**Creación De La Estructura De La Bd Y Aplicación De Restricciones.**

**GA6-220501096-AA2-EV02**

**Script Bases De Datos Del Proyecto.**

**GA6-220501096-AA2-EV03**

**Presentado Por:**

Carlos Andres González Oliva

Alba Liliana Lemus Saldaña

Maycol Valdés Medina

Edwin Andrés Andrade Torres

**Presentado a:**

Oswaldo Bueno

Servicio Nacional de Aprendizaje – Regional Antioquia

Centro de Recursos Renovables La Salada Tecnología En

Análisis Y Desarrollo De Software – 2626977

2023

**Introducción**

La gestión eficiente de información es fundamental en el desarrollo y operación de sistemas y aplicaciones, especialmente en el contexto de empresas y servicios que requieren el manejo de datos relacionales. En este sentido, las bases de datos desempeñan un papel esencial al permitirnos almacenar, organizar y acceder a datos de manera estructurada y segura. El presente trabajo tiene como objetivo diseñar una base de datos utilizando MySQL Workbench para el sistema de gestión de Tecnidomicilios. A través de este diseño, se pretende establecer un marco sólido para el almacenamiento de información, la creación de relaciones entre entidades y la aplicación de restricciones que garanticen la integridad y confiabilidad de los datos.

**Marco Teórico**

**Bases de Datos:**

Las bases de datos son herramientas esenciales en la gestión de información en una amplia gama de aplicaciones, desde sistemas empresariales hasta aplicaciones web y móviles. Estas bases de datos permiten la organización, almacenamiento y recuperación eficiente de datos, lo que es crucial para la toma de decisiones, la automatización de procesos y la generación de informes. Hay varios tipos de bases de datos, pero una de las más utilizadas es la base de datos relacional.

Una base de datos relacional se basa en el modelo de datos relacional, que organiza la información en tablas bidimensionales compuestas por filas (registros) y columnas (atributos). Cada tabla representa una entidad y las relaciones entre las entidades se establecen mediante claves foráneas que hacen referencia a las claves primarias de otras tablas. Este modelo proporciona una estructura clara y coherente para almacenar y relacionar datos, lo que facilita la consulta y la integridad de los datos.

**MySQL Workbench:**

MySQL Workbench es una herramienta de administración de bases de datos ampliamente utilizada para el diseño, desarrollo y gestión de bases de datos MySQL. Ofrece un entorno gráfico que simplifica tareas como la creación de tablas, la definición de relaciones, la ejecución de consultas SQL y la supervisión del rendimiento de la base de datos. MySQL Workbench proporciona una interfaz intuitiva que permite a los desarrolladores y administradores de bases de datos interactuar con el sistema de gestión de bases de datos MySQL de manera eficiente y efectiva.

**Restricciones de Bases de Datos:**

Las restricciones de bases de datos son reglas o condiciones que se aplican a las tablas de una base de datos relacional con el objetivo de garantizar la integridad, la coherencia y la calidad de los datos almacenados. Estas restricciones desempeñan un papel fundamental en la gestión de datos y aseguran que los datos se mantengan consistentes y se adhieran a las reglas específicas definidas por el diseñador de la base de datos. A continuación, se describen algunas de las restricciones utilizadas en el diseño de la base de datos de Tecnidomicilios:

* **Clave Primaria (Primary Key):** La clave primaria es un conjunto de uno o más campos que identifican de manera única cada registro en una tabla para asegurar que estos campos contengan valores únicos, lo que facilita la identificación y recuperación de registros individuales.
* **Clave Foránea (Foreign Key):** La clave foránea establece relaciones entre tablas al hacer referencia a la clave primaria de otra tabla, esto garantiza la integridad referencial y permite mantener coherencia entre los datos relacionados.
* **Restricción CHECK:** La restricción CHECK permite definir condiciones que los valores en una columna deben cumplir. En el diseño de Tecnidomicilios, se utiliza una restricción CHECK en la columna experiencia de la tabla tecnicos para garantizar que el valor sea mayor o igual a cero. Esto asegura que la experiencia laboral de un técnico no sea un valor negativo, lo que sería incoherente.
* **Índices (Indexes):** Los índices son estructuras de datos que mejoran la velocidad de recuperación de datos al crear estructuras de búsqueda optimizadas para columnas específicas. En la tabla servicio.

