



**GUSTAVO CORONEL**  
DESARROLLA SOFTWARE

# **PROGRAMACIÓN**



Microsoft®  
**SQL Server®**

**UNIDAD 02**

## **INTRODUCCIÓN A LA PROGRAMACIÓN**

**GUSTAVO CORONEL**

[www.youtube.com/c/DesarrollaSoftware](http://www.youtube.com/c/DesarrollaSoftware)

[gcoronel@uni.edu.pe](mailto:gcoronel@uni.edu.pe)



# Temas

<b>1</b>	<b>BLOQUE ANÓNIMO .....</b>	<b>3</b>
<b>2</b>	<b>FUNCIONES .....</b>	<b>4</b>
2.1	FUNCIÓN ESCALAR .....	4
2.2	FUNCIÓN DE TABLA EN LÍNEA.....	5
2.3	FUNCIÓN DE TABLA DE MÚLTIPLES INSTRUCCIONES .....	6
<b>3</b>	<b>PROCEDIMIENTOS.....</b>	<b>8</b>
<b>4</b>	<b>ELEMENTOS DE PROGRAMACIÓN .....</b>	<b>10</b>
4.1	VARIABLES.....	10
4.2	SENTENCIA DE ASIGNACIÓN.....	10
<b>5</b>	<b>EJERCICIOS PROPUESTOS .....</b>	<b>10</b>
<b>6</b>	<b>CURSOS VIRTUALES.....</b>	<b>11</b>
6.1	CUPONES .....	11
6.2	FUNDAMENTOS DE PROGRAMACIÓN CON JAVA.....	11
6.3	JAVA ORIENTADO A OBJETOS.....	12
6.4	PROGRAMACIÓN CON JAVA JDBC.....	13
6.5	PROGRAMACIÓN CON ORACLE PL/SQL.....	14



# 1 Bloque anónimo

## Sintaxis

```
BEGIN

    Sentencias a ejecutar

END;
GO
```

## Ejemplo 1

```
BEGIN

    DECLARE @NUM1 INT, @NUM2 INT, @SUMA INT;

    SET @NUM1 = CAST( RAND() * 100 AS INT );
    SET @NUM2 = CAST( RAND() * 100 AS INT );

    SET @SUMA = @NUM1 + @NUM2;

    PRINT CONCAT( 'NUM1 = ', @NUM1 );
    PRINT CONCAT( 'NUM2 = ', @NUM2 );
    PRINT CONCAT( 'SUMA = ', @SUMA );

END;
GO
```



## 2 Funciones

### 2.1 Función Escalar

#### Sintaxis

```
CREATE FUNCTION [ esquema. ] nombre_funcion
(
    [ @parametro tipo_dato [ = valor_defecto ] [ READONLY ]
    [ ,...n ]
]
)
RETURNS tipo_dato_retorno
[ AS ]
BEGIN
    cuerpo_función
    RETURN expression;
END;
```

#### Ejemplo 2

```
CREATE FUNCTION dbo.fn_suma ( @num1 int, @num2 int )
RETURNS int
AS
BEGIN
    DECLARE @suma int;
    SET @suma = @num1 + @num2;
    RETURN @suma;
END;
GO

SELECT dbo.fn_suma( 24, 56 ) as suma;
GO

suma
-----
80
```



## 2.2 Función de tabla en línea

### Sintaxis

```
CREATE FUNCTION [ esquema. ] nombre_función
(
    [ @parametro tipo_dato [ = valor_defecto ] [ READONLY ]
    [ ,...n ]
]
)
RETURNS TABLE
[ AS ]
RETURN [ ( ) sentencia_select [ ) ] ;
```

### Ejemplo 3

```
USE RH;
GO

CREATE FUNCTION dbo.fn_empleados ( @p_dpto int )
RETURNS TABLE
AS
RETURN
    SELECT idempleado, apellido, nombre
    FROM dbo.empleado
    WHERE iddepartamento = @p_dpto;
GO

SELECT * FROM dbo.fn_empleados(103);
GO
```



## 2.3 Función de tabla de múltiples instrucciones

### Sintaxis

```
CREATE FUNCTION [ esquema. ] nombre_función
(
    [ @parametro tipo_dato [ = valor_defecto ] [ READONLY ]
    [ ,...n ]
]
)
RETURNS @return_variable TABLE definición_tipo_tabla
[ AS ]
BEGIN
    cuerpo_funcion
    RETURN;
END;
```

### Ejemplo 4

```
CREATE FUNCTION dbo.fn_catalogo ( )
RETURNS @tabla TABLE
(
    codigo int identity(1,1) primary key not null,
    nombre varchar(50) not null,
    precio money not null
)
AS
BEGIN
    INSERT INTO @tabla(nombre,precio) values('Televisor', 1500.00);
    INSERT INTO @tabla(nombre,precio) values('Refrigeradora', 1450.00);
    INSERT INTO @tabla(nombre,precio) values('Lavadora', 1350.00);
    RETURN;
END;
GO

SELECT * FROM dbo.fn_catalogo();
GO
```



