Proyecto “queCOMEMOS”

TRABAJO PRACTICO BASE DE DATOS

## **Facultad:** Universidad Tecnológica Nacional, General Pacheco

## **Carrera:** TSP

### **Cátedra:** Laboratorio de computación 3

### **Integrante:** German Alejandro Asprino Cabrera

### **Docentes:**

* [Diego Ariel Español](http://frgp.cvg.utn.edu.ar/user/view.php?id=1439&course=1)
* [Ariel Hernan Herrera](http://frgp.cvg.utn.edu.ar/user/view.php?id=6090&course=1)

### **Fecha De Entrega:** 25/07/2017

## **Introducción “quecomemos”:**

QueComemos es una aplicación pensada para esos momentos donde no se sabe que comer en la casa. La aplicación permite manejar recetas precargadas, para realizar búsquedas que nos ayuden a evaluar que comida vamos a hacer. Permite búsquedas de tiempo de preparación, costo, cantidad de macro-nutrientes (proteínas,carbohidratos,calorías,grasas), tipo de comidas (vegetarianas,veganas, celiacas,etc.),por ingredientes en la heladera, por nombre.

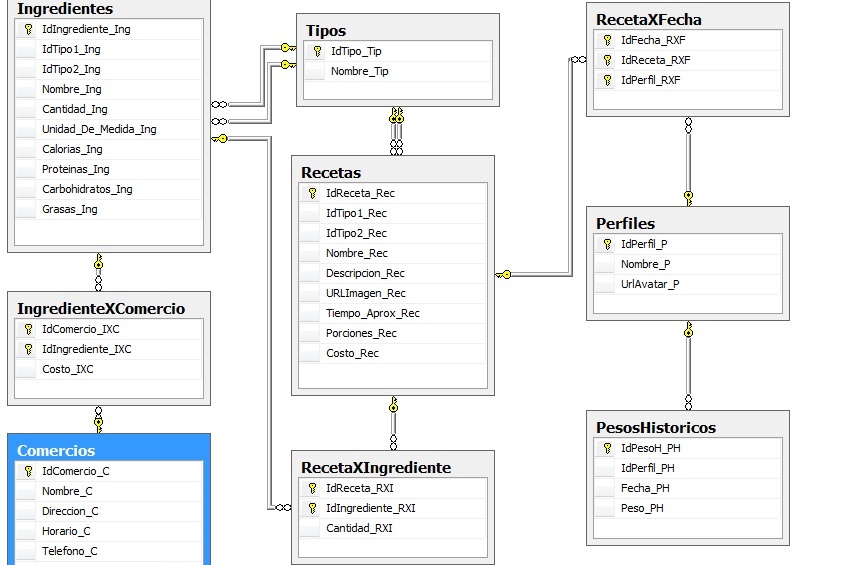
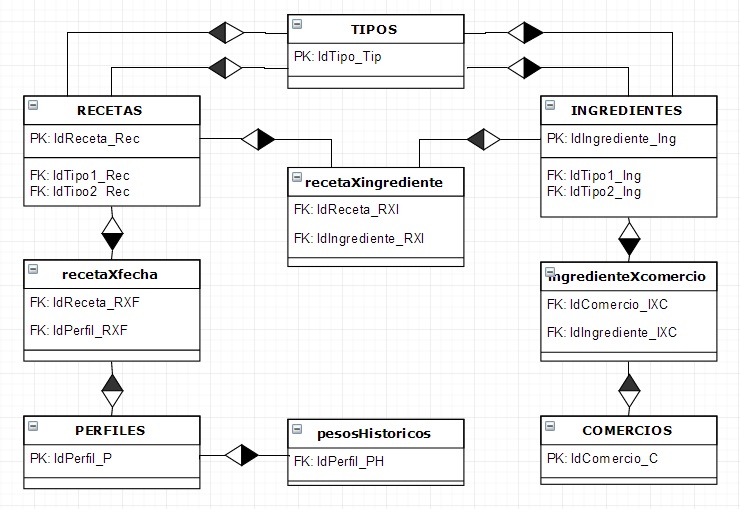
También provee una base de datos de comercios para cargar costos especifico, ver sus horarios y días hábiles.(si los teléfonos de delivery también están incluidos)

La aplicación permite también , de modo opcional, gestionar un perfil en el cual podamos seguir las comidas, la nutrición y el peso en plazo de una semana o mas. (incluye gráficos para ir viendo lo que vamos consumiendo en macro-nutrientes y el peso semanal)  
  