**Script**

-- Crear la base de datos de Tecnidomicilios

CREATE DATABASE Tecnidomicilios;

-- Usar la base de datos creada

USE Tecnidomicilios;

-- Tabla para almacenar información sobre técnicos de servicio

CREATE TABLE `tecnicos` (

`id\_tecnico` int not null auto\_increment*, -- Identificador único del técnico*

`nombres` varchar(50) not null, *-- Nombres del técnico*

`apellidos` varchar(50) not null, *-- Apellidos del técnico*

`compania` varchar(50) not null default "Independiente", *-- Nombre de la compañía del técnico, por defecto Independiente*

`direccion` varchar(100), *-- Dirección del técnico (puede ser nulo)*

`telefono` varchar(11) not null unique, *-- Número de teléfono del técnico con restricción de valor único*

`email` varchar(100) not null unique, *-- Correo electrónico del técnico con restricción de valor único*

`contrasenia` varchar(255) not null, *-- Contraseña del técnico*

`fotografia` varchar(255), *-- URL de la fotografía del técnico*

`experiencia` int check (experiencia >= 0), *-- Años de experiencia laboral del técnico (con restricción CHECK)*

`habilidades` varchar(500), *-- Enumeración de habilidades o especialidades del técnico*

PRIMARY KEY (`id\_tecnico`) *-- Clave primaria*

);

-- Tabla para almacenar información sobre clientes

CREATE TABLE `clientes` (

`id\_cliente` int not null auto\_increment, *-- Identificador único del cliente*

`nombre` varchar(250) not null, *-- Nombre completo del cliente*

`telefono` varchar(11) not null, *-- Número de teléfono del cliente*

`direccion` varchar(100) not null, *-- Dirección del cliente*

`email` varchar(100) not null unique, *-- Correo electrónico del cliente con restricción de valor único*

`contrasenia` varchar(255) not null, *-- Contraseña del cliente*

`fecha\_registro` date not null, *-- Fecha del registro en la aplicación*

PRIMARY KEY (`id\_cliente`) *-- Clave primaria*

);

-- Tabla para almacenar categorías de servicios con sus subcategorías

CREATE TABLE `categoria\_servicio` (

`id\_categoria` int not null auto\_increment, *-- Identificador único de la categoría*

`nombre\_categoria` varchar(100) not null, *-- Nombre de la categoría*

`descripcion` varchar(500*), -- Descripción de la categoría (puede ser nulo)*

`icono` varchar(255*), -- URL o referencia al ícono o imagen de la categoría*

`id\_categoria\_padre` int, *-- ID de la categoría superior (nulo para categorías principales y el número corresponde a la categoría padre (el mismo de id\_categoria))*

PRIMARY KEY (`id\_categoria`), *-- Clave primaria*

FOREIGN KEY (`id\_categoria\_padre`) REFERENCES `categoria\_servicio`(`id\_categoria`) *-- Clave foránea para la categoría superior*

);

-- Tabla para almacenar información sobre servicios ofrecidos

CREATE TABLE `servicio` (

`id\_servicio` int not null auto\_increment, *-- Identificador único del servicio*

`nombre\_servicio` varchar(250) not null, *-- Nombre del servicio*

`descripcion` varchar(500) not null, *-- Descripción del servicio*

`valor` float not null, *-- Costo del servicio*

`id\_cliente` int not null, *-- ID del cliente que ofrece el servicio*

`id\_categoria` int not null, *-- ID de la categoría a la que pertenece el servicio*

`estado` varchar(25), *-- Estado del servicio (por ejemplo, completado, pendiente, cancelado, etc.)*

FOREIGN KEY (`id\_cliente`) REFERENCES `clientes`(`id\_cliente`), *-- Clave foránea para el cliente*

FOREIGN KEY (`id\_categoria`) REFERENCES `categoria\_servicio`(`id\_categoria`), *-- Clave foránea para la categoría*

PRIMARY KEY (`id\_servicio`) *-- Clave primaria*

);

-- Tabla para almacenar solicitudes de servicio

CREATE TABLE `solicitud\_servicio` (

`id\_solicitud` int not null auto\_increment, -- *Identificador único de la solicitud*

`id\_cliente` int not null, *-- ID del cliente que solicita el servicio*

`id\_servicio` int not null, *-- ID del servicio solicitado*

`id\_tecnico` int not null, *-- ID del técnico asignado a la solicitud*

`fecha` date not null, *-- Fecha de la solicitud*

`estado\_solicitud` varchar(25), *-- Estado de la solicitud*

PRIMARY KEY (`id\_solicitud`), *-- Clave primaria*

FOREIGN KEY (`id\_cliente`) REFERENCES `clientes`(`id\_cliente`), *-- Clave foránea para el cliente*

FOREIGN KEY (`id\_servicio`) REFERENCES `servicio`(`id\_servicio`), *-- Clave foránea para el servicio*

