**PRUEBA TECNICA DE CONOCIMIENTOS SQL (PL/SQL – Transact-SQL) .NET**

**Nombre**: \_Edgar Alberto Suarez Mesa

**Fecha**: \_\_30/03/2021\_\_\_\_

**Car Center**

**Servicio de Mantenimiento:**

CAR CENTER es un taller que atiende a sus clientes a nivel nacional, prestando el servicio de reparación de vehículos con mecánicos altamente calificados y proporcionando todos los repuestos necesarios en el mantenimiento.

El taller provee a sus CLIENTES una plataforma donde pueden registrarse e inscribir sus VEHICULOS. De esta manera, cuando el cliente requiere un mantenimiento, puede solicitarlo a través de la plataforma e incluso subir fotos del estado de su vehículo para que pueda iniciarse una valoración del arreglo.

**Facturación:**

Dado el número de identificación del CLIENTE se genera una FACTURA para el cliente, la cual incluye el cobro de todos los MANTENIMIENTOS es estado “*terminado*”. La FACTURA debe contener todos los datos del cliente (primer nombre, segundo nombre, primer apellido, segundo apellido, , tipo de documento, documento, celular, dirección y correo electrónico), los datos del mecánico que hizo el mantenimiento (primer nombre, segundo nombre, primer apellido, segundo apellido, , tipo de documento, documento, celular, dirección y correo electrónico, estado), el desglose de los repuestos utilizados (precio por unidad, número de unidades, descuento), el desglose de servicios (precio de la mano de obra, descuento), y el total de la factura.

A este total se le aplica el 19 % de IVA.

En la facturación se debe tener en cuenta:

- Se aplica un descuento del 50% sobre los servicios de mano de obra cuando el valor total de los repuestos sea superior a $3.000.000.

- El valor de mano de obra de cada uno de los tipos de servicio tiene un valor mínimo y máximo definido.

- No se debe emitir facturas que superen el presupuesto que estableció el CLIENTE (en caso de que él haya establecido ese límite).

En este contexto se necesita crear un CRUD (crear, consultar, actualizar, y eliminar) para la funcionalidad de clientes:

**Entregables:**

1. Definición de Módulos

-Módulo Administración: Agregar y editar información de tiendas, mecánicos y servicios (mano de obra).

-Módulo Clientes: Permite agregar y editar clientes, así como asociar y/o deshabilitar vehículos asociados al cliente.

-Módulo inventarios: Ingreso y administración de inventarios de repuestos.

-Módulo Mantenimientos: Agendamiento e información sobre mantenimientos.

-Módulo Facturación: Generación de facturas.

1. Tablas a crear con columnas y tipos

(Ver archivo adjunto Prueba\_HC\_CarCenter.pdf y Prueba\_HC\_CarCenter.sql)

DROP TABLE Clientes;

/

CREATE TABLE Clientes (

id\_cliente NUMBER ,

primer\_nombre VARCHAR2(255),

segundo\_nombre VARCHAR2(255),

primer\_apellido VARCHAR2(255),

segundo\_apellido VARCHAR2(255),

tipo\_de\_documento VARCHAR2(255),

documento VARCHAR2(255),

celular VARCHAR2(255),

direccion VARCHAR2(255),

correo\_electronico VARCHAR2(255),

valor\_maximo\_factura NUMBER,

CONSTRAINT Clientes\_pk PRIMARY KEY (id\_cliente)

)

/

DROP TABLE Vehiculos;

/

CREATE TABLE Vehiculos (

id\_vehiculo NUMBER ,

placa VARCHAR2(255),

marca VARCHAR2(255),

modelo VARCHAR2(255),

cilindraje NUMBER,

CONSTRAINT Vehiculos\_pk PRIMARY KEY (id\_vehiculo)

)

/

DROP TABLE Clientes\_Vehiculos;

/

CREATE TABLE Clientes\_Vehiculos (

id\_cliente\_vehiculo NUMBER ,

id\_cliente NUMBER,

id\_vehiculo NUMBER,

estado VARCHAR2(255),

CONSTRAINT Clientes\_Vehiculos\_pk PRIMARY KEY (id\_cliente\_vehiculo)

)

/

DROP TABLE Tienda;