## Ejemplo 5

```
USE RH;
GO

CREATE FUNCTION dbo.fn_planilla ( )
RETURNS @planilla TABLE
(
    codigo int primary key not null,
    nombre varchar(50) not null,
    plan_actual money not null,
    plan_proyectada money not null
)
AS
BEGIN
    INSERT INTO @planilla
    SELECT
        d.iddepartamento as codido,
        d.nombre as nombre,
        SUM(e.sueldo) as "planilla actual",
        cast(SUM(e.sueldo * 1.15) as money) as "planilla proyectada"
    FROM dbo.departamento as d
    JOIN dbo.empleado as e
    ON d.iddepartamento = e.iddepartamento
    GROUP BY d.iddepartamento, d.nombre;
    RETURN;
END;
GO

SELECT * FROM dbo.fn_planilla ( )
GO
```



## 3 Procedimientos

### Sintaxis

```
CREATE { PROC | PROCEDURE } [ esquema.] nombre_procedimiento
    [ @parametro tipo_dato [ = default ] [ OUT | OUTPUT ] [ READONLY ] ]
    [ ,...n ]
AS
BEGIN

    cuerpo del procedimiento

END;
```

### Ejemplo 6: Procedimiento simple

```
USE EDUCA;
GO

CREATE PROCEDURE dbo.usp_lista_cursos
AS
BEGIN
    SET NOCOUNT ON;
    SELECT * FROM dbo.curso;
END;
GO

EXEC dbo.usp_lista_cursos;
GO
```





### Ejemplo 7: Procedimiento con parámetros de entrada

```
CREATE PROCEDURE dbo.usp_suma ( @num1 int, @num2 int )
AS
BEGIN
    DECLARE @suma int;
    SET @suma = @num1 + @num2;
    SELECT @num1 NUM1, @num2 NUM2, @suma SUMA;
END;
GO

EXEC dbo.usp_suma 54, 76;
GO
```

### Ejemplo 8: Procedimiento con parámetro de salida

```
USE EDUCA;
GO

CREATE PROCEDURE dbo.usp_precio ( @p_idcurso int, @p_precio money OUT )
AS
BEGIN
    SELECT @p_precio = cur_precio
    FROM dbo.CURSO
    WHERE cur_id = @p_idcurso;
END;
GO

BEGIN
    DECLARE @precio money;
    EXEC dbo.usp_precio 3, @precio OUT;
    PRINT CONCAT( 'PRECIO: ', @precio );
END;
GO

PRECIO: 1815.00
```



## 4 Elementos de programación

### 4.1 Variables

**Sintaxis:**

```
DECLARE  
    @nombre_variable [AS] data_type [ = value ] [ , ... ]
```

### 4.2 Sentencia de asignación

**Instrucción SET**

```
SET @nombre_variable = expresión;  
  
SET @nombre_variable = ( sentencia_select );
```

**Sentencia SELECT**

```
SELECT @nombre_variable = ( sentencia SELECT ), . . .  
FROM . . .
```

## 5 Ejercicios propuestos

1. Desarrolle una función que permita calcular el promedio de 3 números.
2. Desarrolle una función que reporte de la cantidad de alumnos matriculados y las vacantes disponibles de un determinado curso. Base de datos EDUTEC.
3. Desarrolle una función que dado el código de un curso reporte los alumnos que tienen saldo pendiente. Base de datos EDUCA.
4. Desarrolle un procedimiento que reporte por cada curso la cantidad de alumnos matriculados, el importe recaudado y el importe que falta cobrar.
5. Desarrollar un procedimiento que permita obtener la cantidad de artículos vendidos de una determinada categoría. Base de datos: NORTHWIND.
6. Desarrollar una función que permita convertir soles a dólares y viceversa.



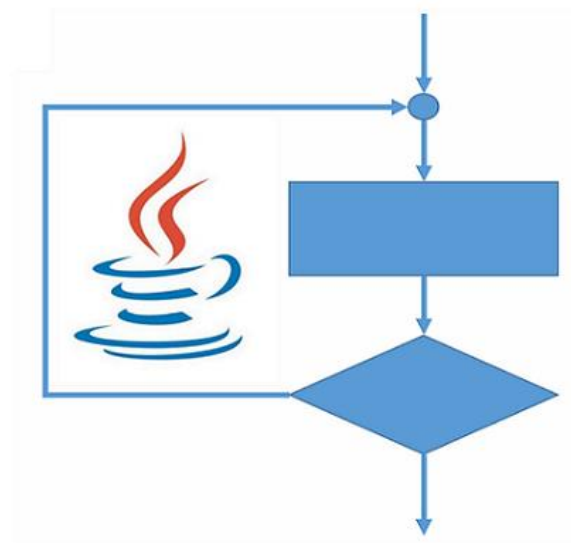
## 6 CURSOS VIRTUALES

### 6.1 CUPONES

En esta URL se publican cupones de descuento:

<http://gcoronelc.github.io>

### 6.2 FUNDAMENTOS DE PROGRAMACIÓN CON JAVA



Tener bases sólidas de programación muchas veces no es fácil, creo que es principalmente por que en algún momento de tu aprendizaje mezclas la entrada de datos con el proceso de los mismos, o mezclas el proceso con la salida o reporte, esto te lleva a utilizar malas prácticas de programación que luego te serán muy difíciles de superar.

En este curso aprenderás las mejores prácticas de programación para que te inicies con éxito en este competitivo mundo del desarrollo de software.

URL del Curso: <https://n9.cl/gcoronelc-java-fund>

Avance del curso: <https://n9.cl/gcoronelc-fp-avance>

Cupones de descuento: <http://gcoronelc.github.io>



## 6.3 JAVA ORIENTADO A OBJETOS



### **CURSO PROFESIONAL DE JAVA ORIENTADO A OBJETOS**

---

**Eric Gustavo Coronel Castillo**

[www.desarrollasoftware.com](http://www.desarrollasoftware.com)

**I N S T R U C T O R**

En este curso aprenderás a crear software aplicando la Orientación a Objetos, la programación en capas, el uso de patrones de software y Swing.

Cada tema está desarrollado con ejemplos que demuestran los conceptos teóricos y finalizan con un proyecto aplicativo.

URL del Curso: <https://bit.ly/2B3ixUW>

Avance del curso: <https://bit.ly/2RYGXIt>

Cupones de descuento: <http://gcoronelc.github.io>



## 6.4 PROGRAMACIÓN CON JAVA JDBC



### PROGRAMACIÓN DE BASE DE DATOS ORACLE CON JAVA JDBC

---

**Eric Gustavo Coronel Castillo**

[www.desarrollasoftware.com](http://www.desarrollasoftware.com)

**I N S T R U C T O R**

En este curso aprenderás a programar bases de datos Oracle con JDBC utilizando los objetos Statement, PreparedStatement, CallableStatement y a programar transacciones correctamente teniendo en cuenta su rendimiento y concurrencia.

Al final del curso se integra todo lo desarrollado en una aplicación de escritorio.

URL del Curso: <https://bit.ly/31apy0O>

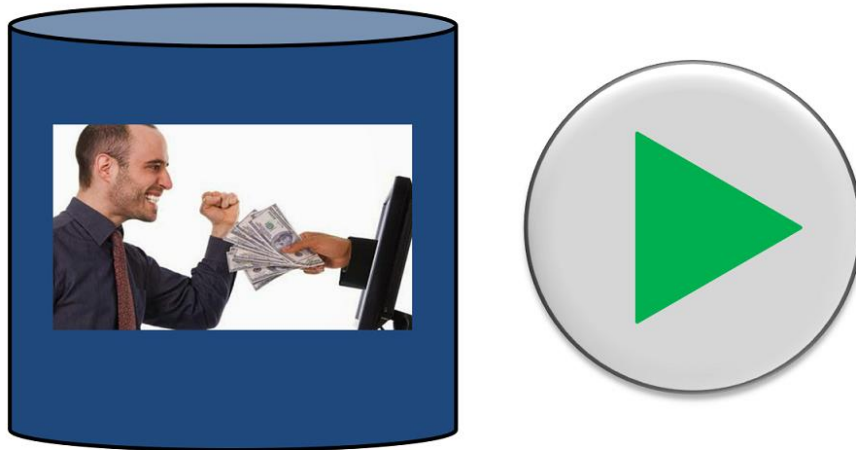
Avance del curso: <https://bit.ly/2vatZOT>

Cupones de descuento: <http://gcoronelc.github.io>



## 6.5 PROGRAMACIÓN CON ORACLE PL/SQL

### ORACLE PL/SQL



En este curso aprenderás a programar las bases de datos ORACLE con PL/SQL, de esta manera estarás aprovechando las ventajas que brinda este motor de base de datos y mejorarás el rendimiento de tus consultas, transacciones y la concurrencia.

Los procedimientos almacenados que desarrolles con PL/SQL se pueden ejecutarlos de Java, C#, PHP y otros lenguajes de programación.

URL del Curso: <https://bit.ly/2YZjfxT>

Avance del curso: <https://bit.ly/3bcigYb>

Cupones de descuento: <http://gcoronelc.github.io>