Puede seguirse el proyecto en la pagina de Github: <https://github.com/hellfishg/queComemos>

**INDICE**

1. DER Y ESQUEMA SQL SERVER. Pág: 3
2. CREACIÓN DE LA BASE DE DATOS Y TABLAS Pág 4-5
3. RELACIONA ENTRE TABLAS Pág 6
4. CARGA DE DATOS Pág 7-8
5. PROCEDIMIENTOS Y CONSULTAS Pág 8-11
6. TRIGGER Pág 12

**1)**DER

**1)** SQL VISTA:

**2)**   
  
CREATE DATABASE queComemos

on ( NAME='queComemos\_dat',FILENAME='C:\HellDocs\queComemos\Base de datos\queComemos.dbo')

GO

-----------------------------------------------------

USE queComemos

GO

-----------------------------------------------------

CREATE TABLE Recetas (

IdReceta\_Rec INT IDENTITY(1,1) NOT NULL,

IdTipo1\_Rec INT NOT NULL,

IdTipo2\_Rec INT NOT NULL,

Nombre\_Rec VARCHAR (50) NOT NULL,

Descripcion\_Rec VARCHAR (max) NULL,

URLImagen\_Rec VARCHAR (max) NULL DEFAULT 'NONE',

Tiempo\_Aprox\_Rec INT NOT NULL,

Porciones\_Rec INT NOT NULL,

Costo\_Rec INT NOT NULL,

CONSTRAINT PK\_REC PRIMARY KEY (IdReceta\_Rec)

)

GO

-----------------------------------------------------

CREATE TABLE Ingredientes (

IdIngrediente\_Ing INT IDENTITY(1,1) NOT NULL,

IdTipo1\_Ing INT NOT NULL,

IdTipo2\_Ing INT NOT NULL,

Nombre\_Ing VARCHAR(50) NOT NULL,

Cantidad\_Ing INT NOT NULL,

Unidad\_De\_Medida\_Ing VARCHAR(11) CHECK ( Unidad\_De\_Medida\_Ing IN('Gramos','Unidades','Mililitros')) NOT NULL,

Calorias\_Ing DECIMAL (10,2) NULL,

Proteinas\_Ing DECIMAL (10,2) NULL,

Carbohidratos\_Ing DECIMAL (10,2) NULL,

Grasas\_Ing DECIMAL (10,2) NULL,

CONSTRAINT PK\_ING PRIMARY KEY (IdIngrediente\_Ing)

)

GO

-----------------------------------------------------

CREATE TABLE RecetaXIngrediente (

IdReceta\_RXI INT NOT NULL,

IdIngrediente\_RXI INT NOT NULL,

Cantidad\_RXI DECIMAL (10,2) NOT NULL,

CONSTRAINT PK\_RXI PRIMARY KEY (IdReceta\_RXI,IdIngrediente\_RXI)

)

GO

-----------------------------------------------------

CREATE TABLE Tipos (

IdTipo\_Tip INT IDENTITY(1,1) NOT NULL,

Nombre\_Tip VARCHAR (50) NOT NULL,

CONSTRAINT PK\_TIP PRIMARY KEY (IdTipo\_Tip)

)

GO

-----------------------------------------------------

CREATE TABLE Perfiles (

IdPerfil\_P INT IDENTITY(1,1) NOT NULL,

Nombre\_P VARCHAR (50) NOT NULL,

UrlAvatar\_P VARCHAR (max) NULL,

CONSTRAINT PK\_PER PRIMARY KEY (IdPerfil\_P)

)

GO

------------------------------------------------------

CREATE TABLE RecetaXFecha (

IdFecha\_RXF DATE NOT NULL,

IdReceta\_RXF INT NOT NULL,

IdPerfil\_RXF INT NOT NULL,

CONSTRAINT PK\_RXF PRIMARY KEY (IdFecha\_RXF,IdReceta\_RXF,IdPerfil\_RXF)

)

GO

-----------------------------------------------------------------------------------------

CREATE TABLE PesosHistoricos (

IdPesoH\_PH INT IDENTITY(1,1) NOT NULL,

IdPerfil\_PH INT NOT NULL,

Fecha\_PH DATE NOT NULL,

Peso\_PH INT NOT NULL,

CONSTRAINT PK\_PEHI PRIMARY KEY (IdPesoH\_PH)

