PROCESO DE GESTIÓN DE FORMACIÓN PROFESIONAL INTEGRAL

FORMATO GUÍA DE APRENDIZAJE

1. IDENTIFICACIÓN DE LA GUIA DE APRENDIZAJE

- Denominación del Programa de Formación: Análisis y desarrollo de software
- Código del Programa de Formación: 228118
- Nombre del Proyecto Formativo : construcción de software integrador de tecnologías orientadas a servicios.
- Fase del Proyecto: Ejecución
- Actividad de Proyecto Formativo: Codificar los módulos del software
- Competencia: Desarrollar la solución de software de acuerdo con el diseño y metodologías de desarrollo
- Resultados de Aprendizaje: Codificar el software de acuerdo con el diseño establecido
- Duración de la Guía de Aprendizaje (horas): 340 Horas

2. PRESENTACIÓN

Estimado aprendiz, el SENA extiende una cordial bienvenida a la octava guía de aprendizaje que comprende la competencia técnica de: desarrollar la solución de software de acuerdo con el diseño y metodologías de desarrollo.

En palabras de Avison y Fitzgerald (1995), una metodología es una serie de procedimientos y herramientas que ayudan a los desarrolladores a implementar nuevos sistemas de información y están formadas por fases que guían a los desarrolladores a elegir las técnicas más apropiadas en cada momento del proyecto.

Para el desarrollo de las actividades planteadas en esta guía, contará con el acompañamiento de los instructores asignados al programa, los cuales de forma continua y permanente lo orientarán con las pautas necesarias para el logro de las actividades de aprendizaje, brindando herramientas básicas de tipo conceptual y metodológico. Los instructores programarán encuentros de asesoría virtual, para brindar orientaciones específicas relacionadas con las temáticas a desarrollar en las actividades. Es importante que organice su tiempo, dada la exigencia que demanda la realización de esta guía de aprendizaje. No olvide revisar y explorar los materiales de estudio del programa.

Por consiguiente, se presentan cada una de las acciones de aprendizaje que le permitirán desarrollar lo anteriormente mencionado.

3. FORMULACIÓN DE LAS ACTIVIDADES DE APRENDIZAJE

En este apartado se describirán las actividades de aprendizaje para cada una de las competencias que plantea la fase de ejecución del proyecto formativo: construcción de software integrador de tecnologías orientadas a servicios.

3.1 Actividades de reflexión inicial:

En grupos de 5 personas genero una discusión respondiendo a las preguntas ¿Por qué las aplicaciones móviles son importantes para la sociedad?, ¿Qué beneficio puedo obtener al convertirme en desarrollador de apps?, ¿Cuáles son las aplicaciones móviles más conocidas y que más se usan en la actualidad?, ¿Qué tipo de aplicación movil mejoraría mi vida? Elaboro una presentación en Presentación en PowerPoint con las respuestas seleccionadas y las comparto con los demás grupos.

- Ambiente Requerido: ambiente de formación con computadores
- Materiales: Guía de aprendizaje, equipos con sistema operativo Windows y conectividad a internet.
- Duracion: 2 horas

3.2 Actividades de contextualización e identificación de conocimientos necesarios para el aprendizaje:

Actividad Individual:

- Leo detenidamente el documento “Casos de Éxito Aplicaciones moviles.pdf” este documento contiene una serie de artículos sobre generalidades de las apps para dispositivos móviles, cuáles serían las características de una aplicación exitosa, y la descripción de dos casos de éxito, Angry Birds y whatsaap. El documento se encuentra en la carpeta de la fase Ejecución en la plataforma, además buscar en internet información que profundice o brinde mayor información sobre los temas leídos”

- Respondo las siguientes preguntas en el Foro que se encuentra en la plataforma.

    - ¿Qué fue lo que más le llamó la atención de los diferentes artículos leidos?
    - ¿Cuales son las características fundamentales que debe tener una aplicación exitosa?

- ¿Por qué es importante incursionar eln el mundo del desarrollo de aplicaciones y videojuegos para dispositivos móviles?
- ¿Qué aplicación podría crear que solucione un problema real actual?
- ¿Qué puede aprender del caso de Rovio?
- ¿Qué puedo aprender del caso de Whatsapp?

- Participo activamente en la lluvia de ideas dirigida por el instructor.
- Leo el documento “Temario Desarrollo de Aplicaciones en la plataforma Android.pdf”
- Participo en la explicación del instructor y en la sesión de preguntas sobre el temario.

Ambiente requerido: ambiente de formación con computadores

Materiales de formación: Guía de aprendizaje, equipos con sistema operativo Windows y conectividad a internet.

Material de apoyo: Documento Temario Desarrollo de Aplicaciones en la plataforma Android.pdf”

Duración de la actividad: 2 horas.

3.3 Actividades de apropiación:

Esta actividad se propone para el desarrollo de los diferentes aspectos claves acerca de la elaboración de aplicaciones móviles para Android.

Elementos para tener en el desarrollo del taller:

Taller

Elabore una investigación corta usando los materiales disponibles en la biblioteca o internet acerca del desarrollo de aplicaciones móviles para Android.

● ¿Qué es Android?

● Definición del concepto de APK.

● ¿Qué es el Android SDK?

● ¿Cuál es el lenguaje utilizado para desarrollar aplicaciones para Android?

● ¿Qué IDEs de desarrollo existen para codificar?

- Definir el concepto de Android multiusuario realizar una breve descripción de este concepto
- Definir el concepto de mínimo privilegio
- ¿Cuáles son los componentes esenciales de una aplicación Android?

Ambiente requerido: ambiente de formación con computadores

Materiales de formación: Guía de aprendizaje, equipos con sistema operativo Windows y conectividad a internet.

Material de apoyo: Documento Temario Desarrollo de Aplicaciones en la plataforma Android.pdf"

Duración de la actividad: 6 horas.

3.4 Actividades de Transferencia el Conocimiento:

3.4.1 Concepotos básicos Kotlin

- En esta actividad los aprendices realizaran los codelabs planteados en el material "Concepotos Basicos Kotlin".

Ambiente requerido: ambiente de formación con computadores

Materiales de formación: Guía de aprendizaje, equipos con sistema operativo Windows y conectividad a internet

Material de apoyo: Link "Concepotos Basicos Kotlin"

Duración de la actividad: 18 horas.

3.4.2 Actividad: Activities e Interfaces

• En esta actividad los aprendices realizarán ejercicios prácticos sobre el empleo y uso de las activities o interfaces de usuario. Atiendo a la demostración de método realizada por el instructor sobre cada uno de los temas identificados en el temario de la actividad.Para tal actividad consulte los medios didácticos entregados por el instructor.

Para tal actividad consulte el siguiente enlace:

https://developer.android.com/guide/components/activities/?hl=es-419

• Ambiente Requerido: ambiente de formación con computadores

• Materiales: Guía de aprendizaje No 1, documento de apoyo: Activities e Interfaces en formato PDF

(Carpeta de material de apoyo), equipos con sistema operativo Windows y conectividad a internet.

3.4.2 Actividad: Activities e Intents

• En esta actividad los aprendices realizarán ejercicios prácticos sobre el empleo y uso de las activities e intents. Atiendo a la demostración de método realizada por el instructor sobre cada uno de los temas identificados en el temario de la actividad.Para tal actividad consulte los medios didácticos entregados por el instructor.Para tal actividad consulte el siguiente enlace: https://developer.android.com/guide/components/intents- filters?hl=es-419

• Ambiente Requerido: ambiente de formación con computadores

• Materiales: Guía de aprendizaje No 1, documento de apoyo: Intents en formato PDF (Carpeta de material de apoyo), equipos con sistema operativo Windows y conectividad a internet.

3.4.3 Actividad: Componentes Interfaz Gráfica

• En esta actividad los aprendices realizarán ejercicios prácticos sobre el uso de los diferentes componentes de la interfaz gráfica empleados en Android. Atiendo a la demostración de método realizada por el instructor sobre cada uno de los temas identificados en el temario de la actividad.Para tal actividad consulte los medios didácticos entregados por el instructor.Para tal actividad consulte los medios didácticos entregados por el instructor. Para tal actividad consulte el siguiente enlace: https://developer.android.com/guide/topics/ui/?hl=es-419

• Materiales: Guía de aprendizaje No 1, documento de apoyo: Componentes Interfaz Gráfica formato PDF (Carpeta de material de apoyo), equipos con sistema operativo Windows y conectividad Internet

3.4.4 Actividad: Activities – Views

• En esta actividad los aprendices realizarán ejercicios prácticos sobre el uso de los diferentes componentes de la interfaz gráfica empleados en Android. Atiendo a la demostración de método realizada por el instructor sobre cada uno de los temas identificados en el temario de la actividad.Para tal actividad consulte los medios didácticos entregados por el instructor.Para tal actividad consulte los medios didácticos entregados por el instructor. Para tal actividad consulte el siguiente enlace:
https://www.movilzona.es/tutoriales/android/desarrollo/curso-de-desarrollo-android-tema-views-tipos-de-views-mas-habituales-y-sus-propiedades/

• Materiales: Guía de aprendizaje No 1, documento de apoyo: Views en Android en formato PDF (Carpeta de material de apoyo), equipos con sistema operativo Windows y conectividad a internet.

3.4.5 Actividad: Layouts - ImageViews – OnClickListener

• En esta actividad los aprendices realizarán ejercicios prácticos sobre el uso de los diferentes tipos de layouts en Android. Atiendo a la demostración de método realizada por el instructor sobre cada uno de los temas identificados en el temario de la actividad.Para tal actividad consulte los medios didácticos entregados por el instructor. Para tal actividad consulte los medios didácticos entregados por el instructor. Para tal actividad consulte el siguiente enlace: http://www.hermosaprogramacion.com/2014/09/android-layouts-views/

• Materiales: Guía de aprendizaje No 1, documento de apoyo: Layouts en Android en formato PDF (Carpeta de material de apoyo), equipos con sistema operativo Windows y conectividad a internet.

3.4.6 Actividad: Proyecto Final

En esta actividad los aprendices realizarán un proyecto final empleando todos los conceptos vistos durante el desarrollo de la formación del lenguaje de programación en Android. Para tal actividad consulte los medios didácticos entregados por el instructor Tomando como referencia la planeación pedagógica y las orientaciones para elaborar guías de aprendizaje citado en la guía de desarrollo curricular

Materiales de formación: : Guía de aprendizaje, equipos con sistema operativo Windows y conectividad a internet.

Material de apoyo:

Tutorial 1: Como hacer un App en android desde cero

Tutorial 2: Como actualizar el SDK de android (SDK Manager)

Tutorial 3: Como crear un emulador en Android (AVD Manager)

Tutorial 4 Como Instalar y Configurar los emuladores de Genymotion

Tutorial 5: Como Utilizar mi Celular para emular mis aplicaciones Android

Tutorial 6: Como crear la interfaz gráfica de mi aplicación Android (LinearLayout)

Tutorial 7: Propiedad Weight en LinearLayout

Tutorial 8: Como crear la interfaz grafica de la aplicacion android - RelativeLayout

Tutorial 9: Como agregar un scroll a la interfaz de mi aplicacion Android - ScrollView

Tutorial 10: Como hacer una App en Android -onClickListener funcion del boton

Tutorial 11: Como usar RadioButton y CheckBox en Android

Tutorial 12 Como usar un Spinner en mi aplicacion Android

Tutorial 13 Como usar un ListView en mi Aplicacion Android

Tutorial 13 1 ListView con ArrayList

Tutorial 14: Como cambiar entre pantallas en Android (Navegacion entre Activities)

Tutorial 15: Como pasar Informacion de un Activity a otro en Android

Tutorial 16: Como Animar un boton personalizado en android

Tutorial 17: Como usar fragments en Android

Tutorial 18: Como usar Fragments Dinámicos

Tutorial 19 Como agregar sonido a un boton en Android

Tutorial 20: Como hacer un SplashScreen o pantalla de inicio en Android

Tutorial 21: Swipe Views Como Hacer Un ViewPager En Android

Tutorial 22: Como implementar un Navigation Drawer en Android Studio

Tutorial 23: Como Guardar datos en Android - SharedPreferences

Tutorial 24: SQLite en Android- Interfaz de Usuario

Tutorial 25: SQLite en Android 2- Como crear la tabla e insertar datos

Tutorial 26: SQLite en Android 3 -Como buscar, eliminar y actualizar datos en la tabla

Tutorial 27: Como conectarse a una base de datos MySQL con la Aplicacion Android

Tutorial 28: Como conectar a una base de datos MySQL con la aplicación Android

Tutorial 29: Como conectar a una base de datos MySQL con la aplicación Android

Tutorial 30: Google Maps

Tutorial 31: Volley en Android Studio

Evidencias de aprendizaje: APK de la app del proyectyo con el producto mínimo viable definid por el instructor.

Duración de la actividad: 120 horas.

4. PLANTEAMIENTO DE EVIDENCIAS DE APRENDIZAJE PARA LA EVALUACIÓN EN EL PROCESO FORMATIVO.

Fase del proyecto formativo	Actividad del proyecto formativo	Actividad de Aprendizaje	Evidencias de Aprendizaje	Criterios de Evaluación	Técnicas e Instrumentos de Evaluación
				Los criterios de evaluación se encuentran en el formato instrumento de evaluación para consultar estos criterios viste la plataforma Territorium Contenido del Curso/ III Trimestre 2024/ Programación Dispositivos Móviles / Instrumentos de Evaluación Por cada una de las actividades propuestas hay un instrumento de evaluación	

5. GLOSARIO DE TÉRMINOS

ADB y ADB shell, Android Depuration Bridge o Puente de Depuración de Android. Es la

herramienta utilizada para enviar comandos desde un ordenador a un dispositivo con este sistema. El shell es un programa Terminal que hace de traductor de nuestras órdenes, actúa sobre una línea de comandos.

• AOSP, Android Open Source Project o Proyecto de Código Abierto Android. Las propias siglas ya lo indican, Android es un sistema de código abierto, lo que quiere decir que cualquiera puede descargar su código fuente y modificarlo a su gusto. Este detalle precisamente es el que hace que Android sea un grande entre los grandes, consiguiendo diferenciarse así de su competidor más directo, iOS de Apple, que se trata de un sistema cerrado.

• APK, Android Package Archive o Paquete de Archivo de Android, es la extensión de los paquetes de los programas que operan en este sistema. Pueden ser abiertos y modificados fácilmente.

• APN, Access Point Name o Nombre del Punto de Acceso, son los datos para poder acceder a las redes 2G, 3G para el uso de datos e internet, y además hacer uso de mensajes multimedia MMS desde el móvil que tiene cada operadora de forma única. En la mayoría de los casos estos datos vienen ya configurados.

• APP, abreviatura de la palabra inglesa application, que quiere decir aplicación.

• ARM, Advanced RISC Machines o Máquinas RISC Avanzadas, es una familia de microprocesadores RISC diseñados por la empresa Acom Computers. Son microprocesadores muy usados en los dispositivos móviles gracias a su gran característica, su bajo consumo.

• OTA, Over The Air o Sobre El Aire, es la acción de descargar datos a través del aire. Las actualizaciones oficiales de Android normalmente se realizan mediante OTA, al igual que la instalación de aplicaciones y sus actualizaciones.

• SDK, Software Development Kit o Kit de Desarrollo de Software, es un conjunto de herramientas de desarrollo que permite al programador crear aplicaciones para un sistema, en este caso Android.

• BOOT, es el proceso que se realiza en el inicio de un ordenador o dispositivo computarizado como puede ser un móvil Android. Su función es la de cargar el programa de arranque que carga toda la configuración del hardware para un correcto funcionamiento.

• Bootloader, es el boot encargado del arranque en los dispositivos móviles.

• Brick o semibrick, también conocida como pantalla de la muerte, es la consecuencia de cargar una ROM de forma errónea. En casi la totalidad de los casos es recuperable.

• Class 2, 4, 6, 10, son las velocidades de las tarjetas SD usadas por los dispositivos móviles, las clases 2 trabajan a 2Mb, la clase 4 a 4Mb, y así sucesivamente.

• Dalvik, es una máquina virtual encargada de la ejecución de las aplicaciones en la versión 2.1 y anteriores de Android.

• Fastboot, es un bootloader con el que podemos hacer flasheos de bajo nivel.

• Hard Reset o Factory Reset, es el proceso de dejar el teléfono tal y como estaba justo al momento de comprarlo como de fábrica, o bien para recuperarlo a como esta ajusto después de haber instalado una rom. HardSPL, otro bootloader, en este caso ha sido modificado para dejarnos cargar una ROM a nuestra preferencia.

• Kernel, es el núcleo de los sistemas que se encarga de realizar las operaciones relativas a la gestión del sistema operativo (gestión de memoria, planificador, etc.)

• Launcher, es la parte encargada de la interfaz gráfica de usuario de Android, es totalmente personalizable como ya hemos visto.

• Wipe, del verbo limpiar en inglés, es lo que comúnmente llamamos a "formatear" en el ámbito de los ordenadores, hacer wipe implica la pérdida de los datos almacenados

6. REFERENTES BILBIOGRÁFICOS

- **https://desarrolloweb.com/articulos/introduccion-android.html**
- **http://androidzone.org/android-para-principiantes/**
- **http://androidzone.org/diccionario-android/**
- **https://www.youtube.com/watch?v=Be5yKKIWVrE**
- **http://www.sgoliver.net/blog/curso-de-programacion-android/indice-de-contenidos/**
- **https://www.tutorialspoint.com/android/**
- **http://www.hermosaprogramacion.com/android/**

7. CONTROL DEL DOCUMENTO

	Nombre	Cargo	Dependencia	Fecha
Autor (es)				

8. CONTROL DE CAMBIOS (diligenciar únicamente si realiza ajustes a la guía)

	Nombre	Cargo	Dependencia	Fecha	Razón del Cambio
Autor (es)					



PROCESO DE GESTIÓN DE FORMACIÓN PROFESIONAL INTEGRAL

FORMATO GUÍA DE APRENDIZAJE

1. IDENTIFICACIÓN DE LA GUIA DE APRENDIZAJE

- Denominación del Programa de Formación: TÉCNICO EN PROGRAMACIÓN DE SOFTWARE
- Código del Programa de Formación: 233104
- Nombre del Proyecto (si es formación Titulada): DESARROLLO DE SOFTWARE A LA MEDIDA PARA EL SECTOR EMPRESARIAL
- Fase del Proyecto (si es formación Titulada): Planeación
- Actividad de Proyecto (si es formación Titulada): Estructurar la base de datos de acuerdo al diseño y el motor de base de datos seleccionado
- Competencia: Administrar base de datos de acuerdo con los estándares y requisitos técnicos
- Resultados de Aprendizaje Alcanzar:
- Construir la base de datos según requerimientos del cliente.
- Programar sentencias SQL en un Sistema Manejador de Bases de Datos según requerimientos del cliente
- Duración de la Guía: 144 Horas

2. PRESENTACIÓN

La información es el bien más preciado de una organización, es por ello que los datos deben estar almacenados en un Sistema Manejador de Bases de Datos, en esta guía nos centraremos en el manejo de Bases de Datos a través de instrucciones SQL.