/

CREATE TABLE Tienda (

id\_tienda NUMBER ,

nombre\_tienda VARCHAR2(255),

direccion VARCHAR2(255),

telefono VARCHAR2(255),

CONSTRAINT Tienda\_pk PRIMARY KEY (id\_tienda)

)

/

DROP TABLE Mecanico;

CREATE TABLE Mecanico (

id\_mecanico NUMBER ,

primer\_nombre VARCHAR2(255),

segundo\_nombre VARCHAR2(255),

primer\_apellido VARCHAR2(255),

segundo\_apellido VARCHAR2(255),

tipo\_de\_documento VARCHAR2(255),

documento VARCHAR2(255),

celular VARCHAR2(255),

direccion VARCHAR2(255),

correo\_electronico VARCHAR2(255),

estado VARCHAR2(255),

id\_tienda NUMBER,

CONSTRAINT Mecanico\_pk PRIMARY KEY (id\_mecanico)

)

/

DROP TABLE Repuestos;

CREATE TABLE Repuestos (

id\_repuesto NUMBER ,

repuesto VARCHAR2(255),

cantidad\_inventario NUMBER,

valor\_unitario NUMBER,

estado VARCHAR2(255),

valor\_minimo\_descuento NUMBER,

porcentaje\_descuento\_aplicable NUMBER,

CONSTRAINT Repuestos\_pk PRIMARY KEY (id\_repuesto)

)

/

DROP TABLE Servicios;

CREATE TABLE Servicios (

id\_servicio NUMBER ,

descripcion\_servicio VARCHAR2(255),

valor\_minimo\_servicio NUMBER,

valor\_maximo\_servicio NUMBER,

valor\_minimo\_descuento NUMBER,

porcentaje\_descuento\_aplicable NUMBER,

CONSTRAINT Servicios\_pk PRIMARY KEY (id\_servicio)

)

/

DROP TABLE Mantenimientos;

CREATE TABLE Mantenimientos (

id\_mantenimiento NUMBER ,

id\_cliente\_vehiculo NUMBER,

id\_tienda NUMBER,

fecha DATE,

estado VARCHAR2(255),

CONSTRAINT Mantenimientos\_pk PRIMARY KEY (id\_mantenimiento)

)

/

DROP TABLE Mantenimientos\_Repuestos;

/

CREATE TABLE Mantenimientos\_Repuestos (

id\_mantenimiento NUMBER,

id\_repuesto NUMBER,

cantidad NUMBER,

valor\_unitario NUMBER,

subtotal NUMBER,

porcentaje\_descuento\_aplicable NUMBER,

descuento\_aplicado NUMBER,

valor\_total NUMBER

)

/

DROP TABLE Mantenimientos\_Servicios;

/

CREATE TABLE Mantenimientos\_Servicios (

id\_mantenimiento NUMBER,

id\_servicio NUMBER,

id\_mecanico NUMBER,

subtotal NUMBER,

porcentaje\_descuento\_aplicable NUMBER,

descuento\_aplicado NUMBER,

valor\_total NUMBER

)

/

DROP TABLE Factura;

/

CREATE TABLE Factura (

id\_factura NUMBER ,

id\_cliente NUMBER,

fecha\_factura DATE,

subtotal NUMBER,

valor\_descuento NUMBER,

iva NUMBER,

valor\_total NUMBER,

observaciones VARCHAR2(255),

CONSTRAINT Factura\_pk PRIMARY KEY (id\_factura)

)

/

DROP TABLE Factura\_Mantenimiento;

/

CREATE TABLE Factura\_Mantenimiento (

id\_factura NUMBER,

id\_mantenimiento NUMBER

)

/

ALTER TABLE Factura\_Mantenimiento ADD FOREIGN KEY ( id\_factura ) REFERENCES Factura ( id\_factura )

/

ALTER TABLE Factura\_Mantenimiento ADD FOREIGN KEY ( id\_mantenimiento ) REFERENCES Mantenimientos ( id\_mantenimiento )

/

ALTER TABLE Clientes\_Vehiculos ADD FOREIGN KEY ( id\_cliente ) REFERENCES Clientes ( id\_cliente )

/

ALTER TABLE Mantenimientos\_Servicios ADD FOREIGN KEY ( id\_servicio ) REFERENCES Servicios ( id\_servicio )

