



BASE DE DATOS AVANZADO I

Unidad 4: Programación Transact SQL

Tema 8: Procedimientos Almacenados - Parte II





Tema 08: Procedimientos Almacenados

4.1. Tema 8: Procedimientos Almacenados

4.1.4. Anidamiento de procedimientos almacenados.

4.1.5. Operaciones con procedimientos almacenados





Capacidades

1. Implementa procedimientos almacenados, funciones, en todos sus tipos, y triggers para garantizar el alto rendimiento de la base de datos, al hacer las transacciones





1. Procedimientos almacenados

Ventajas

- Es más rápido que un batch porque:
 - Ha sido analizado previamente
 - Se han resuelto las referencias a los objetos a utilizar.
 - No se necesita construir el árbol de búsqueda sino que utiliza el generado al momento de crearlo (compilarlo).
 - No se necesita crear un plan de búsqueda porque éste fue generado al momento de crearlo.
 - Elevan la seguridad debido a que puede darse acceso al procedimiento, sin necesidad de darle a los objetos.
 - Mejora el mantenimiento y distribución de versiones.

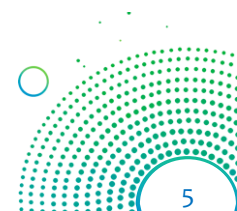




1. Procedimientos almacenados

Desventajas

- Demasiados procedimientos pueden ocasionar:
 - Un procesamiento excesivo por el lado del servidor.
 - Se necesita un personal con más experiencia para el desarrollo y asociación con un aplicativo.
 - Demoras al momento de realizar una migración a otro gestor de base de datos.
 - Aprender la sintaxis equivalente del otro gestor de base de datos para transformar el pseudocódigo.