Para SQL, se realizará la construcción de diferentes bases de datos y tablas mediante utilización de sentencias DDL (Lenguaje de Definición de Datos) y DML (Lenguaje de Manipulación de Bases de Datos), estas sentencias se apicaran específicamente con el gestor de bases de datos MySQL; aunque esté trabajando en plataformas diferentes, la sintaxis básica de SQL es estándar.

3. FORMULACIÓN DE LAS ACTIVIDADES DE APRENDIZAJE

3.1. Actividades de Reflexión inicial.

Vamos a jugar stop por equipos de cinco aprendices, tenga en cuenta las siguientes indicaciones:

- Organice equipos de cinco integrantes
- Cada integrante del equipo debe contar con una hoja y lápiz o lapicero
- Como equipo deben seleccionar cinco temas(palabras), *ej nombre, color, animal, entre otros*
- Una vez elegidos los temas se sortea una letra entre los jugadores y comienza la ronda
- Por medio de votación de los jugadores, las respuestas de cada tema serán marcadas como validas o invalidadas
- Los puntajes serán los siguientes 10 puntos para cada respuesta valida, 5 para respuestas repetidas y 0 para las invalidas
- Se repite el mismo proceso para cada ronda, se siguieren 4 rondas.
- Al finalizar las 4 rondas realicen diligencien la siguiente tabla:

Letra/ Ronda 1	Palabra 1	Palabra 2	Palabra 3	Palabra 4	Palabra 5
Cantidad de Respuestas Validas					
Cantidad de Respuestas Repetidas					
Cantidad de Respuestas invalidas					
Letra/ Ronda 2	Palabra 1	Palabra 2	Palabra 3	Palabra 4	Palabra 5
Cantidad de Respuestas Validas					
Cantidad de Respuestas Repetidas					
Cantidad de Respuestas invalidas					

Una vez finalizada la recolección de información por equipo, con su instructor construyan una tabla similar con la información de todos los equipos, una vez finalizada la actividad realicen un conversatorio sobre los resultados obtenidos, específicamente los datos coincidentes, valores máximos, mínimos, entre otros.

3.2 Actividades de contextualización e identificación de conocimientos necesarios para el aprendizaje.)

Actividad 1. Fundamentar conceptos para diferenciar Bases de Datos Relacionales y no Relacionales
Tipo de actividad: Individual
Tiempo estimado para la actividad: 10 horas

Tomando como referencia el material suministrado por su instructor y sus investigaciones realizadas, en un documento de texto solucione los siguientes interrogantes:

1. Que tipos de bases de datos existen, mencione sus características, ventajas y desventajas
2. Qué es un Sistema manejador de bases de Datos
3. Cuáles son los principales motores de bases de datos existentes y sus características.
4. Que es un Modelo Entidad Relación y modelo relacional
5. Mencione y explique los tipos de relaciones que existen
6. ¿Para qué sirve la normalización?
7. ¿Qué es la integridad de los datos?
8. Construya la línea de tiempo la historia y las generaciones de las bases de datos sistematizadas.

Entregable Informe con Documento de texto con la solución a los interrogantes planteados

3.3. Actividades de apropiación del conocimiento (Conceptualización y Teorización).

Actividad 2. Construir Instrucciones SQL para solucionar requerimientos del cliente
Tipo de Actividad: Individual
Tiempo estimado para la actividad: 50 horas

Para esta actividad nos centraremos en el documento “ENUNCIADOS.doc”, recuerde que dicho documento contiene diversos casos empresariales que requieren ser solucionados, para esta actividad su instructor asignará gradualmente los enunciados para que sean realizadas las siguientes actividades:

- Elaboración del DER, (Diagrama Entidad Relación), estos diagramas deben encontrarse en un UNICO documento de texto.
- DDL: Bloc de notas por cada enunciado desarrollado, con el código SQL utilizado para crear la Base de datos y las respectivas tablas
- DML: En los blocs de notas construidos anteriormente, incluir las instrucciones de inserción de TRES registros por tabla(entidad) y las instrucciones construidas para dar solución a las consultas planteadas por su instructor, es decir que por cada enunciado asignado por su instructor se tendrá un documento txt con la creación de las entidades y la inserción de los registros respectivos.

Actividad 3. Construir consultas SQL a partir de enunciados dados
Tipo de Actividad: Individual
Tiempo estimado para la actividad: 20 horas

Lo invito a que revise el documento adjunto denominado “LaboratorioSQL.doc” donde encontrará diversas actividades, las cuales le permitirán construir sentencias SQL para la definición y manipulación del modelo de base de datos.

3.4. Actividades de transferencia del conocimiento.

Actividad 4. Solucionar requerimientos codificando instrucciones SQL
Tipo de Actividad: Individual
Tiempo estimado para la actividad: 15 horas

Estas dos actividades se desarrollarán con control de tiempo simulando una competencia, se plantea la asignación de cada uno de los ítems con periodos específicos de tiempo, al finalizar el primer ítem se valora el resultado y se procede a realizar la siguiente actividad y así sucesivamente.

1. Realice las actividades propuestas en el documento denominado “Actividad3.doc”, ubicado en el espacio actividades.
2. Realice las actividades propuestas en el documento denominado “Actividad4.doc”, ubicado en el espacio actividades.

Actividad 5. Aplicar DML Y DDL al Caso de estudio Alquiler de Motocicletas
Tipo de Actividad: Por equipos de Desarrollo
Tiempo estimado para la actividad: 50 horas

Tomando como referencia el documento denominado “CasodeEstudioAlquilerMotocicletas” ubicado en la carpeta Actividades/ 4_Transferencia, realicen las siguientes actividades en documento de texto:

1. Obtención de los datos Relevantes
2. Elaboración del diagrama Entidad Relación
3. Elaboración del Diagrama Entidad Relación
4. Relación de las tablas con sus respectivos campos
5. Implementación de la base de datos y tablas en MYSQL
6. Implementación del proyecto, colección y documentos en firebase

4. PLANTEAMIENTO DE EVIDENCIAS DE APRENDIZAJE PARA LA EVALUACIÓN EN EL PROCESO FORMATIVO.

Fase del proyecto formativo	Actividad del proyecto formativo	Actividad de Aprendizaje	Evidencias de Aprendizaje	Criterios de Evaluación	Técnicas e Instrumentos de Evaluación
			Evidencias de Conocimiento:	Crea bases de datos teniendo en cuenta	**Lista de verificación**

			Fundamentar conceptos para diferenciar Bases de Datos Relacionales y no Relacionales: Documento de texto con la solución a los interrogantes planteados **Evidencias de Desempeño** **Solucionar requerimientos codificando instrucciones SQL:** Documento de Word incluya la instrucción utilizada junto con la respuesta arrojada por el sistema. **Evidencias de Producto:** **Construir Instrucciones SQL para solucionar requerimientos del cliente:** UNICO documento de texto con el DER, (Diagrama Entidad Relación) de los requerimientos planteados. Bloc de notas por cada requerimiento planteado con las instrucciones DDL y DML utilizadas. **Construir consultas SQL a partir de enunciados dados:** Documento de texto con nombre “Solucion_Laboratorio_SQL ” y Archivo txt con el nombre “Script_Laboratorio_SQL”. **Aplicar DML Y DDL al Caso de estudio Alquiler de Motocicletas:** Documento de texto con la solución a las actividades planteadas	lenguaje de definición de datos, requerimientos del sistema y tecnología seleccionada. Realiza operaciones sobre los objetos de la base de datos aplicando las instrucciones SQL.	

5. GLOSARIO DE TÉRMINOS

Administración de Base de Datos Relacional: Técnica de administración de bases de datos en la que la información se guarda en tablas de datos bidimensionales. El programa que funciona con dos tablas de datos al mismo tiempo relaciona la información mediante vinculaciones establecidas por una columna o campos comunes.

Acceso: La manera en la cual los archivos o conjunto de datos son referenciados por la computadora.

Base de Datos: Es una serie de datos organizados y relacionados entre sí, los cuales son recolectados y explotados por los Sistemas de Información de una empresa o negocio en particular.

Campo o atributo: Unidad básica de una base de datos, un campo puede ser, por ejemplo, el nombre de una persona. Los nombres de los campos no pueden empezar con espacios en blanco y caracteres especiales. No pueden llevar puntos, ni signos de exclamación o corchetes. Si pueden tener espacios en blanco en el medio.

Clave candidata: Conjunto de atributos de una tabla que identifican unívocamente cada tupla de la tabla.

Clave primaria: Clave candidata que se escoge como identificador de las tuplas.

Clave alternativa: Cualquier clave candidata que no sea primaria.

Clave externa o secundaria: Atributo de una tabla relacionado con una clave de otra tabla

Datos estadísticos: estos almacenan información estadística sobre los datos en la base de datos.

Dominio: El dominio de un campo es el conjunto de valores aceptables que puede contener. Por ejemplo, el dominio del campo sexo estará ajustado a los dos valores que puede contener este dato, es decir, femenino y masculino.

Esquema: Es la aplicación de un Modelo de Datos a la realidad, esa representación puede ser en forma gráfica o lingüística y va a representar los objetos, sus interrelaciones, las operaciones que se pueden realizar sobre ellos y las restricciones que afecten al sistema. Resume las características más importantes del sistema de información para el objetivo que se persigue.

Entidad: Es la representación de un objeto o concepto del mundo real que se describe en una base de datos.

Foreing Key: En el contexto de bases de datos relacionales una clave foránea o clave ajena (o foreing key FK) es una limitación referencial entre dos tablas.

El DBMS: es un conjunto de programas que se encargan de manejar la creación y todos los accesos a las bases de datos.

ELIMINACION: Es una solicitud de eliminación que se expresa de forma muy parecida a una consulta. Sin embargo, en vez de presentar tuplas al usuario, quitamos las tuplas seleccionadas de la base de datos. Sólo puede eliminar tuplas completas; no se puede eliminar únicamente valores de determinados atributos.

Facilidad de Consultas: Permitir al usuario hacer cuestiones sencillas a la base de datos. Este tipo de consultas tienen como misión proporcionar la información solicitada por el usuario de una forma correcta y rápida.

Gestor de Base de Datos: Es un conjunto de programas que permiten crear y mantener una base de datos, asegurando su integridad, confidencialidad y seguridad

Independencia de los datos: Se refiere a la protección contra los programas de aplicaciones que pueden originar modificaciones cuando se altera la organización física y lógica de las bases de datos.

La manipulación de datos: la manipulación de datos es la recuperación de información almacenada en la base de datos

Lenguaje de consulta: Son los lenguajes en el que los usuarios solicitan información de la base de datos. Estos lenguajes son generalmente de más alto nivel que los lenguajes de programación. Los lenguajes de consulta pueden clasificarse como procedimentales y no procedimentales

Manipulación de Base de Datos: Usando la base de Datos -- el usuario puede añadir, borrar y modificar información a la base de datos así como también hacer consultas

Modelo de base de datos orientado a objetos: es una adaptación a los sistemas de bases de datos. Se basa en el concepto de encapsulamiento de datos y código que opera sobre estos en un objeto.

Modelos de Red: Este modelo permite la representación de muchos a muchos de una Base de Datos. El modelo de red evita redundancia en la información, a través de la incorporación de un tipo de registro denominado el conector.

Nivel lógico: definición de las estructuras de datos que constituyen la base de datos

Reglas de Integridad: Son restricciones que definen los estados de consistencias de las bases de datos.

DBMS: Es un conjunto de programas que se encarga de manejar la cremación y todos los acceso a las bases de datos.

DDL: Es un lenguaje de definición de datos

Recuperación: Proporcionar como mínimo el mismo nivel de recuperación que los sistemas de bases de datos actuales. De forma que, tanto en caso de fallo de hardware como de fallo de software, el sistema pueda retroceder hasta un estado coherente de los datos.

Relaciones: Las relaciones son conexiones lógicas que existen entre las tablas y los atributos y se pueden clasificar por diferentes criterios.

Sistema de Administración de Base de Datos: Es el software que controla la organización, almacenamiento, recuperación, seguridad e integridad de los datos en una base de datos

SISTEMA GESTOR DE BASE DE DATOS: Es un conjunto de programas que permiten crear y mantener una base de datos, asegurando su integridad, confidencialidad y seguridad.

Transacción: es una secuencia de operaciones de acceso a la base de datos que constituye una unidad lógica de ejecución.

Transacciones compartidas: Las transacciones compartidas soportan grupos de usuarios en estaciones de trabajo, los cuales desean coordinar sus esfuerzos en tiempo real, los usuarios pueden compartir los resultados intermedios de una base de datos. La transacción compartida permite que varias personas intervengan en una sola transacción.

Tuplas: Es la representación de una fila en una de las tablas que se está almacenando datos. Y las cuales serán llamadas por los administradores de Base de Datos en el tiempo de ejecución de un sistema.

Usuario final: es quien accesa a las bases de datos por medio de un lenguaje de consulta o de programas de aplicación.

6. REFERENTES BILBIOGRÁFICOS

- Abraham Silberschatz, et. Al. Fundamentos de Bases de Datos. 4ª Edición. 2004.
- Sánchez, J. Principios sobre bases de datos. http://www.jorgesanchez.net/bd/bdrelacional.pdf
- Pons, O. et al. Introducción a los sistemas de bases de datos. Paraninfo. 2008.
- Elmasri, R., Navathe, S.B., Fundamentos de Sistemas de Bases de Datos, 5ª edición, Pearson Education, 2008.
- https://dev.mysql.com/doc/refman/8.0/en/
- Reglas de normalización: http://www.youtube.com/watch?v=p_UGm8UYaew
- http://www.youtube.com/watch?v=n8PMCxxuXr4
- http://www.youtube.com/watch?v=TSeN_BtuEKM
- Craig Larman. UML y patrones, editorial Prentice Hall, 2007.
- G. Booch, J. Rumbaugh, I. Jacobson. El lenguaje unificado de modelado: manual de referencia. Addison-Wesley, 1999.
- James Rumbaugh. Object Oriented Modeling and Design. Pearson Education, 2002
- Schmuller, Joseph. Aprendiendo UML en 24 horas. Prentice Hall
- Scott W. Ambler. Introducción a los Objetos, 3ra. edición, 2003
  Disponible en la Biblioteca digital SENA (Solo está autorizado y disponible desde la red del SENA)
- Kimmel, Paul. Manual de UML. Mexico: McGraw-Hill España, 2010. p i.
- Ejercicios Realización de consultas SQL . https://josejuansanchez.org/bd/ejercicios-consultas-sql/

7. CONTROL DEL DOCUMENTO

	Nombre	Cargo	Dependencia	Fecha
Autor (es)	**Magda Milena García Gamboa**	**Instructor**	**CSET-Regional Santander**	**Noviembre de 2022**

8. CONTROL DE CAMBIOS (diligenciar únicamente si realiza ajustes a la guía)

	Nombre	Cargo	Dependencia	Fecha	Razón del Cambio
Autor (es)					

PROCESO DE GESTIÓN DE FORMACIÓN PROFESIONAL INTEGRAL

FORMATO GUÍA DE APRENDIZAJE

1. IDENTIFICACIÓN DE LA GUIA DE APRENDIZAJE

- Denominación del Programa de Formación: Análisis y desarrollo de software.
- Código del Programa de Formación: 228118
- Nombre del Proyecto: Construcción de software integrador de tecnologías orientadas a servicios.
- Fase del Proyecto: Ejecución
- Actividad de Proyecto: Codificar los módulos del software.
- Competencia: Evaluar requisitos de la solución de software de acuerdo con metodologías de análisis y estándares.
- Resultados de Aprendizaje Alcanzar: Desarrollar procesos lógicos a través de la implementación de algoritmos.
- Duración de la Guía: 80 horas

  Formación directa: 60 horas

  Trabajo Autónomo: 20 Horas

2. PRESENTACIÓN

En el mundo actual, donde la información y la tecnología son fundamentales, la capacidad de desarrollar procesos lógicos eficientes es crucial. Los algoritmos, como herramientas para resolver problemas, juegan un papel fundamental en este proceso. En esta presentación, exploraremos cómo desarrollar procesos lógicos a través de la implementación de algoritmos y cómo esta habilidad puede ser beneficiosa en diversos campos.

Esta guía le permitirá al aprendiz introducirse en el mundo de la programación a través de una serie de conocimientos que les darán las herramientas necesarias para obtener los procesos mentales adecuados para aprender a programar. Es muy importante que los aprendices realicen a conciencia las actividades propuestas para que puedan adquirir los conocimientos y la lógica necesaria.

Por consiguiente, se presentan cada una de las acciones de aprendizaje que le permitirán desarrollar lo anteriormente mencionado.

3. FORMULACIÓN DE LAS ACTIVIDADES DE APRENDIZAJE

En este apartado se describirán las actividades de aprendizaje para cada una de las competencias que plantea la fase de ejecución del proyecto formativo: construcción de software integrador de tecnologías orientadas a servicios.

3.1 3.1 Actividades de reflexión inicial: Analizar los elementos básicos de la lógica y la construcción de algoritmos

Antes de iniciar las actividades propuestas, conforme grupos, máximo de 3 personas.

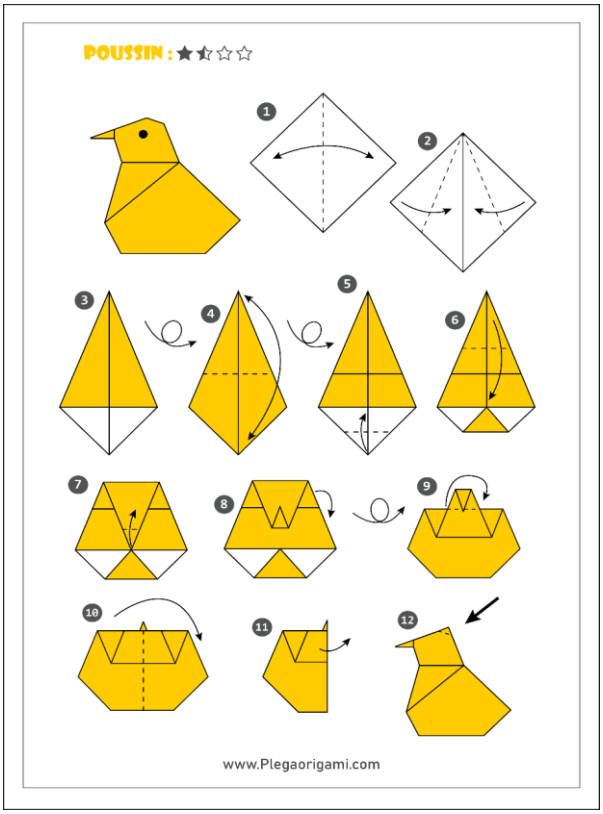
Materiales: Hojas de papel, lápices, esferos, tablero, marcadores

Detengamos el tiempo por un segundo y recordemos cómo hacemos figuras de origami. Para ello se sugiere elaborar una pollito. Sigan los pasos que se les indican a continuación:

Paso 1. Con la hoja de papel realice la figura propuesta

Paso 2. Junto a sus compañeros de grupo escriba en una hoja como sería el paso a paso para elaborar la figura de un animal en papel.

