por espacios de etiquetas de caché para esta entrada.
checksum	varchar(255)	La suma de comprobación de invalidación de la etiqueta cuando se guardó esta entrada.

cache_data**cache_datos**

Almacenamiento para la API de caché.

Columna	Tipo	Comentarios
cid (Primaria)	varchar(255)	Clave principal: ID de caché único.
data	longblob	Una colección de datos a caché.
expire	int(11)	Una marca de tiempo de Unix que indica cuándo debe caducar la entrada de caché, o -1 para nunca.
created	decimal (14,3)	Una marca de tiempo con una precisión de milisegundos que indica cuando la entrada de caché fue creado.
serialized	smallint(6)	Una bandera para indicar si el contenido está serializado (1) o no (0).
tags	longtext	Lista separada por espacios de etiquetas de caché para esta entrada.

checksum	varchar(255)	La suma de comprobación de invalidación de la etiqueta cuando se guardó esta entrada.
----------	--------------	---

cache_default**cache por defecto**

Almacenamiento para la API de caché.

Columna	Tipo	Comentarios
cid (Primaria)	varchar(255)	Clave principal: ID de caché único.
data	longblob	Una colección de datos a caché.
expire	int(11)	Una marca de tiempo de Unix que indica cuándo debe caducar la entrada de caché, o -1 para nunca.
created	decimal (14,3)	Una marca de tiempo con una precisión de milisegundos que indica cuando la entrada de caché fue creado.
serialized	smallint(6)	Una bandera para indicar si el contenido está serializado (1) o no (0).
tags	longtext	Lista separada por espacios de etiquetas de caché para esta entrada.
checksum	varchar(255)	La suma de comprobación de invalidación de la etiqueta cuando se guardó esta entrada.

cache_discovery**cache de descubrimiento**

Almacenamiento para la API de caché.

Columna	Tipo	Comentarios
cid (Primaria)	varchar(255)	Clave principal: ID de caché único.
data	longblob	Una colección de datos a caché.
expire	int(11)	Una marca de tiempo de Unix que indica cuándo debe caducar la entrada de caché, o -1 para nunca.
created	decimal (14,3)	Una marca de tiempo con una precisión de milisegundos que indica cuando la entrada de caché fue creado.
serialized	smallint(6)	Una bandera para indicar si el contenido está serializado (1) o no (0).
tags	longtext	Lista separada por espacios de etiquetas de caché para esta entrada.
checksum	varchar(255)	La suma de comprobación de invalidación de la etiqueta cuando se guardó esta entrada.

cache_dynamic_page_cache

Almacenamiento para la API de caché

Columna	Tipo	Comentarios
cid (Primaria)	varchar(255)	Clave principal: ID de caché único.
data	longblob	Una colección de datos a caché.
expire	int(11)	Una marca de tiempo de Unix que indica cuándo debe caducar la entrada de caché, o -1 para nunca.
created	decimal (14,3)	Una marca de tiempo con una precisión de milisegundos que indica cuando la entrada de caché fue creado.
serialized	smallint(6)	Una bandera para indicar si el contenido está serializado (1) o no (0).

tags	longtext	Lista separada por espacios de etiquetas de caché para esta entrada.
checksum	varchar(255)	La suma de comprobación de invalidación de la etiqueta cuando se guardó esta entrada.

cache_entity**Entidad cache**

Almacenamiento para la API de caché

Columna	Tipo	Comentarios
cid (Primaria)	varchar(255)	Clave principal: ID de caché único.
data	longblob	Una colección de datos a caché.
expire	int(11)	Una marca de tiempo de Unix que indica cuándo debe caducar la entrada de caché, o -1 para nunca.
created	decimal(14,3)	Una marca de tiempo con una precisión de milisegundos que indica cuando la entrada de caché fue creado.
serialized	smallint(6)	Una bandera para indicar si el contenido está serializado (1) o no (0).
tags	longtext	Lista separada por espacios de etiquetas de caché para esta entrada.
checksum	varchar(255)	La suma de comprobación de invalidación de la etiqueta cuando se guardó esta entrada.

cache_menu**Menú de cache**

Almacenamiento para la API de caché

Columna	Tipo	Comentarios
cid (Primaria)	varchar(255)	Clave principal: ID de caché único.
data	longblob	Una colección de datos a caché.
expire	int(11)	Una marca de tiempo de Unix que indica cuándo debe caducar la entrada de caché, o -1 para nunca.
created	decimal(14,3)	Una marca de tiempo con una precisión de milisegundos que indica cuando la entrada de caché fue creado.
serialized	smallint(6)	Una bandera para indicar si el contenido está serializado (1) o no (0).
tags	longtext	Lista separada por espacios de etiquetas de caché para esta entrada.
checksum	varchar(255)	La suma de comprobación de invalidación de la etiqueta cuando se guardó esta entrada.

cache_page**página de caché**

Almacenamiento para la API de caché

Columna	Tipo	Comentarios
cid (Primaria)	varchar(255)	Clave principal: ID de caché único.
data	longblob	Una colección de datos a caché.
expire	int(11)	Una marca de tiempo de Unix que indica cuándo debe caducar la entrada de caché, o -1 para nunca.
created	decimal(14,3)	Una marca de tiempo con una precisión de milisegundos que indica cuando la entrada de caché fue creado.

