



Punto de control: Resultado

Estimado/a alumno/a:

Rango de NOTA:

0 a 5 - 1
6 a 14 - 2
15 a 17 - 3
18 a 20 - 4
21 a 22 - 5
23 a 24 - 6
25 a 26 - 7
27 a 28 - 8
29 - 9
30 - 10

2P - Variables SQL - CORRECTAS: 0 de 2

1) Variables SQL:

Permiten almacenar un valor y recuperarlo más adelante para emplearlos en otras sentencias. **CORRECTA**
no, específicas de batch o procedimiento y deben ser declaradas antes
Son específicas de cada consulta y son ejecutadas automáticamente al realizar la conexión.

Comienzan con "@" (arroba) seguido del nombre (sin espacios), dicho nombre no puede contener cualquier carácter. **INCORRECTA** **si puede tener cualquier caracter**

Una variable debe ser declarada al final para usarse en el código anterior. **se declara antes de usarse**

Más de una respuesta de las ANTERIORES es correcta.

Todas las respuestas ANTERIORES son correctas.

Ninguna de TODAS las respuestas ANTERIORES es correcta.

2) Variables SQL:

sentencias después de ser declarada

Permiten almacenar un valor y recuperarlo en sentencias anteriores a ser declarada para emplearlos en otras sentencias.

no son automáticas ni ligadas a una conexión

Son específicas de cada consulta y son ejecutadas automáticamente al realizar la conexión.

Comienzan con "@" (arroba) seguido del nombre (sin espacios), dicho nombre no puede contener cualquier carácter. **INCORRECTA**

Una variable debe ser declarada al final para usarse en el código anterior.

Más de una respuesta de las ANTERIORES es correcta.

NINGUNA ES CORRECTA



1) Variables SQL - Características:

- Puede declarar varias variables en una misma sentencia. **CORRECTA**

Existen variables globales en SQL Server.

Una variable declarada existe dentro de todo el entorno de trabajo.

Una variable a la cual no se le ha asignado un valor contiene "0" (Cero).

Más de una respuesta de las ANTERIORES es correcta.

Todas las respuestas ANTERIORES son correctas.

Ninguna de TODAS las respuestas ANTERIORES es correcta.

2) Variables SQL - Características:

Podés declarar una o varias variables, en una o varias sentencias

- Solo se puede declarar varias variables en una misma sentencia.

- Existen variables globales en SQL Server. **No existen variables definidas por el usuario que sean globales**

- Una variable declarada existe dentro de todo el entorno de trabajo. **no, solo local**

- Una variable a la cual no se le ha asignado un valor contiene "null". **CORRECTA**

Más de una respuesta de las ANTERIORES es correcta.

Todas las respuestas ANTERIORES son correctas.

Ninguna de TODAS las respuestas ANTERIORES es correcta.

2P - Procedimientos Almacenados - CORRECTAS: 1 de 2

1) Procedimientos Almacenados:

almacena lógica T-SQL para ser ejecutado de forma separada

- Es un grupo de sentencias T-SQL compiladas dentro de un plan de una query. **INCORRECTA**

- Son un método de encapsular tareas repetitivas que no involucran variables definidas por el motor de SQL para cálculos intermedios. **si involucran variables definidas por sql**

- Son variables de control de flujo de ejecución, para la implementación de bloques condicionales o repetitivos. **no, es la definición de estructuras de control**

Son querys y variables que encapsulan código para su reutilización.

Un procedimiento puede contener consultas y variables, pero no se define como "queries y variables"

Más de una respuesta de las ANTERIORES es correcta.

Todas las respuestas ANTERIORES son correctas.

Ninguna de TODAS las respuestas ANTERIORES es correcta. **CORRECTA**

2) Procedimientos Almacenados:



repetitivos.

Son módulos o rutinas que encapsulan código para su reutilización.

- Mas de una respuesta de las ANTERIORES es correcta. **CORRECTA**

Todas las respuestas ANTERIORES son correctas.

Ninguna de TODAS las respuestas ANTERIORES es correcta.

2P - Procedimientos Almacenados - Características - CORRECTAS: 2 de 2

1) Procedimientos Almacenados - Características:

- Aceptan parámetros de entrada. **CORRECTA** puede usar RETURN, pero solo devuelve un valor escalar
 - Devuelven mas de un valor de retorno (escalar) que indica el éxito o falla de su ejecución.
 - No pueden llamar a otros SPs (o sea dentro de un SP no se puede llamar a otros SP).
 - No pueden devolver valores en la forma de parámetros de salida.
- Mas de una respuesta de las ANTERIORES es correcta.
- Todas las respuestas ANTERIORES son correctas.
- Ninguna de TODAS las respuestas ANTERIORES es correcta.

