



VERDAD, BELLEZA, PROBIIDAD

UAT



Facultad de Ingeniería  
Arturo Narro Siller

Coordinación de Carrera  
Desarrollo de Prácticas de Laboratorio  
R-OP-33-15-01

ACADEMIA / LÍNEA DE INVESTIGACIÓN	MATERIA / PROYECTO
Tratamiento de la información	Base de Datos Avanzadas Corporativas

EXAMEN No.	NOMBRE DE EXAMEN	DURACIÓN (HORAS)	ALUMNOS POR GRUPO DE TRABAJO
D	PROYECTO FINAL DE SISTEMA DE FACTURACION		1

### 1. INTRODUCCIÓN

La realización de su examen requiere de todos sus conocimientos básicos adquiridos durante el curso empleando los cursores, procedimientos almacenados, pivots en la base de datos y aplicar conocimientos básicos de la solución FRONT END en proyecto de PHP.

### 2. OBJETIVO(S)

El alumno empleara sus conocimientos básicos y avanzados en base de datos sql, y conocimientos básicos de PHP.

### 3. EQUIPO REQUERIDO

1 Computadora

### 4. MATERIAL REQUERIDO

Aplicaciones MYSQL  
Aplicaciones SQL SERVER  
XAMP  
PHP

### 5. RECOMENDACIONES DE SEGURIDAD

Desarrollo de la aplicación FRONT END, y de BACK END

### 6. PROCEDIMIENTO

Generar los procedimientos almacenados necesarios para realizar el CRUD de la información de los Proveedores.  
Generar los procedimientos almacenados para la generación de la Compra.

Generar los reportes de los Ventas:

En donde usted indique la fecha inicial y la fecha final y devuelva todas las ventas que correspondan con ese rango de fechas.

En donde se ingrese el nombre del cliente y devuelva los artículos que se compran ese cliente.

Generar la Estadística de los Clientes devolviendo la información de la siguiente manera:

Cliente	Mouse	Teclado	Tarjeta
aBc	500	0	0

Generar le reporte estadístico General de los artículos:

Articulo	Stock	Vendido	Comprado	Existencia	Costo	Precio Máximo	PERDIDA/Ganancia
.....							

Donde la columna de PERDIDA/GANACIA será si el precio máximo es mayor que el costo si es igual o menor será PERDIDA.



VERDAD, BELLEZA, PROBIIDAD

UAT



Facultad de Ingeniería  
Arturo Narro Siller

Coordinación de Carrera  
Desarrollo de Prácticas de Laboratorio  
R-OP-33-15-01

ELABORÓ / DIRECTOR DE LA INVESTIGACIÓN	FIRMA	FECHA DE ELABORACIÓN
Zubiri Valdez Hedson Leonardo		25-11-21

## REPORTE DE PRÁCTICA.

### 1. Generar los procedimientos almacenados necesarios para realizar el CRUD de la información de los Proveedores.

Antes de desarrollar los procedimientos almacenados se asegura el que campo de ID se aumente de manera automática en la tabla proveedor de MySQL y SQL Server.

En el caso de **MySQL** accedemos a las tablas y después a la estructura de dbo.proveedor.



Debemos cambiar el campo idProveedor:

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra	Acción
1	idProveedor	int(4)			No	Ninguna			Cambiar  Eliminar  Más
2	nombreProveedor	varchar(20)	utf8mb4_general_ci		Sí	NULL			Cambiar  Eliminar  Más

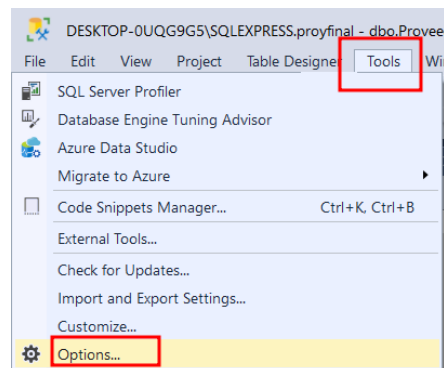
Entre las opciones debemos activar A\_I, es decir el Auto Increment:



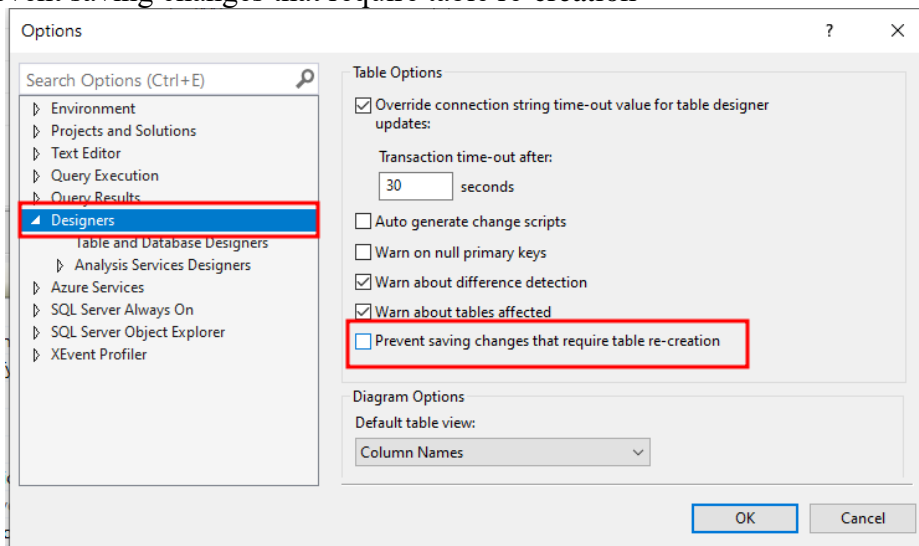
Se puede identificar que el cambio se encuentra en el campo:

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra	Acción
1	idProveedor	int(4)			No	Ninguna		AUTO_INCREMENT	Cambiar  Eliminar  Más
2	nombreProveedor	varchar(20)	utf8mb4_general_ci		Sí	NULL			Cambiar  Eliminar  Más
3	RUCProveedor	varchar(11)	utf8mb4_general_ci		Sí	NULL			Cambiar  Eliminar  Más

En el caso de **SQL Server** debemos acceder a Tools/Options:



Para evitar un mensaje de error al querer cambiar el campo debemos buscar el apartado Designers y activar “Prevent saving changes that require table re-creation”



Accedemos en la jerarquía de bases de datos de sql server y buscamos la sección de tablas y la tabla proveedores.