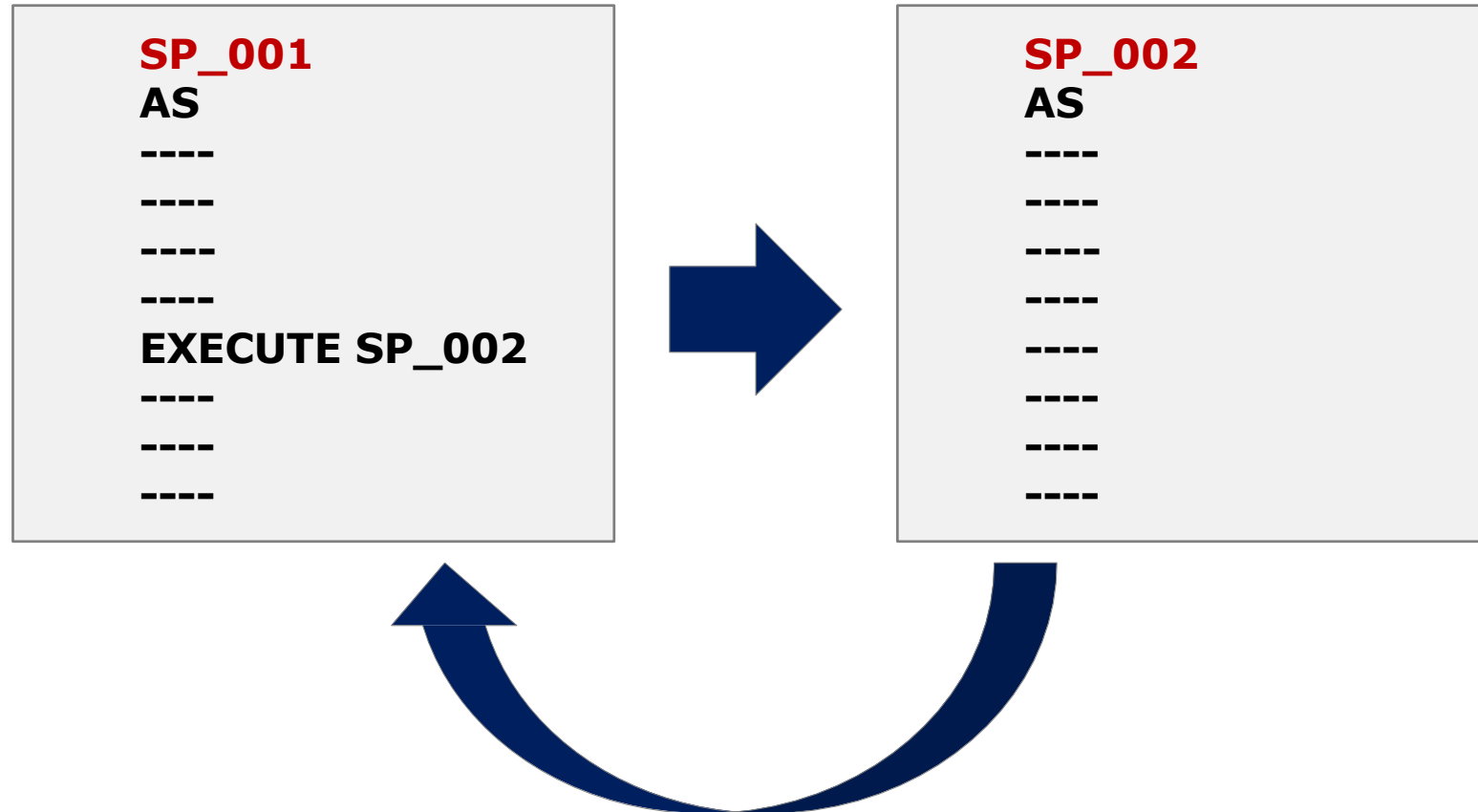


2. Anidamiento de procedimientos almacenados

- La anidación se origina cuando un procedimiento llama a otro. El nivel de anidación se incrementa cuando el procedimiento llamado, inicia la ejecución y disminuye, cuando el procedimiento llamado, finaliza la ejecución
- Si se supera el máximo de 16 niveles de anidación, el procedimiento no se ejecuta correctamente.
- El nivel de anidación actual se almacena en la variable global @@nestlevel

2. Anidamiento de procedimientos almacenados

Estructura de Invocación Anidada





2. Anidamiento de procedimientos almacenados

Ejemplo de Anidamiento de procedimientos

- Crear un procedimiento para ingresar nuevas categorías.

```
Use Negocios
go

--Procedimiento de ingreso de categorias
Create Procedure USP_IngCat
    @p_IdCategoria int,
    @p_NombreCategoria varchar(50),
    @p_Descripcion Varchar(Max)
As
Begin
    Insert Compras.categorias
    Values
        (@p_IdCategoria, @p_NombreCategoria, @p_Descripcion)
End
go
```





2. Anidamiento de procedimientos almacenados

Ejemplo de Anidamiento de procedimientos

- Crear un procedimiento para actualizar determinada categoría.

```
--Procedimiento para Actualizar categoria
Create Procedure USP_ActCat
    @p_IdCategoria int,
    @p_NombreCategoria varchar(50),
    @p_Descripcion Varchar(Max)
As
Begin
    Update Compras.categorias
    Set      NombreCategoria = @p_NombreCategoria,
            Descripcion = @p_Descripcion
    Where IdCategoria = @p_IdCategoria
End
go
```





2. Anidamiento de procedimientos almacenados

Ejemplo de Anidamiento de procedimientos

- Crear un procedimiento que invoque a uno de los 02 anteriores, según sea el caso, haciendo las validaciones respectivas.

```
Create Procedure USP_TranCat
    @p_tipoTran int, @p_IdCategoria int,
    @p_NombreCategoria varchar(50), @p_Descripcion Varchar(Max)
As
Begin
    If @p_tipoTran=1 --Insertar
    Begin
        If Not Exists (Select * from Compras.categorias Where IdCategoria = @p_IdCategoria)
            Execute USP_IngCat @p_idCategoria, @p_NombreCategoria, @p_Descripcion
        Else
            Raiserror ('El Id Categoria ya existe',10,1)
        End
    End
    If @p_tipoTran=2 --Actualizar
    Begin
        If Exists (Select * from Compras.categorias Where IdCategoria = @p_IdCategoria)
        Begin
            If @p_NombreCategoria is Null
                Set @p_NombreCategoria = (Select NombreCategoria From Compras.categorias
                                            Where IdCategoria= @p_IdCategoria)
            If @p_Descripcion is null
                Set @p_Descripcion = (Select Descripcion From Compras.categorias
                                      Where IdCategoria= @p_IdCategoria)
            Execute USP_ActCat @p_idCategoria, @p_NombreCategoria, @p_Descripcion
        End
        Else
            Raiserror ('El Id Categoria No existe',10,1)
        End
    End
End
```