)

GO

-----------------------------------------------------------------------------------------

CREATE TABLE Comercios (

IdComercio\_C INT IDENTITY(1,1) NOT NULL,

Nombre\_C VARCHAR (50) NOT NULL,

Direccion\_C VARCHAR (50) NULL,

Horario\_C VARCHAR (max) NULL,

Telefono\_C VARCHAR (20) NULL,

Dias\_C NCHAR (7) NULL DEFAULT 'lmijvsd',

CONSTRAINT PK\_COM PRIMARY KEY (IdComercio\_C)

)

GO

---------------------------------------------------------------------------------------CREATE TABLE IngredienteXComercio (

IdComercio\_IXC INT NOT NULL,

IdIngrediente\_IXC INT NOT NULL,

Costo\_IXC INT NOT NULL,

CONSTRAINT PK\_IXC PRIMARY KEY (IdComercio\_IXC,IdIngrediente\_IXC)

)

GO

---------------------------------------------------------------------------------------

**3)**

--CLAVES FORANEAS:

ALTER TABLE Recetas

ADD CONSTRAINT FK\_REC1 FOREIGN KEY (IdTipo1\_Rec) REFERENCES Tipos (IdTipo\_Tip)

GO

ALTER TABLE Recetas

ADD CONSTRAINT FK\_REC2 FOREIGN KEY (IdTipo2\_Rec) REFERENCES Tipos (IdTipo\_Tip)

GO

---------------------------------------------------------------------------------------

ALTER TABLE Ingredientes

ADD CONSTRAINT FK\_ING1 FOREIGN KEY (IdTipo1\_Ing) REFERENCES Tipos (IdTipo\_Tip)

GO

ALTER TABLE Ingredientes

ADD CONSTRAINT FK\_ING2 FOREIGN KEY (IdTipo2\_Ing) REFERENCES Tipos (IdTipo\_Tip)

GO

---------------------------------------------------------------------------------------

ALTER TABLE RecetaXIngrediente

ADD CONSTRAINT FK\_RXI1 FOREIGN KEY (IdReceta\_RXI) REFERENCES Recetas (IdReceta\_Rec)

GO

ALTER TABLE RecetaXIngrediente

ADD CONSTRAINT FK\_RXI2 FOREIGN KEY (IdIngrediente\_RXI) REFERENCES Ingredientes (IdIngrediente\_Ing)

GO

---------------------------------------------------------------------------------------

ALTER TABLE IngredienteXComercio

ADD CONSTRAINT FK\_IXC1 FOREIGN KEY (IdComercio\_IXC) REFERENCES Comercios (IdComercio\_C)

GO

ALTER TABLE IngredienteXComercio

ADD CONSTRAINT FK\_IXC2 FOREIGN KEY (IdIngrediente\_IXC) REFERENCES Ingredientes (IdIngrediente\_Ing)

GO

---------------------------------------------------------------------------------------

ALTER TABLE RecetaXFecha

ADD CONSTRAINT FK\_RXF1 FOREIGN KEY (IdReceta\_RXF) REFERENCES Recetas (IdReceta\_Rec)

GO

ALTER TABLE RecetaXFecha

ADD CONSTRAINT FK\_RXF2 FOREIGN KEY (IdPerfil\_RXF) REFERENCES Perfiles (IdPerfil\_P)

GO

---------------------------------------------------------------------------------------

ALTER TABLE PesosHistoricos

ADD CONSTRAINT FK\_PH FOREIGN KEY (IdPerfil\_PH) REFERENCES Perfiles (IdPerfil\_P)

GO

---------------------------------------------------------------------------------------

**4)**  
--IngredientexReceta:

/\*

Recetas cargadas:

Sopa de calabaza 40,3

1 calabaza

100 g queso rallado

50 g fideos municiones

2 calditos knor

3 huevos

Tarta de jamón y queso 30,2

150 g de jamón

200 g de queso cremoso

3 huevos

1 tapa de tarta

Pollo con papas al horno 60,4

4 papas

1 pollo

20 mL aceite

Sal

Ensalada de pollo 30,3

1/4 pollo

1 lechuga

3 tomates

20 g queso rallado

3 huevos

10 mL salsa de soja

10 mL jugo de limón

\*/

INSERT INTO RecetaXIngrediente (IdReceta\_RXI, IdIngrediente\_RXI, Cantidad\_RXI)