2) Procedimientos Almacenados - Características:

- No aceptan parámetros de entrada.
 - Devuelven un valor de retorno (escalar) que indica el éxito o falla de su ejecución. **CORRECTA**
 - No pueden llamar a otros SPs (o sea dentro de un SP no se puede llamar a otros SP).
 - No pueden devolver valores en la forma de parámetros de salida.
- Mas de una respuesta de las ANTERIORES es correcta.
- Todas las respuestas ANTERIORES son correctas.
- Ninguna de TODAS las respuestas ANTERIORES es correcta.

2P - Procedimientos Almacenados - Ventajas - CORRECTAS: 1 de 1

1) Procedimientos Almacenados - Ventajas:

Encapsulan la lógica de negocio y crean piezas de código reutilizable por la aplicación.

Todas las aplicaciones pueden usar los mismos procedimientos para asegurar un acceso consistente a los datos.

Evitan la exposición de los detalles de las tablas al usuario, haciendo innecesario el acceso a las tablas en



Todas las respuestas ANTERIORES son correctas. **CORRECTA**

Ninguna de TODAS las respuestas ANTERIORES es correcta.

2P - Procedimientos Almacenados - Parametros - CORRECTAS: 1 de 2

1) Procedimientos Almacenados - Parametros:

`CREATE PROCEDURE sp_ejemplo @nombre
VARCHAR(50), @resultado INT OUTPUT`

Cuando se definen parámetros de entrada y salida, estos siempre van finalizados por el signo @, con nombre del parámetro anteponiéndose, y luego una designación del tipo de dato.

obligatorio OUTPUT para devolver información

Los parámetros de salida deben incluir la palabra clave OUTPUT para diferenciar de los de entrada.

Los Procedimientos son más flexibles cuando le incluimos parámetros en su definición, de modo tal que pueda crear una lógica de comportamiento más genérica. **Es el objetivo del uso de parámetros**

permiten hacia

Los parámetros de entrada impiden pasar información al SP. Estos valores son ingresados como variables locales al momento de la ejecución dentro del procedimiento almacenado.

Mas de una respuesta de las ANTERIORES es correcta. **CORRECTA**

Todas las respuestas ANTERIORES son correctas.

Ninguna de TODAS las respuestas ANTERIORES es correcta.

2) Procedimientos Almacenados - Parametros:

Cuando se definen parámetros de entrada y salida, estos siempre van finalizados por el signo @, con nombre del parámetro anteponiéndose, y luego una designación del tipo de dato.

Los parámetros de salida deben incluir la palabra clave OUTPUT para diferenciar de los de entrada.

Los Procedimientos nunca pueden incluir parámetros en su definición, de modo tal que pueda crear una lógica de comportamiento más genérica. **si pueden tener parámetros**

permiten

Los parámetros de entrada impiden pasar información al SP. Estos valores son ingresados como variables locales al momento de la ejecución dentro del procedimiento almacenado. **INCORRECTA**

Mas de una respuesta de las ANTERIORES es correcta.

Todas las respuestas ANTERIORES son correctas.

Ninguna de TODAS las respuestas ANTERIORES es correcta. **CORRECTA**

2P - Tipos de funciones - CORRECTAS: 0 de 2

1) Tipos de funciones - Funciones Escalares:

Devuelven un solo tipo de valor definido en la cláusula "Returns".

Este tipo de funciones es sintácticamente similar a funciones del tipo COUNT(), MAX(). **INCORRECTA**



Todas las respuestas ANTERIORES son correctas.

Ninguna de TODAS las respuestas ANTERIORES es correcta.

2) Tipos de funciones - Funciones Tabulares en Línea:

Devuelve una tabla que es el resultado de una única sentencia SELECT. **CORRECTA**

X Devuelven un solo tipo de valor definido en la cláusula "Returns". **devuelven una tabla completa**

X Es similar a un procedimiento, salvo que puede referenciarse como si se tratara de una tabla.

INCORRECTA **no es igual a un procedimiento**

X Este tipo de funciones es sintácticamente similar a funciones del tipo COUNT(), MAX(). **count y max son funciones agregadas escalares**

Más de una respuesta de las ANTERIORES es correcta.

Todas las respuestas ANTERIORES son correctas.