2. Anidamiento de procedimientos almacenados

Ejemplo de Anidamiento de procedimientos

- Ahora ejecutar el procedimiento principal para ingresar (opción 1).



```
--Ejecutar para insertar
Execute USP_TranCat '1','55','Frutas','Alimentos sanos'
go
```

108 %

Messages

(1 row affected)



2. Anidamiento de procedimientos almacenados

Ejemplo de Anidamiento de procedimientos


- Comprobamos.

```
Select * from Compras.categorias
go
```

108 %

Results Messages

	IdCategoria	NombreCategoria	Descripcion
1	1	Bebidas	Gaseosas, cafe, te, cervezas y maltas
2	2	Condimentos	Salsas dulces y picantes, delicias, comida para ...
3	3	Reposteria	Postres, dulces y pan dulce
4	4	Lacteos	Quesos
5	5	Granos/Cereales	Pan, galletas, pasta y cereales
6	6	Carnes	Carnes preparadas
7	7	Frutas/Verduras	Frutas secas y queso de soja
8	8	Pescado/Marisco	Pescados, mariscos y algas
9	55	Frutas	Alimentos sanos





2. Anidamiento de procedimientos almacenados

Ejemplo de Anidamiento de procedimientos

- Ejecutar el procedimiento principal para actualizar (opción 2).



```
--Ejecutar para actualizar
Execute USP_TranCat '2','55',null,'Dulces sanos (Sportacus)'
go
```

108 %

Messages

(1 row affected)

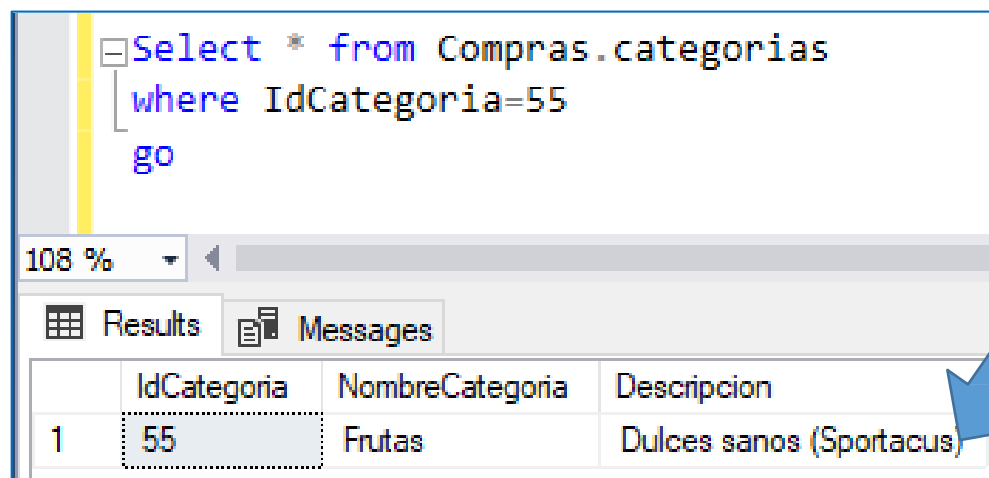




2. Anidamiento de procedimientos almacenados

Ejemplo de Anidamiento de procedimientos

- Comprobamos.



```
Select * from Compras.categorias  
where IdCategoria=55  
go
```

108 %

Results Messages

	IdCategoria	NombreCategoria	Descripcion
1	55	Frutas	Dulces sanos (Sportacus)

3. Operaciones con procedimientos almacenados

Algunos casos

- Procedimientos almacenados con uso de cursores.
- Procedimientos almacenados con transacciones explícitas.
- Procedimientos almacenados con consultas complejas.





3. Operaciones con procedimientos almacenados

Algunos casos

- Procedimientos almacenados con uso de cursores.

Implementar un Procedimiento Almacenado que reciba como parámetro el Id del Cliente, para luego mostrar los pedidos emitidos a dicho Cliente.