--Ensalada de pollo:

SELECT 1 , 3 , 0.25 UNION

SELECT 1 , 12 , 1 UNION

SELECT 1 , 15 , 3 UNION

SELECT 1 , 18 , 20 UNION

SELECT 1 , 16 , 3 UNION

SELECT 1 , 5 , 10 UNION

SELECT 1 , 11 , 10 UNION

--Sopa de calabaza:

SELECT 2 , 9 , 1 UNION

SELECT 2 , 18 , 100 UNION

SELECT 2 , 4 , 50 UNION

SELECT 2 , 1 , 2 UNION

SELECT 2 , 16 , 3 UNION

--Tarta de jamon y queso:

SELECT 3 , 2 , 150 UNION

SELECT 3 , 17 , 200 UNION

SELECT 3 , 16 , 3 UNION

SELECT 3 , 6 , 1 UNION

--Pollo con papas al horno:

SELECT 4 , 13 , 4 UNION

SELECT 4 , 3 , 1 UNION

SELECT 4 , 7 , 20 UNION

SELECT 4 , 14 , 1

GO

-----------------------------------------------------------------------------------------

--Comercios

INSERT INTO Comercios (Nombre\_C, Direccion\_C, Horario\_C, Telefono\_C, Dias\_C)

SELECT 'Super Nuevo Estilo' ,'San Lorenzo 12343', '09:00 a 14:00', '4-713-3456', 'lmijvs0' UNION

SELECT 'Super Coto' ,'Balbin 3900', '09:00 a 21:00', '4-713-7456', 'lmijvsd' UNION

SELECT 'Pizzeria Simoqueña' ,'Matheu 3600', '18:00 a 23:30', '4-713-7856', '0mijvsd' UNION

SELECT 'Dietetica Tomillo' , 'Malaver 4567' , '15:00 a 19:00' , '4-713-9812' , 'lmijv'

GO

---------------------------------------------------------------------------------------

--IngredientesxComercios

INSERT INTO IngredientexComercio (IdComercio\_IXC, IdIngrediente\_IXC, Costo\_IXC)

SELECT 1, 1, 5 UNION

SELECT 1, 2, 10 UNION

SELECT 1, 3, 20 UNION

SELECT 2, 3, 330 UNION

SELECT 2, 4, 12 UNION

SELECT 2, 5, 1 UNION

SELECT 3, 5, 32 UNION

SELECT 3, 7, 56 UNION

SELECT 1, 6, 89 UNION

SELECT 4, 4, 32 UNION

SELECT 4, 7, 56 UNION

SELECT 4, 6, 89

GO

---------------------------------------------------------------------------------------

--PesosHistoricos

INSERT INTO PesosHistoricos (IdPerfil\_PH, Fecha\_PH , Peso\_PH)

SELECT 1, '16-07-2017', 75 UNION

SELECT 1, '17-07-2017', 70 UNION

SELECT 1, '18-07-2017', 72 UNION

SELECT 1, '19-07-2017', 75 UNION

SELECT 1, '20-07-2017', 80 UNION

SELECT 2, '16-07-2017', 40 UNION

SELECT 2, '17-07-2017', 50 UNION

SELECT 2, '18-07-2017', 65

GO

---------------------------------------------------------------------------------------  
  
**5)**

--Procedimientos:

CREATE procedure PROC\_COM\_1

@Id\_Com char (4)

AS

SELECT Nombre\_Ing, Costo\_IXC FROM Ingredientes

INNER JOIN IngredienteXComercio

ON Ingredientes.IdIngrediente\_Ing = IngredienteXComercio.IdIngrediente\_IXC

INNER JOIN Comercios

on IngredienteXComercio.IdComercio\_IXC = Comercios.IdComercio\_C

WHERE IdComercio\_C = @Id\_Com

GO

---------------------------------------------------------------------------------------

CREATE procedure PROC\_COM\_2

@Nombre\_C VARCHAR (50)

AS

SELECT IdComercio\_C FROM Comercios

WHERE Nombre\_C LIKE @Nombre\_C

GO

----------------------------------------------------------------------------------------- Busca las cantidades de un ingrediente dado según la receta.