/

ALTER TABLE Mantenimientos\_Repuestos ADD FOREIGN KEY ( id\_repuesto ) REFERENCES Repuestos ( id\_repuesto )

/

ALTER TABLE Mantenimientos\_Repuestos ADD FOREIGN KEY ( id\_mantenimiento ) REFERENCES Mantenimientos ( id\_mantenimiento )

/

ALTER TABLE Mantenimientos\_Servicios ADD FOREIGN KEY ( id\_mantenimiento ) REFERENCES Mantenimientos ( id\_mantenimiento )

/

ALTER TABLE Mantenimientos\_Servicios ADD FOREIGN KEY ( id\_mecanico ) REFERENCES Mecanico ( id\_mecanico )

/

ALTER TABLE Mantenimientos ADD FOREIGN KEY ( id\_tienda ) REFERENCES Tienda ( id\_tienda )

/

ALTER TABLE Clientes\_Vehiculos ADD FOREIGN KEY ( id\_vehiculo ) REFERENCES Vehiculos ( id\_vehiculo )

/

ALTER TABLE Factura ADD FOREIGN KEY ( id\_cliente ) REFERENCES Clientes ( id\_cliente )

/

ALTER TABLE Mantenimientos ADD FOREIGN KEY ( id\_cliente\_vehiculo ) REFERENCES Clientes\_Vehiculos ( id\_cliente\_vehiculo )

/

1. Definición de llaves Primarias y Foráneas

(Ver archivo adjunto Prueba\_HC\_CarCenter.pdf)

1. Modelo E/R

(Ver archivo adjunto Prueba\_HC\_CarCenter.pdf)