3. Operaciones con procedimientos almacenados

Ejemplo con uso de parámetros y cursores

```
--Fase 01: Crear el Plan
Create Procedure USP_ProCur
    @p_idCliente varchar(10)
As
Begin
    Declare CursorCliPed Cursor For Select IdPedido, FechaPedido from Ventas.pedidoscabe
        Where IdCliente = @p_idCliente

    Declare @vc_idped int, @vc_fecped date
    Open CursorCliPed
    Fetch CursorCliPed Into @vc_IdPed, @vc_fecped
    Print space(5) + 'Pedido' + space(10) + 'Fecha'
    Print Replicate('=',50)
    While @@FETCH_STATUS = 0
    Begin
        Print space(5) + cast(@vc_Idped as Varchar(10)) + space(10) +
            Cast(@vc_fecped As varchar(12))
        Fetch CursorCliPed Into @vc_IdPed, @vc_fecped
    End
    Close CursorCliPed
    Deallocate CursorcliPed
End
```

6,1)



3. Operaciones con procedimientos almacenados

Ejemplo con uso de parámetros y cursores

- Resultado:

```
--Fase 02: Ejecutar plan  
Execute USP_ProCur 'ALFKI'  
go
```

Messages	
Pedido	Fecha
=====	
10643	1997-08-25
10692	1997-10-03
10702	1997-10-13
10835	1998-01-15
10952	1998-03-16
11011	1998-04-09



3. Operaciones con procedimientos almacenados

Ejemplo de procedimientos almacenados con transacciones explícitas

- Implementar un Procedimiento Almacenado que permita recibir como parámetro los campos de la tabla cliente para luego, mediante una transacción explícita, registrar un nuevo cliente, invocando a otro procedimiento almacenado.
- Se confirmará la transacción cuando el cliente no es del Idpaís 002, caso contrario genere un error de severidad grave, controlando y deshaciendo la transacción.





3. Operaciones con procedimientos almacenados

Ejemplo de procedimientos almacenados con transacciones explícitas. Solución

- Paso 1. Implementar el procedimiento simple de ingreso de cliente.

```
CREATE PROCEDURE sp_nuevoCliente
    @idcli varchar(5),
    @nom    varchar(40),
    @dir    varchar(60),
    @idpa   char(3),
    @fono   varchar(25)
AS
BEGIN
    insert into Ventas.clientes
    values (@idcli ,@nom ,@dir,@idpa ,@fono)
END
GO
```



3. Operaciones con procedimientos almacenados

Ejemplo de procedimientos almacenados con transacciones explícitas. Solución

- Paso 02. Implementar el procedimiento principal



3. Operaciones con procedimientos almacenados

```

Create Procedure sp_Ingresar
    @idcli varchar(5), @nom varchar(40),
    @dir varchar(60), @idpa char(3), @fono varchar(25)
As
Begin
    Begin Try
        Begin Tran t1
            exec sp_nuevoCliente @idcli ,@nom ,@dir,@idpa ,@fono
            if @idpa != '002'
                commit tran t1
            else
                Raiserror('Id país no permitido',16,1)
        End Try
        Begin Catch
            print error_message()
            rollback tran t1
        End Catch
    end
go
```

3. Operaciones con procedimientos almacenados

Resultado: Transacción cancelada

```
Exec sp_Ingresar 'EXTRA', 'Alfredo Velazques', 'Jr. Las Casonas #878', '002', '983476334'  
go
```

Messages

(1 row affected)
Id país no permitido

```
Select * From Ventas.clientes  
where IdCliente='EXTRA'  
go
```

143 %

Results Messages

IdCliente	NomCliente	DirCliente	idpais	fonoCliente
-----------	------------	------------	--------	-------------

3. Operaciones con procedimientos almacenados

Resultado: Transacción confirmada

```
Exec sp_Ingresar 'EXTRA', 'Alfredo Velazques', 'Jr. Las Casonas #878', '001', '983476334'  
go
```

Messages

(1 row affected)

```
Select * From Ventas.clientes  
where IdCliente='EXTRA'  
go
```

143 %

Results

Messages

	IdCliente	NomCliente	DirCliente	idpais	fonoCliente
1	EXTRA	Alfredo Velazques	Jr. Las Casonas #878	001	983476334



Ejercicio

- En el departamento de sistemas, se solicita implementar un procedimiento almacenado que reciba como parámetro el nombre de la entidad que se quiere consultar (EMPLEADO, CLIENTE y PRODUCTO) y muestre la información solicitada. Se invocará a los procedimientos de consulta de cada entidad.





