

PROYECTO DE USAID

PUENTES PARA EL EMPLEO

Conexión - Universidad de Oriente UNIVO

Centro de Desarrollo de Software

Creación de Conexiones a Bases de Datos

MANUAL DEL PARTICIPANTE

Nivel 1

2019

El Salvador, Centroamérica



La reproducción de este material ha sido posible gracias al generoso apoyo del pueblo de los Estados Unidos a través de la Agencia de los Estados Unidos para el Desarrollo Internacional (USAID). El contenido es responsabilidad exclusiva del [Nombre del Centro de Formación]; no refleja necesariamente la opinión de USAID o del Gobierno de los Estados Unidos

ÍNDICE

[1. INFORMACIÓN DEL MÓDULO 2](#_Toc510183692)

[1.1 Campo de Especialidad 2](#_Toc510183693)

[1.2 Código del Módulo 2](#_Toc510183694)

[1.3 Duración del módulo 2](#_Toc510183695)

[2. DESCRIPCIÓN GENERAL DEL MÓDULO 3](#_Toc510183696)

[2.1 Descripción 3](#_Toc510183697)

[2.2 Competencia del módulo 3](#_Toc510183698)

[2.3 Criterios de evaluación 3](#_Toc510183699)

[2.4 Criterios de promoción 3](#_Toc510183700)

[3. DESARROLLO DE UNIDADES DE APRENDIZAJE 3](file:///C:\Users\Alexiscr\OneDrive%20-%20Programa%20Oportunidades%20FGK\Nivel%201%20C%23%20MVC\Manuales\M.1.10%20Creacion%20de%20conexiones%20a%20bases%20de%20datos.docx#_Toc510183701)

[3.1 Unidad 1: Elementos de Conexion 3](#_Toc510183702)

[3.1.1 Interpretar las partes de una cadena de conexión. 3](#_Toc510183703)

[Actividad Propuesta 1 4](#_Toc510183704)

[3.2.1 Interpretar una cadena de conexión para diferentes gestores. 5](#_Toc510183705)

[3.2.2 Formato de cadena de conexión básica 5](#_Toc510183706)

[Actividad Propuesta 2 6](#_Toc510183707)

[3.3.1 Construir una cadena de conexión 7](#_Toc510183708)

[3.2.2 Formato de cadena de conexión MySql 7](#_Toc510183709)

[3.2.2 Formato de cadena de conexión SQL Express 8](#_Toc510183710)

[Actividad Propuesta 3 9](#_Toc510183711)

[3.4.1 Identifica los errores en una cadena de conexión 10](#_Toc510183712)

[Actividad Propuesta 4 11](#_Toc510183713)

[3. GLOSARIO 12](#_Toc510183714)

[4. PLAN DE EVALUACIONES 14](#_Toc510183715)

[5. JORNALIZACIÓN DEL MÓDULO 15](#_Toc510183716)

[6. FUENTES DE INFORMACIÓN Y MATERIALES DE APOYO 16](#_Toc510183717)

# 1. INFORMACIÓN DEL MÓDULO

## 1.1 Campo de Especialidad

* Desarrollador de Aplicaciones Web usando C# ASP.net con MVC
* Desarrollador de Aplicaciones Móviles con C# Xamarin
* Programador Analista en Lenguaje PHP
* Programador Analista en Lenguaje JAVA.

## 1.2 Código del Módulo

ECBD

## 1.3 Duración del módulo

16 Horas





# 2. DESCRIPCIÓN GENERAL DEL MÓDULO

## 2.1 Descripción

Este módulo está orientado para desarrollar habilidades para la creación de conexiones a bases de datos a partir de análisis de la sintaxis básica de código SQL para crear una conexión segura y funcional, con lo cual se pretende obtener información de bases de datos para poder procesarlos a través de una aplicación.

## 2.2 Competencia del módulo

Elaborar conexiones a bases de datos

## 2.3 Criterios de evaluación

* Identifica la sintaxis de una cadena de conexión
* Identifica los componentes necesarios de una conexión para diferentes gestores de bases de datos
* Elabora la sintaxis de una cadena de conexión
* Detecta los posibles errores en la ejecución de una cadena de conexión

## 2.4 Criterios de promoción

Para dar por aprobado un módulo el participante deberá:

* Haber asistido al 100% de las clases del módulo.
* Desarrollar las actividades que se presentan en el manual.
* Obtener un promedio final mayor o igual a 7.0 en las actividades del módulo.