serialized	smallint(6)	Una bandera para indicar si el contenido está serializado (1) o no (0).
tags	longtext	Lista separada por espacios de etiquetas de caché para esta entrada.
checksum	varchar(255)	La suma de comprobación de invalidación de la etiqueta cuando se guardó esta entrada.

cache_render**procesamiento de caché**

Almacenamiento para la API de caché

Columna	Tipo	Comentarios
cid (Primaria)	varchar(255)	Clave principal: ID de caché único.
data	longblob	Una colección de datos a caché.
expire	int(11)	Una marca de tiempo de Unix que indica cuándo debe caducar la entrada de caché, o -1 para nunca.
created	decimal(14,3)	Una marca de tiempo con una precisión de milisegundos que indica cuando la entrada de caché fue creado.
serialized	smallint(6)	Una bandera para indicar si el contenido está serializado (1) o no (0).
tags	longtext	Lista separada por espacios de etiquetas de caché para esta entrada.
checksum	varchar(255)	La suma de comprobación de invalidación de la etiqueta cuando se guardó esta entrada.

Comment**Comentario**

La tabla base para las entidades de comentarios.

Columna	Tipo	Comentarios
cid (Primaria)	int(10)	
comment_type	varchar(32)	El ID de la entidad objetivo.
uuid	varchar(128)	
langcode	langcode	

comment_entity_statistics**comentar estadísticas de la entidad**

Mantiene estadísticas de la entidad y comentarios posts para mostrar...

Columna	Tipo	Comentarios
entity_id (Primaria)	int(10)	La entity_id de la entidad para la que se compilan las estadísticas.
entity_type (Primaria)	varchar(32)	El tipo de entidad de la entidad a la que este comentario es una respuesta.
field_name (Primaria)	varchar(32)	El nombre de campo del campo que se utilizó para agregar este comentario.
cid	int(11)	El comment.cid del último comentario.
last_comment_timestamp	int(11)	La marca de tiempo de Unix del último comentario que se publicó dentro de este nodo, desde comment.changed
last_comment_name	varchar(60)	El nombre del último autor para publicar un comentario en este nodo, desde comment.name.

last_comment_uid	int(10)	El ID de usuario del último autor para publicar un comentario en este nodo, desde comentario.uid.
comment_count	int(10)	El número total de comentarios sobre esta entidad.

comment_field_data**comentar el campo de datos**

La tabla de datos para las entidades de comentarios.

Columna	Tipo	Comentarios
cid (Primaria)	int(10)	
comment_type	varchar(32)	El ID de la entidad objetivo.
langcode (Primaria)	varchar(12)	
status	tinyint(4)	
pid	int(10)	El ID de la entidad objetivo.
entity_id	int(10)	El ID de la entidad objetivo.
subject	varchar(64)	
uid	int(10)	El ID de la entidad objetivo.
name	varchar(60)	
mail	varchar(254)	
homepage	varchar(255)	
hostname	varchar(128)	
created	int(11)	
changed	int(11)	
thread	varchar(255)	
entity_type	varchar(32)	
field_name	varchar(32)	
default_langcode	tinyint(4)	

comment__comment_body

Almacenamiento de datos para el campo de comentario comment_body.

Columna	Tipo	Comentarios
bundle	varchar(128)	El paquete de instancia de campo al que pertenece esta fila, utilizado al eliminar una instancia de campo.
deleted (Primaria)	tinyint(4)	Un valor booleano que indica si este elemento de datos se ha eliminado
entity_id (Primaria)	int(10)	La identificación de la entidad a la que se adjunta estos datos.
revision_id	int(10)	La identificación de la revisión de la entidad a la que se adjunta estos datos, que para una versión sin versión. El tipo de entidad es el mismo que el ID de la entidad
langcode (Primaria)	varchar(32)	El código de idioma para este elemento de datos.
delta (Primaria)	int(10)	El número de secuencia para este elemento de datos, utilizado para campos de valor múltiple.
comment_body_value	longtext	
comment_body_format	varchar(255)	

Config

La tabla base para los datos de configuración.

Columna	Tipo	Comentarios
collection (Primaria)	varchar(255)	Clave principal: colección de objetos de configuración.
name (Primaria)	varchar(255)	Clave principal: Config. Nombre del objeto.
data	longblob	Un objeto de configuración en serie de datos.

file_managed

La tabla base para las entidades de archivo.

Columna	Tipo	Comentarios
fid (Primaria)	int(10)	
uuid	varchar(128)	
langcode	varchar(12)	
uid	int(10)	El ID de la entidad objetivo.
filename	varchar(255)	
uri	varchar(255)	
filemime	varchar(255)	
filesize	bigint(20)	
status	tinyint(4)	
created	int(11)	
changed	int(11)	

file_usage

Seguimiento de donde se utiliza un archivo.

Columna	Tipo	Comentarios
fid (Primaria)	int(10)	ID de archivo.
module (Primaria)	varchar(50)	El nombre del módulo que está utilizando el archivo.
type (Primaria)	varchar(64)	El nombre del tipo de objeto en el que se utiliza el archivo.
id (Primaria)	varchar(64)	La clave principal del objeto utilizando el archivo.
count	int(10)	La cantidad de veces que este archivo es utilizada por este objeto.

Flood

Flood controla el umbral de eventos, como el número.