FOREIGN KEY (`id\_tecnico`) REFERENCES `tecnicos`(`id\_tecnico`) -- *Clave foránea para el técnico*

);

-- Tabla para almacenar historial de servicios

CREATE TABLE `historial\_servicios` (

`id\_historial` int not null auto\_increment, *-- Identificador único del registro en el historial*

`id\_cliente` int not null, *-- ID del cliente al que pertenece el historial*

`id\_servicio` int not null, *-- ID del servicio solicitado o utilizado*

`fecha\_solicitud` date not null, -- *Fecha en que se solicitó o utilizó el servicio*

`estado` varchar(25), *-- Estado del servicio (por ejemplo, completado, pendiente, cancelado, etc.)*

PRIMARY KEY (`id\_historial`), *-- Clave primaria*

FOREIGN KEY (`id\_cliente`) REFERENCES `clientes`(`id\_cliente`), *-- Clave foránea para el cliente*

FOREIGN KEY (`id\_servicio`) REFERENCES `servicio`(`id\_servicio`) *-- Clave foránea para el servicio*

);

-- Tabla para almacenar comentarios sobre servicios

CREATE TABLE `comentarios` (

`id\_comentario` int not null auto\_increment, -- *Identificador único del comentario*

`id\_servicio` int not null, -- *ID del servicio al que se refiere el comentario*

`id\_cliente` int not null, *-- ID del cliente que escribió el comentario*

`contenido` text not null, -- *Contenido del comentario*

`calificacion` int not null, -- *Calificación del servicio*

`fecha\_publicacion` datetime not null, *-- Fecha de publicación del comentario*

PRIMARY KEY (`id\_comentario`), -- *Clave primaria*

FOREIGN KEY (`id\_servicio`) REFERENCES `servicio`(`id\_servicio`), *-- Clave foránea para el servicio*

FOREIGN KEY (`id\_cliente`) REFERENCES `clientes`(`id\_cliente`) -- *Clave foránea para el cliente*

);

-- Índices para mejorar el rendimiento en las búsquedas--

-- Índice en la columna 'estado' de la tabla 'servicio'

CREATE INDEX idx\_estado ON `servicio`(`estado`);

-- Índice en la columna 'email' de la tabla 'tecnicos'

CREATE INDEX idx\_email\_tecnicos ON `tecnicos`(`email`);

-- Índice en la columna 'email' de la tabla 'clientes'

CREATE INDEX idx\_email\_clientes ON `clientes`(`email`);

-- Índice en la columna 'estado\_solicitud' de la tabla 'solicitud\_servicio'

CREATE INDEX idx\_estado\_solicitud ON `solicitud\_servicio`(`estado\_solicitud`);

-- Índice en la columna 'estado' de la tabla 'historial\_servicios'

CREATE INDEX idx\_estado\_historial ON `historial\_servicios`(`estado`);

**Conclusión**

En este trabajo, hemos diseñado una base de datos utilizando MySQL Workbench para el sistema de gestión de Tecnidomicilios, en el cual se incluyen las tablas para almacenar información sobre técnicos de servicio, clientes, categorías de servicios, servicios ofrecidos, solicitudes de servicio, historial de servicios y comentarios sobre servicios, y se han aplicado restricciones de base de datos para asegurar la integridad de los datos. El diseño de esta base de datos proporciona una base sólida para el desarrollo de la aplicación Tecnidomicilios, permitiendo un almacenamiento organizado y seguro de datos relacionados con la interacción entre técnicos y clientes. Además, facilita la realización de consultas y operaciones sobre la información de manera eficiente y confiable, lo que contribuye a la eficacia y calidad del servicio ofrecido por la plataforma.

**Bibliografía**

Torres Remon, M. Á. (2012). *Programación Transact SQL Server 2012*. Editorial Macro.

García, A. (2003). Manual práctico de sql. *Omega*.

Oracle. (s.f.). *Getting Started with MySQL Database Service. Oracle.* <https://docs.oracle.com/en-us/iaas/mysql-database/doc/getting-started.html>

Valencia, E. A. R. (2015). Diseño de bases de datos distribuidas con MySQL Workbench 5.2 CE. *Área de Ciencias y Tecnología*.

**Anexos**

Anexo 1: Script SQL

- En este anexo se presenta el script SQL utilizado para crear la base de datos "Tecnidomicilios" y sus tablas.

Anexo 2: Modelo Entidad-Relación (ER)

- Este anexo contiene el diagrama del Modelo Entidad-Relación (ER) que representa la estructura de la base de datos.