## 3.1 Elementos de Conexion

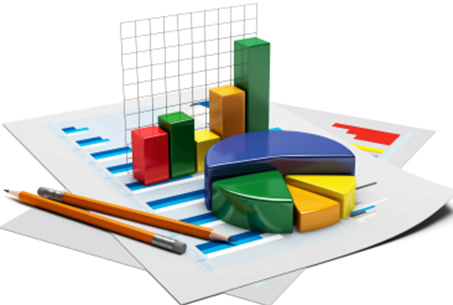
# 3. DESARROLLO DE UNIDADES DE APRENDIZAJE

## 3.1.1 Interpretar las partes de una cadena de conexión.

**¿Qué es un Dato?**

Un dato es la representación de una variable que puede ser cuantitativa o cualitativa, indican un valor que se le asigna a las cosas.

En informática, un dato es la expresión general que va a describir aquellas características de la entidad sobre la que opera. En la estructura de datos, un dato es la más mínima parte de la información.



Los datos pueden ser generados de forma automática y acumulativa con diferentes tipos de programas informáticos o bien tienen que ser siempre ingresados para formar una base de datos.

Los datos que se ingresan en una base pueden ser de diversos tipos, según la información que se acumule en dicha base.

**¿Que es una Base de Datos?**

Una base de datos es un conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso.



Cada base de datos se compone de una o más tablas que guarda un conjunto de datos. Cada tabla tiene una o más columnas y filas.

Las columnas guardan una parte de la información sobre cada elemento que queramos guardar en la tabla, cada fila de la tabla conforma un registro.

**Tipo de almacenamiento de datos:**

El almacenamiento es la manera de cómo se guarda información en una computadora, existen dos formas de almacenar información:

**Almacenamiento Primario:** en el cual interviene la memoria de la computadora, ya que es el espacio que tenemos disponible para guardar toda la información que introducimos; siempre y cuando tenga un suministro constante de energía eléctrica, ya que de no ser así toda esa información se pierde.

**Almacenamiento Secundario:** el cual consiste en almacenar la información en forma permanente hasta que el usuario lo requiera.

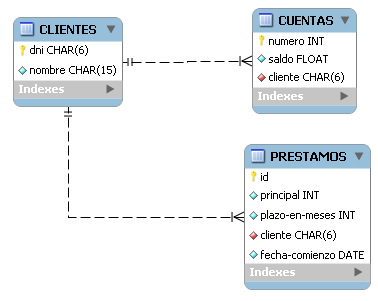
**Tecnlogias de Base de Datos:**

En lo que respecta a las tecnologías de bases de datos se pueden encontrar 2 tipos:

* Las Bases de Datos Relacionales.
* BD No Relacionales.

Bases de Datos Relacionales (SQL):

Una base de datos relacional es una colección de elementos de datos organizados en un conjunto de tablas formalmente descritas desde la que se puede acceder a los datos o volver a montarlos de muchas maneras diferentes sin tener que reorganizar las tablas de la base.



La base de datos relacional fue inventada por E.F. Codd en IBM en 1970.

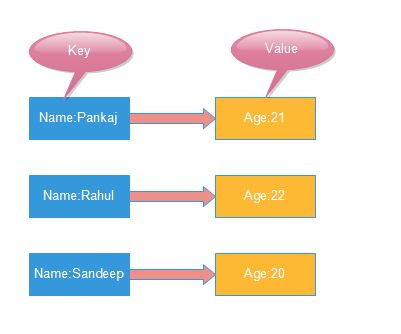
**Ejemplos:**

Algunos ejemplos de base de datos relaciones tenemos:

* Oracle,
* PostgreSQL,
* MySQL,
* Microsoft SQL

Base de Datos No Relaciones.