Columna	Tipo	Comentarios
fid (Primaria)	int(11)	ID único del evento de inundación.
event	varchar(64)	Nombre del evento (por ejemplo, contacto).
identifier	varchar(128)	Identificador del visitante, como una dirección IP o un nombre de host.
timestamp	int(11)	Marca de tiempo del evento.

expiration	int(11)	Marca de tiempo de caducidad. Los eventos caducados se purgan en la ejecución cron.
------------	---------	---

History

Un registro de qué usuarios han leído qué nodos.

Columna	Tipo	Comentarios
uid (Primaria)	int(11)	Los users.uid que leen el nid del nodo.
nid (Primaria)	int(10)	El node.nid que fue leído.
timestamp	int(11)	La marca de tiempo de Unix en la que se produjo la lectura.

key_value

Tabla genérica de almacenamiento de valor-clave.

Columna	Tipo	Comentarios
collection (Primaria)	varchar(128)	Una colección con nombre de pares de clave y valor.
name (Primaria)	varchar(128)	La clave del par clave-valor. Como KEY es una palabra clave reservada de SQL, nombrefue elegido en su lugar.
value	longblob	El valor.

key_value_expire

Tabla genérica de almacenamiento de clave / valor con vencimiento.

Columna	Tipo	Comentarios
collection (Primaria)	varchar(128)	Una colección con nombre de pares de clave y valor.
name (Primaria)	varchar(128)	La clave del par clave / valor.
value	longblob	El valor del par clave / valor.
expire	int(11)	El tiempo transcurrido desde la época de Unix en segundos cuando este elemento caduco. Por defecto al tiempo máximo posible.

locales_location

Información de ubicación para cadenas fuente.

Columna	Tipo	Comentarios
lid (Primaria)	int(11)	Identificador único de esta ubicación.
sid	int(11)	Identificador único de esta cadena.
type	varchar(50)	El tipo de ubicación (archivo, configuración, ruta, etc)
name	varchar(255)	Escriba la información de ubicación dependiente (nombre de archivo, ruta, etc.).
version	varchar(20)	Versión de Drupal donde se encontró la ubicación.

locales_source

Lista de cadenas fuente en inglés.

Columna	Tipo	Comentarios
lid (Primaria)	int(11)	Identificador único de esta cadena.
source	blob	La cadena original en inglés.
context	varchar(255)	El contexto al que se aplica esta cadena.
version	varchar(20)	Versión de Drupal donde se usó por última vez la cadena (para optimización de locales).

locales_target

Almacenas versiones traducidas de cadenas.

Columna	Tipo	Comentarios
lid (Primaria)	int(11)	ID de cadena fuente. Referencias locales_source.lid.
translation	blob	Valor de cadena de traducción en este idioma.
language (Primaria)	varchar(12)	Código de lenguaje. Referencias language.langcode.
customized	int(11)	Booleano que indica si la traducción es personalizada para este sitio.

locale_file

Información de estado de importación de archivos para la traducción de la interfaz.

Columna	Tipo	Comentarios
project (Primaria)	varchar(255)	Un nombre corto único para identificar el proyecto al que pertenece el archivo.
langcode (Primaria)	varchar(12)	Código de idioma de esta traducción. Referencias language.langcode.
filename	varchar(255)	Nombre del archivo importado.
version	varchar(128)	Etiqueta de versión del archivo importado.
uri	varchar(255)	URI del archivo remoto, el archivo local resultante o el archivo importado localmente.
timestamp	int(11)	Marca de tiempo Unix del archivo importado.
last_checked	int(11)	La marca de tiempo de Unix de la última vez que se confirmó que esta traducción es el más reciente lanzamiento disponible.

menu_link_content

La tabla base para las entidades menu_link_content.

Columna	Tipo	Comentarios
id (Primaria)	int(10)	
bundle	varchar(32)	
uuid	varchar(128)	
langcode	varchar(12)	

menu_link_content_data

La tabla de datos para las entidades menu_link_content.

Columna	Tipo	Comentarios
id (Primaria)	int(10)	
bundle	varchar(32)	
langcode (Primaria)	varchar(12)	
title	varchar(255)	
description	varchar(255)	
menu_name	varchar(255)	
link_uri	varchar(2048)	La URI del enlace.
link_title	varchar(255)	El texto del enlace.
link_options	longblob	Serie serializada de opciones para el enlace.
external	tinyint(4)	
rediscover	tinyint(4)	
weight	int(11)	
expanded	tinyint(4)	
enabled	tinyint(4)	
parent	varchar(255)	
changed	int(11)	
default_langcode	tinyint(4)	

menu_tree

Contiene la jerarquía del árbol del menú.