Paso 3. ¡Ahora reflexionemos! ¿Qué pasaría si cambiamos los materiales o la forma del animal? ¿Cambiaría el paso a paso



Escriban sus conclusiones:
Algo que para algunos es sencillo, puede que para otros sea difícil. En la vida diaria encontraremos una serie de actividades que requieren de pasos para su solución. ¿Lo has notado?
descrito?

3.2 Actividades de contextualización e identificación de conocimientos necesarios para el aprendizaje:

Duración: 2 horas.

Después de la explicación del instructor, me reúno en parejas y consulto información para las siguientes actividades de investigación:

Historia de la Lógica Proposicional:

Investigar y presentar una breve historia de la lógica proposicional, destacando sus orígenes y su evolución a lo largo del tiempo.

- Aplicaciones de la Lógica Proposicional en la Vida Real:

Identificar ejemplos de aplicación de la lógica proposicional en la vida cotidiana, como sistemas de seguridad, electrónica digital, inteligencia artificial, entre otros.

- Importancia de los Algoritmos:

Discutir la importancia de los algoritmos en la resolución de problemas, desde aplicaciones simples hasta complejas, y su impacto en diferentes áreas, como la informática, las finanzas y la medicina.

- **Evidencia de conocimiento:**

Elementos para tener en cuenta en el documento:

- Se deben seguir las normas básicas de presentación de un documento escrito, es decir el documento debe tener como mínimo una portada, introducción, objetivos.

Lineamientos generales para la entrega de la evidencia:

- Productos para entregar: documento tipo informe
- Extensión: Libre.
- Formato: PDF.

3.3 Actividades de apropiación:

En esta actividad usted encontrará 4 actividades de Aprendizaje, desglosadas cada una de ellas en subactividades. Desarrolle cada una de ellas, evidenciando lo propuesto.

3.3.1 Actividad de Aprendizaje 1: Identificar los conocimientos necesarios para elaborar algoritmos básicos

Siga las indicaciones que se presentan a continuación:

1. Conforme un equipo de trabajo como máximo de 4 personas.

2. Elegir a quien llevará la venda, es decir, el que hará el papel de gallinita ciega.

3. El compañero(a), una vez elegido debe ponerse un pañuelo en los ojos, de forma que no pueda ver nada.

4. Los compañeros(as) que pueden ver deberán guiar al compañero a algún lugar de la sede, indicado por el instructor.

Nota: Tenga en cuenta que su compañero no podrá ver por ende deberá guiarlo paso a paso hasta el lugar de destino.

5. Comparta con sus compañeros la experiencia de guiar y ser guiado.

6. A continuación, observe el siguiente video: que_es_un_algoritmo.mp4 https://acortar.link/ez1ieR contenido en la carpeta de material de apoyo.

7. Comparta y socialice con sus compañeros de clase si lo que hicieron al guiar al compañero de la venda, tiene algo que ver con lo visto en el video.

3.3.2 Actividad de Aprendizaje 2: Identificar los conocimientos básicos necesarios para elaborar proposiciones simples y compuestas

1. Observen el video proposiciones y operaciones básicas.mp4, que se encuentra en el material de apoyo.
2. En grupos de dos personas realicen la lectura del documento denominado **lectura_introducción_a_la_lógica**, que se encuentra en el material de apoyo.
3. Elaboren en su cuaderno 10 enunciados diferentes y traten de observar si son o no proposiciones. Clasifíquenlas en compuestas o simples al frente de cada enunciado.
4. Socialicen con los demás grupos los enunciados trabajados y los resultados obtenidos.

3.3.3 Actividad de Aprendizaje 3: Conceptos claves para resolver problemas lógicos proposicionales.

En un curso de lógica proposicional, los estudiantes enfrentan dificultades para resolver problemas complejos que involucran la construcción de tablas de verdad, la simplificación de expresiones

lógicas y la demostración de teoremas. En grupo de trabajo realizaremos un foro con las siguientes preguntas:

1. Define qué es una proposición y proporciona un ejemplo.
2. Explica la diferencia entre una proposición simple y una proposición compuesta.
3. ¿Qué es una tabla de verdad y cómo se utiliza para evaluar proposiciones compuestas?
4. Describe qué es una tautología y da un ejemplo.
5. ¿Qué es una contradicción en el contexto de la lógica proposicional? Proporciona un ejemplo.
6. Define qué es una implicación lógica y proporciona un ejemplo.
7. ¿Qué son las leyes de De Morgan en lógica proposicional? Proporciona las expresiones equivalentes.
8. ¿Cuál es la diferencia entre una implicación material y una equivalencia lógica?
9. Explica qué significa que dos proposiciones sean equivalentes en el contexto de la lógica proposicional.
10. ¿Qué son las formas normales en lógica proposicional y cuáles son sus tipos principales

3.3.4 Actividad de Aprendizaje 4: Problemas lógicos proposicionales Simboliza los siguientes enunciados

Dadas las siguientes proposiciones:

a) No vi la película, pero leí la novela
b) Ni vi la película ni leí la novela
c) No es cierto que viese la película y leyese la novela
d) Vi la película, aunque no leí la novela
e) No me gusta trasnochar ni madrugar
f) tú estás equivocado o es falsa la noticia que has leído
g) Si no estuvieras loca, no habrías venido aquí
h) Llueve y o bien nieva o sopla el viento
i) está lloviendo y nevando o está soplando el viento
j) Si hay verdadera democracia, entonces no hay detenciones arbitrarias ni otras violaciones de los derechos civiles.
k) Roberto hará el doctorado cuando y solamente cuando obtenga la licenciatura
l) Si viene en tren, llegará antes de las seis. Si viene en coche, llegará antes de las seis. Luego, tanto si viene en tren como si viene en coche, llegará antes de las seis.

Enlaza cada proposición con su formalización

“Llueve” = p, “Hace sol” = q

1	Llueve y hace sol	5A	¬ p
2	Llueve y no hace sol	3B	p ∨ q
3	Llueve o hace sol	1C	p ∧ q
4	Si no llueve, hace sol	2D	p ∧ ¬ q
5	No es cierto que llueva	6E	¬ ¬ q
6	No es cierto que no llueva	7F	q ↔ ¬p
7	Hará sol si y solo si llueve	4G	¬p → q

Construye las tablas de verdad de:

a. ¬p ∧ q
b. ¬p ∧ ¬q
c. (p ∨ ¬q) ∨ p
d. (p → q) ∧ p
e. [¬(p → q) ∨ (p ↔ q)] ∧ [(¬p → q) ∨ ¬p]

3.3.5 Actividad Aprendizaje 5: Identificar los conocimientos necesarios para elaborar algoritmos básicos

Para elaborar esta actividad, deben estar atentos a la explicación de su instructor y seguir los siguientes pasos:

• Observar atentos el video que es un algoritmo y lógica de programación estructura selectiva

• Luego, según lo visto en los videos, identifiquen cada una de las figuras y describan su utilidad o funcionamiento, señalen en qué parte de un diagrama de flujo se pueden utilizar. Escriban en la misma hoja.

• A continuación, con el grupo completo y haciendo una lluvia de ideas, piensen en una acción de la vida cotidiana. Ejemplo, pasos para cambiar un foco (bombillo) y utilizan las figuras dibujadas en el papel.

3.3.6 Actividad de Aprendizaje 6: Problemas Algoritmos

Realice la actividad de manera individual propuesta en los documentos de la unidad en el siguiente orden

- 6.1 Ejercicios Pseudocódigo-diagramas de flujo.pdf
- 6.2 Estructuras de control flowgorithm.pdf
- 6.3 Ciclos flowgorithm.pdf
- 6.4 Arrays flowgorithm.pdf
- 6.5 Funciones y Procedimientos NetBeans.pdf

3.4 Actividades de Transferencia el Conocimiento:

Actividad de Aprendizaje: Cuestionario y problemas de Lógica de Programación y Algoritmos para validar los conocimientos adquiridos por los aprendices.

4. PLANTEAMIENTO DE EVIDENCIAS DE APRENDIZAJE PARA LA EVALUACIÓN EN EL PROCESO FORMATIVO.

Fase del proyecto formativo	Actividad del proyecto formativo	Actividad de Aprendizaje	Evidencias de Aprendizaje	Criterios de Evaluación	Técnicas e Instrumentos de Evaluación
			Evidencia de producto: Taller de estructuras de control en la solución de algoritmos. GA3-220501096-AA1-EV01	Emplea lenguaje de programación para la solución de problemas de lógica proposicional según el análisis realizado.	Lista de chequeo. IE-GA3-220501096-AA1-EV01.
			Evidencia de producto: Taller de	Emplea lenguaje de programación para la solución de problemas de lógica proposicional	Lista de chequeo. IE- GA3-2205010

			funciones y procedimientos en la solución de algoritmos. GA3-220501096-AA1-EV02	según el análisis realizado.	96-AA1-EV02

5. GLOSARIO DE TÉRMINOS

Lógica Proposicional: Parte de la lógica que estudia las proposiciones simples y las operaciones que se pueden realizar sobre ellas (como la conjunción, la disyunción y la negación).

Algoritmo: Conjunto finito de instrucciones precisas y ordenadas que permiten llevar a cabo una actividad a resolver un problema.

Variable: Un símbolo que puede representar cualquier valor o elemento dentro de un conjunto dado.

Condición: Una expresión lógica cuyo valor determina la ejecución de ciertas partes de un algoritmo.

Ciclo/ Bucle: Una estructura de control que permite repetir un conjunto de instrucciones hasta que se cumpla una condición específica.

Condición IF: Una estructura de control que ejecuta cierto bloque de código si se cumple una condición determinada.

Ciclo While: Un bucle que repite un bloque de código mientras una condición específica sea verdadera.

Ciclo For: Un bucle que itera sobre una secuencia de elementos (como una lista o rango) y ejecuta un bloque de código para cada elemento.

Recursión: Un proceso en el que una función se llama a sí misma directa o indirectamente.

Estructura de datos: Forma de organizar y almacenar datos para que puedan ser accedidos y utilizados de manera eficiente.

Pila (Stack): Una estructura de datos que sigue el principio de "último en entrar, primero en salir" (LIFO).

Cola (Queue): Una estructura de datos que sigue el principio de "primero en entrar, primero en salir" (FIFO).

Árbol: Una estructura de datos jerárquica compuesta por nodos conectados entre sí mediante aristas.

Grafo: Una estructura de datos compuesta por un conjunto de nodos (vértices) y un conjunto de aristas que conectan esos nodos.

Complejidad algorítmica: Medida del tiempo y/o espacio necesario para ejecutar un algoritmo en función del tamaño de la entrada.

Orden de complejidad: Una función que describe cómo aumenta el tiempo y/o el espacio requerido por un algoritmo a medida que aumenta el tamaño de la entrada.

Big O Notation: Una notación utilizada para describir el orden de complejidad de un algoritmo en términos asintóticos.

Programación Dinámica: Una técnica para resolver problemas computacionales dividiéndolos en subproblemas más pequeños y resolviéndolos de manera recursiva.

Divide y Vencerás: Un paradigma de diseño de algoritmos que consiste en resolver un problema dividiéndolo en subproblemas más pequeños, resolviendo estos subproblemas de manera independiente y luego combinando sus soluciones.

Heurística: Una técnica que proporciona una solución aproximada a un problema cuando no es posible encontrar una solución óptima en un tiempo razonable.

6. REFERENTES BILBIOGRÁFICOS

Java en Castellano. (2021, Dic 01). Servlets y JSP.

https://programacion.net/articulo/servlets_y_jsp_82 Maida, E. G., & Pacienzia, J. (2015). Metodologías de desarrollo de software. Quintas, A.

F. (2000). *Java 2: manual de usuario y tutorial*. Grupo Editorial RA-MA. Sun. (2021, Dic 01). Java. http://java.sun.com/

Vall Mainou, A. (2010). Desarrollo para internet con tecnología JAVA.

7. CONTROL DEL DOCUMENTO

	Nombre	Cargo	Dependencia	Fecha
Autor (es)				

8. CONTROL DE CAMBIOS (diligenciar únicamente si realiza ajustes a la guía)

	Nombre	Cargo	Dependencia	Fecha	Razón del Cambio
Autor (es)					

Introducción a la base de datos MySQL®

FAVA - Formación en Ambientes Virtuales de Aprendizaje

SENA - Servicio Nacional de Aprendizaje.

ESTRUCTURA DE CONTENIDOS

Pág.
Estructura de contenidos....2
Mapa de contenido....4
Desarrollo de contenidos....5
1. Generalidades de MySQL. 5
1.1. Componentes de MySQL. 5
1.2 Motores de almacenamiento de datos. 6
1.3. La consola de MySQL. 7
1.4 Tipos de datos. 8
3. Creación de la estructura de almacenamiento. 14
3.1. Creación de la base de datos. 15
3.2. Creación de las tablas. 16
4. Modi旳椀cación de la estructura de las tablas. 21
5. Actualización e inserción de registros. 25
5.1. Inserción de registros. 25
5.1.2. Inserción de registros con llaves foráneas. 27
6. Consultas de registros. 29
6.1. Datos de prueba. 29
6.2. Consultas básicas. 30
6.3. Aplicación de 旳椀ltros a las consultas. 33
6.4. Ordenamiento de las consultas. 34
6.5. Introducción a las consultas multi-tablas 35
6.6. Subconsultas. 37
Glosario 40
Bibliografía 41
Control del documento 42

INTRODUCCIÓN A LA BASE DE DATOS MySQL

INTRODUCCIÓN

MySQL es un sistema manejador de bases de datos de libre uso y distribución bajo licencia GPL de los más utilizados y que está disponible para varios sistemas operativos (DUBOIS, 2009).

Su popularidad se debe principalmente a su licencia libre y a su facilidad de uso y administración. Por otra parte ha sido integrada con otras herramientas libres como son Linux, Apache, PHP, entre otras. Esta combinación e integración de tecnologías dio nombre a la plataforma de desarrollo conocida como LAMP (Linux, Apache, MySQL y PHP).

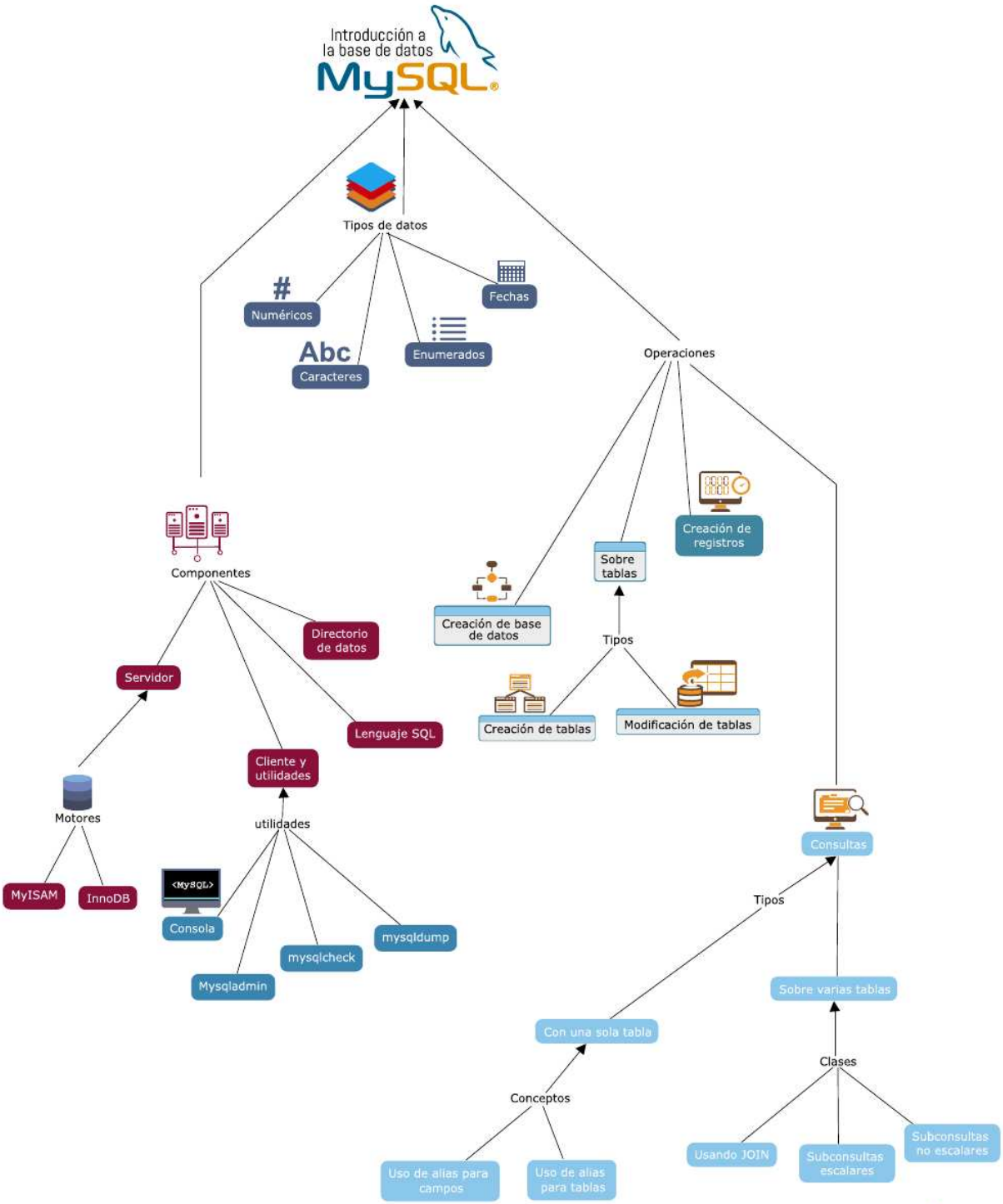
En años recientes MySQL fue adquirida por Sun Microsystems que luego fue adquirida por Oracle. Lo anterior no signi旳椀ca que Mysql deje de ser libre ya que su desarrollo está bajo la licencia GPL. Sin embargo, Oracle ofrece otras versiones de MySQL orientadas a empresas con modelo de licenciamiento de pago.