Este tipo de tecnología funciona con variables o documentos. La colección de datos se parece entre sí pero nunca será igual. No se necesita crear relaciones nuevas cuando queremos añadir atributos nuevos a los objetos de una colección de documentos en una Base de Datos no Relacional.



El tipo de Base de Datos Relacionales usan SQL, mientras que en las Databases no Relacionales se usa JavaScript.

**Ejemplos:**

Algunos ejemplos de base de datos no relacionales tenemos:

* MongoDB,
* REDIS,
* CouchDB.

**Formatos para el almacenamiento de Datos:**

**Formato Binario:**

Habitualmente se piensa en los archivos binarios como una secuencia de bytes, que es lo que implica que dígitos binarios (bits) se agrupen de ocho en ocho comúnmente.



Los archivos binarios que contienen bytes suelen ser interpretados como alguna cosa que no sean caracteres de texto.

Un ejemplo típico son los programas de ordenador compilados; de hecho, las aplicaciones o programas compilados son conocidos como binarios, especialmente entre los programadores.

Pero un archivo binario puede almacenar imágenes, sonido, versiones comprimidas de otros archivos, etc. - en pocas palabras, cualquier tipo de información.

**Formato CSV:**

Los archivos CSV (del inglés comma-separated values) son un tipo de documento en formato abierto sencillo para representar datos en forma de tabla, en las que las columnas se separan por comas (o punto y coma en donde la coma es el separador decimal: Chile, Perú, Argentina, España, Brasil...) y las filas por saltos de línea.

El formato CSV es muy sencillo y no indica un juego de caracteres concreto, ni cómo van situados los bytes, ni el formato para el salto de línea. Estos puntos deben indicarse muchas veces al abrir el archivo, por ejemplo, con una hoja de cálculo.



El formato CSV no está estandarizado. La idea básica de separar los campos con una coma es muy clara, pero se vuelve complicada cuando el valor del campo también contiene comillas dobles o saltos de línea.

**Formato JSON:**

JSON acrónimo de JavaScript Object Notation, es un formato de texto ligero para el intercambio de datos. JSON es un subconjunto de la notación literal de objetos de JavaScript, aunque hoy, debido a su amplia adopción como alternativa a XML, se considera un formato de lenguaje independiente.



JSON acrónimo de JavaScript Object Notation, es un formato de texto ligero para el intercambio de datos. JSON es un subconjunto de la notación literal de objetos de JavaScript aunque hoy, debido a su amplia adopción como alternativa a XML, se considera un formato de lenguaje independiente.

**¿Qué es una Transacción?**

Una transacción puede ser definida como un grupo de tareas. Una sola tarea es la unidad mínima de procesamiento que no puede ser divido en mas.

**Ejemplo:**

Transacciones Bancarias

**Propiedades ACID BD:**

Una transacción es una unidad muy pequeña de un programa y esta puede contener varias tareas de bajo nivel.

Una transacción en un sistema de base de datos debe mantener su Atomicidad, Consistencia, Aislamiento y Durabilidad – comúnmente conocido como propiedades ACID por su acrónimo en ingles de Atomicity, Consistency, Isolation, Durability – con el objetivo de asegurar su precisión e integridad de los datos almacenados.

El concepto de ACID se describe en la sección 4 de la norma ISO/IEC 10026-1:1992. Cada uno de estos atributos se puede medir en un punto de referencia. En general, sin embargo, un gestor de transacciones o monitor está diseñado para realizar el concepto de ACID.

**Atomicity (Atomicidad):**

Esta propiedad indica que una transacción debe ser tratada como unidad atómica, por lo cual todas sus operaciones deberán ejecutarse o de lo contrario ninguna debería hacerlos.

**Ejemplo:**

Considerando la siguiente transacción, se debe transferir 100 de una cuenta X a la cuenta Y.



**Consistencia: (Consistency)**

La base de dados debe permanecer en un estado consistente después de cualquier transacción. Ninguna transacción debe tener ningún efecto adverso sobre los datos que residen en la base de datos.

Si la base de datos se encontraba en un estado consistente antes de la ejecución de una transacción, debe seguir siendo coherente después de la ejecución de la transacción también.