Columna	Tipo	Comentarios
menu_name	varchar(32)	El nombre del menú. Todos los enlaces con el mismo nombre de menú (como & # 39; herramientas & # 39;)
mlid (Primaria)	int(10)	El ID del enlace del menú (mlid) es la clave principal entera.
id	varchar(255)	Nombre único de la máquina: el ID del complemento.
parent	varchar(255)	El ID del complemento para el padre de este enlace.
route_name	varchar(255)	El nombre de la máquina de una ruta de Symfony definida que representa este elemento del menú.
route_param_key	varchar(255)	Una cadena codificada de parámetros de ruta para cargar por ruta.
route_parameters	longblob	Matriz serializada de los parámetros de ruta de este enlace de menú.
url	varchar(255)	La ruta externa a la que apunta este enlace (cuando no se utiliza una ruta).
title	longblob	El título serializado para el enlace. Puede ser un TranslatableMarkup.
description	varchar(255)	La descripción serializada de este enlace, utilizada para las páginas de administración y el atributo del título. Puede ser un TranslatableMarkup.
class	longblob	La clase para este plugin de enlace.

options	longblob	Una matriz serializada de opciones de URL, como una cadena de consulta o atributos HTML.
provider	text	El nombre del módulo que generó este enlace.
enabled	longblob	Una bandera para indicar si el enlace debe representarse en los menús. (0 = un menú deshabilitado elemento que se puede mostrar en las pantallas de administración, 1 = un enlace normal y visible)
discovered	varchar(50)	Una marca para saber si se descubrió el enlace, por lo que se puede purgar en la reconstrucción
expanded	smallint(6)	Marca para saber si este enlace se debe representar como expandido en los menús - expandido los enlaces siempre muestran sus enlaces secundarios, en lugar de solo cuando el enlace está en el camino activo (1 = expandido, 0 = no expandido)
weight	smallint(6)	Enlace de peso entre enlaces en el mismo menú a la misma profundidad.
metadata	smallint(6)	Una matriz serializada de datos que puede ser utilizada por la instancia del complemento.
has_children	int(11)	Indicador que indica si algún enlace habilitado tiene este enlace como principal (1 = habilitado los niños existen, 0 = no niños habilitados).
depth	longblob	La profundidad relativa al nivel superior. Un enlace con el padre vacío tendrá profundidad == 1.
P1	int(10)	El primer mlid en el camino materializado. Si N = profundidad, entonces pN debe ser igual a mlid Si la profundidad > 1, entonces p (N-1) debe ser igual al mlid del enlace principal. Todos los pX donde X > la profundidad debe ser igual a cero. Las columnas p1. p9 también se llaman los padres.
P2	int(10)	El segundo mlid en el camino materializado. Ver p1.
P3	int(10)	El tercer mlido en el camino materializado. Ver p1.
P4	int(10)	El cuarto mlid en el camino materializado. Ver p1.
P5	int(10)	El quinto mlid en el camino materializado. Ver p1.
P6	int(10)	El sexto mlid en el camino materializado. Ver p1.
P7	int(10)	El séptimo mlid en el camino materializado. Ver p1.
P8	int(10)	El octavo mlid en el camino materializado. Ver p1.
P9	int(10)	El noveno mlid en el camino materializado. Ver p1.
form_classes	varchar(255)	meh

Node

La tabla base para las entidades de nodo.

Columna	Tipo	Comentarios
id (Primaria)	int(10)	
vid	int(10)	
type	varchar(32)	El ID de la entidad objetivo.
uuid	varchar(128)	
langcode	varchar(12)	

9. Implementación ETL

Después de hacer un recorrido por el sistema de gestión de contenido (Drupal) para saber su funcionalidad y estar más familiarizados con el motor de datos que se encuentra manejando en la actualidad, “que es lo primordial que se debe tener claro”, se podrá empezar con la implementación de la estrategia para llevar a cabo la finalidad de este documento. (Propia, 2018).

Cabe Aclarar Que Una Extracción, Transformación y carga tiene sus riesgos ya que se cargaran de un sistema a otro donde es necesario tener muchas cosas en cuenta para la satisfacción y correcto ETL como por ejemplo las especificaciones de mapeo, la perdida de datos, falta de políticas de migración, las interferencias en el momento de hacer la migración, entre muchos otros por lo tanto es necesario estar seguros de que estrategia le conviene a su migración para desarrollarlo de la mejor manera y que no tenga muchas complejidades en el camino ni al finalizarlo.(EIO, s.f)

9.1 Estrategia Big Bang

Esta estrategia consiste en un apagado del sistema antiguo y un encendido del nuevo en un día Cero. Los sistemas (antiguo y nuevo) no coexistirán, ya que se sigue utilizando el sistema actual hasta que el proceso de migración de datos no finaliza. Esta estrategia trabaja y hace pruebas en el nuevo sistema con datos no reales mientras se desarrolla la migración. los usuarios comenzarán a utilizar el nuevo aplicativo el mismo día en el que dejarán de tener acceso al antiguo. Sin embargo, esto no quiere decir que no tenga sus lados positivos y negativos los cuales Modelos de prueba (2014) los define de la siguiente manera:

9.1.1 Ventajas

- Útil para la detección de errores cuando se encuentren todos los módulos ya en construcción

- Aplicable antes de la entrega de proyecto para evaluar el trabajo de todo el sistema con diversos escenarios.
- Se puede evaluar la interacción entre módulos para agilizar los procesos.
- Es apta para aplicar diversos escenarios para poder analizar el trabajo de todos los módulos en diversas situaciones

9.1.2 Desventajas

- Dificultan la detección de posibles errores.
- Solo es posible aplicarla hasta que se esté avanzado el proyecto.
- Se tienen que trabajar con todos los módulos, no se puede hacer un análisis individual.
- Si el proyecto es de corto plazo no se tiene mucho tiempo para la aplicación de la prueba y realizar las correcciones.