Para una mejor compresión del recurso es necesario que el aprendiz haya estudiado los recursos de introducción al lenguaje SQL de la actividad de proyecto 6.

Para el desarrollo de este recurso se requiere que se tenga instalado el MySQL en el computador del aprendiz. Para apoyar la instalación existe un video tutorial en el área de materiales de la actividad de proyecto 6.



MAPA DE CONTENIDO



DESARROLLO DE CONTENIDOS

1. Generalidades de MySQL.

MySQL como manejador de bases de datos (SGBD) presenta las siguientes características (DUBOIS,2009):



Velocidad
MySQL es veloz comparado con la mayoría de las bases libres.



Facilidad de uso
MySQL es de alto desempeño pero a la vez fácil de usar.



Soporta el lenguaje SQL
MySQL soporta el lenguaje estructurado de consultas (SQL).



Robustez
El servidor MySQL es multi-hilo y puede atender varios usuarios de manera simultánea.



Portabilidad
MySQL corre en muchos sistemas operativos entre ellos windows, linux, unix.



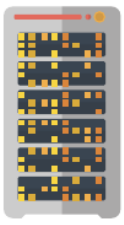
Conectividad
MySQL soporta distintos esquemas de conectividad y las bases de datos pueden ser accedidas desde cualquier sitio de Internet.



Seguridad
MySQL maneja esquemas de seguridad que permiten asignar permisos a nivel de usuario.

1.1. Componentes de MySQL.

Los componentes de MySQL se pueden dividir en cuatro (DUBOIS, 2009):



El servidor.

El programa principal que corre como un servicio tanto en plataformas Linux como windows y es quien coordina todas las operaciones del manejador.



El cliente y las utilidades.

MySQL suministra varias utilidades a saber:

- **MySQL:** Un programa interactivo que permite enviar consultas SQL al servidor. Requiere se puede llamar la consola.
- **mysqladmin:** Un programa administrativo para realizar tareas como subir y bajar el servidor, cambiar la con旳椀guración del servidor, monitorear el estado del mismo, entre otras.
- **mysqldump:** herramienta para hacer copias de seguridad o copiar datos a otros servidores.
- **mysqlcheck:** herramienta para realizar el chequeo, análisis y optimización de la base de datos.



Lenguaje SQL del servidor.

Implementación del lenguaje estándar SQL dentro del servidor.



El directorio de datos.

Es la ubicación donde se almacenan las bases de datos gestionadas por MySQL.

1.2 Motores de almacenamiento de datos.

Muchas de las características de las bases de datos están provistas por el motor de almacenamiento. MySQL trae varios motores pero los más usados son MyISAM e InnoDB (DUBOIS,2009).

1.2.1. Motor MyISAM.

Este es el motor por defecto usado por MySQL y representa cada tabla por medio de tres archivos en el sistema de archivos. Cada archivo tiene un nombre base que es el mismo que el de la tabla y una extensión que indica su función. Por ejemplo para una tabla llamada “clientes” existen los siguientes archivos:

a. **clientes.frm:** es el archivo de formato que contiene la de旳椀nición de la estructura de la tabla.
b. **clientes.MYD:** es el archivo que contiene los datos de la tabla.
c. **clientes.MYI:** es el archivo que contiene los índices de la tabla.

1.2.2. Motor InnoDB.

Este motor de almacenamiento está pensado para bases de datos transaccionales, es decir, que realizan operaciones que requieren ser tratadas como transacciones tipo CRUD (en inglés ACID).

Almacena las tablas en el sistema operativo de la siguiente manera:

a. **Un archivo .frm:** cada tabla de InnoDB es representada por un archivo .frm que contiene la de旳椀nición de la estructura.
b. **Tablespace compartido:** consiste en uno o más archivos que contienen todos los datos de las tables de manera contigua. Por defecto InnoDB almacena los datos de manera contigua.
c. **Tablespace individual:** se puede con旳椀gurar InnoDB para que una tabla tenga un tablespace separado.

1.3. La consola de MySQL.

La consola es la utilidad que permite transmitir las instrucciones al servidor MySQL.

Para invocarla se usa la siguiente sintaxis:

```
$ mysql -h <nombre_del_host> -p -u <nombre_de_usuario>
```

Donde:

- -h <nombre del host>: indica a cual servidor o host conectarse.
- -u <nombre_de_usuario>: indica el nombre de usuario a conectar.
- -p: indica al MySQL que pida el password del usuario.

Una vez ejecutado el comando aparece lo siguiente en pantalla:

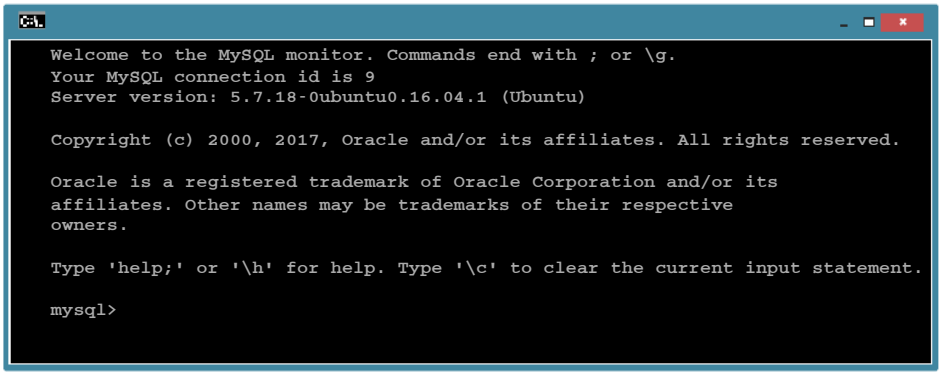


Figura 1.1. Consola de MySQL.

1.4 Tipos de datos.

1.4.1. Tipos Numéricos.

MySQL soporta todos los tipos de datos SQL numéricos estándar: los tipos de datos enteros y los tipos de datos en coma 旳甦otante.

Tipos de datos enteros.

TIPO	Bytes	Valor Mínimo (Con signo/Sin signo)	Valor Máximo (Con signo/Sin signo)
TINYINT	1	-128 0	127 255
BIT (BOOL,BOOLEAN)	Número entero con valor 0 o 1. Sinónimo de TINYINT(1)		
SMALLINT	2	-32768 0	32767 65535
MEDIUMINT	3	-8388608 0	8388607 16777215
INT	4	-2147483648 0	2147483647 4294967295
BIGINT	8	-9223372036854775808 0	9223372036854775807 18446744073709551615

Tabla 1.1. Tipos de datos enteros.

Tipos de datos en coma 旳甦otante.

Tipo	Tamaño
FLOAT (m,d)	Contiene un número en coma 旳甦otante de precisión sencilla. El valor M es la anchura a mostrar y D es el número de decimales.
DOUBLE (m,d)	Contiene un número en coma 旳甦otante de precisión doble. Igual que FLOAT la diferencia es el rango de valores posibles.
DECIMAL (m [,d])	Se usan para guardar valores para los que es importante preservar una precisión exacta, por ejemplo con datos monetarios. Ejemplo: salario DECIMAL(5,2) Si se omite D el valor por defecto es 0, los valores no tendrán punto decimal ni decimales.

Tabla 1.2. Tipos de datos en coma 旳甦otante.

Ejemplo:

Si en la base de Datos se requiere almacenar las edades de las personas, cuyo valor máximo será 100, la opción más adecuada para el tipo de dato sería “TINYINT”.

Si se requiere sistematizar las notas de un colegio y la nota de旳椀nitiva se debe dar en decimales y el valor máximo es 10.00, la opción más adecuada es “FLOAT”.

1.4.2. Tipos de cadenas de caracteres.

Listado de cada uno de los tipos de dato con formato string en MySQL, su ocupación en disco y valores.

Tipo	Tamaño	Sintaxis
CHAR (M)	Los valores válidos para M son de 0 a 255 caracteres. Contiene una cadena de longitud constante. Para mantener la longitud de la cadena, se rellena a la derecha con espacios. Estos espacios se eliminan al recuperar el valor.	PacIdenti旳椀cacion CHAR(10)
VARCHAR (M)	Los valores válidos para M son de 0 a 255 caracteres. Contiene una cadena de longitud variable. Los espacios al 旳椀nal se eliminan.	PacNombres VARCHAR(50)
BLOB	Una longitud máxima de 65.535 caracteres. Válido para objetos binarios como imágenes, 旳椀cheros de texto, audio o video.	PacImagenFoto BLOB
TEXT	Una longitud máxima de 65.535 caracteres. Sirve para almacenar texto plano sin formato. Distingue entre minúsculas y mayúsculas.	PacDescripcion TEXT
TINYBLOB TINYTEXT	Longitud máxima de 255 caracteres.	
MEDIUMBLOB MEDIUMTEXT	Longitud máxima de 16777215 caracteres	
LONGBLOB LONGTEXT	Longitud máxima de 4294967298 caracteres.	

SET	Contiene un conjunto.	SET('valor1','valor2',...)
ENUM	Un objeto de tipo cadena que puede tener cero o más valores, cada uno de los cuales debe estar entre una lista de valores 'valor1', 'valor2', ... Contiene un enumerado. Un objeto de tipo cadena que puede tener un único valor, entre una lista de valores 'valor1', 'valor2', ...,	ENUM('valor1','valor2',...)

Tabla 1.3. Tipos de datos de caracteres.

En tabla se observa que el campo PacIdenti的椀cacion se declaró como un "CHAR" de 10 caracteres porque los documentos de identidad tienen entre 1 y 10 caracteres.

Para almacenar nombres, direcciones e información con máximo de 100 caracteres se recomienda el tipo "VARCHAR". En la tabla anterior el campo PacNombres se declaró varchar(50) porque los nombres tienen una cantidad de caracteres variable.

Para almacenar grandes cantidades de caracteres como descripciones, observaciones, comentarios en este caso se tomaría TEXT o BLOB.

1.4.3. Tipos de fecha y hora.

Los tipos de fecha y hora para representar valores temporales son:

TIPO	RANGO	FORMATO
DATE	Válido para almacenar una fecha con año, mes y día. Su rango oscila entre: '1000-01-01′ y '9999-12-31′.	AAAA-MM-DD
DATETIME	Almacena una fecha y una hora. Su rango oscila entre '1000-01-01 00:00:00′ y '9999-12-31 23:59:59′.	AAAA-MM-DD HH:MM:SS
TIME	Una hora. El rango está entre '-838:59:59' y '838:59:59'.	HH:MM:SS
TIMESTAMP	Almacena una fecha y hora UTC. El rango está entre '1970-01-01 00:00:00' y algún momento del año 2037.	AAAA-MM-DD HH:MM:SS

YEAR (2|4)	Almacena un año dado con 2 ó 4 dígitos de longitud (por defecto son 4). El rango de valores oscila entre 1901 y 2155 con 4 dígitos. Mientras que con 2 dígitos el rango es desde 1970 a 2069 (70-69).	AA ó AAAA

Tabla 1.4. Tipos de datos fecha.

Ejemplo:

Para llevar el control de acceso (con horas, minutos y segundos) de los usuarios a un sistema de Información lo recomendable es tener un campo de tipo “DATETIME”.

1.4. Modi旳椀cadores.

Además de los tipos de datos requiere es necesario conocer algunos modi旳椀cadores que se utilizan para el manejo de los campos. Estos modi旳椀cadores se presentan a continuación:

MODIFICADOR	USO	TIPO DE CAMPO QUE APLICA
AUTO_INCREMENT	El valor se va incrementando automáticamente en cada registro (1,2,3,etc).	Enteros
DEFAULT	Coloca un valor por defecto (el valor se coloca justo detrás de esta palabra).	Todos excepto TEXT y BLOB
NOT NULL	Impide que un campo sea nulo.	Todos
PRIMARY KEY	Hace que el campo se considere llave primaria.	Todos
UNIQUE	65565 bytes. Evita la repetición de valores.	Todos

Tabla 1.5. Modi旳椀cadores.

2. Base de datos didáctica.

En este recurso se presenta cómo hacer uso del motor de Base de Datos MySQL para crear una base de datos. Se utilizará para esto la base de datos “Citas” compuesta por cinco tablas: Pacientes, Medicos, Consultorios, Citas y Tratamientos, como se presenta en la 旳椀gura 2.1.

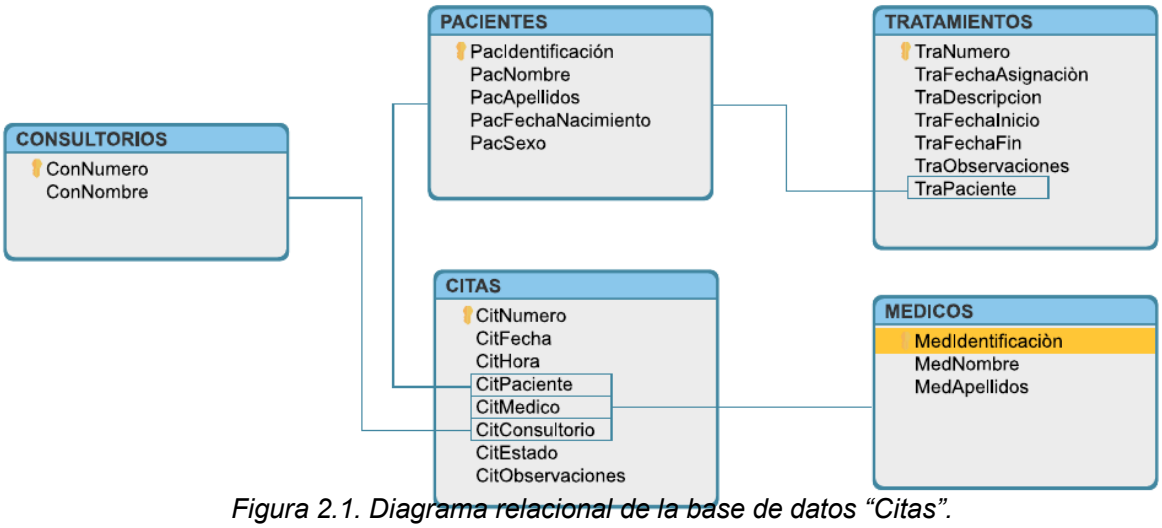


Figura 2.1. Diagrama relacional de la base de datos "Citas".

Se considera importante presentar la estructura de la base de datos detallando las tablas, tipos de datos de los campos y modi旳椀cadores a utilizar. Con el 旳椀n de que pueda proceder a su creación usando SQL como Lenguaje de De旳椀nición de Datos (DDL).

2.1. Tabla "Pacientes".

Tabla "Pacientes"				
Atributo - Campo	Tipo de Dato	Long.	Modificador	Tabla y Campo Foráneo
PacIdentificacion	Char	10	Primary key, not null	
PacNombres	Varchar	50	not null	
PacApellidos	Varchar	50	not null	
PacFechaNacimiento	Date		not null	
PacSexo	Por tener el Modificador enum, no se declara		(ENUM('M,'F'))	

Figura 2.2. De旳椀nición de la tabla Pacientes.

2.2. Tabla “Medicos”.

Tabla "Médicos"				
Atributo - Campo	Tipo de Dato	Long.	Modificador	Tabla y Campo
MedIdentificacion	Char	10	Primary key, not null	
MedNombres	Varchar	50	not null	
MedApellidos	Varchar	50	not null	

2.3. Tabla “Consultorios”.

Tabla “Consultorios”				
Atributo - Campo	Tipo de Dato	Long.	Modificador	Tabla y Campo Foráneo
ConNumero	Int	3	Primary key, not null	
ConNombre	Varchar	50	not null	

Figura 2.4. Definición de la tabla Consultorios.

2.4. Tabla “Tratamientos”.

Tabla “Tratamientos”				
Atributo - Campo	Tipo de Dato	Long.	Modificador	Tabla y Campo Foráneo
TraNumero	Int		Primary key, not null auto_increment	
TraFechaAsignado	Date		not null	
TraDescripcion	Text		not null	
TraFechaInicio	Date		Primer key, not null	
TraFechaFin	Text		not null	
TraPacientes	Char	10	not null	Pacientes (PacientIdentificacion)

Figura 2.5. Definición de la tabla Tratamientos.

2.5. Tabla “Citas”.

Tabla “Citas”				
Atributo - Campo	Tipo de Dato	Long.	Modificador	Tabla y Campo Foráneo
CitNumero	Int		Primary key, auto_increment	
CitFecha	Date		not null	
CitHora	Varchar	10	not null	
CitPaciente	Char	10	not null	Pacientes (PacientIdentificacion)
CitMedico	Char	10	not null	Medicos(MedIdentificacion)
CitConsultorio	Int		not null	Consultorio (ConNumero)
CitEstado	Por tener el Modificador enum, no se declara		(ENUM('Asignada,'Cumplida')) DEFAULT "Asignada"	
CitObservaciones	Text		not null	

Figura 2.6. De旳椀nición de la tabla Citas.

3. Creación de la estructura de almacenamiento.

Para iniciar el proceso de de旳椀nición de datos en MySQL se debe realizar el siguiente procedimiento:

a) Abrir el bloc de notas del equipo para digitar cada una de las instrucciones del Lenguaje de De旳椀nición de Datos, esto con el 旳椀n de ir construyendo el script completo de creación de la base de datos.

b) Iniciar la consola de MySQL como se explica en videotutorial sobre instalación de MySQL.

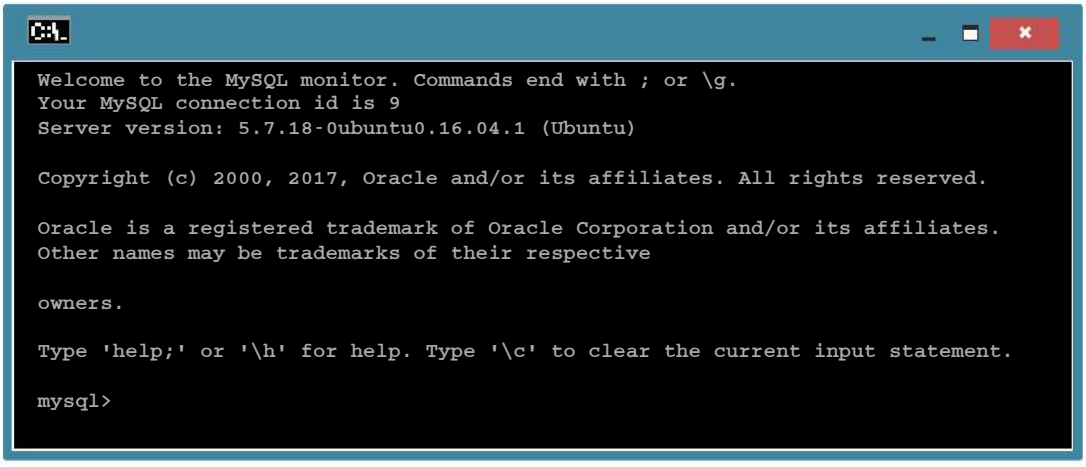


Figura 3.1.Consola MySQL lista para recibir instrucciones.

c) Escribir o pegar las instrucciones en la consola. Las instrucciones deben terminar con punto y coma para que se ejecuten.