1. Consulta de Clientes que han comprado un acumulado $100.000 en los últimos 60 días

SELECT clientes.id\_cliente,

clientes.primer\_nombre || ' ' || clientes.segundo\_nombre || ' ' ||

clientes.primer\_apellido || ' ' || clientes.segundo\_apellido nombreApellidos,

tipo\_de\_documento,

documento,

to\_char(facturado.total\_acumulado, '999,999,999,999.99') valor\_facturado

FROM clientes,

(SELECT id\_cliente, total\_acumulado

FROM (SELECT id\_cliente, SUM(valor\_total) total\_acumulado

from factura

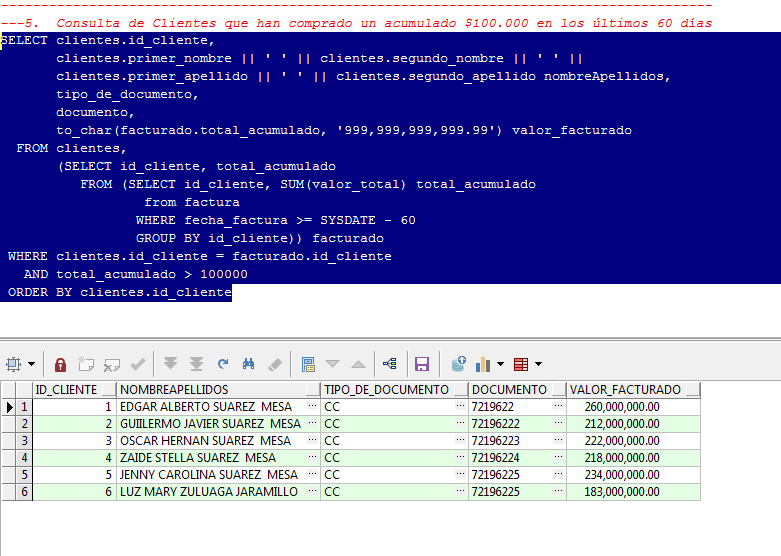
WHERE fecha\_factura >= SYSDATE - 60

GROUP BY id\_cliente)) facturado

WHERE clientes.id\_cliente = facturado.id\_cliente

AND total\_acumulado > 100000

ORDER BY clientes.id\_cliente



1. Consulta de los 100 productos más vendidos en los últimos 30 días

SELECT r.id\_repuesto, r.repuesto, ventas.masvendidos

FROM Repuestos r,

(SELECT mr.id\_repuesto, SUM(mr.cantidad) masvendidos

FROM Mantenimientos\_Repuestos mr

WHERE mr.id\_mantenimiento IN

(SELECT id\_mantenimiento

FROM Mantenimientos m

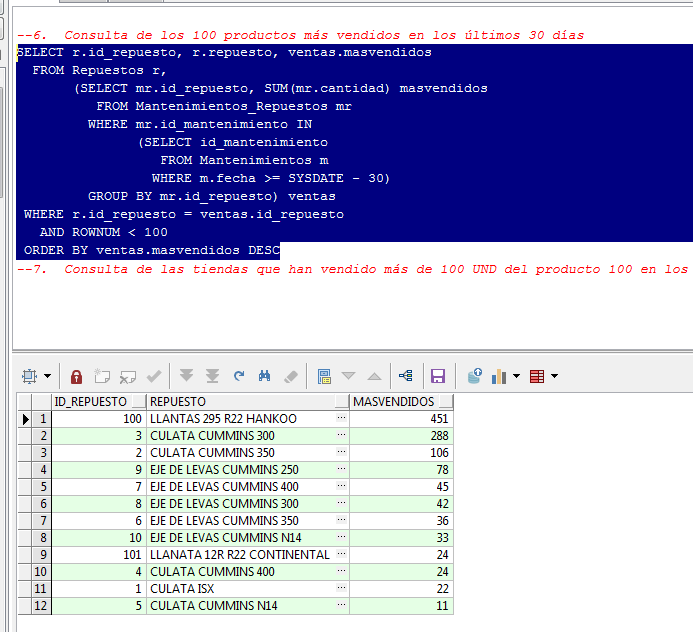
WHERE m.fecha >= SYSDATE - 30)

GROUP BY mr.id\_repuesto) ventas

WHERE r.id\_repuesto = ventas.id\_repuesto

AND ROWNUM < 100

ORDER BY ventas.masvendidos DESC



1. Consulta de las tiendas que han vendido más de 100 UND del producto 100 en los últimos 60 días.

SELECT m.id\_mantenimiento,

m.id\_tienda,

(SELECT t.nombre\_tienda

FROM tienda t

WHERE t.id\_tienda = m.id\_tienda

AND ROWNUM<2) nombretienda,

producto.id\_repuesto,

(

SELECT r.repuesto

FROM Repuestos r

WHERE r.id\_repuesto=producto.id\_repuesto

AND ROWNUM<2

) Nombrerepuesto,

producto.cantidad

FROM Mantenimientos m,

(SELECT mr.id\_mantenimiento, mr.id\_repuesto, mr.cantidad

FROM Mantenimientos\_Repuestos mr

WHERE mr.id\_mantenimiento IN

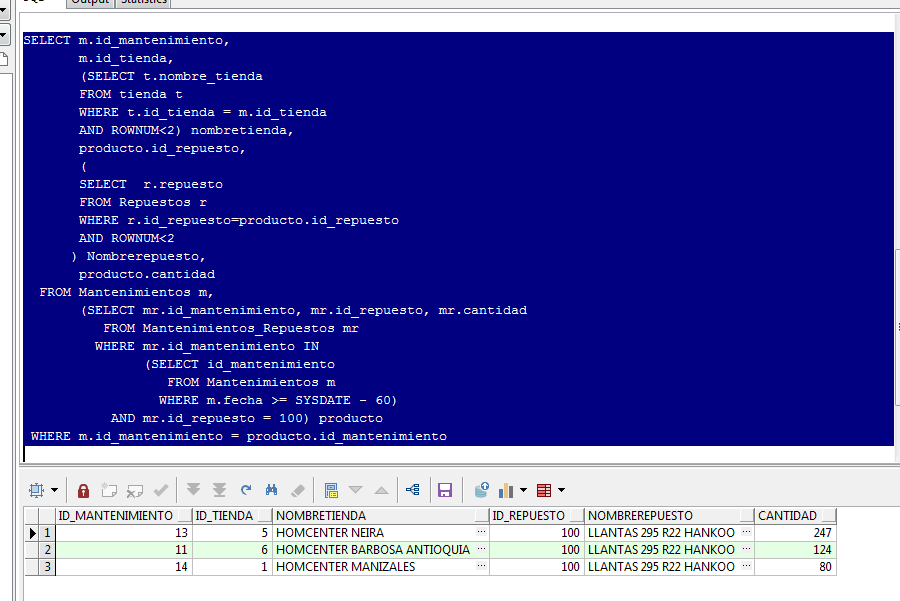
(SELECT id\_mantenimiento

FROM Mantenimientos m

WHERE m.fecha >= SYSDATE - 60)

