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

**Nombre**: Mauricio Sandoval Parra

**Fecha**: 23 de marzo de 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

Se deben tener los siguientes módulos:

* **Módulo de inscripción de usuario**

En este módulo el cliente debe inscribir sus datos personales.

* **Módulo de inscripción de vehículo**

En este módulo el cliente una vez inscrito como usuario en la aplicación puede inscribir su vehículo o en caso de tener más de un vehículo inscribir todos los que tenga.

* **Módulo de inscripción de servicios**

En este módulo un empleado puede realizar la inscripción, modificación y borrado de un servicio.

* **Módulo de inscripción de repuestos**

En este módulo un empleado puede realizar la inscripción, modificación y borrado de un repuesto, además de cargar existencias al inventario cuando llegue nuevo producto al CAR CENTER.

* **Módulo de mantenimientos**

En este módulo los operarios cargaran a un mantenimiento los repuestos y servicios que se le vayan aplicando al vehículo.

* **Módulo de facturación**

En este módulo un empleado podrá generar la factura relacionando a la misma el cliente, el empleado tipo mecánico que lo atendió y el mantenimiento se le realizó a su vez este último ya tiene el contenido de los repuestos y servicios aplicados al vehículo.

1. Tablas a crear con columnas y tipos
2. Definición de llaves Primarias y Foráneas

CREATE TABLE tiendas (

id\_tienda NUMBER(12) NOT NULL,

ciudad VARCHAR2(50) NOT NULL,

nombre VARCHAR2(50) NOT NULL,

direccion VARCHAR2(100) NOT NULL,

telefono VARCHAR2(30) NOT NULL

);

ALTER TABLE tiendas ADD CONSTRAINT tiendas\_pk PRIMARY KEY ( id\_tienda );

/

CREATE TABLE repuestos (

id\_repuesto NUMBER(12) NOT NULL,

id\_tienda NUMBER(12) NOT NULL,

nombre VARCHAR2(100) NOT NULL,

marca VARCHAR2(30) NOT NULL,

precio NUMBER(8,2) NOT NULL,

cantidad NUMBER(3) NOT NULL,

descuento NUMBER(2,2)

);

ALTER TABLE repuestos ADD CONSTRAINT repuestos\_pk PRIMARY KEY ( id\_repuesto );

ALTER TABLE repuestos ADD CONSTRAINT fk\_repuestos\_tiendas FOREIGN KEY ( id\_tienda ) REFERENCES tiendas ( id\_tienda );

/

CREATE TABLE servicios (

id\_servicio NUMBER(12) NOT NULL,

tipo\_servicio VARCHAR2(1) NOT NULL,

servicio VARCHAR2(100) NOT NULL,

precio\_mano\_obra NUMBER(7,2) NOT NULL,

descuento NUMBER(2,2),

valor\_minimo NUMBER(7,2) NOT NULL,

valor\_maximo NUMBER(9,2) NOT NULL

);

ALTER TABLE servicios ADD CONSTRAINT servicios\_pk PRIMARY KEY ( id\_servicio );

/

CREATE TABLE tiendas (

id\_tienda NUMBER(12) NOT NULL,

ciudad VARCHAR2(50) NOT NULL,

nombre VARCHAR2(50) NOT NULL,

direccion VARCHAR2(100) NOT NULL,

telefono VARCHAR2(30) NOT NULL

);

ALTER TABLE tiendas ADD CONSTRAINT tiendas\_pk PRIMARY KEY ( id\_tienda );

/

CREATE TABLE mantenimiento (

id\_mantenimiento NUMBER(12) NOT NULL,

placa VARCHAR2(6) NOT NULL,

id\_tienda NUMBER(12) NOT NULL,

estado VARCHAR2(1)

);

ALTER TABLE mantenimiento ADD CHECK ( estado IN ('P','T'));

ALTER TABLE mantenimiento ADD CONSTRAINT mantenimiento\_pk PRIMARY KEY ( id\_mantenimiento );

ALTER TABLE mantenimiento ADD CONSTRAINT fk\_manteni\_vehi FOREIGN KEY ( placa )REFERENCES vehiculo ( placa );

ALTER TABLE mantenimiento ADD CONSTRAINT fk\_mant\_tienda FOREIGN KEY ( id\_tienda ) REFERENCES tiendas ( id\_tienda );

/

CREATE TABLE mantenimi\_repuest (

id\_repuesto NUMBER(12) NOT NULL,

id\_mantenimiento NUMBER(12) NOT NULL,

cantidad NUMBER(3) NOT NULL

);

ALTER TABLE mantenimi\_repuest ADD CONSTRAINT fk\_mant\_repu FOREIGN KEY ( id\_mantenimiento ) REFERENCES mantenimiento ( id\_mantenimiento );