3.1 Creación de la base de datos.

Como el servicio ya está inicializado, se crea la Base de Datos, para nuestro ejemplo, CITAS.

La sintaxis de la instrucción es:

```
mysql> create database nombre_basedatos;
```

Para crear la base de datos “citas” se procede así:

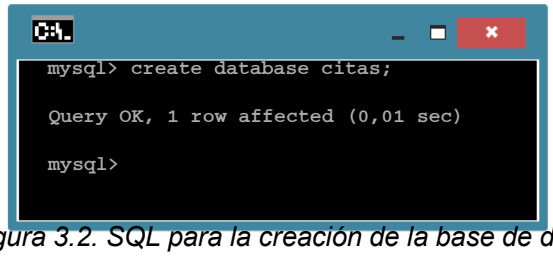


Figura 3.2. SQL para la creación de la base de datos.

Para crear las tablas debemos seleccionar “citas” como base de datos prede旳椀nida. La instrucción es:

```
mysql> use nombre_base_de_datos ;
```

Para este ejemplo se tiene:



Figura 3.3. SQL para seleccionar una base de datos.

Para veri旳椀car que la base de datos fue creada se usa el siguiente comando:

```
mysql> show databases ;
```

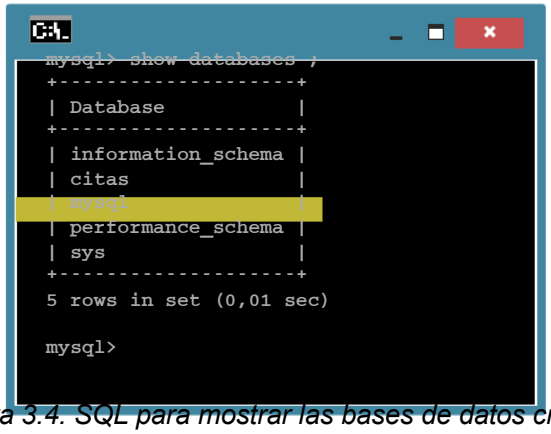


Figura 3.4. SQL para mostrar las bases de datos creadas.

En la pantalla anterior se puede observar que aparece la base de datos “citas”.

3.2 Creación de las tablas.

Una vez de旳椀nida “citas” como base de datos predeterminada, se procede a crear las respectivas tablas.

La sintaxis para la creación de tablas es la siguiente:

```
mysql> create table nombreTabla (
nombrecampo1 tipodatos(tamaño) modi旳椀cador,
nombrecampo2 tipodatos(tamaño) modi旳椀cador,
...
primary key (nombrecampo1)
);
```

3.2.1 Creación de la tabla “Medicos”.

Para la creación de la tabla “Medicos” se procede con el siguiente comando:

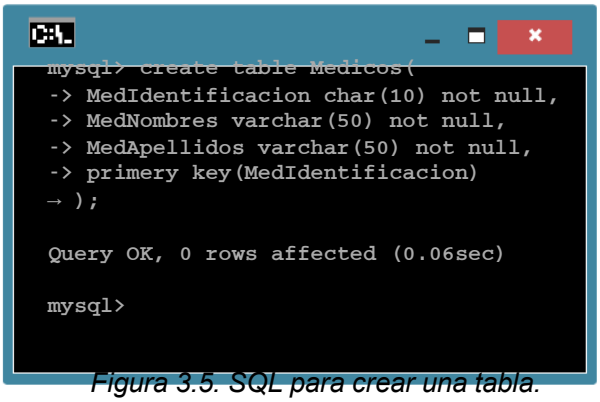


Figura 3.5. SQL para crear una tabla.

Una vez creada se verifica que la tabla aparezca en la base de datos con el siguiente comando:

```
mysql> show tables from base_de_datos ;
```

Al ejecutar el comando en la consola de MySQL muestra los siguiente:

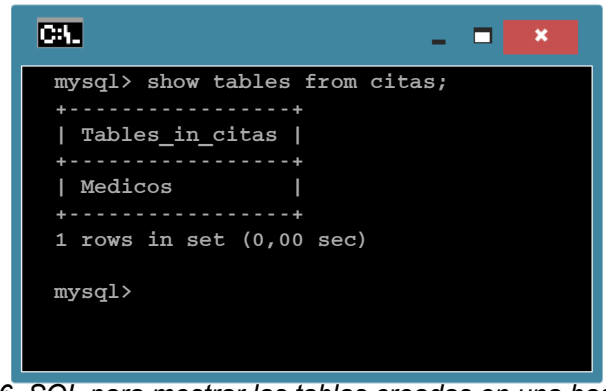


Figura 3.6. SQL para mostrar las tablas creadas en una base de datos.

Se puede observar que aparece la table “Medicos” en el listado.

Para verificar que la estructura de la tabla se usa el comando:

```
mysql> describe nombre_de_tabla ;
```

Para verificar la tabla “Medicos” se procede como se muestra a continuación:

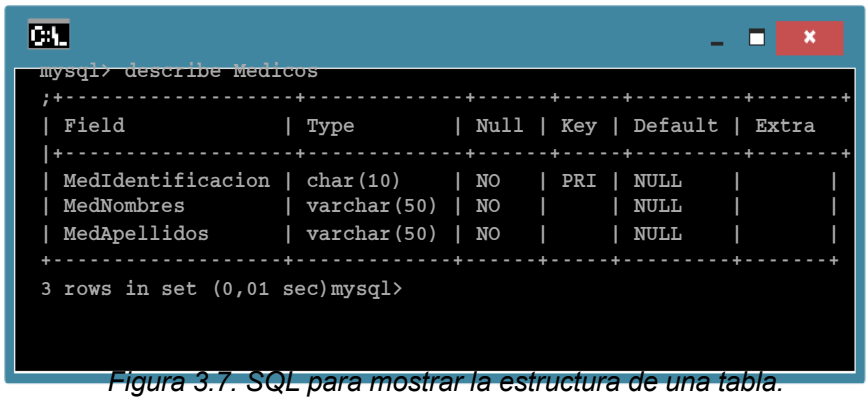


Figura 3.7. SQL para mostrar la estructura de una tabla.

3.2.2 Creación de la tabla "Pacientes".

Para crear la tabla "Pacientes" se procede con el siguiente comando:

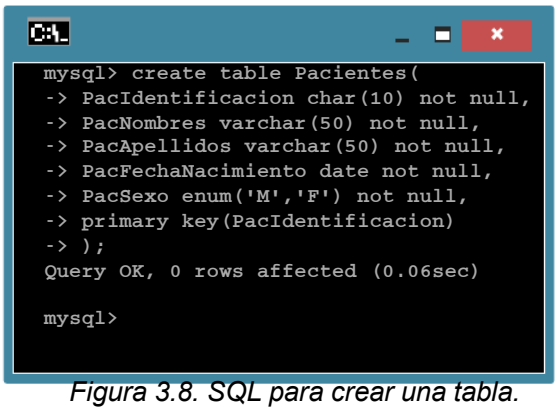


Figura 3.8. SQL para crear una tabla.

Se puede veri旳椀car que la tabla quedó creada con el siguiente comando:

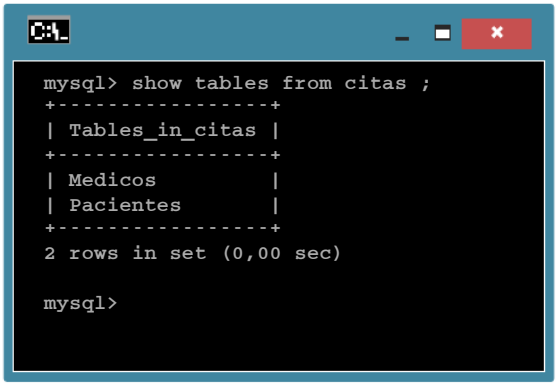


Figura 3.9. SQL para mostrar las tablas de una base de datos.

Para verificar la estructura de la tabla se procede con el siguiente comando:

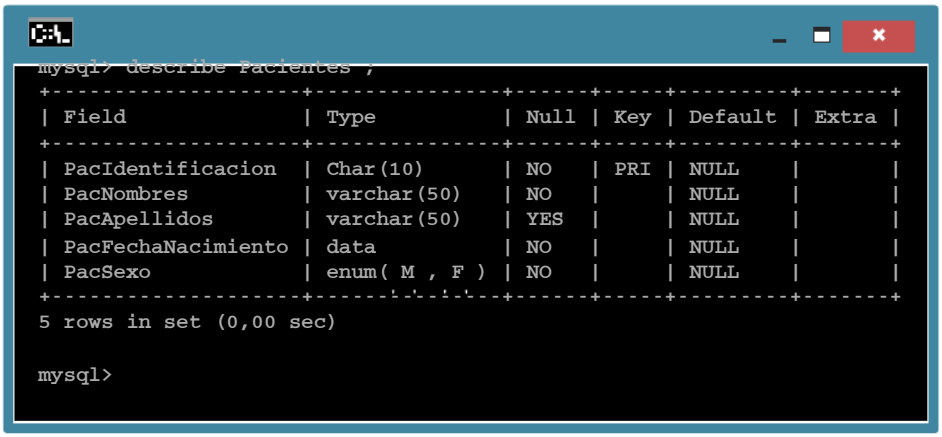


Figura 3.10. SQL para conocer la estructura de una tabla.

3.2.3. Creación tabla "Consultorios".

Para crear la tabla "Consultorios" se usa el siguiente comando:

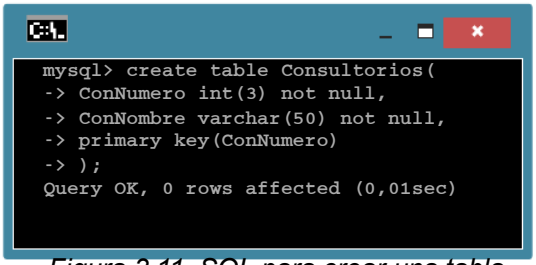


Figura 3.11. SQL para crear una tabla.

Para verificar que la tabla haya sido creada se procede así:

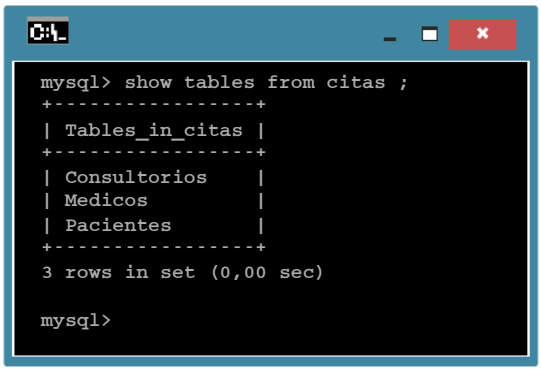


Figura 3.12. SQL para mostrar las tablas de una base de datos.

Para veri旳椀car la estructura de la tabla se usa el siguiente comando:

```
mysql> describe Consultorios ;
+-----------+-------------+------+-----+---------+-------+
| Field     | Type        | Null | Key | Default | Extra |
+-----------+-------------+------+-----+---------+-------+
| ConNumero | int(3)      | NO   | PRI | NULL    |       |
| ConNombre | varchar(50) | NO   |     | NULL    |       |
+-----------+-------------+------+-----+---------+-------+
2 rows in set (0,00 sec)

mysql>
```

Figura 3.13. SQL para mostrar la estructura de una tabla.

3.2.4. Creación tabla “Citas”.

Al revisar la estructura de la tabla “Citas” de旳椀nida en un numeral anterior se puede observar que esta tabla tiene unos modi旳椀cadores nuevos. Además esta tabla contiene llaves foráneas, es decir, está relacionada con otras tablas.

La sintaxis para la creación de una llave foránea es la siguiente:

```
foreign key(nombre_de_campo) references tabla(campo_tabla)
```

Al aplicar la sintaxis anterior a la creación de la tabla “Citas” se tiene lo siguiente:

```
CREATE TABLE citas (
  -> CitNumero int AUTO_INCREMENT,
  -> CitFecha date NOT NULL,
  -> CitHora varchar(10) NOT NULL,
  -> CitPaciente char(10) NOT NULL,
  -> CitMedico char(10) NOT NULL,
  -> CitConsultorio int(3) NOT NULL,
  -> CitEstado enum('Asignada','Cumplida') NOT NULL DEFAULT 'Asignada',
  -> PRIMARY KEY (CitNumero),
  -> FOREIGN KEY (CitPaciente) REFERENCES Pacientes (PacIdentificacion),
  -> FOREIGN KEY (CitMedico) REFERENCES Medicos (MedIdentificacion),
  -> FOREIGN KEY (CitConsultorio) REFERENCES Consultorios (ConNumero)
);
Query OK, 0 row affected (0,11 sec)

mysql>
```

Figura 3.14. SQL para crear una tabla.

4. Modi旳椀cación de la estructura de las tablas.

Para modi旳椀car la estructura de una tabla se usa el siguiente comando:

```
mysql> alter table <nombre tabla> <atributo> <opciones> ;
```

La lista de los atributos o comandos es la siguiente:

ATRIBUTOS	FUNCIÓN	SINTAXIS
MODIFY	Modifica un campo, esto puede ser el tipo de dato y/o el ancho del campo. También se utiliza para agregar y/o eliminar modificaciones.	alter table nombre_tabla MODIFY COLUMN nombre_columna modificador; Para realizar este cambio se deben incluir los modificadores que tiene el campo y los que se van a agregar.
CHANGE	Modifica el nombre de un campo.	alter table nombre_tabla CHAGE COLUMN nombre_columna_actual nuevo_nombre_columna tipo_dato ancho_dato modificador(es);
RENAME	Cambia el nombre de una tabla.	alter table personas rename clientes;
DROP	Borra Columnas.	alter table tabla DROP COLUMN ncolumnaABorrar;
ADD	Adiciona Columnas a una Tabla, es importante informar la ubicación de la nueva columna. Para ello se utiliza AFTER.	alter table nombre_tabla ADD nueva_Columna Tipo_dato Modificador AFTER Nombre_columna;

Figura 4.1. Atributos para modi旳椀car la estructura de una tabla.

Se debe tener cuidado al momento de cambiar la estructura de una tabla ya que estos cambios pueden afectar toda la base de datos. El MySQL veri旳椀ca cada cambio antes de aplicar. Si los cambios rompen las reglas de integridad el motor de base de datos no dejará aplicarlos y en su lugar emitirá un código y mensaje de error.

Para practicar esta instrucción se va a crear la tabla “Tratamientos” con una estructura inicial a la cual posteriormente se le introducirán unos cambios para llegar a una estructura deseada.

La estructura inicial de la tabla es la siguiente:

Campo	Tipo de campo	Longitud	Modificador
Tranumero	Int		Primary Key
TraFechaAsignado	Date		Not null
Descripción	Text		Not null
TraFechaInicio	Varchar	10	Not null
TraObservaciones	Text		Not null
TraTemporal	Varchar	2	
TraPaciente	Char	10	Pacientes (PacIdentificacion)

Figura 4.2. Estructura de la tabla Tratamientos.

Para crearla se usa el siguiente comando:

```
mysql> create table Tratamientos(
    -> TraNumero int,
    -> TraFechaAsignado date not null,
    -> Descripcion text not null,
    -> TraFechaInicio varchar(10);
    -> TraObservaciones text not null,
    -> TraTemporal varchar(2);
    -> TraPaciente char(10) not null,
    -> primary key( TraNumero ),
    -> foreign key ( TraPaciente ) references Pacientes ( PacIdentificacion )
    -> );

Query OK, 0 rows affected (0,14 sec)

mysql>
```

Figura 4.3. SQL para crear una tabla.

Para verificar su estructura se usa el comando “describe” así:

```
mysql> describe Consultorios ;
+------------------+-------------+------+-----+---------+-------+
| Field            | Type        | Null | Key | Default | Extra |
+------------------+-------------+------+-----+---------+-------+
| TraNumero        | int(11)     | NO   | PRI | NULL    |       |
| TraFechaAsignado | date        | NO   |     | NULL    |       |
| Descripcion      | text        | NO   |     | NULL    |       |
| TraFechaInicio   | varchar(10) | YES  |     | NULL    |       |
| TraObservaciones | text        | NO   |     | NULL    |       |
| TraTemporal      | varchar(2)  | NO   |     | NULL    |       |
| TraPaciente      | varchar(10) | NO   | MUL | NULL    |       |
+------------------+-------------+------+-----+---------+-------+
2 rows in set (0,00 sec)

mysql>
```

Figura 4.4. SQL para mostrar la estructura de una tabla.

Los cambios que se necesitan aplicar para llegar a la estructura requerida son los siguientes:

No	Descripción
1	Se requiere que el campo "TraNumero" sea auto-incremental.
2	Se requiere que el campo "Descripcion" sea renombrado a "TraDescripcion" para seguir el estándar de nombres de campos.
3	Cambia el tipo de datos del campo "TraFechaInicio" al tipo "date".
4	Agregar el campo "TraFechaFin" de tipo "date" después de la fecha de inicio.

Figura 4.5. Lista de las modificaciones a realizar a la tabla Tratamientos.

1) Para aplicar el primer cambio: "Se requiere que el campo TraNumero sea autoincremental" se utiliza el siguiente comando:

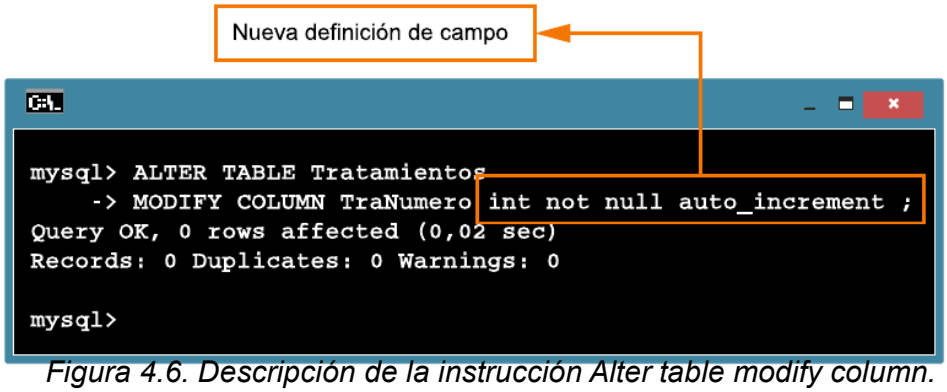


Figura 4.6. Descripción de la instrucción Alter table modify column.