10. Planificación de costos

Etapa	Recursos	Tiempo	Costo
Visión	3	2 semanas	\$ 2.000.000
Análisis	3	3 semanas	\$ 4.000.000
Diseño	3	4 semanas	\$ 3.500.000
Pruebas	3	5 semanas	\$ 2.500.000
ETL	3	4 semanas	\$ 5.000.000
Entrega	3	1 día	\$ 1.000.000
Total			\$18.000.000

10.1 Planificación de Actividades

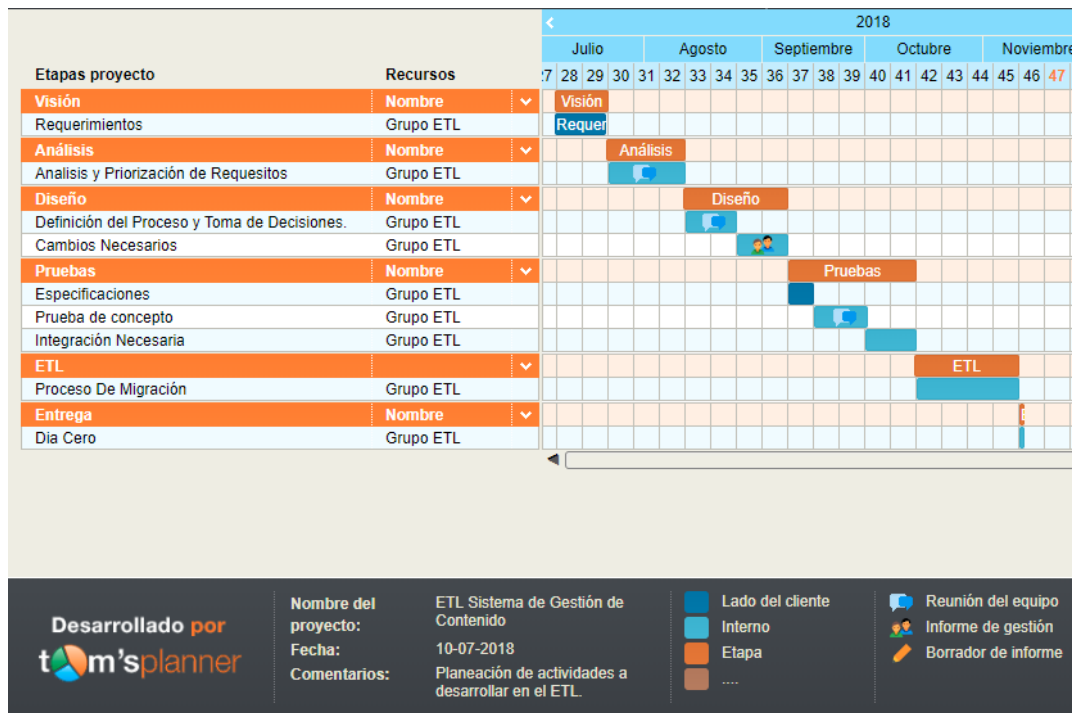


Figura 25. Diagrama de Gantt, Propia Recuperado de <https://www.tomsplanner.es/>

11. Proceso

Para llevar a cabo la migración correspondiente es necesario tener claro:

1. la compatibilidad de MySQL y MariaDB ya que esto facilitara dicho proceso sin mayores errores porque la estructura de la base de datos y los índices de MariaDB son los mismos que los de MySQL. Esto le permite cambiar de MySQL a MariaDB sin tener que alterar sus aplicaciones ya que los datos y las estructuras de datos no tendrán que cambiar, cabe resaltar que en esta migración se utiliza MySQL – 5.6 y MariaDB – 10.3.10
2. Programas Instalados como MariaDB, XAMPP y MySQL. (para ver la instalación de estas herramientas diríjase al punto 5. Proceso De Instalación)
3. El día en el que se va a realizar la migración ya que como se dijo anteriormente es en un día cero donde se migrara y automáticamente se utilizará el “nuevo sistema o bases de datos en él”.

Definiendo estos puntos se podrá empezar con el proceso de ETL en el sistema de gestión de contenidos (DRUPAL):

- Realizar un Backup de la base de datos de DRUPAL es decir, exportar desde MySQL “que es nuestro motor de datos principalmente”, toda la base de datos perteneciente a DRUPAL para poder llevar a cabo la migración a MariaDB.

Nota: Anteriormente XAMPP incluía MySQL pero con la compra que se realizó de este en el 2010, **Michael "Monty" Widenius** decidió trabajar en una nueva alternativa de esta gran creación para que así no se tenga que pagar la licencia que ahora hay que pagar para MySQL por lo tanto XAMPP tuvo una actualización y ahora incluye MariaDB para que sea libre.(TodoBI, 2009).

- Desinstalar el XAMPP que se tiene actualmente para instalar la nueva versión o la actualización, que esta vez incluirá MariaDB. (para ver la instalación del XAMPP diríjase al punto 5 .Proceso De Instalación).
- Se inicia los servidores Apache y MySQL, porque MySQL, pues hasta el momento esto es un error Data ya que en verdad lo que se estaría iniciando es MariaDB pero no se ha hecho esta actualización dentro del XAMPP.
- Se importara la carpeta de la base de datos exportada anteriormente para que este tome todas las tablas y demás del sistema de gestión de contenido (DRUPAL) el cual se desarrollara de la siguiente manera:

Configuración y migración de MariaDB

Después de terminar la instalación nos muestra la siguiente pantalla.