Al dar click derecho sobre la tabla accedemos a la propiedad design:



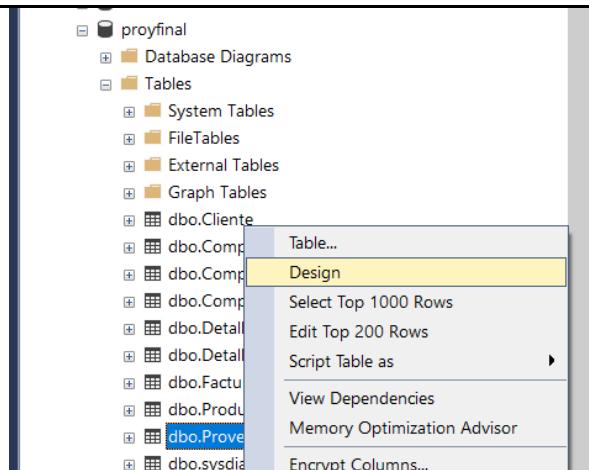
VERDAD, BELLEZA, PROBIIDAD

UAT

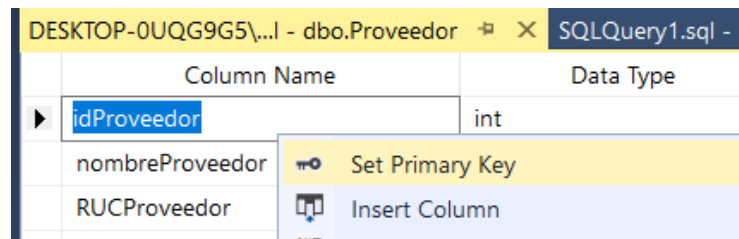


Facultad de Ingeniería  
Arturo Narro Siller

Coordinación de Carrera  
**Desarrollo de Prácticas de Laboratorio**  
**R-OP-33-15-01**

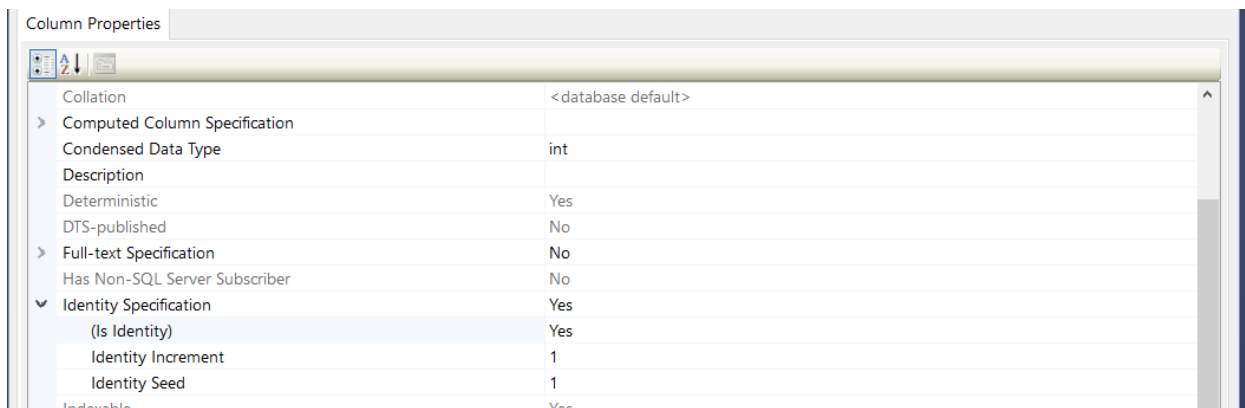


Al dar click derecho indicamos que el campo idProveedor es Primary Key



Aún con el campo idProveedor seleccionado en la parte inferior accedemos a las propiedades de la columna, ahí colocamos en Identity Specification:

- (Is Identity) yes
- Identity Increment 1



Con lo anterior nos aseguramos de trabajar bien con métodos como el de creación de proveedores entre otros. De tal manera que los procedimientos almacenados para operaciones CRUD con la tabla proveedores queda de la siguiente manera:



UAT

## MySQL

### CREATE

DELIMITER \$\$

```
CREATE PROCEDURE crearProveedor(  
    IN nomP varchar(50),  
    IN RUC varchar(11),  
    IN direc varchar(70),  
    IN tel varchar(10),  
    IN fecha date,  
    IN estado tinyint(1)  
)
```

BEGIN

```
INSERT INTO `dbo.Proveedor` (  
    nombreProveedor,  
    RUCProveedor,  
    direccionProveedor,  
    TelefonoProveedor,  
    FechaRegistro,  
    EstadoProveedor  
)
```

```
VALUES (nomP,RUC,direc,tel,fecha,estado);
```

END\$\$

```
CALL crearProveedor("Hedson Zubiri","1111111","Altamira","8332321226","2021-11-24",1);
```

### READ

DELIMITER \$\$

```
CREATE PROCEDURE getProveedores()
```

BEGIN

```
SELECT * FROM `dbo.proveedor`;
```

END \$\$

```
CALL getProveedores();
```

DELIMITER \$\$

```
CREATE PROCEDURE getProveedorByID(IN id int)
```

BEGIN

```
SELECT * FROM `dbo.proveedor` where idProveedor = id;
```

END \$\$

```
CALL getProveedorByID(11);
```



VERDAD, BELLEZA, PROBIIDAD

UAT



Facultad de Ingeniería  
Arturo Narro Siller

Coordinación de Carrera  
Desarrollo de Prácticas de Laboratorio  
R-OP-33-15-01

## UPDATE

DELIMITER \$\$

CREATE PROCEDURE updateProveedor(

IN id int,  
IN nomP varchar(50),  
IN RUC varchar(11),  
IN direc varchar(70),  
IN tel varchar(10),  
IN fecha date,  
IN estado tinyint(1)

)

BEGIN

UPDATE `dbo.Proveedor`

SET

nombreProveedor=nomP,  
RUCProveedor = RUC,  
direccionProveedor= direc,  
TelefonoProveedor = tel,  
FechaRegistro = fecha,  
EstadoProveedor = estado

WHERE idProveedor = id;

END \$\$

CALL updateProveedor(11,"Hedson Zubiri","1","Altamira","8332321226","2021-11-24",0);

## DELETE

DELIMITER \$\$

CREATE PROCEDURE delProveedorByID(IN id int)

BEGIN

DELETE FROM `dbo.proveedor` WHERE idProveedor=id;

END \$\$

CALL delProveedorByID(11);



VERDAD, BELLEZA, PROBIIDAD

UAT



Facultad de Ingeniería  
Arturo Narro Siller

Coordinación de Carrera  
Desarrollo de Prácticas de Laboratorio  
R-OP-33-15-01

## SQL Server

### CREATE

GO

```
CREATE PROCEDURE crearProveedor(  
    @nomP varchar(50),  
    @RUC varchar(11),  
    @direc varchar(70),  
    @tel varchar(10),  
    @fecha date,  
    @estado bit  
)
```

AS

```
INSERT INTO Proveedor(  
    nombreProveedor,  
    RUCProveedor,  
    direccionProveedor,  
    TelefonoProveedor,  
    FechaRegistro,  
    EstadoProveedor  
)
```

```
VALUES (@nomP,@RUC,@direc,@tel,@fecha,@estado);
```

GO

```
EXEC crearProveedor "Hedson Zubiri","1111111","Altamira","8332321226","2021-11-24",1;
```