Ninguna de TODAS las respuestas ANTERIORES es correcta.

2P - Práctica - Procedimientos y Funciones - CORRECTAS: 2 de 4

1) Dado el siguiente código de Store Procedure de SQL indicar el enunciado correcto.

```

CREATE PROCEDURE dbo.InsertarOActualizarCelular
    @Marca NVARCHAR(50),
    @Modelo NVARCHAR(50),
    @Memoria NVARCHAR(20),
    @Procesador NVARCHAR(50)
AS
BEGIN
    SET NOCOUNT ON;
    IF EXISTS (SELECT 1 FROM dbo.Inventariocelulares WHERE Modelo = @Modelo)
    BEGIN
        IF EXISTS (SELECT 1 FROM dbo.Inventariocelulares WHERE Modelo = @Modelo AND Marca <> @Marca)
        BEGIN
            THROW 50002, 'Error: El Modelo ingresado ya existe con una Marca diferente.', 1;
        END
        ELSE
        BEGIN
            UPDATE dbo.Inventariocelulares
            SET Memoria = @Memoria,
                Procesador = @Procesador
            WHERE Modelo = @Modelo;
        END
    END
    ELSE
    BEGIN
        INSERT INTO dbo.Inventariocelulares (Marca, Modelo, Memoria, Procesador)
        VALUES (@Marca, @Modelo, @Memoria, @Procesador);
    END
END;

```

Confeccionar un SP que permita insertar un nuevo dispositivo en la tabla dbo.Inventariodispositivos. Si ya existe un registro con el mismo Marca y Modelo, debe actualizar todos los campos (Memoria, Procesador y Marca). Si el Modelo ya está registrado pero la Marca es diferente, debe actualizar solo los valores de Memoria y Procesador sin generar ningún error.

Confeccionar un SP que permita insertar un nuevo celular en la tabla dbo.Inventariocelulares. Si ya existe un registro con el mismo Modelo pero con una Marca diferente, se debe permitir la inserción de un nuevo registro sin



procesador, y si el valor ingresado es una Marca y Modelo diferente a los ya cargados se debe cargar el nuevo registro, y si los datos son duplicados se generar el error. **CORRECTA**

Confeccionar un SP que permita insertar un nuevo celular en la tabla dbo.Inventariocelulares. Si ya existe un registro con el mismo Modelo, se debe actualizar solo el campo Marca y dejar intactos los valores de Memoria y Procesador. Si se intenta insertar un nuevo registro con un Modelo y Marca duplicados, el procedimiento debe permitir la inserción sin lanzar ningún error.

Ninguno de los enunciados ANTERIORES es correcto. Código Incorrecto

2) Dado el siguiente código de Store Procedure de SQL indicar el enunciado correcto.

```
CREATE TABLE dbo.NumeroPar (
    nro SMALLINT
);
CREATE PROCEDURE InsertarNumerosPares
AS
BEGIN
    DECLARE @i SMALLINT;
    SET @i = 4;
    WHILE @i <= 22
    BEGIN
        IF @i NOT IN (8, 12, 18)
        BEGIN
            INSERT INTO dbo.NumeroPar (nro) VALUES (@i);
        END
        SET @i = @i + 2;
    END
END;
```

Creación de Tabla dbo.NumeroPar y Store Procedure que permita insertar números pares del 4 al 22 en una tabla con el nombre dbo.NumeroPar (nro smallint). Excepto los números 8, 12 y 18. La tabla debe ser creada en el procedimiento. **INCORRECTA** [la tabla se crea afuera](#)

Creación de Tabla dbo.NumeroPar y Store Procedure que permita insertar números pares del 4 al 21 en una tabla con el nombre dbo.NumeroPar (nro smallint). Excepto los números 8, 12 y 18. La tabla debe ser creada fuera del procedimiento. [llega hasta 22](#)

Creación de Tabla dbo.NumeroPar y Store Procedure que permita insertar números pares del 4 al 21 en una tabla con el nombre dbo.NumerosPar (nro smallint). Excepto los números 8, 12 y 18. La tabla debe ser creada fuera del procedimiento. [lo mismo](#)

Creación de Tabla dbo.NumeroPar y Store Procedure que permita insertar números pares del 4 al 21 en una tabla con el nombre dbo.NumeroPar (nro smallint). Excepto los números 8, 12 y 18. La tabla debe ser creada fuera del procedimiento. [lo mismo](#)