**Aislamiento: (Isolation)**

En un sistema de base de datos donde una transacción se ejecuta simultáneamente y en paralelo, la propiedad de aislamiento indica que todas las transacciones se realizaran y ejecutaran como si fuera la única transacción en el sistema. Ninguna transacción afectara a la existencia de ninguna otra transacción.

**Durabilidad: (Durability)**

Las bases de datos deben ser suficientemente duraderas como para mantener todas sus actualizaciones más recientes, incluso si el sistema falla o se reinicia.

Si una transacción actualiza un fragmento de los datos en una base de datos y se compromete, la base de datos retendrá los datos modificados.

Si una transacción se confirma, pero el sistema falla antes de que los datos se puedan escribir en el disco, los datos se actualizarán una vez que el sistema vuelva a la acción.

**Sistema Gestor de Base de Datos.**

Definimos un Sistema Gestor de Bases de Datos o SGBD, también llamado DBMS (Data Base Management System) como una colección de datos relacionados entre sí, estructurados y organizados, y un conjunto de programas que acceden y gestionan esos datos.

La colección de esos datos se denomina Base de Datos o BD, (DB Data Base).

El SGBD es una aplicación que permite a los usuarios definir, crear y mantener la BD y proporciona un acceso controlado a la misma.

Un Sistema Gestor de Base de Datos debe prestar los siguientes servicios:

* Creación y definición de la BD.
* Manipulación de los datos.
* Acceso controlado a los datos de la BD.
* Mantener la integridad y consistencia.
* Acceso compartido a la BD.
* Mecanismos de respaldo y recuperación.

**SGBD Actuales.**

Entre los SGBD mas utilizados en la actualidad tenemos:

* Microsoft SQL Server.
* MySQL.
* MongoDB
* MariaDB
* Oracle
* PostgreSQL
* SQLite



### Actividad Propuesta 1

ACTIVIDAD #1

Identificar Elementos de Conexión

**OBJETIVO:**

Conocer los diferentes gestores de base de datos mas utilizados en la actualidad.

**INTRUCCIONES:**

- Formar equipos de 3 personas.

- Los participantes deberán investigar sobre un SGBD asignado por el entrenador.

- Deberán preparar una exposición corta de un aproximado de 7 minutos, acorde a los lineamientos entregados por el entrenador.

**Temas:**

* Microsoft SQL Server.
* MySQL.
* MongoDB
* MariaDB
* Oracle
* PostgreSQL
* SQLite



**TIEMPO ESTIMADO:**

50 minutos



**PARTICIPANTES:**

Individual



**MATERIALES**

Hoja con preguntas

Hoja papel bond, lapicero.

## 3.2.1 Interpretar una cadena de conexión para diferentes gestores.

**¿Qué es SQLite?**

SQLite es una biblioteca en proceso que implementa un motor de base de datos SQL de auto contenido, sin servidor, con cero configuraciones y transaccional. Se trata de una base de datos, que es cero configuraciones, lo que significa que, al igual que otras bases, no es necesario configurarla en el sistema.



El motor de SQLite no es un proceso independiente como otras bases de datos, esto permite ligarlo estáticamente o dinámicamente según el requisito con su uso.

SQLite accede a los archivos de almacenamiento directamente.

**Breve Historia de SQLite.**

**2000** - D. Richard Hipp diseñó SQLite para el propósito de ninguna administración requerida para operar un programa.

**2000** - en agosto, SQLite 1,0 liberado con GNU Database Manager.

**2011** - Hipp anunció la adición de la interfaz UNQl a SQLite dB y para desarrollar UNQLite (base de datos orientada a documentos).

Ventajas de SQLite:

SQLite no requiere un proceso o sistema de servidor independiente que funcione (sin servidor).

SQLite viene con configuración cero, lo que significa que no se necesita configuración ni administración.

Una base de datos SQLite completa se almacena en un único archivo de disco de cross-plataform.

SQLite es muy pequeño y ligero, menos de 400KiB completamente configurado o menos de 250KiB con funciones opcionales omitidas.

SQLite es auto-contenido, lo que significa que no hay dependencias externas.

Las transacciones de SQLite son totalmente compatibles con el ácido, lo que permite el acceso seguro desde múltiples procesos o subprocesos.