### READ

GO

```
CREATE PROCEDURE getProveedores
```

AS

```
SELECT * FROM Proveedor;
```

GO

```
EXEC getProveedores;
```

GO

```
CREATE PROCEDURE getProveedorByID(@id int)
```

AS

```
SELECT * FROM Proveedor
```

```
WHERE idProveedor=@id;
```

GO

```
EXEC getProveedorByID 11;
```



UAT

## UPDATE

GO

```
CREATE PROCEDURE updateProveedor(  
    @id int,  
    @nomP varchar(50),  
    @RUC varchar(11),  
    @direc varchar(70),  
    @tel varchar(10),  
    @fecha date,  
    @estado bit  
)
```

AS

UPDATE Proveedor

SET

```
    nombreProveedor = @nomP,  
    RUCProveedor = @RUC,  
    direccionProveedor = @direc,  
    TelefonoProveedor = @tel,  
    FechaRegistro = @fecha,  
    EstadoProveedor = @estado
```

WHERE idProveedor = @id;

GO

```
EXEC updateProveedor 11,"Leo Zubiri","1111111","Altamira","8332321226","2021-11-24",1;
```

## DELETE

GO

```
CREATE PROCEDURE delProveedorByID(@id int)
```

AS

```
DELETE FROM Proveedor
```

```
WHERE idProveedor = @id;
```

GO

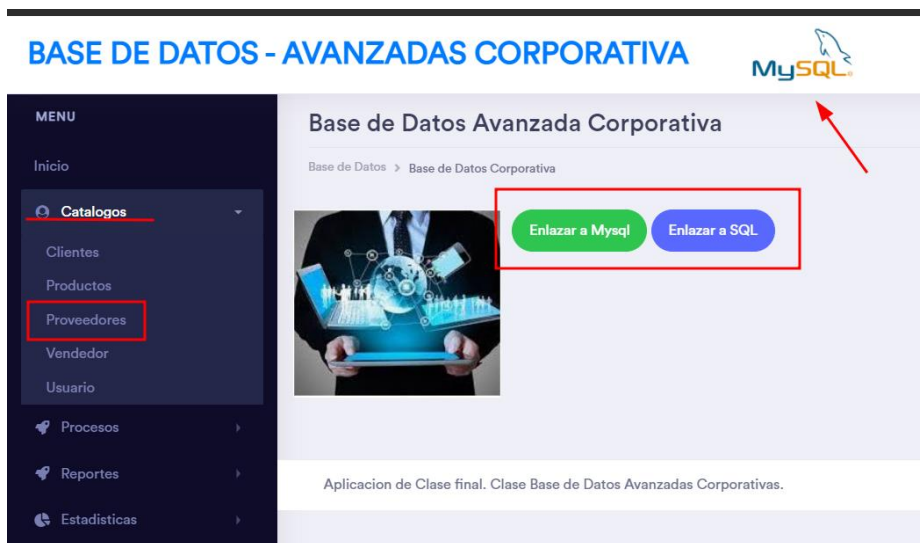
```
EXEC delProveedorByID 11
```

## Relación de los procedimientos almacenados y el Frontend

*Partiendo de que los procedimientos ya se encuentran creados dentro de las respectivas bases de datos.*



En la página principal hay dos botones para cambiar la base de datos a la que se está conectado, así como en todo momento se visualiza una imagen según esté realizada esta configuración:



Las operaciones CRUD de la tabla proveedores se llevan a cabo visualmente desde el menú en Catálogos/Proveedores, las opciones necesarias son mandadas mediante el link de la página.

Por ejemplo, desde la página de inicio según la base de datos a la que estemos conectada se puede leer un 1 o 2:

localhost/proyphp/index.php?server=2

DATOS - AVANZADAS CORPORATIVA



localhost/proyphp/index.php?server=1

DATOS - AVANZADAS CORPORATIVA



Este parámetro servidor es transferido al acceder al catálogo de proveedores para desplegar toda la información según la base de datos.



VERDAD, BELLEZA, PROBIIDAD

UAT



Facultad de Ingeniería  
Arturo Narro Siller

Coordinación de Carrera  
Desarrollo de Prácticas de Laboratorio  
R-OP-33-15-01

localhost/proyphp/proveedores.php?server=1

## BASE DE DATOS - AVANZADAS CORPORATIVA

MySQL

MENU

- Inicio
- Catalogos
- Procesos
- Reportes
- Estadísticas

Agregar Nuevo Proveedor

ID	Nombre	RUC	Direccion	Telefono	Fecha	Estado	Editar   Eliminar
1	David Corporation	12345676543	Santa Rosa 234	044567890	2013-05-01	1	
2	Computer House	32134546445	Av. España 435	044234657	2013-06-03	1	
3	CS Computer	32434567898	Av. Arequipa 145	044152451	2013-06-04	1	
4	Corporation Primattec	23423431241	Av. Crepusculo 456	044354253	2013-07-15	1	

El archivo cargado para esto es proveedores.php. El contenido inicial y lo más relevante de este archivo:

```
<?php
    include "encabezado.php";
    include "left.php";
    include_once "DB/CRUD.php";

    if(isset($_GET["server"])){
        echo '<script type="text/javascript"> setDBimages('.$opcDB.'); </script> ';
        $opcDB = $_GET["server"];
        $proveedor = getData($opcDB);
    }
?>
```

Se incluye los archivos de interfaz y el archivo para las operaciones CRUD. Al momento de ser cargado el archivo se establece el server y se coloca una respectiva imagen en el encabezado. En la variable \$proveedor se obtiene el retorno del método getData( ).

Cuando recibe los resultados en \$proveedor son cargados ese mismo archivo mediante:

```
<div class="table-responsive">
    <table class="table table-bordered">
        <thead class="bg-success">
            <tr>
                <th>ID</th>
                . . .
                <th>Editar | Eliminar</th>
            </tr>
        </thead>
        <tbody>
```



VERDAD, BELLEZA, PROBIIDAD

UAT



Facultad de Ingeniería  
Arturo Narro Siller

Coordinación de Carrera  
Desarrollo de Prácticas de Laboratorio  
R-OP-33-15-01

```
<?php foreach($proveedor as $prov){ ?>
<tr>
    <td><?php echo $prov->idProveedor ?></td>
    . . .
    <td><?php echo $prov->EstadoProveedor ?></td>
    <td>
        <a class="btn table-active text-white" href="<?php
        echo "editProveedor.php?server=".$opcDB."&id=" . $prov->idProveedor
        ?>"
        >✎
        </a> | <a class="btn table-danger text-white" href="<?php
        echo "DB/action.php?action=DEL&server=".$opcDB."&id=".$prov->idProveedor
        ?>">🗑
        </a></td>
    </tr>
<?php } ?>
</tbody>
</table>
</div>
```

El método `getData` se encuentra en el **CRUD.php**, en este archivo se encuentra la conexión de la base de datos, así como todas las funciones necesarias para mandar a llamar los procedimientos almacenados respectivamente:

Para la **conexión** se recibe un 1 para MySQL y un 2 para SQL Server:

```
function conectar($opcDB){
    if($opcDB == 1){ //MySQL
        host="localhost";
        try{
            $GLOBALS['cnn']=new PDO("mysql:host=".$GLOBALS['host'].";dbname=".$GLOBALS['bd'].",
            $GLOBALS['user'], $GLOBALS['password']);
            $GLOBALS['cnn']->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
        }catch (PDOException $e){
            echo "Error!: No se pudo conectar a la bd ".$GLOBALS['bd']."<br/>";
            die();
        }
    }
    elseif($opcDB ==2){ //SQL Server
        $host = "DESKTOP-0UQG9G5\SQLEXPRESS";
        try {
            $GLOBALS['cnn'] = new PDO("sqlsrv:server=$host;database=".$GLOBALS['bd'].");
            $GLOBALS['cnn']->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
        }catch (Exception $e) {
            echo "Ocurrió un error con la base de datos: " . $e->getMessage();
            die();
        }
        if($GLOBALS['bd']) echo "Correcto";
    }
}
```



VERDAD, BELLEZA, PROBIIDAD

UAT



Facultad de Ingeniería  
Arturo Narro Siller

Coordinación de Carrera  
Desarrollo de Prácticas de Laboratorio  
R-OP-33-15-01

Las variables \$GLOBALS hacen referencia a variables fuera de la función:

```
$cnn = null;    //Connection se guarda en una variable
$host="";
$user="root";
$password="";
$bd="proyfinal";
$data=null;
$opcDB=1;
```

La función de **getData** así como el resto de operaciones utilizan PDO debido a que la sintaxis es la misma y lo único que cambia es la sintaxis de la sentencia SQL a ejecutar:

### CREATE

```
function crearProveedor($opcDB,$d){
    $cnn = &$GLOBALS['cnn'];
    conectar($opcDB);

    //Preparamos la consulta sql
    if($opcDB==1) $respuesta = $cnn->prepare("CALL crearProveedor(?,?,?,?,?);");
    else if ($opcDB==2) $respuesta = $cnn->prepare("EXEC crearProveedor ?,?,?,?,?,? ;");

    //Ejecutamos la consulta
    $respuesta->execute([$d[0],$d[1],$d[2],$d[3],$d[4],$d[5]]);
    desconectar();
    return $respuesta;
}
```

### READ

```
function getData($opcDB){
    if($opcDB==1 || $opcDB==2){
        try {
            //Creamos la conexión PDO por medio de una instancia de su clase
            $cnn = &$GLOBALS['cnn'];
            conectar($opcDB);

            //Preparamos la consulta sql
            if($opcDB==1) $respuesta = $cnn->prepare("CALL getProveedores()");
            else if ($opcDB==2) $respuesta = $cnn->prepare("EXEC getProveedores");

            //Ejecutamos la consulta
            $respuesta->execute();
            $results = $respuesta -> fetchAll(PDO::FETCH_OBJ);
            desconectar();
            return $results;

        } catch (Exception $e) { echo $e->getMessage(); return null; }
    }
}
```



VERDAD, BELLEZA, PROBIIDAD

UAT



Facultad de Ingeniería  
Arturo Narro Siller

Coordinación de Carrera  
Desarrollo de Prácticas de Laboratorio  
R-OP-33-15-01

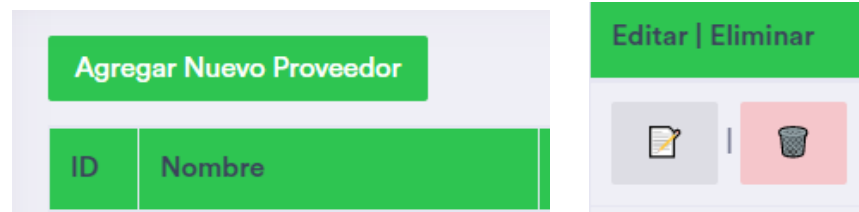
## UPDATE

```
function updateProveedor($id,$opcDB,$d){  
    $cnn = &$GLOBALS['cnn'];  
    conectar($opcDB);  
  
    if($opcDB==1) $respuesta = $cnn->prepare("CALL updateProveedor(?,?,?,?,?,?,?);");  
    else if($opcDB==2) $respuesta = $cnn->prepare("EXEC updateProveedor ?,?,?,?,?,?,? ;");  
  
    //Ejecutamos la consulta  
    $respuesta->execute([$id,$d[0],$d[1],$d[2],$d[3],$d[4],$d[5]]);  
    desconectar();  
    return $respuesta;  
}
```

## DELETE

```
function delProveedor($id,$opcDB){  
    $cnn = &$GLOBALS['cnn'];  
    conectar($opcDB);  
  
    if($opcDB==1) $respuesta = $cnn->prepare("CALL delProveedorByID(?);");  
    else if ($opcDB==2) $respuesta = $cnn->prepare("EXEC delProveedorByID ? ;");  
  
    //Ejecutamos la consulta  
    $respuesta->execute([$id]);  
    desconectar();  
    return $respuesta;  
}
```

El resto de recursos que utilizan estos métodos:





VERDAD, BELLEZA, PROBIDAD

UAT



Facultad de Ingeniería  
Arturo Narro Siller

Coordinación de Carrera  
Desarrollo de Prácticas de Laboratorio

R-OP-33-15-01

### Crear Proveedor

Nombre

RUC

Dirección

Teléfono

Fecha Registro

AAAA-MM-DD

Estado

Activo

Guardar

Volver

### Editar Proveedor

ID

1

Nombre

David Corporation

RUC

12345676543

Dirección

Santa Rosa 234

Teléfono

044567890

Fecha Registro

2013-05-01

Estado

Activo

Guardar

Volver

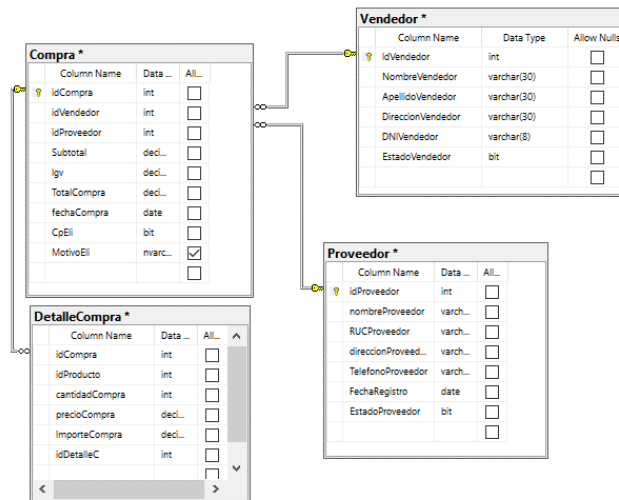


VERDAD, BELLEZA, PROBIIDAD

UAT

## 2. Generar los procedimientos almacenados para la generación de la Compra.

Analizando las tablas:



112 %

Results Messages **Compra**  $\text{subtotal} \times 1.18 = \text{totalCompra}$

	idCompra	idVendedor	idProveedor	Subtotal	Igv	TotalCompra	fechaCompra	CpEli	MotivoEli
1	1	2	3	1680.00	302.40	1982.40	2013-10-03	1	NULL
2	2	2	2	120.00	21.60	141.60	2013-10-04	1	NULL
3	3	2	6	720.00	129.60	849.60	2013-10-04	1	NULL
4	4	2	1	760.00	136.80	896.80	2013-10-04	1	NULL

**DetalleCompra**

	idCompra	idProducto	cantidadCompra	precioCompra	ImporteCompra	idDetalleC
1	1	9	X	300.00	900.00	65
2	1	4	X	355.00	710.00	66
3	1	10	X	70.00	70.00	67
4	2	1		25.00	50.00	68
5	2	10		70.00	70.00	69
6	3	5	2	360.00	720.00	70
7	4	7		60.00	120.00	71
8	4	12		320.00	640.00	72

112 %

Results Messages **Producto** **stock --**

	idProducto	NombreProducto	PCProducto	PVProducto	StockProducto	FechaActProducto	MarcaProducto	UnidMedProducto	EstadoProducto
1	1	Mouse	20.00	25.00	146	2012-08-20 00:00:00.000	Avatec	Unidades	1
2	2	Teclado	30.00	35.00	230	2012-08-21 00:00:00.000	Genius	Unidades	1
3	3	Memoria RAM	140.00	160.00	463	2012-08-22 00:00:00.000	Kingston	Unidades	1
4	4	Disco Duro	350.00	355.00	85	2013-10-01 00:00:00.000	Sata	Unidades	1
5	5	Impresora	350.00	360.00	529	2013-01-16 00:00:00.000	Hp	Unidades	1
6	6	Monitor	450.00	500.00	59	2013-01-22 00:00:00.000	SANSUMG	Unidades	1
7	7	Parlantes	50.00	60.00	200	2013-01-16 00:00:00.000	Hallon	Unidades	1
8	8	Switch 8 Puertos	35.00	45.00	19	2013-01-23 00:00:00.000	Link	Unidades	1
9	9	Procesador Core i7	280.00	300.00	5	2013-01-23 00:00:00.000	Intel	Unidades	1
10	10	Case	60.00	70.00	5	2013-01-28 00:00:00.000	Avatec	Unidades	1
11	11	Router	150.00	180.00	16	2013-05-23 00:00:00.000	Cisco	Unidades	1
12	12	Procesador Core i5	280.00	320.00	13	2013-05-23 00:00:00.000	LG	Unidades	1
13	13	Tarjeta de Video	50.00	80.00	15	2013-05-23 00:00:00.000	Kingston	Unidades	1
14	14	Scanner	250.00	300.00	15	2013-05-26 00:00:00.000	HP	Unidades	1
15	15	Router Cisco	120.00	125.00	3	2013-08-29 00:00:00.000	CISCO	Unidades	1
16	16	Camara Web	25.00	40.00	3	2013-08-29 00:00:00.000	Micronisc	Unidades	1
17	17	sgfhdfhd	34.00	37.40	8	2013-10-01 00:00:00.000	fgfdgfd	unidad	0



VERDAD, BELLEZA, PROBIIDAD

UAT



Facultad de Ingeniería  
Arturo Narro Siller

Coordinación de Carrera  
Desarrollo de Prácticas de Laboratorio  
R-OP-33-15-01

Previamente desde el front se hace una consulta de los productos para saber qué datos mandar:

### SQL SERVER

```
USE ProyFinal  
GO
```

```
CREATE PROCEDURE createCompra(  
    @idVen int,  
    @idProv int,  
    @subTotal decimal(18,2)  
) AS  
INSERT INTO Compra  
VALUES(  
    @idVen,  
    @idProv,  
    @subTotal,  
    @subTotal * 0.18,  
    @subTotal * 1.18,  
    GETDATE(),  
    1,  
    NULL  
)  
GO  
CREATE PROCEDURE insertDetCompra(  
    @idProd int,  
    @cant int,  
    @precio decimal(18,2),  
    @importe decimal(18,2)  
) AS  
INSERT INTO DetalleCompra  
VALUES(  
    (SELECT MAX(idCompra) FROM Compra),  
    @idProd,  
    @cant,  
    @precio,  
    @importe  
)
```

### MySQL

```
DELIMITER $$  
CREATE PROCEDURE createCompra(  
    idVen int,  
    idProv int,  
    subTotal decimal(18, 2)  
) BEGIN  
INSERT INTO `dbo.compra` VALUES(  
    idVen,  
    idProv,  
    subTotal,
```





VERDAD, BELLEZA, PROBIIDAD

UAT



Facultad de Ingeniería  
Arturo Narro Siller

Coordinación de Carrera  
Desarrollo de Prácticas de Laboratorio  
R-OP-33-15-01

```
subTotal * 0.18,  
subTotal * 1.18,  
NOW(),  
1,  
NULL  
)  
END$$ ;  
GO  
  
DELIMITER $$  
CREATE PROCEDURE insertDetCompra(  
    idProd int,  
    cant int,  
    precio decimal(18, 2),  
    importe decimal(18, 2)  
) BEGIN  
INSERT INTO `dbo.detalleCompra` VALUES(  
    (SELECT MAX(idCompra) FROM `dbo.compra`),  
    idProd,  
    cant,  
    precio,  
    importe  
)  
END$$ ;
```

## BASE DE DATOS - AVANZADAS CORPORATIVA



MENU

Inicio

Catalogos

Procesos

Reportes

Estadísticas

Compra

Disco Duro Sata

Mouse Avatec

Teclado Genius

Memoria RAM Kingston

Disco Duro Sata

Impresora Hp

Monitor SANSUMG

Parlantes Halion

Switch 8 Puertos Link

Procesador Core i7 Intel

Case Avatec

Router Cisco

Procesador Core i5 LG

Tarjeta de Video Kingston

Scanner HP

Router Cisco CISCO

Camara Web Micronisc

sghfdhfd fgfdgfd

Confirmar compra



VERDAD, BELLEZA, PROBIIDAD

UAT



Facultad de Ingeniería  
Arturo Narro Siller

Coordinación de Carrera  
Desarrollo de Prácticas de Laboratorio  
R-OP-33-15-01

### 3. Generar los reportes de las Ventas:

1. En donde usted indique la fecha inicial y la fecha final y devuelva todas las ventas que correspondan con ese rango de fechas.