AND mr.id\_repuesto = 100) producto

WHERE m.id\_mantenimiento = producto.id\_mantenimiento



1. Consulta de todos los clientes que han tenido más de un(1) mantenimento en los últimos 30 días.

SELECT clv.id\_cliente\_vehiculo,

clv.id\_cliente,

(

SELECT

cli.primer\_nombre || ' ' || cli.segundo\_nombre || ' ' ||

cli.primer\_apellido || ' ' || cli.segundo\_apellido|| ' ' ||tipo\_de\_documento || ' ' ||documento

FROM Clientes cli

WHERE clv.id\_cliente=cli.id\_cliente

AND ROWNUM<2

) DescribeCliente,

clv.id\_vehiculo,

(

SELECT 'Placa: ' ||v.placa|| ' ' ||v.marca|| ' Modelo: ' ||v.modelo|| ' Cilindraje: ' ||v.cilindraje

FROM Vehiculos v

WHERE v.id\_vehiculo=clv.id\_vehiculo

AND ROWNUM<2

) DESCRIBEVEHICULO

FROM Clientes\_Vehiculos clv

WHERE clv.id\_cliente\_vehiculo

IN (

SELECT id\_cliente\_vehiculo

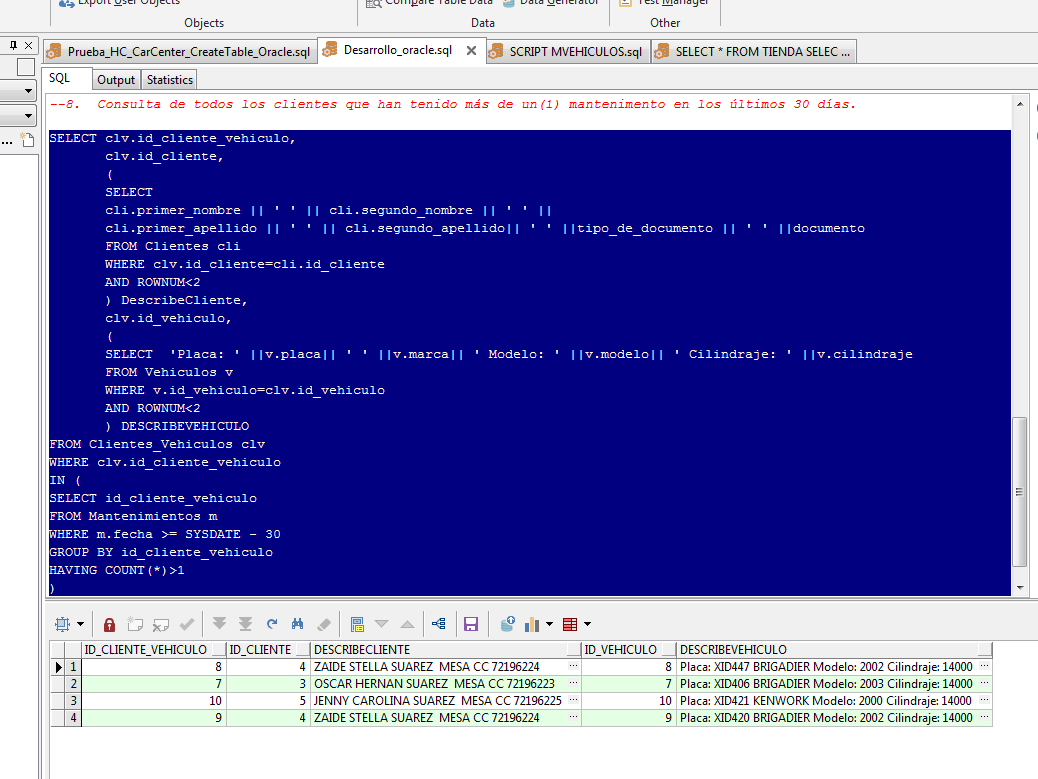
FROM Mantenimientos m

WHERE m.fecha >= SYSDATE - 30

GROUP BY id\_cliente\_vehiculo

HAVING COUNT(\*)>1

)