SQLite soporta la mayoría de las funciones de lenguaje de consulta encontradas en SQL92 (SQL2) Standard.

SQLite está escrito en ANSI-C y proporciona API simple y fácil de usar.

SQLite soporta la mayoría de las funciones de lenguaje de consulta encontradas en SQL92 (SQL2) Standard.

SQLite está disponible en UNIX (Linux, Mac OS-X, Android, iOS) y Windows (Win32, WinCE, WinRT)

**Tipos de Datos en SQLite**

En la mayoría de los motores de bases de datos SQL se utiliza un tipeo rígido, estático.

Con el tecleo estático, el tipo de datos de un valor es determinado por su contenedor - la columna particular en la que se almacena el valor.

SQLite utiliza un sistema de tipo dinámico más general. En SQLite, el tipo de la clase de un valor está asociado al valor mismo, no a su contenedor.

El sistema de tipo dinámico de SQLite es compatible de forma inversa con los sistemas de tipo estático más comunes de otros motores de bases de datos, en el sentido de que las instrucciones SQL que funcionan en las bases con tipo estático deben funcionar de la misma manera en SQLite. Sin embargo, la mecanografía dinámica en SQLite le permite hacer cosas que no son posibles en bases de datos tradicionales de tipo rígido.

* **NULL:** el valor es nulo (NULL).
* **INTEGER:** el valor en un entero con signo.
* **REAL:** El valor es un valor de punto flotante.
* **TEXT:** El valor es una cadena de texto
* **BLOB:** El valor es un BLOB de datos, almacenado exactamente como fue ingresado.

**Lenguajes de los SGBD.**

Los lenguajes van a permitir al administrador de la BD especificar los datos que componen la BD, su estructura, las relaciones que existen entre ellos, las reglas de integridad, los controles de acceso, las características de tipo físico y las vistas externas de los usuarios.

**Lenguaje de definición de datos (LDD o DDL):** Proporcionan la estructura y métodos de almacenamiento para acceder a los datos desde el sistema de base de datos.

* ALTER
* CREATE
* DROP

**Lenguaje de manipulación de datos (LMD o DML):** Permite a los usuarios manipular (añadir / modificar/ eliminar) los datos.

* SELECT
* UPDATE
* DELETE
* INSERT INTO

**DML Comandos:**

**SELECT:** La sentencia SELECT permite consultar los datos almacenados en una tabla de base de datos.

*Sintaxis*:

**SELECT \* FROM** Persona**;**

**INSERT:** Esta sentencia permite agregar uno o más registros a una tabla en una base de datos.

Sintaxis:

**INSERT INTO** 'Persona' ('Id','Nombre','Apellido') **VALUES** (1,'Juan','Perez');

**UPDATE:** Una sentencia UPDATE de SQL es utilizada para modificar los valores de un conjunto de registros existentes en una tabla.

*Sintaxis*:

**UPDATE** Persona **SET** Nombre **=** 'Pedro' **WHERE** Id **=** 1**;**

**DELETE:** Sentencia para borrar uno o más registros existentes en una tabla.

Sintaxis:

**DELETE FROM** Persona **WHERE** Id = '1';

### Actividad Propuesta 2

ACTIVIDAD # 2

Interpretar cadenas de conexión para diferentes gestores

**OBJETIVO:**

Identificar y listar los elementos Generales de Conexión en una cadena de conexión por cada gestor de Base de Datos visto en el capitulo anterior.

**PREGUNTAS GUÍAS:**

1. ¿Qué es una cadena de conexión o paquete de conexión en SQLite?
2. ¿Cuáles elementos tiene la cadena de conexión generalmente o propiedades de un paquete de conexion?
3. ¿Mencione al menos 3 gestores de Base de Datos?
4. ¿Mencione los pasos a seguir para implementar SQLite en Xamarin?

**INTRUCCIONES:**

1. En forma individual, cada participante deberá contestar las interrogantes planteadas
2. Las respuestas deberán ser escritas de manera legible con tinta azul.
3. No se permitirá el uso de material de apoyo y teléfono celular.