#### SQL SERVER

```
USE proyfinal
GO
CREATE PROCEDURE betweenVentaFechas(@fechaIni DATE, @fechaFin DATE)
AS
SELECT v.IdVendedor, v.NombreVendedor, c.NombreCliente, f.Total, f.Fecha
FROM Facturas f
INNER JOIN Cliente c
ON c.IdCliente = f.IdCliente
INNER JOIN Vendedor v
ON v.IdVendedor = f.IdVendedor
WHERE f.Fecha BETWEEN @fechaIni AND @fechaFin
GO
EXEC betweenVentaFechas '2013/01/01', '2013/12/31'
```

	IdVendedor	NombreVendedor	NombreCliente	Total	Fecha
1	1	Andrea	Juan	147.50	2013-01-18 00:00:00
2	1	Andrea	Juan	59.00	2013-01-18 00:00:00
3	1	Andrea	Juan	177.00	2013-01-18 00:00:00
4	1	Andrea	Carlos	177.00	2013-01-18 00:00:00
5	1	Andrea	Carlos	212.40	2013-01-18 00:00:00
6	1	Andrea	Andrea	247.80	2013-01-18 00:00:00
7	1	Andrea	Maria	2855.60	2013-01-18 00:00:00
8	1	Andrea	Maria	637.20	2013-01-18 00:00:00
9	1	Andrea	Carlos	299.72	2013-01-18 00:00:00
10	1	Andrea	Carlos	1917.50	2013-01-18 00:00:00
11	1	Andrea	Andrea	2350.56	2013-01-18 00:00:00
12	1	Andrea	Pamela	708.00	2013-01-18 00:00:00

#### MySQL

```
DELIMITER $$
CREATE PROCEDURE betweenVentaFechas(fechaIni DATE, fechaFin DATE)
BEGIN
SELECT v.IdVendedor, v.NombreVendedor, c.NombreCliente, f.Total, f.Fecha
FROM `dbo.facturas` f
INNER JOIN `dbo.cliente` c
ON c.IdCliente = f.IdCliente
INNER JOIN `dbo.vendedor` v
ON v.IdVendedor = f.IdVendedor
WHERE f.Fecha BETWEEN fechaIni AND fechaFin;
END$$
CALL betweenVentaFechas('2013/01/01', '2013/12/31');
```

	IdVendedor	NombreVendedor	NombreCliente	Total	Fecha
1	1	Andrea	Juan	147.50	2013-01-18 00:00:00
1	1	Andrea	Juan	59.00	2013-01-18 00:00:00
1	1	Andrea	Juan	177.00	2013-01-18 00:00:00
1	1	Andrea	Carlos	177.00	2013-01-18 00:00:00
1	1	Andrea	Carlos	212.40	2013-01-18 00:00:00
1	1	Andrea	Andrea	247.80	2013-01-18 00:00:00



VERDAD, BELLEZA, PROBIIDAD

UAT



Facultad de Ingeniería  
Arturo Narro Siller

Coordinación de Carrera  
Desarrollo de Prácticas de Laboratorio

R-OP-33-15-01

## Código PHP

```
function getDataVentasPeriodo($opcDB,$fechaI,$fechaF){  
    $cnn = &$GLOBALS['cnn'];  
    conectar($opcDB);  
  
    //Preparamos la consulta sql  
    if($opcDB==1) $respuesta = $cnn->prepare("CALL betweenVentaFechas(?, ?);");  
    else if ($opcDB==2) $respuesta = $cnn->prepare("EXEC betweenVentaFechas ?, ?;");  
  
    //Ejecutamos la consulta  
    $respuesta->execute([$fechaI,$fechaF]);  
    $results = $respuesta -> fetchAll(PDO::FETCH_OBJ);  
    desconectar();  
    return $results;  
}
```

El resultado web es el siguiente:

## BASE DE DATOS - AVANZADAS CORPORATIVA



MENU

- Inicio
- Catalogos
- Procesos
- Reportes
- Estadísticas

### Consultar Ventas

Periodo InicialPeriodo Final

Consultar

ID Vendedor	Nombre	Nombre Cliente	Total	Fecha
1	Andrea	Juan	147.50	2013-01-18 00:00:00
1	Andrea	Juan	59.00	2013-01-18 00:00:00
1	Andrea	Juan	177.00	2013-01-18 00:00:00
1	Andrea	Carlos	177.00	2013-01-18 00:00:00
1	Andrea	Carlos	212.40	2013-01-18 00:00:00
1	Andrea	Andrea	247.80	2013-01-18 00:00:00
1	Andrea	Maria	2855.60	2013-01-18 00:00:00



VERDAD, BELLEZA, PROBIIDAD

UAT



Facultad de Ingeniería  
Arturo Narro Siller

Coordinación de Carrera  
Desarrollo de Prácticas de Laboratorio  
R-OP-33-15-01

2. En donde se ingrese el **nombre del cliente** y devuelva los artículos que se compran por ese cliente

#### SQL SERVER

```
use proyfinal
GO
CREATE PROCEDURE getProductosByNomCliente(@nombre varchar(50))
AS
select distinct cl.IdCliente,cl.NombreCliente,detFac.IdProducto,p.NombreProducto from Facturas
fac
inner join Cliente cl
on fac.IdCliente = cl.IdCliente
inner join Detalle detFac
on detFac.num_fact = fac.num_fact
inner join Producto p
on p.IdProducto = detFac.IdProducto
where cl.NombreCliente+' '+cl.ApellidoCliente like '%'+ @nombre + '%';

GO
EXEC getProductosByNomCliente 'Juan Perez Castillo' ;
```

Results		Messages	
1	1	Juan	1 Mouse
2	1	Juan	2 Teclado
3	1	Juan	3 Memoria RAM
4	1	Juan	4 Disco Duro
5	1	Juan	5 Impresora

#### MySQL

```
DELIMITER $$
CREATE PROCEDURE getProductosByNomCliente(nombre varchar(50))
BEGIN
SELECT DISTINCT cl.IdCliente,cl.NombreCliente,d.IdProducto,p.NombreProducto
FROM `dbo.facturas` f
INNER JOIN `dbo.cliente` cl
ON cl.IdCliente = f.IdCliente
INNER JOIN `dbo.Detalle` d
ON d.num_fact = f.num_fact
INNER JOIN `dbo.Producto` p
ON p.IdProducto = d.IdProducto
WHERE CONCAT(cl.NombreCliente,' ',cl.ApellidoCliente) LIKE CONCAT('%',nombre,'%');
END$$

CALL getProductosByNomCliente('Perez Castillo');
```

IdCliente	NombreCliente	IdProducto	NombreProducto
1	Juan	2	Teclado
1	Juan	1	Mouse
1	Juan	3	Memoria RAM
1	Juan	4	Disco Duro
1	Juan	5	Impresora



VERDAD, BELLEZA, PROBIIDAD

UAT



Facultad de Ingeniería  
Arturo Narro Siller

Coordinación de Carrera  
Desarrollo de Prácticas de Laboratorio  
R-OP-33-15-01

### Código PHP

```
function getDataArtsCliente($opcDB,$nombre){
    $cnn = &$GLOBALS['cnn'];
    conectar($opcDB);

    //Preparamos la consulta sql
    if($opcDB==1) $respuesta = $cnn->prepare("CALL getProductosByNomCliente(?);");
    else if ($opcDB==2) $respuesta = $cnn->prepare("EXEC getProductosByNomCliente ? ;");

    //Ejecutamos la consulta
    $respuesta->execute([$nombre]);
    $results = $respuesta -> fetchAll(PDO::FETCH_OBJ);
    desconectar();
    return $results;
}
```

### Resultado web:

#### BASE DE DATOS - AVANZADAS CORPORATIVA



MENU

- Inicio
- Catalogos**
- Procesos
- Reportes
- Estadísticas

### Consultar Compras de Cliente

Nombre y/o apellidos

ID Cliente	Nombre	ID Producto	Producto
1	Juan	2	Teclado
1	Juan	1	Mouse
1	Juan	3	Memoria RAM
1	Juan	4	Disco Duro
1	Juan	5	Impresora

Aplicacion de Clase final. Clase Base de Datos  
Avanzadas Corporativas.

