



Tecnicatura Universitaria  
en Programación

## LABORATORIO DE COMPUTACIÓN II

Unidad Temática N°4:  
Programación SQL Server

Guía de estudio  
1° Año – 2° Cuatrimestre



## Índice

4.1: Introducción a la Programación en SQL Server.....	2
4.2: Manejo de errores.....	2
4.3: Programación aplicada a Procedimientos Almacenados y Funciones definidas por el usuario.....	2
4.4: Triggers.....	3
4.5: Tablas temporales .....	4
BIBLIOGRAFÍA .....	5

#### 4.1: Introducción a la Programación en SQL Server

Utilizando la base de datos Librería realizar las consignas detalladas a continuación.

1. Declarar 3 variables que se llamen `codigo`, `stock` y `stockMinimo` respectivamente. A la variable `codigo` setearle un valor. Las variables `stock` y `stockMinimo` almacenarán el resultado de las columnas de la tabla `artículos` `stock` y `stockMinimo` respectivamente filtradas por el código que se corresponda con la variable `codigo`.
2. Utilizando el punto anterior, verificar si la variable `stock` o `stockMinimo` tienen algún valor. Mostrar un mensaje indicando si es necesario realizar reposición de artículos o no.
3. Modificar el ejercicio 1 agregando una variable más donde se almacene el precio del artículo. En caso que el precio sea menor a \$500, aplicarle un incremento del 10%. En caso de que el precio sea mayor a \$500 notificar dicha situación y mostrar el precio del artículo.
4. Declarar dos variables enteras, y mostrar la suma de todos los números comprendidos entre ellos. En caso de ser ambos números iguales mostrar un mensaje informando dicha situación
5. Mostrar nombre y precio de todos los artículos. Mostrar en una tercer columna la leyenda 'Muy caro' para precios mayores a \$500, 'Accesible' para precios entre \$300 y \$500, 'Barato' para precios entre \$100 y \$300 y 'Regalado' para precios menores a \$100.
6. Modificar el punto 2 reemplazando el mensaje de que es necesario reponer artículos por una excepción.

#### 4.2: Manejo de errores

1. Modificar el ejercicio 2 de la sección 1.1 reemplazando los mensajes mostrados en consola con `print`, por excepciones. Verificar el comportamiento en el SQL Server Management.
2. Modificar el ejercicio anterior agregando las cláusulas de `try catch` para manejo de errores, y mostrar el mensaje capturado en la excepción con `print`.

#### 4.3: Programación aplicada a Procedimientos Almacenados y Funciones definidas por el usuario

1. Programar procedimientos almacenados que permitan realizar las siguientes tareas:
  - a. Mostrar los artículos cuyo precio sea mayor o igual que un valor que se envía por parámetro.
  - b. Ingresar un artículo nuevo, verificando que la cantidad de stock que se pasa por parámetro sea un valor mayor a 30 unidades y menor que 100. Informar un error caso contrario.
  - c. Mostrar un mensaje informativo acerca de si hay que reponer o no stock de un artículo cuyo código sea enviado por parámetro
  - d. Actualizar el precio de los productos que tengan un precio menor a uno ingresado por parámetro en un porcentaje que también se envíe por parámetro. Si no se modifica ningún elemento informar dicha situación
  - e. Mostrar el nombre del cliente al que se le realizó la primer venta en un parámetro de salida.
  - f. Realizar un select que busque el artículo cuyo nombre empiece con un valor enviado por parámetro y almacenar su nombre en un parámetro de salida. En caso que haya varios artículos ocurrirá una excepción que deberá ser manejada con try catch.
2. Programar funciones que permitan realizar las siguientes tareas:
  - a. Devolver una cadena de caracteres compuesto por los siguientes datos: Apellido, Nombre, Telefono, Calle, Altura y Nombre del Barrio, de un determinado cliente, que se puede informar por codigo de cliente o email.
  - b. Devolver todos los artículos, se envía un parámetro que permite ordenar el resultado por el campo precio de manera ascendente ('A'), o descendente ('D').
  - c. Crear una función que devuelva el precio al que quedaría un artículo en caso de aplicar un porcentaje de aumento pasado por parámetro.

#### 4.4: Triggers

1. Crear un desencadenador para las siguientes acciones:
  - a. Restar stock DESPUES de INSERTAR una VENTA
  - b. Para no poder modificar el nombre de algún artículo
  - c. Insertar en la tabla HistorialPrecio el precio anterior de un artículo si el mismo ha cambiado
  - d. Bloquear al vendedor con código 4 para que no pueda registrar ventas en el sistema.

#### 4.5: Tablas temporales

1. Crear un procedimiento almacenado que devuelva la primera y la última venta en una sola tabla
2. Crear un procedimiento almacenado que devuelva en una sola tabla las facturas realizadas en días impares dentro de un mes / año pasado por parámetro

## BIBLIOGRAFÍA

Gorman K., Hirt A., Noderer D., Rowland-Jones J., Sirpal A., Ryan D. & Woody B (2019) Introducing Microsoft SQL Server 2019. Reliability, scalability, and security both on premises and in the cloud. Packt Publishing Ltd. Birmingham UK

Microsoft (2021) SQL Server technical documentation. Disponible en: <https://docs.microsoft.com/en-us/sql/sql-server/?view=sql-server-ver15>

Opel, A. & Sheldon, R. (2010). Fundamentos de SQL. Madrid. Editorial Mc Graw Hill

Varga S., Cherry D., D'Antoni J. (2016). Introducing Microsoft SQL Server 2016 Mission-Critical Applications, Deeper Insights, Hyperscale Cloud. Washington. Microsoft Press



### Atribución-No Comercial-Sin Derivadas

Se permite descargar esta obra y compartirla, siempre y cuando no sea modificado y/o alterado su contenido, ni se comercialice. Referenciarlo de la siguiente manera:  
Universidad Tecnológica Nacional Facultad Regional Córdoba (S/D). Material para la Tecnicatura Universitaria en Programación, modalidad virtual, Córdoba, Argentina.