**TIEMPO ESTIMADO:**

30 minutos



**PARTICIPANTES:**

Individual



**MATERIALES**

Hoja con preguntas

Hoja papel bond, lapicero.

## 3.3.1 Construir una cadena de conexión

Basados en el capitulo anterior podemos construir nuestra propia cadena de conexión con los parámetros generales, realizaremos la cadena de conexión para Sql y Mysql en C#.

## 3.3.2 Formato de cadena de conexión MySql

“Recordar antes de iniciar debemos descargar el MySQL CONECTOR/NET que provee el la conexion de C# con MySQL ”

Al iniciar la clase donde colocaremos nuestra cadena de conexión haremos lo siguien:

using MySql.Data.MySqlClient;

Que es la declaración que usaremos una base de datos MySql.

Para hacer la conexion crearemos lo que se llama conection string o cadena de conexion, usaremos: MySQLConecctionStringBuilder para lograrlo.

MySqlConnectionStringBuilder builder = new MySqlConnectionStringBuilder();

builder.Server = "localhost";

builder.UserID = "root";

builder.Password = "";

builder.Database = "mydb01";

La cadena de conexion sirve para especificar al conector aque instancia de MySQL nos conectaremos y para ello proporcionamos el usuario y password, ademas de la base de datos en la cual trabajaremos.

El siguiente paso es crear una conexion utilizando MySqlConnection y convertiendo la cadena de conexion de objeto a cadena usando builder.toString(), despues creamos un comando utilizando la instancia de conexion, asignamos el CommandText que es codigo SQL, despues abrimos la conexion y ejecutamos la consulta, se usa ExecuteNonQuery para comandos que no devuelven resultados como Insert, Delete, Update.

MySqlConnection conn = new MySqlConnection(builder.ToString());

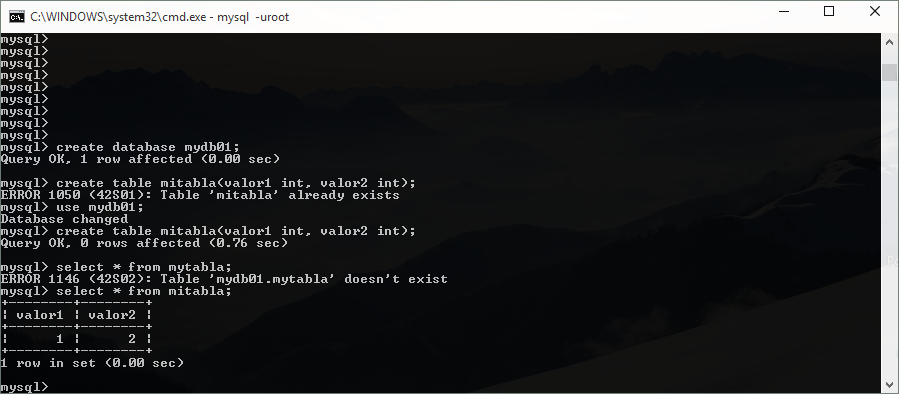
MySqlCommand cmd = conn.CreateCommand();

cmd.CommandText = "INSERT INTO mitabla (valor1,valor2) value (1,2)";

conn.Open();

cmd.ExecuteNonQuery();

Como resultado final al consultar la BD nos mostrara lo siguiente:

****

## 3.3.3 Formato de cadena de conexión SQL Express

Al iniciar la clase donde colocaremos nuestra cadena de conexión haremos lo siguien:

using System.Data.SqlClient;

Que es la declaración que usaremos una base de datos SqlExpress.

Para hacer la conexion crearemos lo que se llama conection string o cadena de conexion, usaremos: SqlConnection para lograrlo.

MySqlConnectionStringBuilder builder = new MySqlConnectionStringBuilder();

SqlConnection conexion = new SqlConnection("server=DIEGO-PC\\SQLEXPRESS ; database=base1 ; integrated security = true");

conexion.Open();

MessageBox.Show("Se abrió la conexión con el servidor SQL Server y se seleccionó la base de datos");

La cadena de conexión sirve para especificar al conector a que instancia de SQL nos conectaremos y para ello proporcionamos el tipo de seguridad que usaremos, en este caso autenticación de sistema, ademas de la base de datos en la cual trabajaremos.