Ninguno de los enunciados ANTERIORES es correcto. Código Incorrecto **CORRECTA**

3) Dado el siguiente código de Store Procedure de SQL indicar el enunciado correcto.



```

BEGIN
DECLARE @i SMALLINT;
SET @i = 4;
WHILE @i < 22
BEGIN
    IF @i NOT IN (8, 12, 18)
    BEGIN
        INSERT INTO dbo.NumeroPar (nro) VALUES (@i);
    END
    SET @i = @i + 2;
END
END;

```

Creacion de Tabla dbo.NumeroPar y Store Procedure que permita insertar números pares del 4 al 22 en una tabla con el nombre dbo.NumeroPar (nro smallint). Excepto los números 8, 12 y 18. La tabla debe ser creada fuera del procedimiento.

Creacion de Tabla dbo.NumeroPar y Store Procedure que permita insertar números pares del 4 al 22 en una tabla con el nombre dbo.NumeroPar (nro smallint). Excepto los números 8, 12 y 18. La tabla debe ser creada en el procedimiento.

Creacion de Tabla dbo.NumeroPar Store Procedure que permita insertar números pares del 4 al 21 en una tabla con el nombre dbo.NumeroPar (nro smallint). Excepto los números 8, 12 y 18. La tabla debe ser creada fuera del procedimiento. **CORRECTA**

X Creacion de Tabla dbo.NumeroPar y Store Procedure que permita insertar números pares del 4 al 21 en una tabla con el nombre dbo.NumeroPar (nro smallint). Excepto los números 8, 12 y 18. La tabla debe ser creada fuera del procedimiento. **INCORRECTA**

Ninguno de los enunciados ANTERIORES es correcto. Codigo Incorrecto

4) Dado el siguiente código de Store Procedure de SQL indicar el enunciado correcto.

```

CREATE TABLE dbo.NumeroPar (
    nro SMALLINT
);
CREATE PROCEDURE InsertarNumeroPar
    @numero SMALLINT
AS
BEGIN
    IF @numero % 2 <> 0
    BEGIN
        THROW 50001, 'El número ingresado no es par. Inserción fallida.', 1;
    END
    ELSE
    BEGIN
        INSERT INTO dbo.NumeroPar (nro) VALUES (@numero);
    END
END;
EXEC InsertarNumeroPar @numero = 4;

```



sólo se deberá insertar si el número es par. De lo contrario lanzar una excepción. La tabla debe ser creada en el procedimiento.

Creacion de Store Procedure que inserte a la tabla dbo.NumeroPar el número ingresado automaticamente, pero sólo se deberá insertar si el número es par. De lo contrario lanzar una excepción. La tabla debe ser creada fuera del procedimiento.

Creacion de Store Procedure que inserte a la tabla dbo.NumeroPar el número ingresado por parámetro, pero sólo se deberá insertar si el número es par. De lo contrario lanzar una excepción. La tabla debe ser creada fuera del procedimiento. **CORRECTA**

Ninguno de los enunciados ANTERIORES es correcto. Código Incorrecto

2P - Practica - Triggers - CORRECTAS: 1 de 1

1) ¿Qué hace el siguiente trigger INSTEAD OF DELETE en la tabla Employees?

```
CREATE TRIGGER trg_PreventDelete
ON Employees
INSTEAD OF DELETE
AS
BEGIN
    PRINT 'No se permite eliminar empleados de esta tabla.';
END;
```

El trigger permite la eliminación de empleados, pero registra la acción en una tabla de auditoría.

El trigger impide la eliminación de registros en la tabla Employees y muestra un mensaje de error.
CORRECTA

El trigger permite la eliminación de empleados y muestra un mensaje de error después de la eliminación.

El trigger reemplaza el contenido de la tabla Employees por una nueva tabla temporal y muestra un mensaje de error.

Ninguno de los enunciados ANTERIORES es correcto.

2P - Seguridad - General SQLServer - CORRECTAS: 0 de 1

1) Entidades de seguridad y seguridad de objetos de base de datos:

El ámbito de influencia de una entidad de seguridad no depende del ámbito de su definición: Windows, servidor o base de datos; y de si la entidad de seguridad es indivisible o es una colección.

Un Inicio de sesión de Windows es un ejemplo de entidad de seguridad divisible y un Grupo de Windows es un ejemplo de una del tipo colección. **INCORRECTA**



Todas las respuestas ANTERIORES son correcta.