### 4. Generar la Estadística de los Clientes devolviendo la información de la siguiente manera:

Cliente	Mouse	Teclado	Tarjeta
aBc	500	0	0



VERDAD, BELLEZA, PROBIIDAD

UAT



Facultad de Ingeniería  
Arturo Narro Siller

Coordinación de Carrera  
Desarrollo de Prácticas de Laboratorio

R-OP-33-15-01

## MYSQL

DELIMITER \$\$

CREATE PROCEDURE CantProductosClientes()

BEGIN

set @cols = null;

SELECT GROUP\_CONCAT(DISTINCT CONCAT(  
' SUM(IF(p.IdProducto=',IdProducto,', detFac.CantidadVenta, 0))AS `',NombreProducto,``'  
) ) INTO @cols FROM `dbo.producto`;

set @SQL = CONCAT(  
'SELECT cl.NombreCliente,',@cols,  
' FROM `dbo.facturas` fac  
inner join `dbo.cliente` cl  
on fac.IdCliente = cl.IdCliente  
inner join `dbo.detalle` detFac  
on detFac.num\_fact = fac.num\_fact  
inner join `dbo.producto` p  
on p.IdProducto = detFac.IdProducto  
GROUP BY cl.NombreCliente;'  
);

PREPARE stmt FROM @SQL;

execute stmt;

deallocate prepare stmt;

END\$\$

CALL CantProductosClientes();

NombreCliente	Mouse	Case	Router	Procesador Core i5	Tarjeta de Video	Scanner	Router Cisco	Camara Web	sgghdfhd	Teclado	Memoria RAM	Disco Duro	Impresora	Monitor	Parlantes	Switch 8 Puentes	Procesador Core i7
Andrea	0	0	0	0	0	0	0	0	0	0	0	11	3	3	0	0	0
Arturo	0	0	0	0	0	0	0	0	0	0	0	6	0	0	0	0	0
Carlos	0	0	0	0	0	0	0	0	0	4	0	8	20	6	0	0	0
David	0	0	0	0	1	0	0	1	0	0	0	0	1	0	2	0	0
Hobby	2	1	0	3	1	0	1	0	0	12	14	4	5	9	4	0	1
Javier	0	0	0	0	0	0	0	0	0	0	0	3	9	0	0	0	0
Juan	12	0	0	0	0	0	0	0	0	20	21	22	37	0	0	0	0
Luis	11	4	4	4	1	0	0	0	0	20	4	7	6	15	9	1	2
Maria	6	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0
Pamela	23	0	0	0	2	5	0	0	0	33	22	26	26	23	21	0	12
Paola	1	0	0	0	0	0	0	0	0	4	3	0	1	1	0	0	0
Paty	5	0	0	0	0	0	0	0	0	3	0	3	1	3	2	0	0
Perla	10	0	0	0	0	0	0	0	0	0	4	0	2	0	12	0	0
Roberto	7	5	0	0	0	0	0	0	0	15	12	13	8	7	0	0	0



VERDAD, BELLEZA, PROBIIDAD

UAT



Facultad de Ingeniería  
Arturo Narro Siller

Coordinación de Carrera  
Desarrollo de Prácticas de Laboratorio  
R-OP-33-15-01

## SQL SERVER

```
use proyfinal
GO
CREATE PROCEDURE CantProductosClientes
AS
declare @query varchar(4000)
declare @cols varchar(2000)
declare @restrictions varchar(2000)

--Obtenemos las columnas del pivot
SET @restrictions = STUFF(
(
SELECT DISTINCT ',isnull(' + QUOTENAME(p.[NombreProducto])+',0) ' +
QUOTENAME(p.[NombreProducto])
FROM [dbo].[Producto] p FOR XML
PATH(''), TYPE
).value('','nvarchar(max)'), 1, 1, '');

SET @cols = STUFF(
(
SELECT DISTINCT ',' + QUOTENAME(p.[NombreProducto])
FROM [dbo].[Producto] p FOR XML
PATH(''), TYPE
).value('','nvarchar(max)'), 1, 1, '');

--CONCATENAR QUERY FINAL
SET @query = 'SELECT [NombreCliente], '+@restrictions+
'from (
select cl.[NombreCliente],p.[NombreProducto],detFac.[CantidadVenta] from
[dbo].[Facturas] fac
inner join [dbo].[Cliente] cl
on fac.IdCliente = cl.IdCliente
inner join [dbo].[Detalle] detFac
on detFac.num_fact = fac.num_fact
inner join [dbo].[Producto] p
on p.IdProducto = detFac.IdProducto
)x pivot (sum(CantidadVenta) for NombreProducto in ('+@cols+')) n'
EXECUTE(@query);

GO
EXEC CantProductosClientes;
```

	NombreCliente	Camara Web	Case	Disco Duro	Impresora	Memoria RAM	Monitor	Mouse	Parlantes	Procesador Core i5	Procesador Core i7	Router Cisco	Router	Scanner	sgfhdhfd	Switch 8 Puertos	Tarjeta de Video	Teclado
1	Andrea	0	0	11	3	0	3	0	0	0	0	0	0	0	0	0	0	0
2	Arturo	0	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	Carlos	0	0	8	20	0	6	0	0	0	0	0	0	0	0	0	0	4
4	David	1	0	0	1	0	0	0	2	0	0	0	0	0	0	0	1	0
5	Hobby	0	1	4	5	14	9	2	4	3	1	1	0	0	0	0	1	12
6	Javier	0	0	3	9	0	0	0	0	0	0	0	0	0	0	0	0	0
7	Juan	0	0	22	37	21	0	12	0	0	0	0	0	0	0	0	0	20
8	Luis	0	4	7	6	4	15	11	9	4	2	0	4	0	0	1	1	20
9	Maria	0	0	0	2	0	0	6	0	0	0	0	0	0	0	0	0	0
10	Pamela	0	0	26	26	22	23	23	21	0	12	0	0	5	0	0	2	33
11	Paola	0	0	0	1	3	1	1	0	0	0	0	0	0	0	0	0	4
12	Paty	0	0	3	1	0	3	5	2	0	0	0	0	0	0	0	0	3
13	Perla	0	0	0	2	4	0	10	12	0	0	0	0	0	0	0	0	0
14	Roberto	0	5	13	8	12	7	7	0	0	0	0	0	0	0	0	0	15