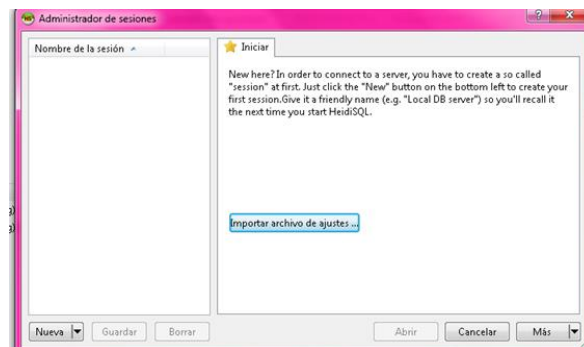


Figura 26. Configuración Recuperada de Propia

Configuración de administración de secciones y se ingresa la contraseña.

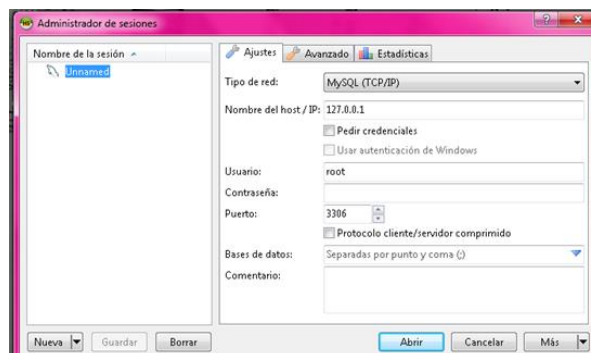


Figura 27. Acceso Base de Datos Recuperado de Propia

Creación de la base de datos.

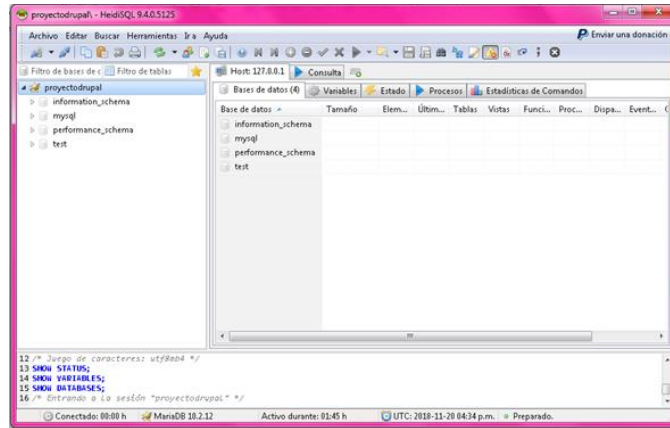


Figura 28. Creación Recuperada de Propia

Se busca la opción para cargar la base de datos SQL.

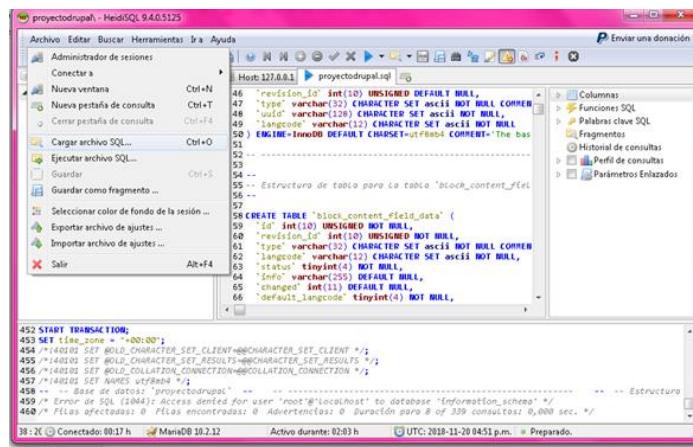


Figura 29. Importación Recuperada de Propia

Seleccionamos la base de datos.

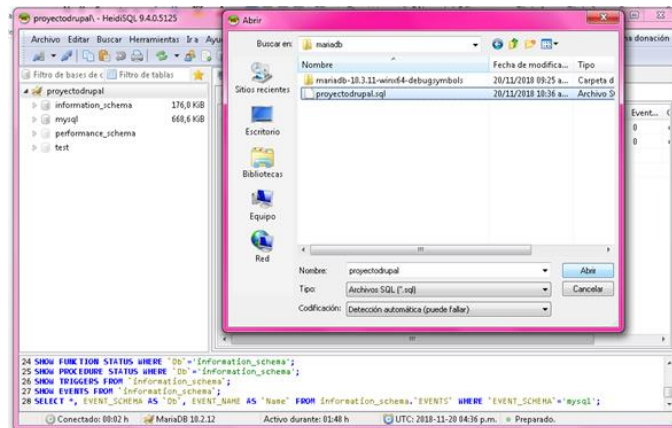


Figura 30. Buscar Archivo Recuperado de Propia

A continuación, se muestra el proceso del cargue de los datos de la DB.