Ejercicio: Solución

- Se creará un procedimiento de consulta para cada entidad con parámetro vacío.
- Procedimiento de consulta de Empleados.

```
CREATE PROCEDURE uspConsultarEmpleados
AS
BEGIN
    SELECT E.IdEmpleado          [Id del Empleado],
           E.NomEmpleado + space(1) + E.ApeEmpleado [Empleado],
           DateDiff(yy,E.FecNac,getdate())           [Edad],
           E.DirEmpleado          [Dirección],
           D.nomDistrito          [Distrito],
           C.desCargo             [Cargo],
           DateDiff(yy,E.FecContrata,getdate())      [Tiempo de Servicio]
    FROM RRHH.empleados E JOIN RRHH.Distritos D
    ON E.idDistrito = D.idDistrito JOIN RRHH.Cargos C
    ON E.idCargo = C.idcargo
END
GO
```

Ejercicio: Solución

- Procedimiento de consulta de Clientes.

```
- Create Procedure uspConsultarClientes
  As
- Begin
-   Select  C.IdCliente      [Id del Cliente],
            C.NomCliente    [Cliente],
            C.DirCliente     [Dirección],
            P.NombrePais     [País],
            C.fonoCliente    [Teléfono]
  From Ventas.clientes C Join Ventas.países P
    On C.idpais = P.Idpais
- End
go
```

Ejercicio: Solución

- Procedimiento de consulta de Productos.

```

Create Procedure uspConsultarProductos
As
Begin
    Select  P.IdProducto      [Id del Producto],
            P.NomProducto     [Nombre del Producto],
            PV.NomProveedor    [Razón Social del Proveedor],
            C.NombreCategoria  [Categoría],
            P.PrecioUnidad     [Precio Unitario],
            Case
                When P.UnidadesEnExistencia = 0 Then 'Sin Stock'
                Else str(P.UnidadesEnExistencia)
            End As [Stock]
    From Compras.productos P Join Compras.categorias C
        On P.IdCategoria = C.IdCategoria Join Compras.proveedores PV
        On P.IdProveedor = PV.IdProveedor
    Order By 6 desc
End
go

```

Ejercicio: Solución

- Procedimiento de consulta según valor del parámetro y que invoque a los otros procedimientos.

```
[-] Create Procedure uspConsultar
  @p_EntidadAconsultar varchar(max)
  As
  [-] Begin
  [-]   If upper(@p_EntidadAconsultar)='CLIENTE'
  [-]       Execute uspConsultarClientes
  [-]   Else If upper(@p_EntidadAconsultar)='EMPLEADO'
  [-]       Execute uspConsultarEmpleados
  [-]   Else If upper(@p_EntidadAconsultar)='PRODUCTO'
  [-]       Execute uspConsultarProductos
  [-]   Else
  [-]       RaiseError ('Consulta de Entidad No Implementada',10,1)
  [-] End
go
```

Ejercicio: Solución

- Ahora ejecutar:

```
--Ejecutar SP para consultar Empleado  
Execute uspConsultar 'Empleado'  
go
```

Results		Messages					
	Id del Empleado	Empleado	Edad	Dirección	Distrito	Cargo	Tiempo de Servicio
1	1	Nancy Davolio	50	Calle Las Magnolias 123	Rimac	Auxiliar de Ventas	8
2	2	Andrew Fuller	66	Av Abancay 234	Lima	Representante de Ventas	7
3	3	Janet Leverling	55	Av. Riva Agüero 233	San Miguel	Supervisor de Ventas	7
4	4	Margaret Peacock	60	Calle las flores 411	Rimac	Representante de Ventas	7
5	5	Steven Buchanan	63	Av. Brasil 222	Ate	Supervisor de Ventas	7
6	6	Michael Suyama	55	Calle Zafiro 344	Lima	Auxiliar de Ventas	7
7	7	Robert King	58	Jr Puni 322	Lima	Ejecutivo de Ventas	7
8	8	Laura Callahan	60	Dinthilac 123	Rimac	Auxiliar de Ventas	7
9	9	Anne Dodsworth	49	Av Los fresnos 2334	Ate	Representante de Ventas	7