2) Para aplicar el segundo cambio: se requiere que el campo *Descripcion* sea renombrado a *TraDescripcion* se utiliza el comando "change" como se describe a continuación:

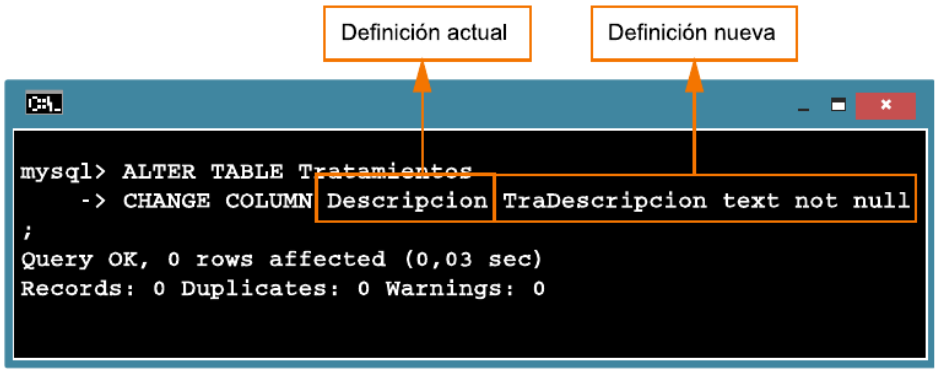


Figura 4.7. Descripción de la instrucción Alter table change column.

3) Para aplicar el tercer cambio: “Cambiar el tipo de dato del campo “TraFechaInicio” al tipo date “ se utiliza el comando “modify” como se describe a continuación:

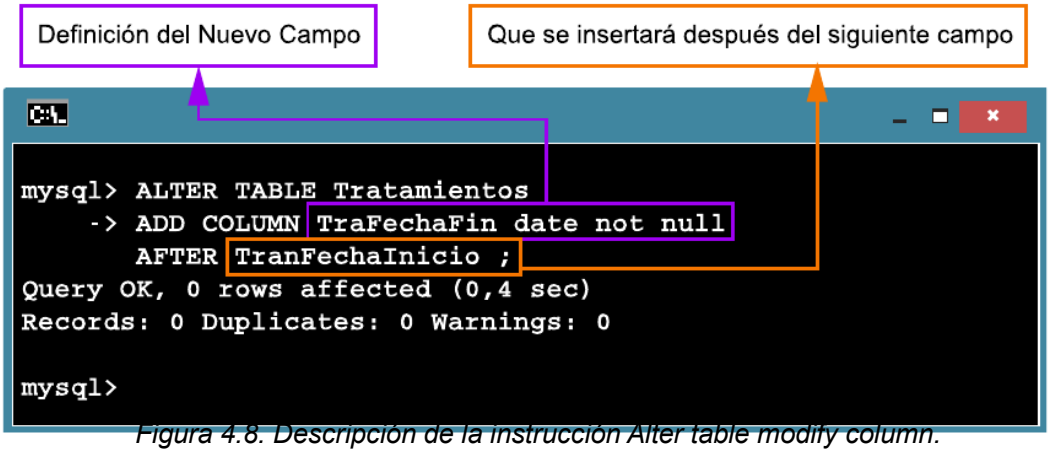


Figura 4.8. Descripción de la instrucción Alter table modify column.

4) Para aplicar el cuarto cambio: “Agregar el campo TraFechaFin de tipo date después de la fecha de inicio.” se aplica el siguiente comando:

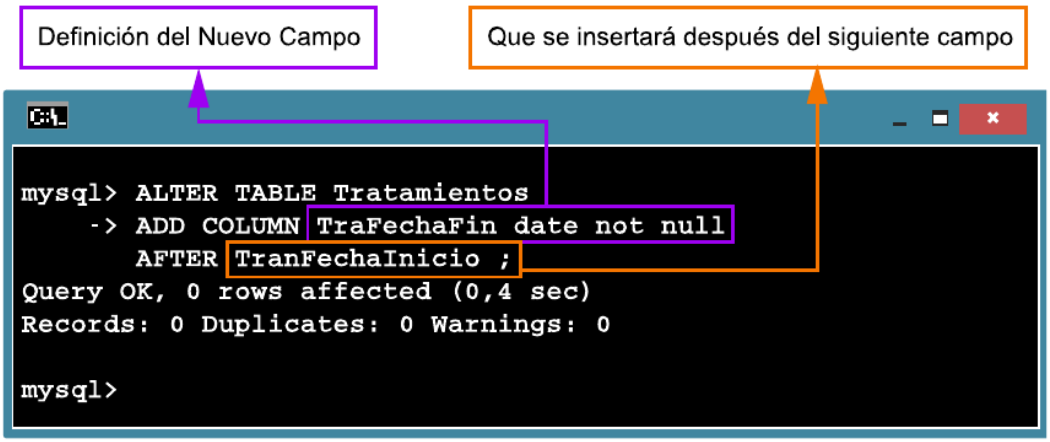


Figura 4.9. Descripción de la instrucción Alter table add column.

5) Para aplicar el quinto cambio: “Eliminar el campo TraTemporal” se procede con el comando “drop column” como se describe a continuación:

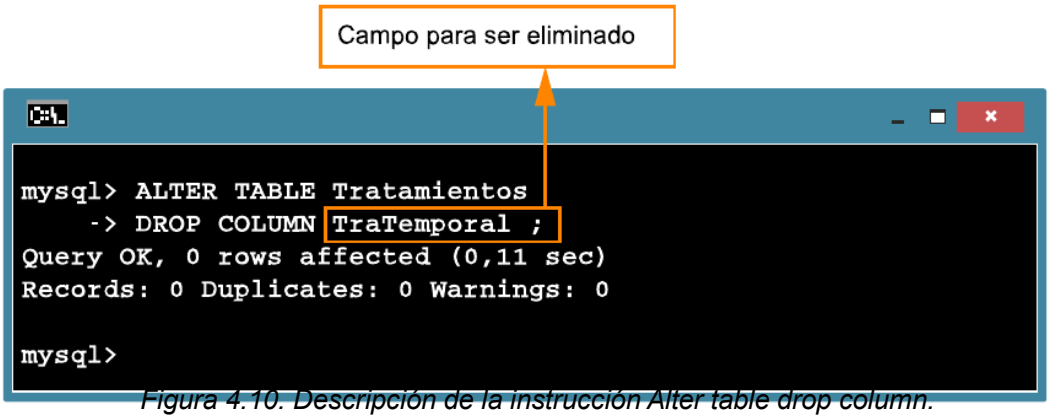


Figura 4.10. Descripción de la instrucción Alter table drop column.

Por último se verifica la nueva estructura de la tabla usando el comando “describe” como se muestra a continuación:

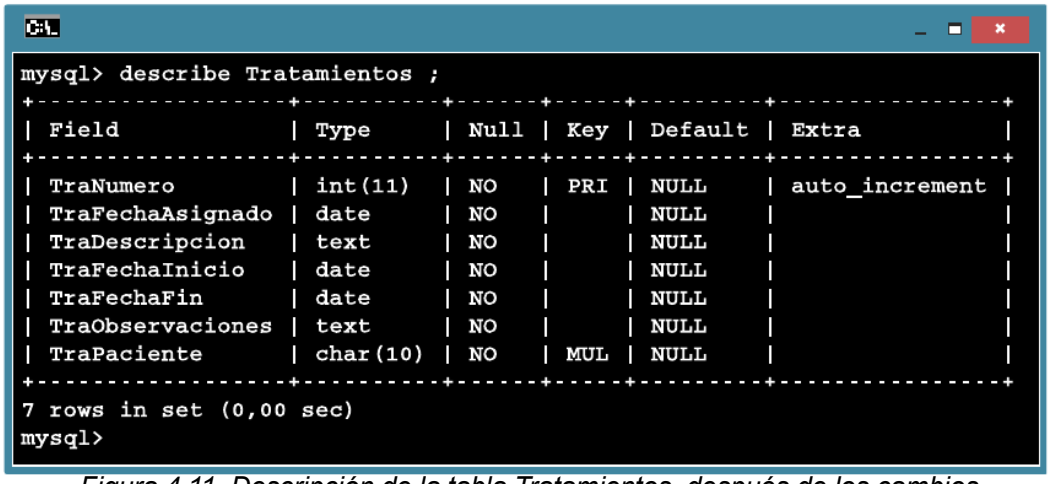


Figura 4.11. Descripción de la tabla Tratamientos después de los cambios.

5. Actualización e inserción de registros.

5.1 Inserción de registros.

Para realizar las operaciones de Inserción de Registros se utiliza la sentencia “INSERT INTO” del lenguaje de manipulación de datos (D.M.L.).

El comando “INSERT INTO” permite incluir los datos en cada uno de los campos que se tienen en las tablas de la base de datos creada. Su sintaxis es:

```
mysql> INSERT INTO <nombre tabla> (<campo1>, <campo2>, …, <campoN>)
       VALUES (<valor1>,<valor2>, … , <valorN>) ;
```

Para los establecer lo valores que tomarán los campos se debe tener en cuenta lo siguiente:

a) Los campos tipo texto, char o varchar deben estar encerrados entre comillas. Ejemplo: ‘Gomez Rodriguez’, ‘Carrera 42 45 -11 Apto 1102’.

b) Los campos tipo entero debe ir sin ningún cambio.

c) Los campos tipo decimal o 昀敳otante: la parte entera debe estar separada de la parte decimal por un punto “.”. Lo anterior ya que la coma es usada como separador de campos.

d) Los campos tipo fecha: pueden tomar varios tipos de formatos. El formato prede昀椀nido es ‘AAAA-MM-DD’ donde AAAA es el año , MM el mes y DD el día. El valor debe además ir encerrado entre comillas. Por ejemplo el 13 de Julio de 2017 quedaría expresado como ‘2017-07-13’ .

5.1.1 Inserción de registros sin llaves foráneas.

Para insertar el siguiente registro :

Identificación	91.222.333
Nombres	Carlos Jesús
Apellidos	Rodríguez Cala
Fecha Nacimiento	23 de Enero de 1970
Sexo	M

Figura 5.1. Registro ejemplo a insertar.

Se utiliza el siguiente comando:

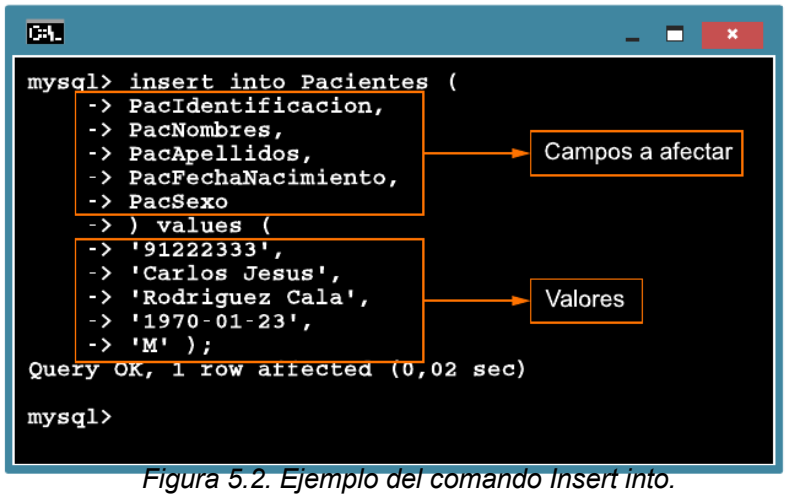


Figura 5.2. Ejemplo del comando Insert into.

5.1.2 Inserción de registros con llaves foráneas.

Las llaves foráneas son reglas de integridad que se deben cumplir para que la base de datos sea consistente.

Para insertar registros en tablas que tienen llaves foráneas se debe identi的椀car el valor de esas llaves para incluirlos en la sentencia de inserción.

Por ejemplo, para insertar un registro en la tabla “Tratamientos” se debe determinar la llave primaria del paciente en la tabla “Pacientes”.

Para insertar el siguiente registro:

TraFechaAsignado	“2017-07-13”
TraDescripcion	Tratamiento de Conductos
TraFechainicio	“2017-08-01”
TraFechaFin	“2017-08-03”
TraObservaciones	Paciente con hipertensión
TraPaciente	91.222.333

Figura 5.3. Ejemplo de registro con llaves foráneas a insertar.

Se procede así:

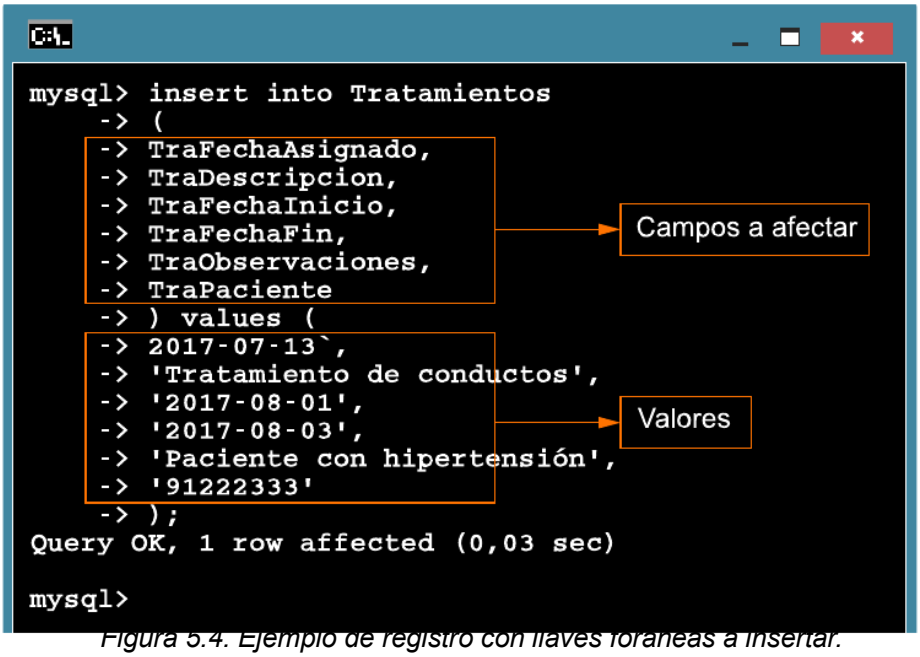


Figura 5.4. Ejemplo de registro con llaves foráneas a insertar.

Para la inserción del anterior registro no se tuvo que de旳椀nir el valor del campo “TraNumero” ya que fue de旳椀nido como “auto_increment”, es decir, que se auto incrementa con cada registro nuevo. Para veri旳椀car que el registro fue insertado se ejecuta una consulta como se muestra en la 旳椀gura 5.5.

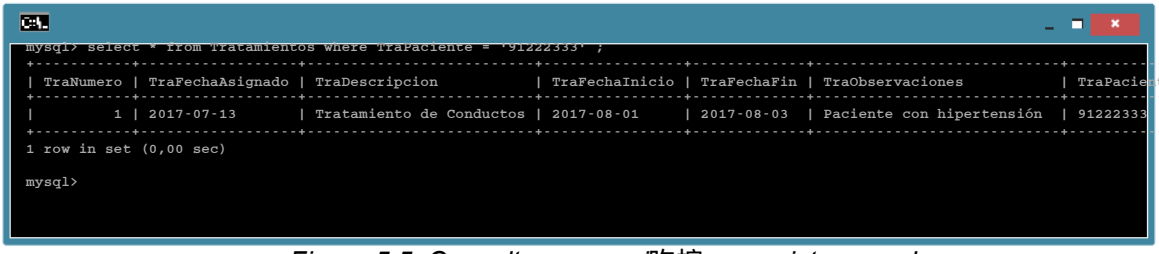


Figura 5.5. Consulta para veri旳椀car registro creado.

6. Consultas de registros.

Para realizar las operaciones de Consulta de Registros se utiliza la sentencia SELECT del lenguaje de manipulación de datos (D.M.L.).

La sentencia SELECT permite visualizar la información de la Base de Datos, los datos que se presentan corresponden a una o más 旳椀las de una tabla o requiere a una o más 旳椀las de una o más tablas.

La sintaxis básica es:

```
SELECT <lista de campos >
FROM <tablas >
WHERE <condiciones>
GROUP BY <campos de agrupamiento>
HAVING <condiciones sobre los grupos >
ORDER BY <ordenamientos>
LIMIT <cantidad de registros a traer >
OFFSET <número de página >
```

6.1 Datos de prueba.

Para el ejemplo se procederá con los siguientes registros de prueba:

IDENTIFICACIÓN	NOMBRES	APELLIDOS	FECHA NACIMIENTO	SEXO
29.234.333	Alejandra Marcela	Díaz Granados	1980-03-24	F
1098.343.678	Juan Antonio	Pérez Pereira	1978-08-09	M
37.456.298	Diana Marcela	Estévez	1985-09-06	F
51.890.654	Adriana María	Pataquiva	1990-12-24	F
91.222.333	Carlos Jesús	Rodríguez Cala	1970-01-23	M

Figura 6.1. Conjunto de registros de ejemplo.

TraFechaAsignado	TraDescripción	TraFechaInicio	TraFechaFin	TraObservaciones	TraPaciente
2017-07-13	Tratamiento de Conductos	2017-08-01	2017-08-03	Paciente con hipertensión	91.222.333
2017-07-02	Profilaxis	2017-07-02	2017-07-02	Sin novedad	51.890.654
2017-06-05	Resina	2017-06-05	2017-06-05	Paciente con sensibilidad	37.456.298
2017-05-23	Profilaxis	2017-05-23	2017-05-23	Sin novedad	29.234.333

Figura 6.2. Conjunto de registros de ejemplo.

6.2 Consultas básicas.

Para hacer un ejemplo de consultas básicas se tomará la tabla pacientes. Esta consulta puede generarse de diferentes maneras dependiendo de los datos que se necesiten visualizar.

Si se quiere visualizar todos los registros y campos de la tabla se usa el asterisco (*) el cual indica al intérprete de comandos que extraiga todos los campos.