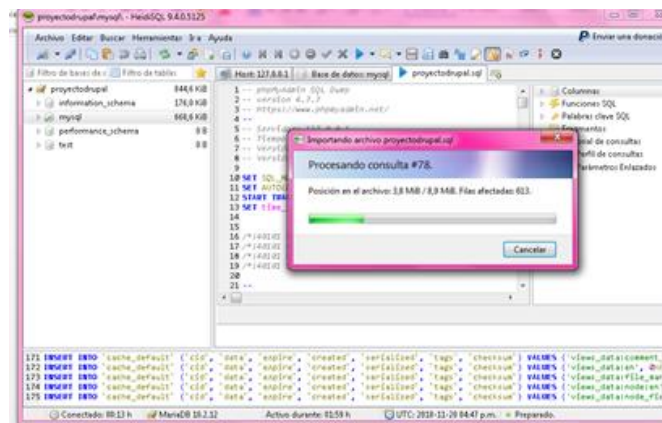


Figura 31. Carga Recuperado de Propia

Cargue de la base de datos completa.

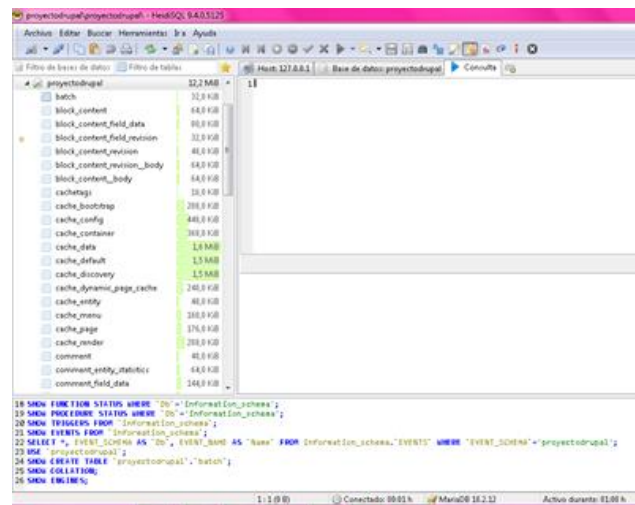


Figura 32. Resultado Recuperado de Propia

Visualización de una de las tablas de la base de datos 'batch'.

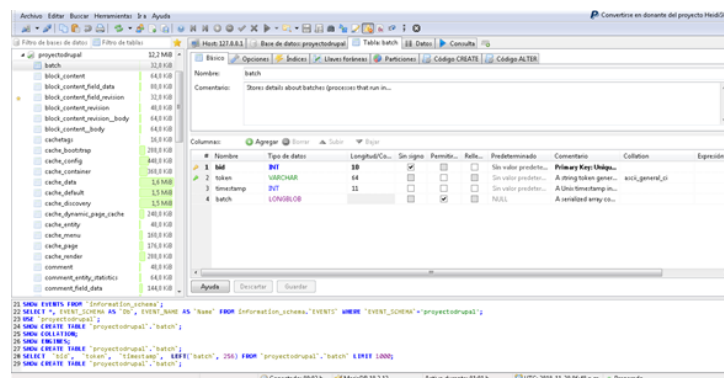


Figura 33. Tabla Batch Recuperado de Propia

Visualización de una de las tablas de la base de datos ‘cache_data’.

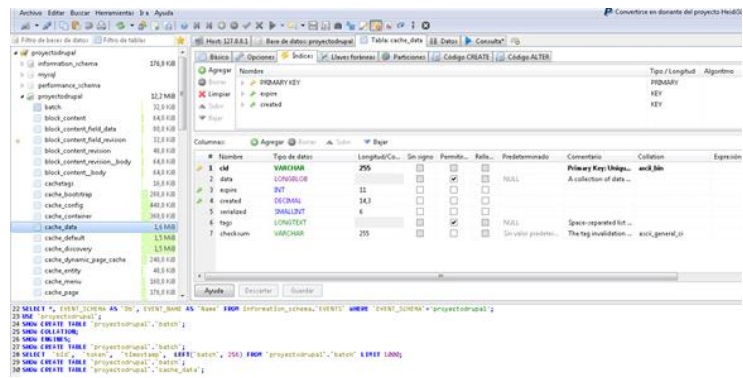


Figura 34. Tabla Cache_Data Recuperado de Propia

Visualización de una de las tablas de la base de datos ‘bloc_content_body’.

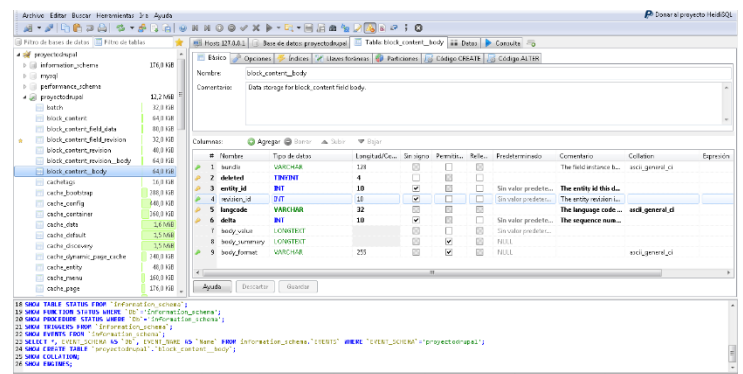


Figura 35. Tabla bloc_content_body Recuperado de Propia

Visualización de una de las tablas de la base de datos ‘history’.