1. Procedimiento que reste la cantidad de productos del inventario de las tiendas cada que se presente una venta.

CREATE OR REPLACE PROCEDURE egreso\_inventario(p\_id\_repuesto repuestos.id\_repuesto%TYPE,

p\_cantidad\_venta NUMBER) IS

v\_cantidad\_inventario repuestos.cantidad\_inventario%type;

v\_error number;

BEGIN

*--- Procedimiento que reste la cantidad de productos del inventario*

*--- de las tiendas cada que se presente una venta.*

*--Valor existencias en el inventario*

BEGIN

SELECT cantidad\_inventario

INTO v\_cantidad\_inventario

FROM repuestos

WHERE id\_repuesto = p\_id\_repuesto;

EXCEPTION

WHEN OTHERS THEN

v\_error := SQLCODE;

END;

*--Egreso Inventario*

BEGIN

UPDATE repuestos

SET cantidad\_inventario = v\_cantidad\_inventario - p\_cantidad\_venta

WHERE id\_repuesto = p\_id\_repuesto;

COMMIT;

EXCEPTION

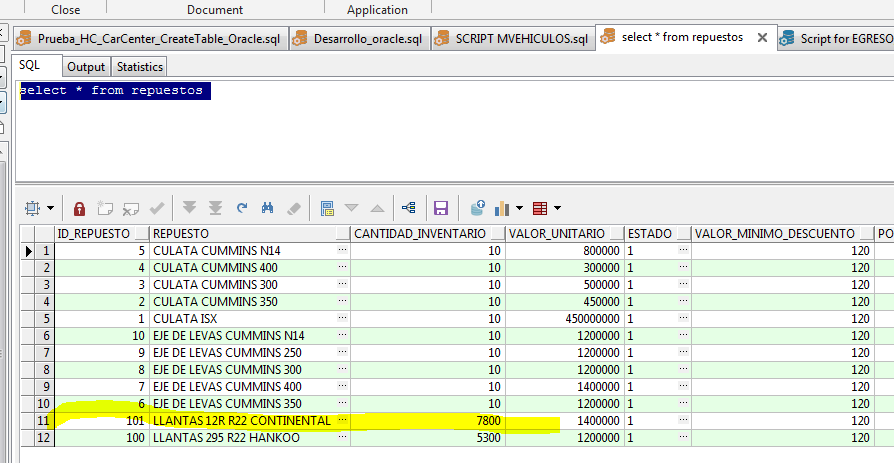
WHEN OTHERS THEN

v\_error := SQLCODE;

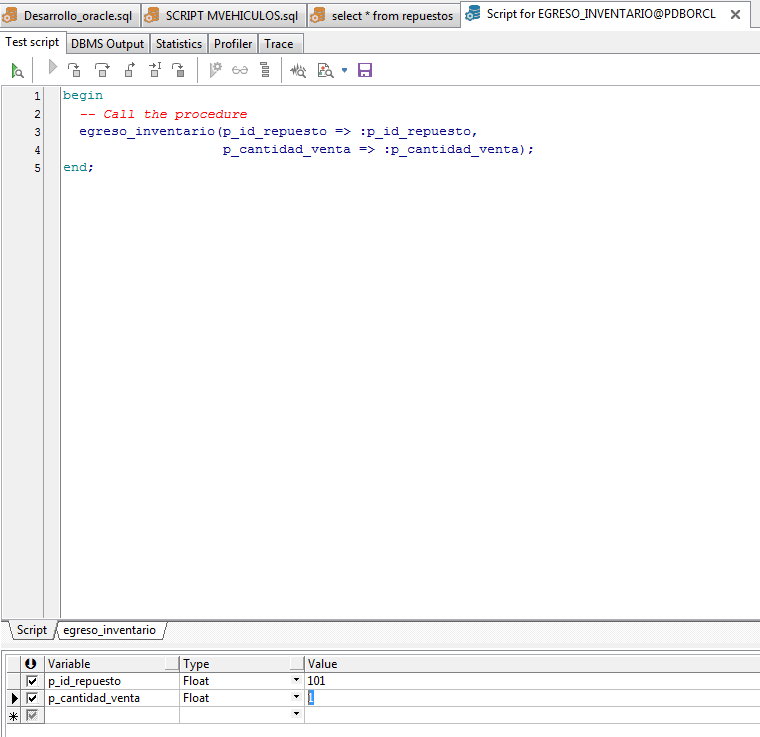
END;