VERDAD, BELLEZA, PROBIIDAD

UAT



Facultad de Ingeniería  
Arturo Narro Siller

Coordinación de Carrera  
Desarrollo de Prácticas de Laboratorio

R-OP-33-15-01

## Código PHP

```
function getNombreProductos($opcDB){
    $cnn = &$GLOBALS['cnn'];
    conectar($opcDB);

    //Preparamos la consulta sql
    if($opcDB==1) $respuesta = $cnn->prepare("SELECT * FROM `dbo.producto`");
    else if ($opcDB==2) $respuesta = $cnn->prepare("SELECT * FROM producto");

    //Ejecutamos la consulta
    $respuesta->execute();
    $results = $respuesta -> fetchAll(PDO::FETCH_OBJ);
    desconectar();
    return $results;
}

function getEstClientes($opcDB){
    $cnn = &$GLOBALS['cnn'];
    conectar($opcDB);

    //Preparamos la consulta sql
    if($opcDB==1) $respuesta = $cnn->prepare("CALL CantProductosClientes()");
    else if ($opcDB==2) $respuesta = $cnn->prepare("EXEC CantProductosClientes; ");

    //Ejecutamos la consulta
    $respuesta->execute();
    $results = $respuesta -> fetchAll(PDO::FETCH_OBJ);
    desconectar();
    return $results;
}
```

## Resultado WEB

BASE DE DATOS - AVANZADAS CORPORATIVA



Estadística de Cliente-Artículos																
Nombre Cliente	Mouse	Teclado	Memoria RAM	Disco Duro	Impresoras	Monitor	Parlantes	Switch & Puertos	Procesador Core i7	Casa	Router	Procesador Core i5	Tarjeta de Video	Scanner	Router Cloud	Camera Web
Andrea	0	0	0	11	3	3	0	0	0	0	0	0	0	0	0	0
Arturo	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0	0
Carlos	0	4	0	8	20	6	0	0	0	0	0	0	0	0	0	0
David	0	0	0	0	1	0	2	0	0	0	0	0	1	0	0	1
Hobby	2	12	14	4	5	9	4	0	1	1	0	3	1	0	1	0
Javier	0	0	0	3	9	0	0	0	0	0	0	0	0	0	0	0
Juan	12	20	21	22	37	0	0	0	0	0	0	0	0	0	0	0
Luis	11	20	4	7	6	15	9	1	2	4	4	4	1	0	0	0
Maria	6	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0
Demanda	21	32	25	26	26	21	21	0	12	0	0	0	2	0	0	0



## 5. Generar el reporte estadístico General de los artículos:

Articulo | Stock | Vendido | Comprado | Existencia | Costo | Precio Máximo | PERDIDA/Ganancia  
.....

Donde la columna de PERDIDA/GANACIA será si el precio máximo es mayor que el costo si es igual o menor será PERDIDA

```
GO
CREATE PROCEDURE estGeneralArts
AS

DROP TABLE IF EXISTS #TempProducto
CREATE TABLE #TempProducto(
    Articulo varchar(30),
    Stock int,
    Vendido int,
    Comprado int,
    Existencia bit,
    Costo decimal(9,1),
    PrecioMaximo decimal(9,1),
    PerdidaGanancia varchar(30)
)

DECLARE @nom varchar(30),
@stock int,
@vend int,
@comp int,
@exis bit,
@costo decimal(9,2),
@precio decimal(9,2),
@perd_gan decimal(9,2),
@txt_perd_gan varchar(30);

DECLARE miCursor CURSOR FOR
SELECT p.NombreProducto,p.StockProducto,
(SELECT count(CantidadVenta)
FROM Detalle
WHERE Detalle.IdProducto = p.IdProducto
),
0,
p.EstadoProducto,p.PCProducto,p.PVProducto,p.PVProducto
FROM Producto p;

OPEN miCursor
FETCH NEXT FROM miCursor
INTO @nom, @stock, @vend, @comp, @exis, @costo, @precio, @perd_gan
WHILE @@FETCH_STATUS = 0 BEGIN
FETCH NEXT FROM miCursor
INTO @nom, @stock, @vend, @comp, @exis, @costo, @precio, @perd_gan
SET @comp = @stock + @vend
```



VERDAD, BELLEZA, PROBIIDAD

UAT



Facultad de Ingeniería  
Arturo Narro Siller

Coordinación de Carrera  
Desarrollo de Prácticas de Laboratorio  
R-OP-33-15-01

```
IF @perd_gan > @costo
SET @txt_perd_gan = 'GANANCIA'
ELSE
SET @txt_perd_gan = 'PERDIDA'

INSERT INTO #TempProducto VALUES(
    @nom, @stock, @vend, @comp, @exis, @costo, @precio, @txt_perd_gan
)

END
SELECT * FROM #TempProducto
CLOSE miCursor
DEALLOCATE miCursor
GO

EXEC estGeneralArts
```

	Articulo	Stock	Vendido	Comprado	Existencia	Costo	PrecioMaximo	PerdidaGanancia
1	Teclado	230	36	266	1	30.0	35.0	GANANCIA
2	Memoria RAM	463	22	485	1	140.0	160.0	GANANCIA
3	Disco Duro	85	31	116	1	350.0	355.0	GANANCIA
4	Impresora	529	40	569	1	350.0	360.0	GANANCIA
5	Monitor	59	26	85	1	450.0	500.0	GANANCIA
6	Parlantes	200	19	219	1	50.0	60.0	GANANCIA
7	Switch 8 Puertos	19	1	20	1	35.0	45.0	GANANCIA
8	Procesador Core i7	5	7	12	1	280.0	300.0	GANANCIA
9	Case	5	4	9	1	60.0	70.0	GANANCIA
10	Router	16	1	17	1	150.0	180.0	GANANCIA
11	Procesador Core i5	13	3	16	1	280.0	320.0	GANANCIA
12	Tarjeta de Video	15	5	20	1	50.0	80.0	GANANCIA
13	Scanner	15	2	17	1	250.0	300.0	GANANCIA
14	Router Cisco	3	1	4	1	120.0	125.0	GANANCIA

BASE DE DATOS - AVANZADAS CORPORATIVA



Agregar Nuevo Proveedor								
Articulo	Stock	Vendido	Comprado	Existencia	Costo	PrecioMaximo	PerdidaGanancia	
Teclado	230	36	266	1	30.0	35.0	GANANCIA	
Memoria RAM	463	22	485	1	140.0	160.0	GANANCIA	
Disco Duro	85	31	116	1	350.0	360.0	GANANCIA	
Impresora	529	40	569	1	350.0	360.0	GANANCIA	

LINK DEL PROYECTO EN GITHUB

<https://github.com/Leo-Zubiri/PHP-MYSQL-SQLSERVER>