La instrucción es:

```
foreign key(nombre_de_campo) references tabla(campo_tabla)
mysql> SELECT * FROM Pacientes ;
```

Al ejecutar el comando en la consola de MySQL aparece los siguiente:
Figura 6.3. SQL para hacer consultas a la base de datos.

Para limitar el número de registros se usa el comando “LIMIT x”. Ejemplo:

```
mysql> select * from Pacientes ;
+------------------+------------------+---------------+--------------------+----------+
| PacIdentificacion | PacNombres       | PacApellidos  | PacFechaNacimiento | PacSexo |
+------------------+------------------+---------------+--------------------+----------+
| 1098343678       | Juan Antonio     | Perez Pereira | 1978-08-09         | M       |
| 29234333         | Alejandra Marcela | Diaz Granados | 1980-03-24         | F       |
| 37456298         | Diana Marcela    | Estevez       | 1985-09-06         | F       |
| 51890654         | Adriana María    | Pataquiva     | 1990-12-24         | F       |
| 91222333         | Carlos Jesus     | Rodriguez Cala | 1970-01-23         | M       |
+------------------+------------------+---------------+--------------------+----------+
5 rows in set (0,00 sec)mysql>
```

Figura 6.4. SQL para hacer consultas a la base de datos con límite.

La consulta anterior limita el resultado a máximo tres registros.

El asterisco indica al intérprete de comandos seleccionar todos los campos. Esto no es adecuado ya que puede generar consultas pesadas que consumen valiosos recursos del servidor. Para especi旳椀car los campos se procede así:

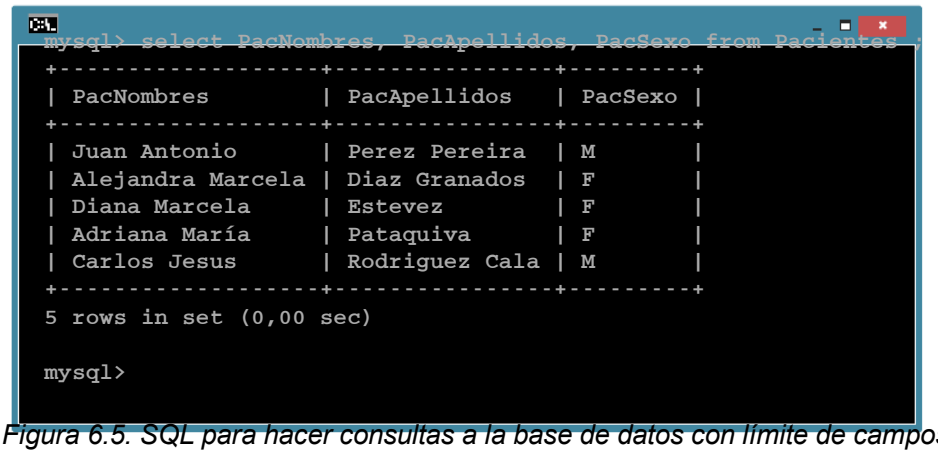


Figura 6.5. SQL para hacer consultas a la base de datos con límite de campos.

En la anterior consulta solo se trajeron los campos especi旳椀cados.

Se pueden cambiar los nombres de los campos para que sean más entendibles para el usuario 旳椀nal a través de un “alias”.

La sintaxis para cuando se usan alias es la siguiente:

```
mysql> SELECT <nombre_campo> AS nombre_alterno_de_campo FROM <tabla>;
```

Para el ejemplo anterior se puede mejorar los nombres de los campos así:

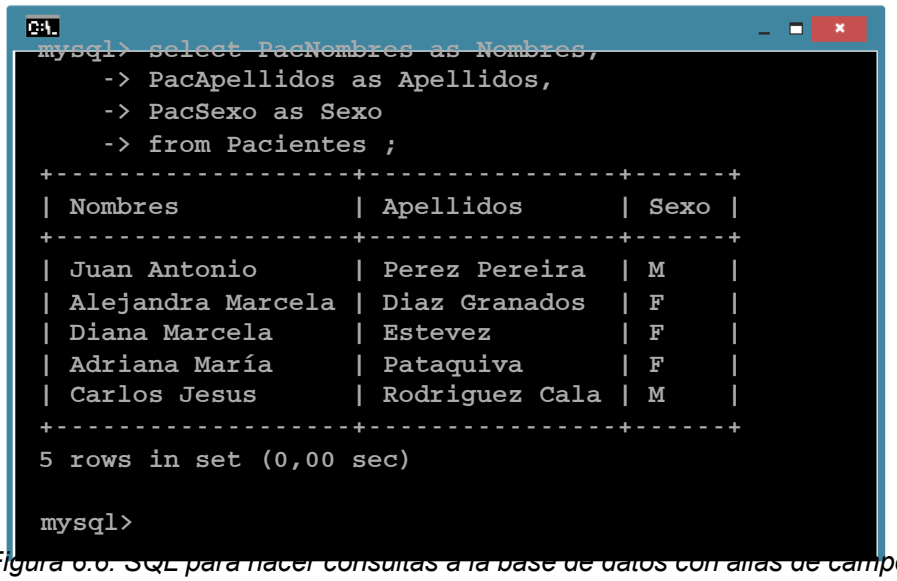


Figura 6.6. SQL para hacer consultas a la base de datos con alias de campos.

Las tablas tambien pueden tener un alias que será de gran utilidad cuando se realicen los JOIN o enlaces. La sintaxis es la siguiente:

```
mysql> SELECT <nombre_campo> FROM <tabla> [AS] nombre_alterno_de_tabla;
```

Para el ejemplo anterior se puede aplicar un alias a la tabla así:

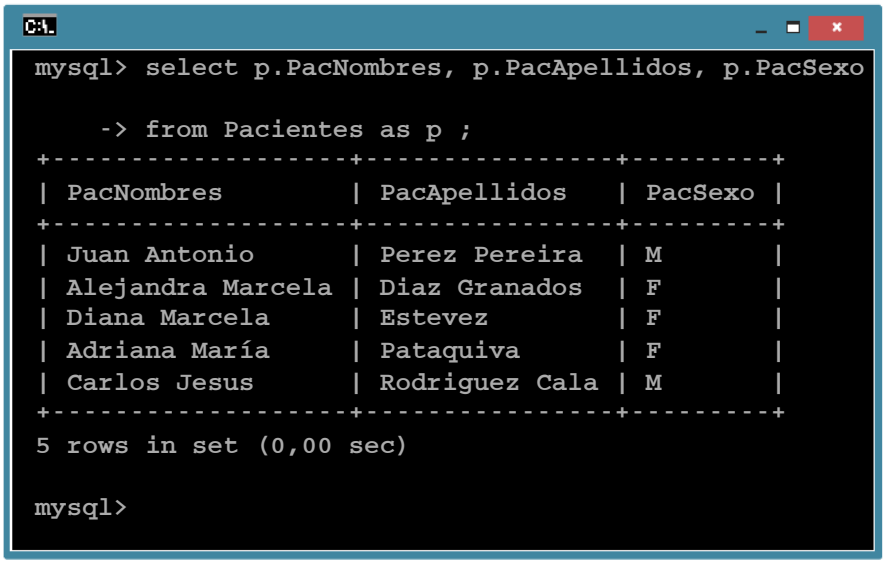
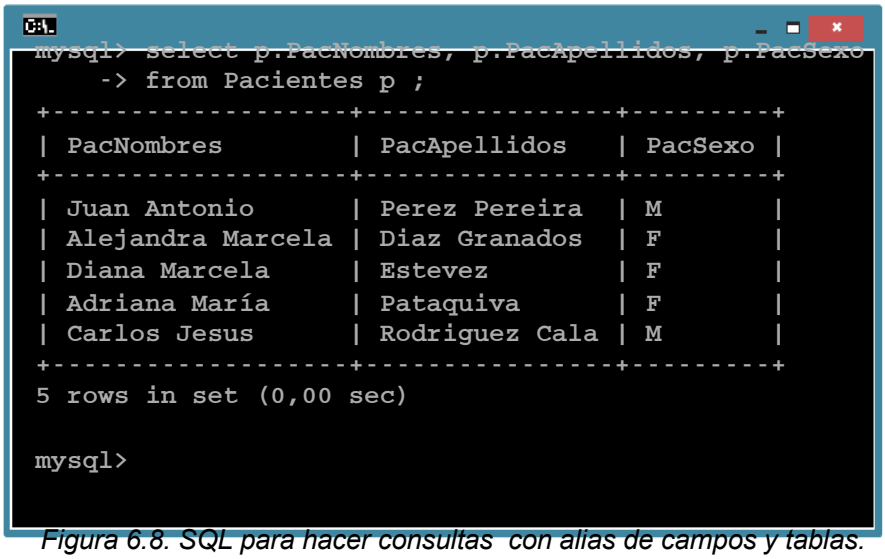


Figura 6.7. SQL para hacer consultas a la base de datos con alias de campos y tablas.

El pre旳椀jo “AS” se puede omitir. Para el ejemplo requiere funciona lo siguiente:



```
mysql> select p.PacNombres, p.PacApellidos, p.PacSexo
    -> from Pacientes p ;
+-------------------+----------------+---------+
| PacNombres        | PacApellidos   | PacSexo |
+-------------------+----------------+---------+
| Juan Antonio      | Perez Pereira  | M       |
| Alejandra Marcela | Diaz Granados  | F       |
| Diana Marcela     | Estevez        | F       |
| Adriana María     | Pataquiva      | F       |
| Carlos Jesus      | Rodriguez Cala | M       |
+-------------------+----------------+---------+
5 rows in set (0,00 sec)

mysql>
```

Figura 6.8. SQL para hacer consultas con alias de campos y tablas.

6.3 Aplicación de 旳椀ltros a las consultas.

El 旳椀ltro como su nombre lo indica limita el número de registros a mostrar a través de la incorporación de una o varias condiciones lógicas.

En el lenguaje SQL los 旳椀ltros se implementan con la cláusula “WHERE” seguido de una o más preguntas lógicas. Estas preguntas se evalúan una a una y se aplican operaciones de matemática booleana para determinar el resultado lógico que siempre deberá ser “verdadero” o “falso”

La sintaxis es:

```
SELECT <campos>
FROM <tablas>
WHERE <bloque de condiciones>
```

Ejemplo, para seleccionar los pacientes de sexo femenino se procede con la siguiente consulta.

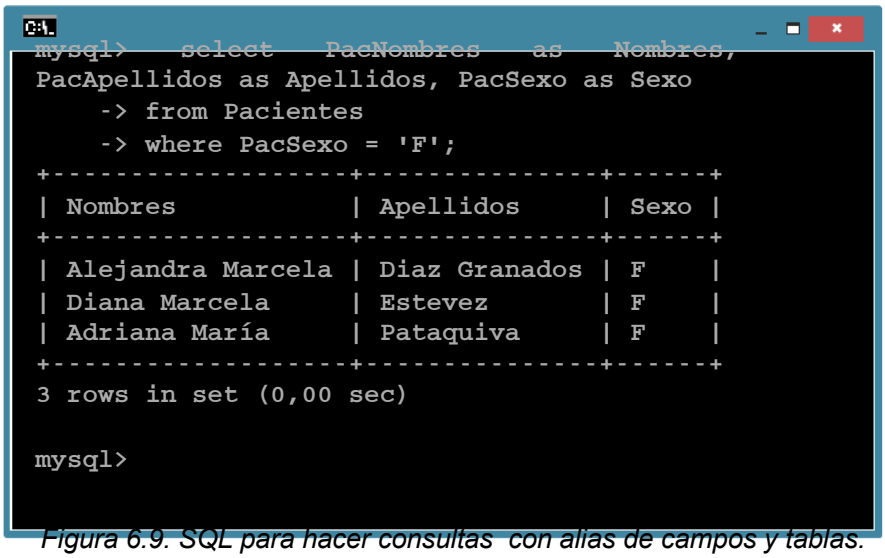


Figura 6.9. SQL para hacer consultas con alias de campos y tablas.

En el ejemplo anterior la pregunta " PacSexo = 'F' " se hace a cada registro de la tabla "Pacientes". Aquellos registros en los cuales la pregunta es verdadera son extraidos.

Se pueden hacer múltiples preguntas apoyados en los operadores lógicos AND y OR.

Las preguntas se pueden agrupar con paréntesis. De la misma forma como se evalúan las expresiones aritméticas las expresiones booleanas encerradas en paréntesis se evalúan primero.

6.4 Ordenamiento de las consultas.

Los resultados de las consultas se pueden ordenar en un orden especi的椀cado con el comando "ORDER BY".

La sintaxis es la siguiente:

```
SELECT <campos>
FROM <tablas>
WHERE <bloque de condiciones>
ORDER BY <lista de los campos separados por coma> ASC |
DESC
```

En el siguiente ejemplo la consulta es ordenada por dos campos: PacNombres y PacApellidos.

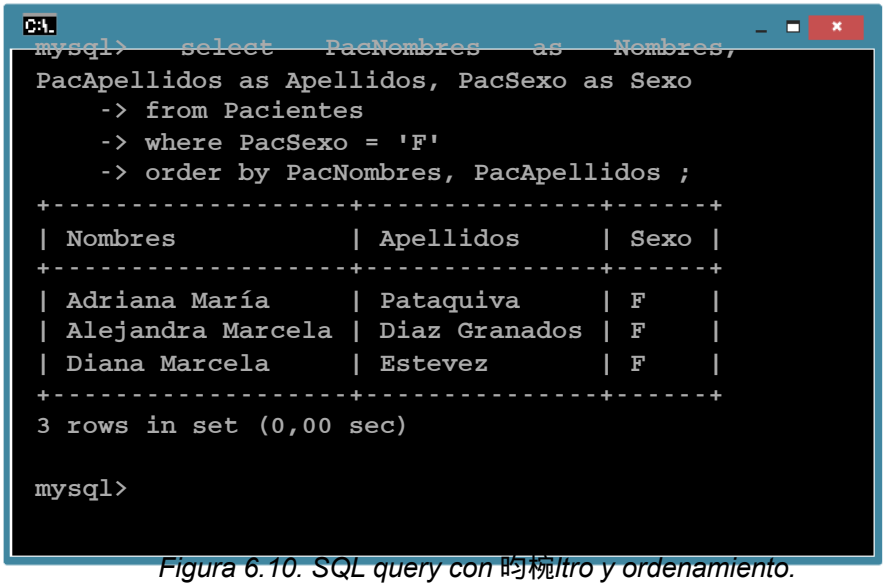


Figura 6.10. SQL query con filtro y ordenamiento.

Por defecto el ordenamiento es ascendente. Si se requiere en orden descendente se procede aplicando el modificador "DESC" así:

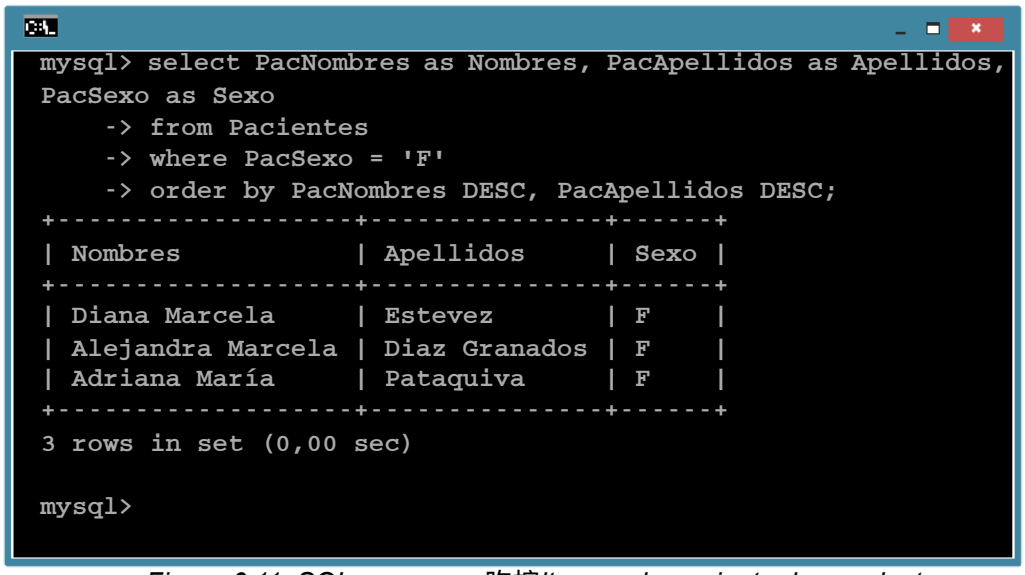


Figura 6.11. SQL query con filtro y ordenamiento descendente.

6.5 Introducción a las consultas multi-tablas

En muchos casos se requiere que la consulta extraiga la información de varias tablas al

mismo tiempo.

En estos casos MySQL y en general el lenguaje SQL suministra la opción “JOIN”. La sintaxis es la siguiente:

```
SELECT <campos>
FROM <tabla1 alias t1 > JOIN <tabla2 alias t2 > ON <condición de unión>
WHERE <bloque de condiciones>
```

Ejemplo: se requiere listado con la cédula, nombres, apellidos y la descripción del tratamiento realizado por cada paciente.

Identificación	Nombres	Apellidos	Tratamiento	Observaciones
91.222..333	Carlos Jesús	Rodríguez Cala	Tratamiento de conductos	Paciente con hipertensión

Figura 6.12. Ejemplo para consultas multi-tablas.

Los campos nombres y apellidos están contenidos en la tabla “Pacientes”.
Los campos descripción y observaciones del tratamiento está contenido en la tabla “Tratamientos”.
La cédula es la llave primaria en la tabla “Pacientes” y es una llave foránea en la tabla “Tratamientos”.