CREATE procedure PROC\_REC\_1

@IdReceta\_Rec char (4)

AS

SELECT Nombre\_Ing , Cantidad\_RXI FROM Recetas

INNER JOIN RecetaXIngrediente ON IdReceta\_RXI = IdReceta\_Rec

INNER JOIN Ingredientes ON IdIngrediente\_Ing = IdIngrediente\_RXI

WHERE IdReceta\_Rec = @IdReceta\_Rec

GO

---------------------------------------------------------------------------------------

--Busca la cantidad de porciones segun el ID\_RECETA

CREATE procedure PROC\_REC\_2

@IdReceta\_Rec char (4)

AS

SELECT Porciones\_Rec FROM Recetas

WHERE IdReceta\_Rec = @IdReceta\_Rec

GO

---------------------------------------------------------------------------------------

-- RECETAS.setConsulta(string recetaNombre)

CREATE procedure PROC\_REC\_3

@Nombre\_Rec VARCHAR (50)

AS

SELECT IdReceta\_Rec FROM Recetas

WHERE Nombre\_Rec LIKE @Nombre\_Rec

GO

---------------------------------------------------------------------------------------

--Regresa una tabla con todos los macronutrientes :

CREATE procedure PROC\_MACRO

AS

SELECT IdReceta\_Rec,

SUM(

CAST((( nullif(Calorias\_Ing,0) /Cantidad\_Ing ) \*Cantidad\_RXI) / Porciones\_Rec as INT)

) AS CaloriasXreceta ,

SUM(

CAST((( nullif(Carbohidratos\_Ing,0)/Cantidad\_Ing ) \*Cantidad\_RXI) / Porciones\_Rec as INT)

) AS CarbohidratosXreceta ,

SUM(

CAST((( nullif(Proteinas\_Ing,0)/Cantidad\_Ing ) \*Cantidad\_RXI) / Porciones\_Rec as INT)

) AS ProteinasXreceta ,

SUM(

CAST((( nullif(Grasas\_Ing,0)/Cantidad\_Ing ) \*Cantidad\_RXI) / Porciones\_Rec as INT)

) AS GrasasXreceta

FROM Ingredientes

INNER JOIN RecetaXIngrediente ON IdIngrediente\_RXI = IdIngrediente\_Ing

INNER JOIN Recetas ON IdReceta\_Rec = IdReceta\_RXI

GROUP BY IdReceta\_Rec

GO

---------------------------------------------------------------------------------------

--Consulta hasta cuantas calorias en todas las recetas:

CREATE procedure PROC\_BUSC\_CALORIAS

@Num int

AS

SELECT \* FROM

(

SELECT

Nombre\_Rec,

Tiempo\_Aprox\_Rec,

SUM(

CAST((( nullif(Calorias\_Ing,0) /Cantidad\_Ing ) \*Cantidad\_RXI) / Porciones\_Rec as INT)

)

AS CaloriasXreceta

FROM Ingredientes

INNER JOIN RecetaXIngrediente ON IdIngrediente\_RXI = IdIngrediente\_Ing

INNER JOIN Recetas ON IdReceta\_Rec = IdReceta\_RXI

GROUP BY IdReceta\_Rec, Nombre\_Rec, Tiempo\_Aprox\_Rec

)

AS Total

WHERE CaloriasXreceta < @Num

GO

---------------------------------------------------------------------------------------

--Busqueda de receta por fecha de un perfil especifico:

CREATE procedure PROC\_FECHA\_PERF

@Fecha VARCHAR (11) , @Nombre\_P VARCHAR (50)

AS

SELECT Nombre\_Rec FROM Perfiles

INNER JOIN RecetaXFecha ON IdPerfil\_RXF = IdPerfil\_P

INNER JOIN Recetas ON IdReceta\_Rec = IdReceta\_RXF

WHERE IdFecha\_RXF = @Fecha AND Nombre\_P LIKE @Nombre\_P

GO

---------------------------------------------------------------------------------------

--Consulta de las recetas de una fecha, de un perfil y toda la suma de sus macronutrientes.