Ejercicio: Solución

- Ahora ejecutar:

```
--Ejecutar SP para consultar Cliente  
Execute uspConsultar 'Cliente'  
go
```

Results		Messages			
	Id del Cliente	Cliente	Dirección	País	Teléfono
1	ALFKI	Alfreds Futterkiste	Obere Str. 57	Argentina	030-0074321
2	ALLI	Alfredo Kimball	Ca Paujile 123	Peru	98765567
3	ANATR	Ana Trujillo Emparedados y helados	Avda. de la Constitucion 2222	España	(5) 555-4729
4	ANTON	Antonio Moreno Taqueria	Mataderos 2312	Colombia	(5) 555-3932
5	AROUT	Around the Hom	120 Hanover Sq.	USA	(71) 555-7788

Ejercicio: Solución

- Ahora ejecutar:

```
--Ejecutar SP para consultar Producto  
Execute uspConsultar 'Producto'  
go
```

Results Messages						
	Id del Producto	Nombre del Producto	Razón Social del Proveedor	Categoría	Precio Unitario	Stock
1	5	Mezcla Gumbo del chef Anton	New Orleans Cajun Delights	Condimentos	21	Sin Stock
2	17	Cordero Alice Springs	Pavlova, Ltd.	Carnes	39	Sin Stock
3	29	Salchicha Thüringer	Plutzer Lebensmittelgroßmärkte AG	Carnes	135	Sin Stock
4	31	Queso gorgonzola Telino	Formaggi Fortini s.r.l.	Lacteos	12	Sin Stock
5	53	Empanada de carne	Gday, Mate	Carnes	32	Sin Stock
6	75	Cerveza Klosterbier Rhönbräu	Plutzer Lebensmittelgroßmärkte AG	Bebidas	8	125
7	40	Carne de cangrejo de Boston	New England Seafood Cannery	Pescado/Marisco	18	123
8	6	Memelada de grosellas de la abuela	Grandma Kellys Homestead	Condimentos	25	120

Ejercicio: Solución

- Ahora ejecutar:

```
--Ejecutar SP para consultar Entidad no implementada  
Execute uspConsultar 'Proveedor'  
go
```

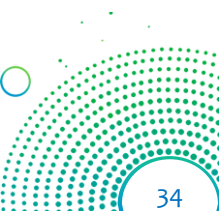
 Messages

Consulta de Entidad No Implementada



Conclusiones

- Un procedimiento almacenado de SQL Server es un grupo de una o varias instrucciones Transact-SQL o una referencia a un método de Common Runtime Language (CLR) de Microsoft
- .NET Framework.
- Los procedimientos se asemejan a las construcciones de otros lenguajes de programación.
 - Los procedimientos almacenados pueden invocar a otros procedimientos.





Bibliografía

Microsoft (2017) Procedimientos almacenados (motor de base de datos). Recuperado de: <https://docs.microsoft.com/es-es/sql/relational-databases/stored-procedures/stored-procedures-database-engine?view=sql-server-2017>



GRACIAS

**SEDE LIMA CENTRO**

Av. Uruguay 514
Cercado – Lima
Teléfono: 419-2900

SEDE INDEPENDENCIA

Av. Carlos Izaguirre 233
Independencia – Lima
Teléfono: 633-5555

SEDE BREÑA

Av. Brasil 714 – 792
(CC La Rambla – Piso 3)
Breña – Lima
Teléfono: 633-5555

SEDE TRUJILLO

Calle Borgoño 361
Trujillo
Teléfono: (044) 60-2000

SEDE SAN JUAN DE LURIGANCHO

Av. Próceres de la Independencia 3023-3043
San Juan de Lurigancho – Lima
Teléfono: 633-5555

SEDE BELLAVISTA

Av. Mariscal Oscar R. Benavides 3866 – 4070
(CC Mall Aventura Plaza)
Bellavista – Callao
Teléfono: 633-5555

SEDE AREQUIPA

Av. Porongoche 500
(CC Mall Aventura Plaza)
Paucarpata - Arequipa
Teléfono: (054) 60-3535