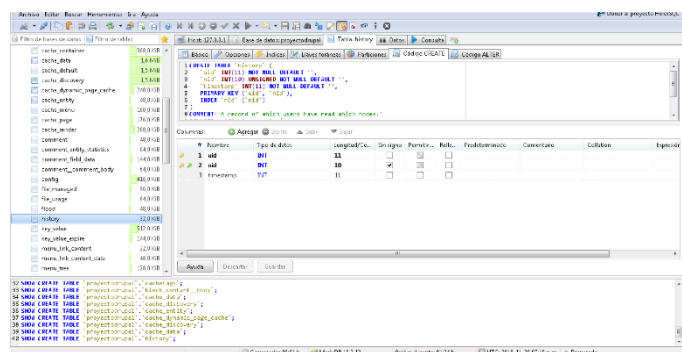


Figura 36. Tabla history Recuperado de Propia

Visualización de una de las tablas de la base de datos 'user_roles'.

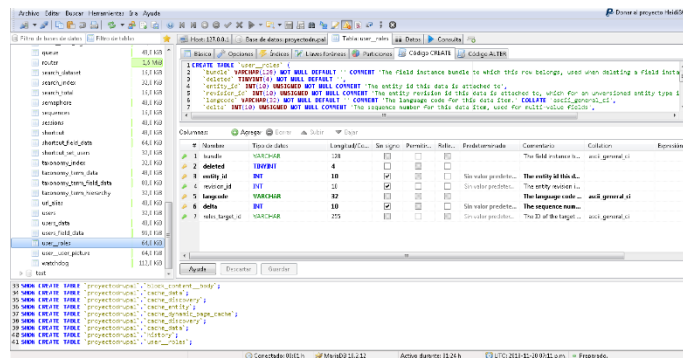


Figura 37. Tabla User_roles Recuperado de Propia

una vez realizado esto se podrá entrar a aquel sistema el cual ingresara a su pantalla principal y demás sin errores.

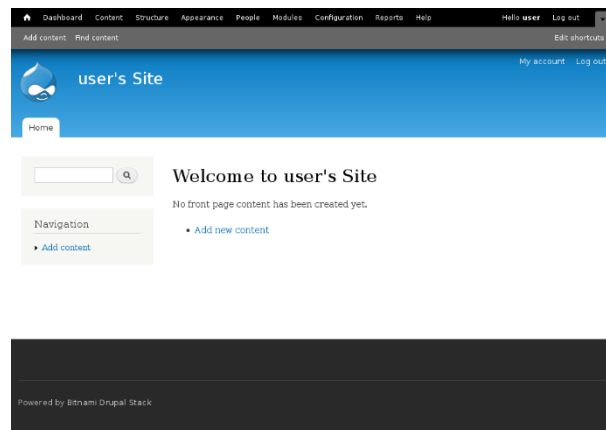


Figura 38. Inicio Drupal Recuperado de Propia

Al finalizar los puntos anteriores se podrá notar que de una manera fácil cambiamos de MySQL a MariaDB dentro de un sistema X en este caso DRUPAL sin tener alteraciones en su migración ya que MariaDB es un derivado con mejoras y cambios muy mínimos de MySQL hasta el momento.

12. Conclusión

Después de realizar el ETL correspondiente al sistema de gestión de contenido (Drupal) utilizando nuestras dos herramientas principales (MySQL – MariaDB0) podemos concluir que la migración no es tan compleja como al principio del documento se creía ya que estos dos motores de bases de datos van muy de la mano y sus diferencias son muy pocas por lo tanto al momento de hacer una migración de este tipo no será muy compleja además de esto la importancia de realizar un ETL con buenas bases ya que este es una parte muy importante para muchas empresas que necesiten aplicarlo, es necesario tener en cuenta que desde el inicio de requerimientos se va desarrollando todo el ETL y si este no es seguro y confiable puede que aquel no sea muy factible como se espere y esto sería de gran gravedad tanto para la empresa como para el grupo que realiza aquella operación.

LISTA DE REFERENCIAS

EcuRed (2018). Drupal Recuperado de <https://www.ecured.cu/Drupal>

Drupal (s.f). Página Oficial Recuperado de <https://www.drupal.org/>

Drupal (2018). Documentación, 3.2. Concept: Additional Tools

Recuperado de https://www.drupal.org/docs/user_guide/en/install-tools.html

Culturacion (s.f). Que es y para qué sirve MySQL Recuperado de <http://culturacion.com/que-es-y-para-que-sirve-mysql/>

EIO IT Business Solutions (s.f). Migración de Datos Recuperado de <https://www.eio.cat/migracion-de-datos-buenas-practicas/?lang=es>

Modelos de Prueba (2014). Prueba de Software Big Bang Recuperado de <https://modelosdepruebasw.wordpress.com/>

TodoBI Business Intelligence (s,f). MariaDB la alternativa a MySQL Recuperado de <http://www.todobi.com/2009/11/mariadb-la-alternativa-mysql.html>

Soporte Nerion (s.f). Que es MariaDB y mejoras de MySQL Recuperado de <https://www.nerion.es/soporte/tutoriales/que-es-mariadb-y-mejoras-sobre-mysql/>

Panoply (2018). MariaDB vs MySQL: Comparación en profundidad Recuperado de <https://blog.panoply.io/a-comparative-vmariadb-vs-mysql>

Drupal (2018). Documentación 1.1 Concept: Drupal as a Content Management System
Recuperado de https://www.drupal.org/docs/user_guide/en/understanding-drupal.html

Drupal (2018). Documentación 3.1 Concept: Server Requirements Recuperado de

https://www.drupal.org/docs/user_guide/en/install-requirements.html