CREATE procedure PROC\_FECHA\_REC

@IdPerfil\_P INT, @Fecha DATE

AS

SELECT

Nombre\_Rec AS Receta,

SUM(

CAST((( nullif(Calorias\_Ing,0) /Cantidad\_Ing ) \* Cantidad\_RXI) / Porciones\_Rec AS decimal(16,0))

) AS Calorias ,

SUM(

CAST((( nullif(Proteinas\_Ing,0) /Cantidad\_Ing ) \* Cantidad\_RXI) / Porciones\_Rec AS decimal(16,0))

) AS Proteinas ,

SUM(

CAST((( nullif(Carbohidratos\_Ing,0) /Cantidad\_Ing ) \* Cantidad\_RXI) / Porciones\_Rec AS decimal(16,0))

) AS Carbohidratos ,

SUM(

CAST((( nullif(Grasas\_Ing,0) /Cantidad\_Ing ) \* Cantidad\_RXI) / Porciones\_Rec AS decimal(16,0))

) AS Grasas

FROM Ingredientes

INNER JOIN RecetaXIngrediente ON IdIngrediente\_RXI = IdIngrediente\_Ing

INNER JOIN Recetas ON IdReceta\_Rec = IdReceta\_RXI

INNER JOIN RecetaXFecha ON IdReceta\_RXF = IdReceta\_Rec

WHERE IdPerfil\_RXF = @IdPerfil\_P AND IdFecha\_RXF = @Fecha

GROUP BY Nombre\_Rec

GOb

---------------------------------------------------------------------------------------

--Busca la ultima fecha donde se cargo un peso de un perfil determinado por ID.

CREATE procedure PROC\_PESO\_FECHA

@IdPerfil\_P int

AS

SELECT Fecha\_PH , Peso\_PH, IdPesoH\_PH, IdPerfil\_PH FROM PesosHistoricos

WHERE IdPerfil\_PH = @IdPerfil\_P

ORDER BY Fecha\_PH DESC

GO

---------------------------------------------------------------------------------------

--Consultas sueltas:

---------------------------------------------------------------------------------------

--Genera una tabla donde muestra todas los macronutrientes consumidos en las fechas de una perfil especifico.

SELECT IdPerfil\_RXF, IdFecha\_RXF, Calorias\_Ing,Proteinas\_Ing,Carbohidratos\_Ing,Grasas\_Ing FROM Ingredientes

INNER JOIN RecetaXIngrediente ON IdIngrediente\_RXI = IdIngrediente\_Ing

INNER JOIN Recetas ON IdReceta\_Rec = IdReceta\_RXI

INNER JOIN RecetaXFecha ON IdReceta\_RXF = IdReceta\_Rec

WHERE IdPerfil\_RXF = 1 AND IdFecha\_RXF = '2017-07-18'

---------------------------------------------------------------------------------------

--Busqueda de ingredientes por receta

SELECT Nombre\_Ing , Cantidad\_RXI FROM Recetas

INNER JOIN RecetaXIngrediente ON IdReceta\_RXI = IdReceta\_Rec

INNER JOIN Ingredientes ON IdIngrediente\_Ing = IdIngrediente\_RXI

---------------------------------------------------------------------------------------

--Consulta hasta calorias.

SELECT \* FROM

(

SELECT

R1.Nombre\_Rec,

R1.Tiempo\_Aprox\_Rec,

SUM(

CAST(((Calorias\_Ing/Cantidad\_Ing ) \*Cantidad\_RXI) / R1.Porciones\_Rec as INT)

)

AS CaloriasXreceta

FROM Ingredientes

INNER JOIN RecetaXIngrediente ON IdIngrediente\_RXI = IdIngrediente\_Ing

INNER JOIN Recetas R1 ON R1.IdReceta\_Rec = IdReceta\_RXI

GROUP BY R1.IdReceta\_Rec, R1.Nombre\_Rec, R1.Tiempo\_Aprox\_Rec

)

AS Total

WHERE CaloriasXreceta < 800

---------------------------------------------------------------------------------------

--Consulta la ultima fecha de una perfil especifico.

SELECT Fecha\_PH , Peso\_PH, IdPesoH\_PH, IdPerfil\_PH FROM PesosHistoricos

WHERE IdPerfil\_PH = 1

ORDER BY Fecha\_PH DESC

---------------------------------------------------------------------------------------

**6)**  
--TRIGGER:

CREATE TRIGGER UPDATE\_INGREDIENTE ON Ingredientes

AFTER UPDATE

AS

BEGIN

SET NOCOUNT ON;

BEGIN

UPDATE IngredientexComercio

SET Costo\_IXC =(SELECT CAST ((Cantidad\_Ing \* Costo\_IXC) AS INT) FROM Ingredientes

INNER JOIN IngredientexComercio ON IdIngrediente\_IXC = IdIngrediente\_Ing)

END

END

GO

---------------------------------------------------------------------------------------------------------------------------------------