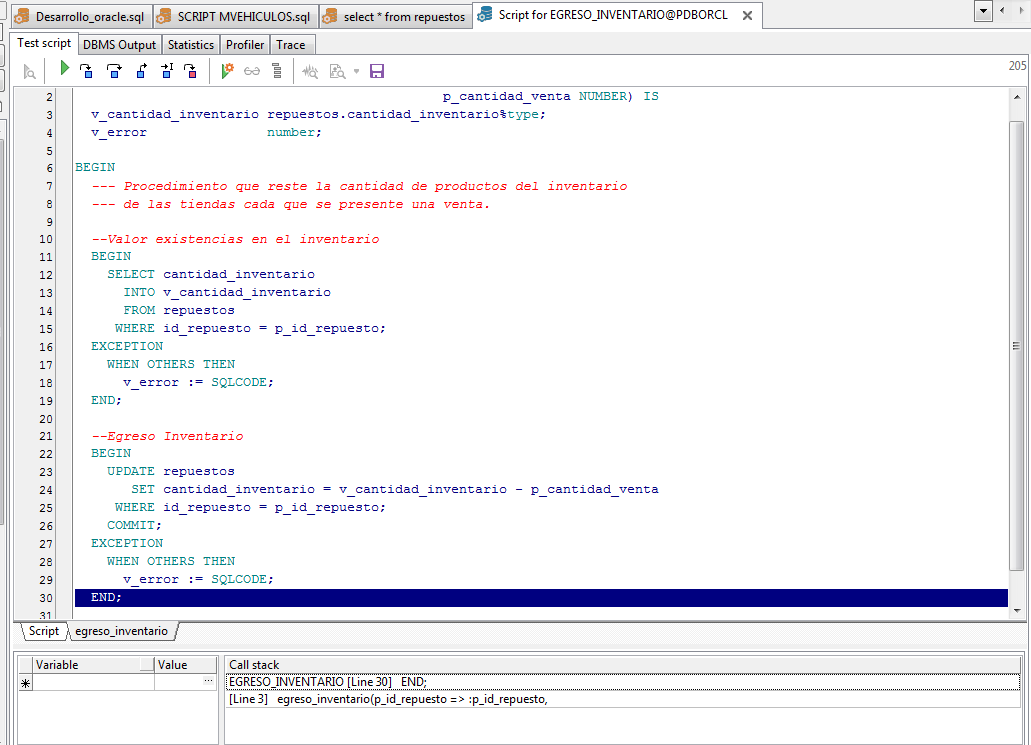
END egreso\_inventario;