Ninguna de TODAS las respuestas ANTERIORES es correcta. **CORRECTA**

2P - Seguridad - Usuarios y Esquemas - CORRECTAS: 0 de 2

1) Usuario y esquema dbo:

dbo no es un usuario

- +** El usuario dbo es una tabla de seguridad de usuario especial que hay en cada base de datos.
- ×** Todos los administradores de SQL Server, los miembros del rol fijo de servidor sysdba, el inicio de sesión sa y los propietarios de la base de datos especifican las bases de datos como el usuario dbo. **INCORRECTA**
- +** El usuario dbo tiene todos los permisos en la base de datos master y se limitar o desactivar. **master no, ni eliminar ni desactivar**

El rol fijo de base de datos db_owner no es lo mismo que la cuenta de usuario que se registra como el propietario de la base de datos. **CORRECTA**

Mas de una respuesta de las ANTERIORES es Correcta.

Todas las respuestas ANTERIORES son correcta.

Ninguna de TODAS las respuestas ANTERIORES es correcta.

2) Inicio de sesión sa:

base de datos no, sino sql server

- ×** La cuenta sa se crea de forma predeterminada cuando se instala una base de datos. **INCORRECTA**
- +** La base de datos predeterminada de la cuenta sa es msdb. **no, es master**
- +** El inicio de sesión, la cuenta sa es miembro del rol fijo de nivel de servidor sysdba.
- +** Este inicio de sesión, la cuenta sa tiene todos los permisos en el servidor y puede limitarse. **no se puede limitar**

Mas de una respuesta de las ANTERIORES es Correcta.

Todas las respuestas ANTERIORES son correcta.

Ninguna de TODAS las respuestas ANTERIORES es correcta. **CORRECTA**

2P - Seguridad - ROLES - CORRECTAS: 1 de 1

1) Rol público de base de datos y de servidor:

Cada inicio de sesión pertenece al rol fijo de servidor public y cada usuario de base de datos pertenece al rol de base de datos public.

Cuando a un usuario o inicio de sesión no se le han concedido ni denegado permisos concretos para un elemento protegible, hereda los permisos para ese elemento concedidos a public.

El rol fijo de servidor public y el de base de datos public no pueden quitarse, pero se puede revocar los permisos de los roles public.



Todas las respuestas ANTERIORES son correcta. **CORRECTA**

Ninguna de TODAS las respuestas ANTERIORES es correcta.

2P - Seguridad - Elementos protegibles - CORRECTAS: 1 de 1

1) Elementos protegibles => Ámbito protegible: servidor.

Grupo de disponibilidad, Punto de conexión, Iniciar sesión, Rol del servidor, Enlace de servicio remoto.

Grupo de disponibilidad, Clave simétrica, Iniciar sesión, Rol del servidor, Base de datos.

Grupo de disponibilidad, Punto de conexión, Certificate, Rol del servidor, (Base de datos) Rol.

Grupo de disponibilidad, Punto de conexión, Iniciar sesión, Rol del servidor, Base de datos. **CORRECTA**

Mas de una respuesta de las ANTERIORES es Correcta.

Todas las respuestas ANTERIORES son correcta.

Ninguna de TODAS las respuestas ANTERIORES es correcta.

2P - Seguridad - Jerarquía de Cifrado - CORRECTAS: 0 de 1

1) Jerarquía de cifrado => Conceptos:

Las claves maestras de base de datos se protegen mediante la clave asimétricas maestra de servicio.

La clave maestra de servicio y todas las claves maestras de base de datos son claves asimétricas.

La clave maestra de servicio y todas las claves maestras de base de datos son claves simétricas.

CORRECTA

El módulo de Administración extensible de claves (EKM) mantiene las claves asimétricas o simétricas fuera de SQL Server. **INCORRECTA**

Mas de una respuesta de las ANTERIORES es Correcta.

Todas las respuestas ANTERIORES son correcta.

Ninguna de TODAS las respuestas ANTERIORES es correcta.

2P - Teoria - Triggers - CORRECTAS: 0 de 2

1) Triggers: Nivel de base de datos

Responden a la necesidad de garantizar la integridad y consistencia de los datos dentro de nuestras tablas de usuario.

No ayudan a mantener las reglas de diseño de nuestra base de datos.

Nos proporcionarán mecanismos para garantizar que nuestra base de datos está diseñada e implementada



TODAS LAS RESPUESTAS ANTERIORES SON CORRECTAS. CORRECTA

Ninguna de TODAS las respuestas ANTERIORES es correcta.