Otras variantes de conexión serian:

Seguridad estándar: Server=myServerAddress;Database=myDataBase;User Id=myUsername;  
Password=myPassword;

Conectar a una instancia Sql: Server=myServerName\myInstanceName;Database=myDataBase;User Id=myUsername;

Password=myPassword;

Conectarse mediante una dirección IP: Data Source=190.190.200.100,1433;Network Library=DBMSSOCN;  
Initial Catalog=myDataBase;User ID=myUsername;Password=myPassword;

## 3.4.1 Identifica los errores en una cadena de conexión

Una vez hecha la cadena de conexión parte importante del proceso es la identificación de errores en la conexión, para ello nos guiaremos por los codigos de error que nos indicaran cual componente esta causando el error.

**Listado de codigos de errores comunes:**

SqlException.Number = 11001: Host Desconocido.

SqlException.Number = 18456: Error de inicio de sesión.

ConnectRetryCount: El valor predeterminado es 1, El intervalo es de 0 a 255.

ConnectRetryInterval: El valor predeterminado es 1 segundo. El intervalo es de 1 a 60.

“Los valores del punto anterior deben cumplir el siguiente orden: Connection Timeout = ConnectRetryCount \* ConnectionRetryInterval”

### Actividad Propuesta 3

ACTIVIDAD # 3

Crear cadenas de conexión para diferentes gestores

**OBJETIVO:**

Crear cadenas de conexión por cada gestor de Base de Datos visto en el capitulo anterior.

**PREGUNTAS GUÍAS:**

1. Cree un script de conexión básico para Sql usando la BD prueba1
2. Cree un script de conexión básico para MySql usando la BD prueba1

**INTRUCCIONES:**

1. En forma individual, cada participante deberá Resolver los ejercicios planteados
2. Debera dejar evidencia en forma de captura de pantalla y el script de la resolución de los ejercicios.
3. No se permitirá el uso de material de apoyo y teléfono celular.



**TIEMPO ESTIMADO:**

60 minutos



**PARTICIPANTES:**

Individual



**MATERIALES**

Hoja de ejercicios

PC.

Software Visual Studio

# 3. GLOSARIO

Base de Datos:

Una base de datos o banco de datos es un conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso.

Cadena de conexión:

es una cadena que contiene información acerca de una fuente de datos (generalmente un motor de base de datos), además de

incluir la información necesaria para conectarse a la misma.

SQL:

SQL (por sus siglas en inglés Structured Query Language; en español lenguaje de consulta estructurada) es un lenguaje específico del dominio que da acceso a un sistema de gestión de bases de datos relacionales que permite especificar diversos tipos de operaciones en ellos

MYSQL:

es un sistema de gestión de bases de datos relacional desarrollado bajo licencia dual: Licencia pública general/Licencia comercial por Oracle Corporation y está considerada como la base datos de código abierto más popular del mundo

ERROR: Diferencia entre el valor real o exacto de una magnitud y el que resulta del cálculo hecho por una persona o por una máquina.

# 4. PLAN DE EVALUACIONES

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Item | Actividad y  Tema evaluado | Tipo de evaluación | Semana | Ponderación |
| 1 | Actividad 1 | Prueba escrita | 1 | 20% |
| 2 | Actividad 2 | Archivo de ejecución del script | 2 | 40% |
| 3 | Actividad 3 | Prueba Practica | 3 | 40% |
| **TOTAL** | | | | **100%** |

# 5. JORNALIZACIÓN DEL MÓDULO

|  |  |  |  |
| --- | --- | --- | --- |
| SEMANA | SUB COMPETENCIA | ESCENARIO DE APRENDIZAJE | DURACIÓN EN HORAS |
| 1 | Tema 1.1 |  | 4 horas |
| 2 | Tema 1.2 |  | 10 horas |
| 3 | Tema 1.2 |  | 10 horas |
| **TOTAL DE HORAS DEL MÓDULO** | |  | **24 HORAS** |

# 

# 6. FUENTES DE INFORMACIÓN Y MATERIALES DE APOYO