**DATOS INVENTARIO ACTUAL**

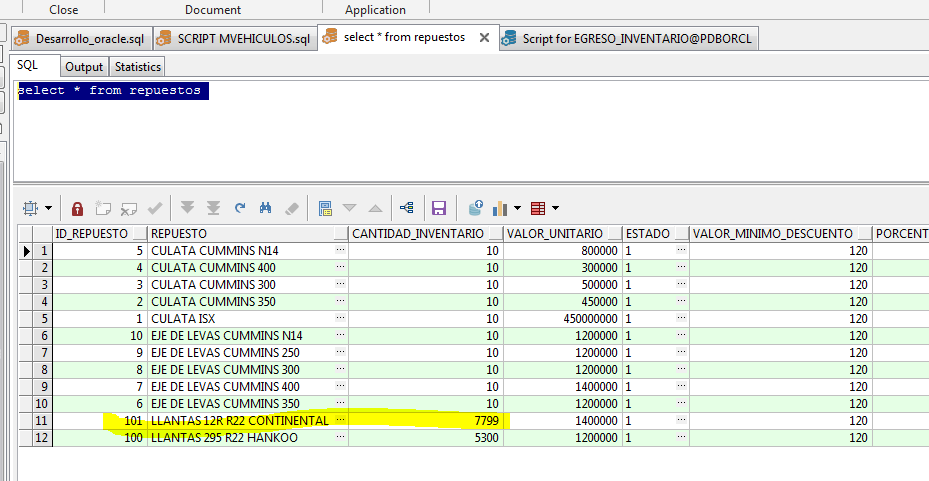


**EJECUCION PROCEDURE**





**SE ACTUALIZO EL INVENTARIO**



**Convenciones**

* Lenguaje: Oracle PL/SQL ó Transact-SQL

**Por favor tener en cuenta:**

* Subir solución (Back-front-scripts) en repositorio Git y adjuntar URL

https://github.com/Cabobe/PruebaHC

Conceptos

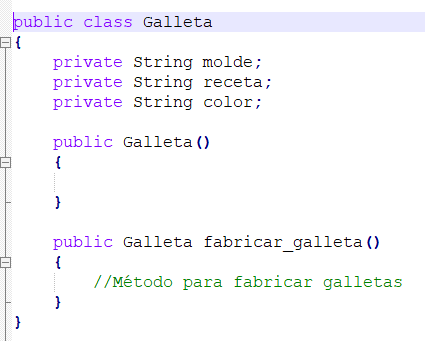
1. ¿Cuál es Componente básico para la ejecución de un programa en .Net?
2. Eclipse
3. Framework
4. Compilador
5. Ninguna de las anteriores
6. Describa de manera sintética:
7. Que es una clase

Una clase es la abstracción de una entidad del mundo real, en la abstracción se incluyen las propiedades y métodos asociados a la entidad que se quiere representar.

1. Que es un objeto

Es el tipo de dato resultado de la implementación de una clase, una clase es la definición de una entidad en el desarrollo de una aplicación, mientras que el objeto surge de la definición de la clase, se crea en momento de ejecución de la aplicación y puede cambiar sus atributos durante la ejecución.

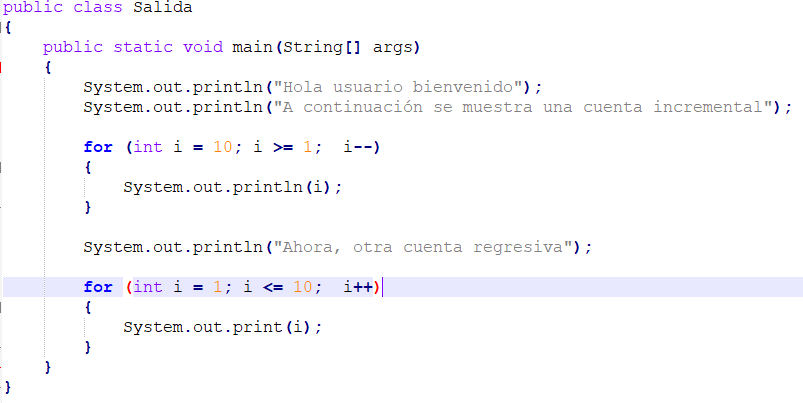
1. En el siguiente ejemplo de código:



el método Galleta es:

* 1. El método principal de la clase
  2. Un singleton de la clase
  3. El constructor de la clase
  4. Ninguno de los anteriores

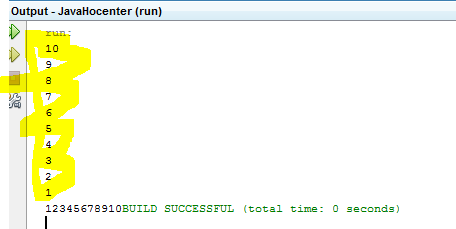
1. en el siguiente ejemplo de código:



la salida del primer loop es:

1. "a,b,c,d,e,f,g,h,i,j"
2. "2,4,6,8,10,12,14,16,18,20"
3. "1,2,3,4,5,6,7,8,9,10"
4. "10,9,8,7,6,5,4,3,2,1"
5. 5. ninguna de las anteriores

Pero sale hacia abajo



1. ¿Qué es un API?

Application Programming Interface. Es un módulo de software que se comunica o interactua con otro. Una API esta formado por paquetes son un conjunto de funciones, interfaces, protocolos y comandos que permiten la integración entre aplicaciones y sistemas operativos

1. ¿Qué es un servicio Rest y Cuál es su diferencia con un SOAP?

REST: Es la comunicación de un Cliente a un servidor a través de un protocolo HTTP

El protocolo de comunicación se hace a través de mensajes. Si el mensaje es enviado en **formato JSON es REST. S**i el mensaje enviado en **formato XML es SOAP. El servicio lo recepcionara y devolverá una respuesta.**

REST se considera una alternativa más rápida en los escenarios basados en la Web