2) Triggers: Nivel de base de datos

Responden a la necesidad de garantizar la integridad y disponibilidad de los datos dentro de nuestras tablas de usuario.

- No ayudan a mantener las reglas de actualización de nuestra base de datos. INCORRECTA

Nos proporcionarán mecanismos para garantizar que nuestra base de datos está diseñada e implementada de acuerdo a los estándares de la instalación por defecto.

Tienen dos alcances diferenciados, a nivel de servidor y a nivel de base de datos. CORRECTA

Más de una respuesta de las ANTERIORES es correcta.

Todas las respuestas ANTERIORES son correctas.

Ninguna de TODAS las respuestas ANTERIORES es correcta.

2P - Transacciones y Bloqueos - CORRECTAS: 1 de 2

1) Transacciones Explícitas:

Es aquella en la que se define al final de la transacción para validar la propiedad ACID.

Reciben el nombre de tablas definidas por el usuario.

- Se inician y finalizan precisamente mediante comandos como BEGIN TRANSACTION, COMMIT, y ROLLBACK. CORRECTA

Deben realizarse un control automático de ejecución para cumplir con la propiedad ACID.

Más de una respuesta de las ANTERIORES es correcta.

Todas las respuestas ANTERIORES son correctas.

Ninguna de TODAS las respuestas ANTERIORES es correcta.

2) Características ACID: => Aislamiento (Isolation)

Esta propiedad asegura que una transacción se ejecute de manera completa o no se ejecute en absoluto.

Garantiza que una transacción lleve a la base de datos de un estado válido a otro estado válido.

Asegura que las transacciones concurrentes se ejecuten de manera que no afecten entre sí. CORRECTA

Garantiza que los cambios realizados por la transacción se guardan de forma que no se perderán, garantizando la integridad de los datos.

Más de una respuesta de las ANTERIORES es correcta.

- Todas las respuestas ANTERIORES son correctas. INCORRECTA



1) Características de los cursos - instrucciones:

DECLARE => define el comportamiento de desplazamiento y la consulta utilizada para generar el conjunto de atributos sobre el que opera el cursor.

OPEN => llena el conjunto de atributos que deben ser analizados para el comportamiento de las sentencias.

FETCH => devuelve el valor de un atributo de la tabla del conjunto de resultados.

CLOSE => libera el conjunto de resultados actual asociado al cursor. **CORRECTA**

Más de una respuesta de las ANTERIORES es correcta.



Todas las respuestas ANTERIORES son correctas. **INCORRECTA**

Ninguna de TODAS las respuestas ANTERIORES es correcta.

2P - Practica - Cursos - CORRECTAS: 1 de 1

1) ¿Qué sentencia es necesaria para abrir un cursor en SQL?



OPEN nombre_del_cursor **CORRECTA**

ACTIVATE nombre_del_cursor

START nombre_del_cursor

EXECUTE nombre_del_cursor

Ninguno de los enunciados ANTERIORES es correcto

RESUMEN

TOTAL CORRECTAS: 14 de 30 **INSUFICIENTE**

2P - Variables SQL	CORRECTAS: 0 de 2
2P - Variables SQL - Características	CORRECTAS: 2 de 2
2P - Procedimientos Almacenados	CORRECTAS: 1 de 2
2P - Procedimientos Almacenados - Características	CORRECTAS: 2 de 2
2P - Procedimientos Almacenados - Ventajas	CORRECTAS: 1 de 1
2P - Procedimientos Almacenados - Parámetros	CORRECTAS: 1 de 2
2P - Tipos de funciones	CORRECTAS: 0 de 2
2P - Práctica - Procedimientos y Funciones	CORRECTAS: 2 de 4
2P - Práctica - Triggers	CORRECTAS: 1 de 1
2P - Seguridad - General SQLServer	CORRECTAS: 0 de 1
2P - Seguridad - Usuarios y Esquemas	CORRECTAS: 0 de 2
2P - Seguridad - ROLES	CORRECTAS: 1 de 1
2P - Seguridad - Elementos protegibles	CORRECTAS: 1 de 1
2P - Seguridad - Jerarquía de Cifrado	CORRECTAS: 0 de 1
2P - Teoría - Triggers	CORRECTAS: 0 de 2



BARRIOS, LUCAS DIEGO

[Mi perfil](#) · [Ayuda](#) · [Cerrar sesión](#)[SALIR](#)

UNLaM - DIIT - UEaD. MieL Materias Interactivas en Línea - 2024 - [Aviso legal](#)