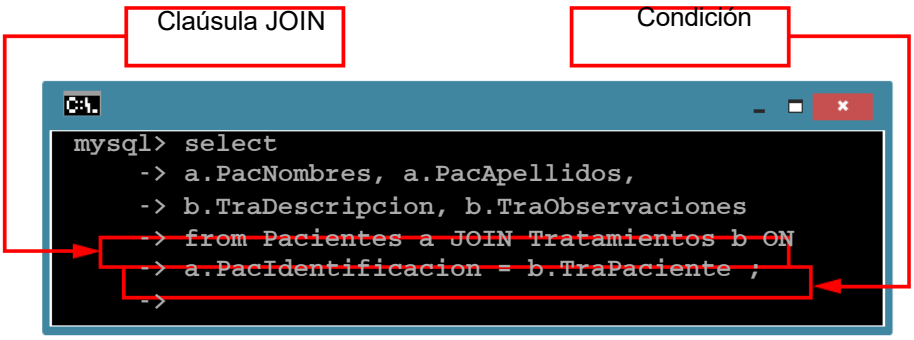
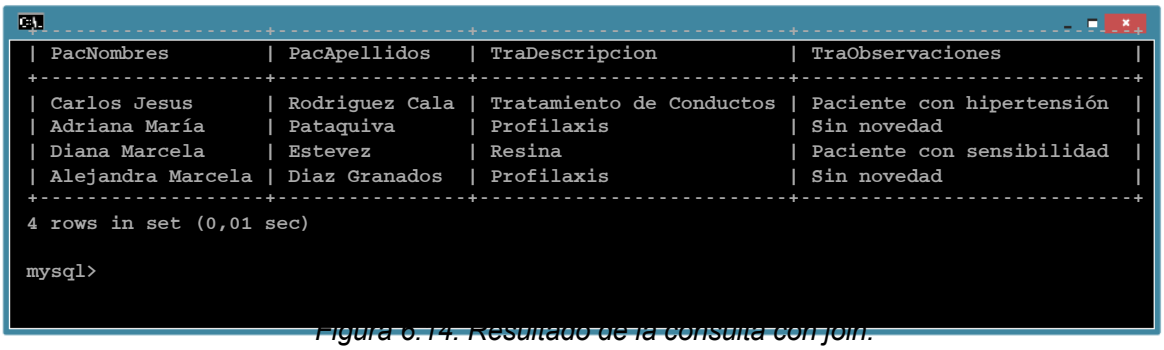


Figura 6.13. Ejemplo de una consulta con join.

Para hacer el enlace o “join” de las tablas se procede así:

El resultado es el siguiente:

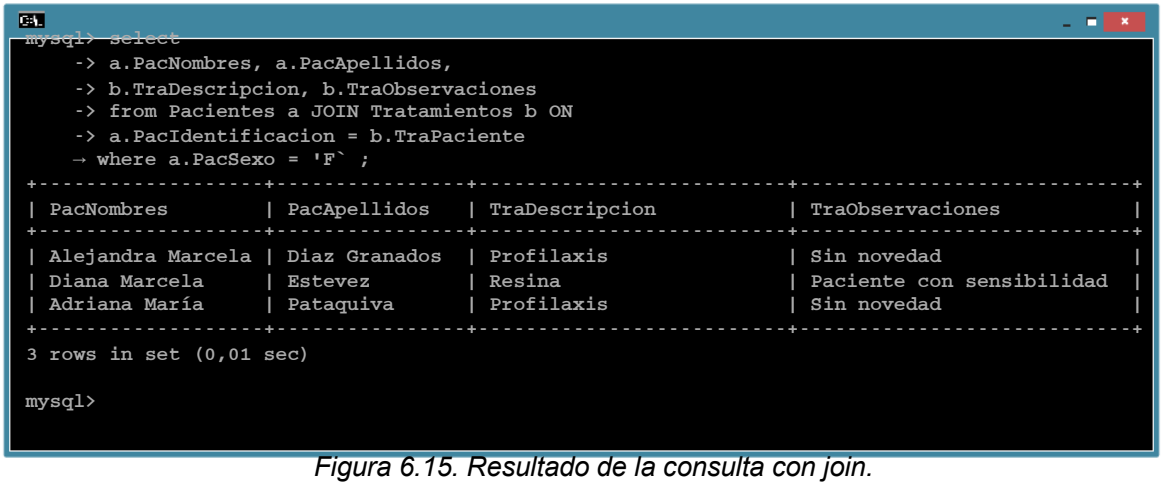


```
+-------------------+---------------+--------------------------+----------------------------+
| PacNombres        | PacApellidos  | TraDescripcion           | TraObservaciones           |
+-------------------+---------------+--------------------------+----------------------------+
| Carlos Jesus      | Rodriguez Cala | Tratamiento de Conductos | Paciente con hipertensión  |
| Adriana María     | Pataquiva     | Profilaxis               | Sin novedad                |
| Diana Marcela     | Estevez       | Resina                   | Paciente con sensibilidad  |
| Alejandra Marcela | Diaz Granados | Profilaxis               | Sin novedad                |
+-------------------+---------------+--------------------------+----------------------------+
4 rows in set (0,01 sec)

mysql>
```

Figura 6.14. Resultado de la consulta con join.

El enlace o "join" genera una tabla a la cual se le pueden aplicar otros 昀椀ltros. Para el ejemplo anterior se puede restringir la consulta sólo a los pacientes femeninos. En este caso el comando quedaría así:



```
mysql> select
    -> a.PacNombres, a.PacApellidos,
    -> b.TraDescripcion, b.TraObservaciones
    -> from Pacientes a JOIN Tratamientos b ON
    -> a.PacIdentificacion = b.TraPaciente
    → where a.PacSexo = 'F` ;
+-------------------+---------------+--------------------------+----------------------------+
| PacNombres        | PacApellidos  | TraDescripcion           | TraObservaciones           |
+-------------------+---------------+--------------------------+----------------------------+
| Alejandra Marcela | Diaz Granados | Profilaxis               | Sin novedad                |
| Diana Marcela     | Estevez       | Resina                   | Paciente con sensibilidad  |
| Adriana María     | Pataquiva     | Profilaxis               | Sin novedad                |
+-------------------+---------------+--------------------------+----------------------------+
3 rows in set (0,01 sec)

mysql>
```

Figura 6.15. Resultado de la consulta con join.

6.6. Subconsultas.

Las subconsultas son como su nombre lo indica una consulta que se lleva a cabo dentro de otra consulta. Esto se presenta cuando se requiere consultar un dato o varios datos que están en distintas tablas pero no se quiere o no se puede realizar un JOIN que sería la forma más organizada de realizar esta labor.

6.6.1. Subconsultas con resultados escalares.

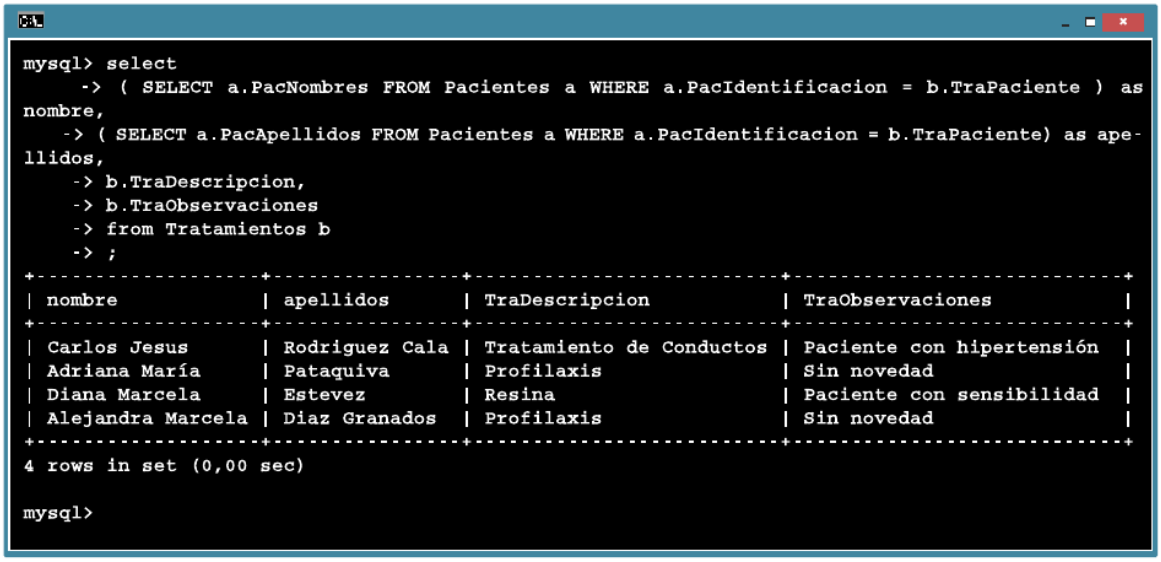
MySQL y en general el lenguaje SQL permite realizar subconsultas con la siguiente sintaxis:

```
SELECT * FROM tabla1 WHERE campo1 = (SELECT campo1 FROM t2);
```

Como se puede observar la subconsulta debe estar contenida entre un paréntesis. El lenguaje SQL primero resuelve la subconsulta y luego compara los resultados.

En este caso la subconsulta debe arrojar un único valor de lo contrario no se podrá realizar la comparación y el MySQL generará un error.

Ejemplo. Se puede recorrer la tabla Tratamientos y traer el nombre del paciente sin necesidad de hacer JOIN de la siguiente manera:



```
mysql> select
    -> ( SELECT a.PacNombres FROM Pacientes a WHERE a.PacIdentificacion = b.TraPaciente ) as
nombre,
   -> ( SELECT a.PacApellidos FROM Pacientes a WHERE a.PacIdentificacion = b.TraPaciente) as ape-
llidos,
    -> b.TraDescripcion,
    -> b.TraObservaciones
    -> from Tratamientos b
    -> ;
+------------------+---------------+-------------------------+----------------------------+
| nombre           | apellidos     | TraDescripcion          | TraObservaciones           |
+------------------+---------------+-------------------------+----------------------------+
| Carlos Jesus     | Rodriguez Cala | Tratamiento de Conductos | Paciente con hipertensión  |
| Adriana María    | Pataquiva     | Profilaxis              | Sin novedad                |
| Diana Marcela    | Estevez       | Resina                  | Paciente con sensibilidad  |
| Alejandra Marcela | Diaz Granados | Profilaxis              | Sin novedad                |
+------------------+---------------+-------------------------+----------------------------+
4 rows in set (0,00 sec)

mysql>
```

Figura 6.16. Ejemplo de una consulta con subconsultas.

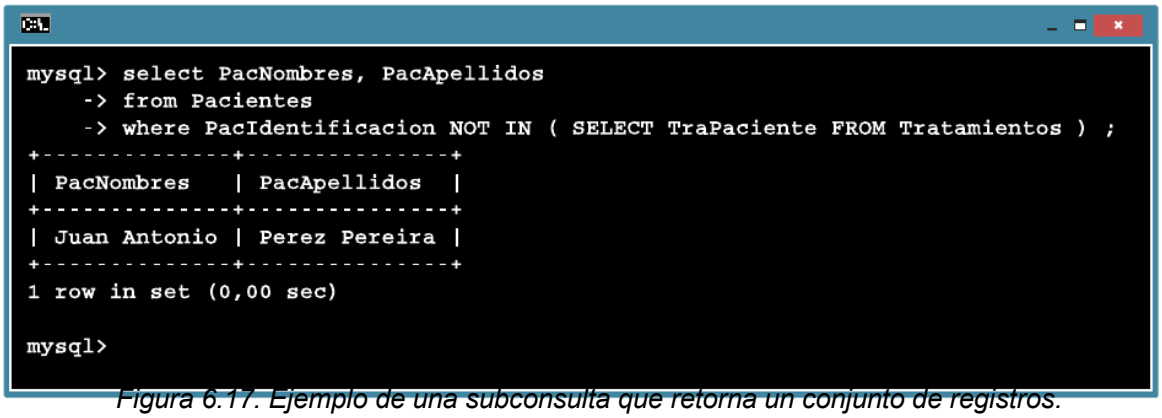
6.6.2. Subconsultas que retornan conjuntos de registros.

Las subconsultas pueden arrojar múltiples datos. En estos casos ya no se pueden comparar con los comparadores escalares como “=” , “>” , “<”. Se deben comparar con operadores de conjuntos. La sintaxis es la siguiente:

```
SELECT * FROM tabla1 WHERE campo1 [NOT] IN (SELECT campo1 FROM t2);
```

Esta sintaxis indica que la cláusula WHERE es verdadera si el campo1 de la tabla1 está contenido en el resultado de la sub consulta.

Ejemplo. Se puede determinar cuales pacientes aún no tienen un tratamiento así:



```
mysql> select PacNombres, PacApellidos
    -> from Pacientes
    -> where PacIdentificacion NOT IN ( SELECT TraPaciente FROM Tratamientos ) ;
+--------------+---------------+
| PacNombres   | PacApellidos  |
+--------------+---------------+
| Juan Antonio | Perez Pereira |
+--------------+---------------+
1 row in set (0,00 sec)

mysql>
```

Figura 6.17. Ejemplo de una subconsulta que retorna un conjunto de registros.

Además del comparador IN existen otros comparadores como ANY, ALL, SOME.

GLOSARIO

Alias: nombre alterno que pueden tomar los campos de una tabla o las tablas mismas para efectos de simpli旳椀car o mejorar su descripción.

Condición: partes que en su conjunto forman un 旳椀ltro.

Consulta: conjunto de instrucciones en lenguaje SQL que realizan una consulta a la base de datos.

D.D.L.: acrónimo de Data De旳椀nition Language. Subconjunto del lenguaje SQL utilizado para crear, modi旳椀car o borrar los componentes de las bases de datos.

D.M.L: acrónimo de Data Manipulation Language. Subconjunto del lenguaje SQL que trata la forma con los registros son creados, actualizados, consultados o borrados.

Filtro: condición que limita la cantidad de registros en una consulta.

GPL: acrónimo de General Public License. Tipo de licenciamiento abierto para software.

Join: unión entre una o más tablas.

Llave primaria: campo de una tabla cuyo valor identi旳椀ca inequívocamente un registro.

Llave foránea: campo de una tabla cuyo valor apunta a la llave primaria de otra tabla.

SQL: acrónimo de Structured Query Language. Lenguaje para el manejo de bases de datos.

5) Para aplicar el quinto cambio: “Eliminar el campo TraTemporal” se procede con el comando “drop column” como se describe a continuación:

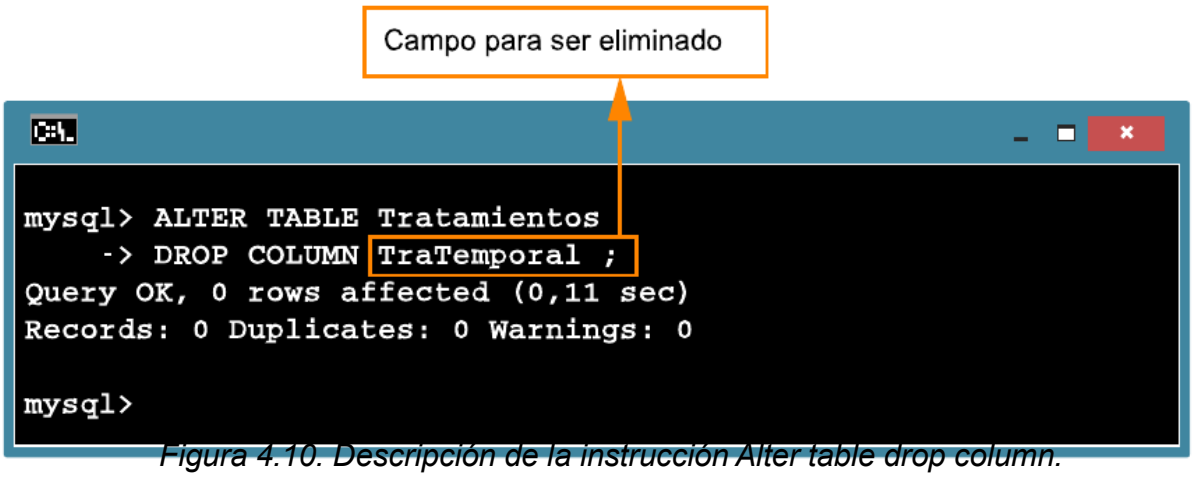
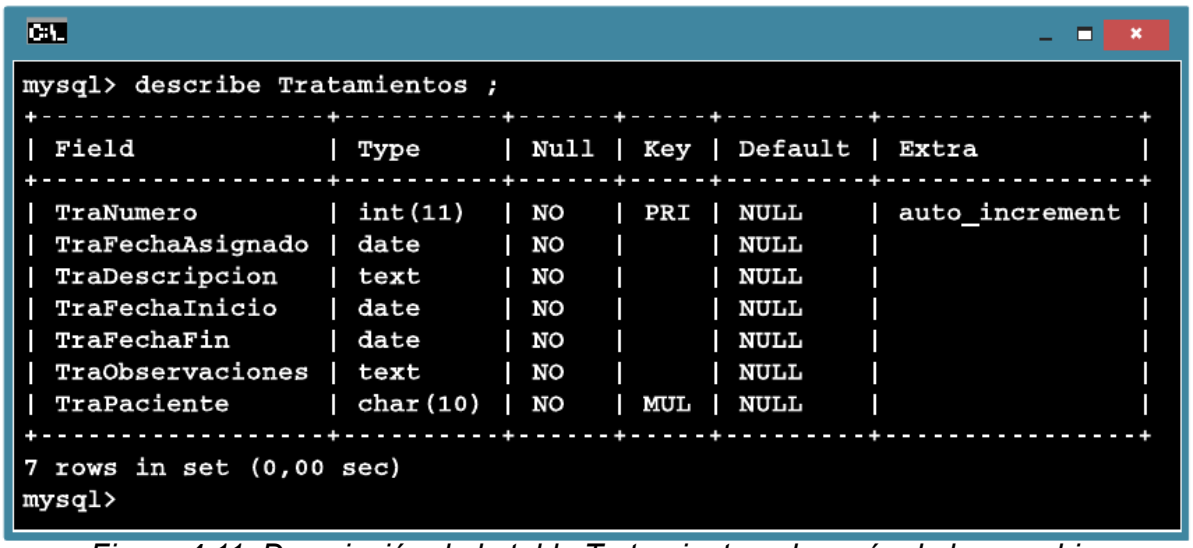


Figura 4.10. Descripción de la instrucción Alter table drop column.

Por último se verificca la nueva estructura de la tabla usando el comando “describe” como se muestra a continuación:



```
mysql> describe Tratamientos ;
+------------------+----------+------+-----+---------+----------------+
| Field            | Type     | Null | Key | Default | Extra          |
+------------------+----------+------+-----+---------+----------------+
| TraNumero        | int(11)  | NO   | PRI | NULL    | auto_increment |
| TraFechaAsignado | date     | NO   |     | NULL    |                |
| TraDescripcion   | text     | NO   |     | NULL    |                |
| TraFechaInicio   | date     | NO   |     | NULL    |                |
| TraFechaFin      | date     | NO   |     | NULL    |                |
| TraObservaciones | text     | NO   |     | NULL    |                |
| TraPaciente      | char(10) | NO   | MUL | NULL    |                |
+------------------+----------+------+-----+---------+----------------+
7 rows in set (0,00 sec)
mysql>
```

Figura 4.11. Descripción de la tabla Tratamientos después de los cambios.

5. Actualización e inserción de registros.

5.1 Inserción de registros.

Para realizar las operaciones de Inserción de Registros se utiliza la sentencia “INSERT INTO” del lenguaje de manipulación de datos (D.M.L.).