ALTER TABLE mantenimi\_repuest ADD CONSTRAINT fk\_repu\_mant FOREIGN KEY ( id\_repuesto ) REFERENCES repuestos ( id\_repuesto );

/

CREATE TABLE mantemini\_servicios (

id\_servicio NUMBER(12) NOT NULL,

id\_mantenimiento NUMBER(12) NOT NULL

);

ALTER TABLE mantemini\_servicios ADD CONSTRAINT fk\_mante\_servic FOREIGN KEY ( id\_mantenimiento ) REFERENCES mantenimiento ( id\_mantenimiento );

ALTER TABLE mantemini\_servicios ADD CONSTRAINT fk\_servic\_mante FOREIGN KEY ( id\_servicio ) REFERENCES servicios ( id\_servicio );

/

CREATE TABLE persona (

id\_persona NUMBER(12) NOT NULL,

primer\_nombre VARCHAR2(20) NOT NULL,

segundo\_nombre VARCHAR2(20),

primer\_apellido VARCHAR2(20) NOT NULL,

segundo\_apellido VARCHAR2(20),

tipo\_documento VARCHAR2(2) NOT NULL,

numero\_documento NUMBER(15) NOT NULL,

celular NUMBER(10),

direccion VARCHAR2(100),

correo\_electronico VARCHAR2(50)

);

ALTER TABLE persona ADD CONSTRAINT persona\_pk PRIMARY KEY ( id\_persona );

/

CREATE TABLE cliente (

id\_cliente NUMBER(12) NOT NULL,

id\_persona NUMBER(12) NOT NULL,

valor\_limite NUMBER(9,2)

);

ALTER TABLE cliente ADD CONSTRAINT cliente\_pk PRIMARY KEY ( id\_cliente );

ALTER TABLE cliente ADD CONSTRAINT fk\_cli\_pers FOREIGN KEY ( id\_persona ) REFERENCES persona ( id\_persona );

/

CREATE TABLE empleado (

id\_empleado NUMBER(12) NOT NULL,

id\_persona NUMBER(12) NOT NULL,

tipo\_empleado VARCHAR2(10) NOT NULL,

estado VARCHAR2(1)

);

ALTER TABLE empleado ADD CONSTRAINT empleados\_pk PRIMARY KEY ( id\_empleado );

ALTER TABLE empleado ADD CONSTRAINT fk\_emp\_pers FOREIGN KEY ( id\_persona ) REFERENCES persona ( id\_persona );

/

CREATE TABLE fotos (

id\_foto NUMBER(12) NOT NULL,

ruta\_foto VARCHAR2(200) NOT NULL

);

ALTER TABLE fotos ADD CONSTRAINT fotos\_pk PRIMARY KEY ( id\_foto );

/

CREATE TABLE vehiculo (

placa VARCHAR2(6) NOT NULL,

id\_cliente NUMBER(12) NOT NULL,

marca VARCHAR2(20) NOT NULL,

modelo VARCHAR2(20) NOT NULL,

ano NUMBER(4) NOT NULL,

color VARCHAR2(10) NOT NULL,

tipo VARCHAR2(20),

cant\_puertas NUMBER(1) NOT NULL

);

ALTER TABLE vehiculo ADD CONSTRAINT vehiculo\_pk PRIMARY KEY ( placa );

ALTER TABLE vehiculo ADD CONSTRAINT fk\_vehiculo\_cliente FOREIGN KEY ( id\_cliente ) REFERENCES cliente ( id\_cliente );

/

CREATE TABLE vehiculo\_fotos (

placa VARCHAR2(6) NOT NULL,

id\_foto NUMBER(12) NOT NULL

);

ALTER TABLE vehiculo\_fotos ADD CONSTRAINT fk\_fot\_vehi FOREIGN KEY ( id\_foto ) REFERENCES fotos ( id\_foto );

ALTER TABLE vehiculo\_fotos ADD CONSTRAINT fk\_vehi\_fotos FOREIGN KEY ( placa ) REFERENCES vehiculo ( placa );

/

CREATE TABLE factura (

numero\_factura NUMBER(12) NOT NULL,

fecha DATE NOT NULL,

id\_cliente NUMBER(12) NOT NULL,

id\_empleado NUMBER(12) NOT NULL,

id\_mantenimiento NUMBER(12) NOT NULL,

subtotal NUMBER(9,2),

impuesto NUMBER(9,2),

descuento NUMBER(8,2),

total NUMBER(9,2)

);

ALTER TABLE factura ADD CONSTRAINT factura\_pk PRIMARY KEY ( numero\_factura );

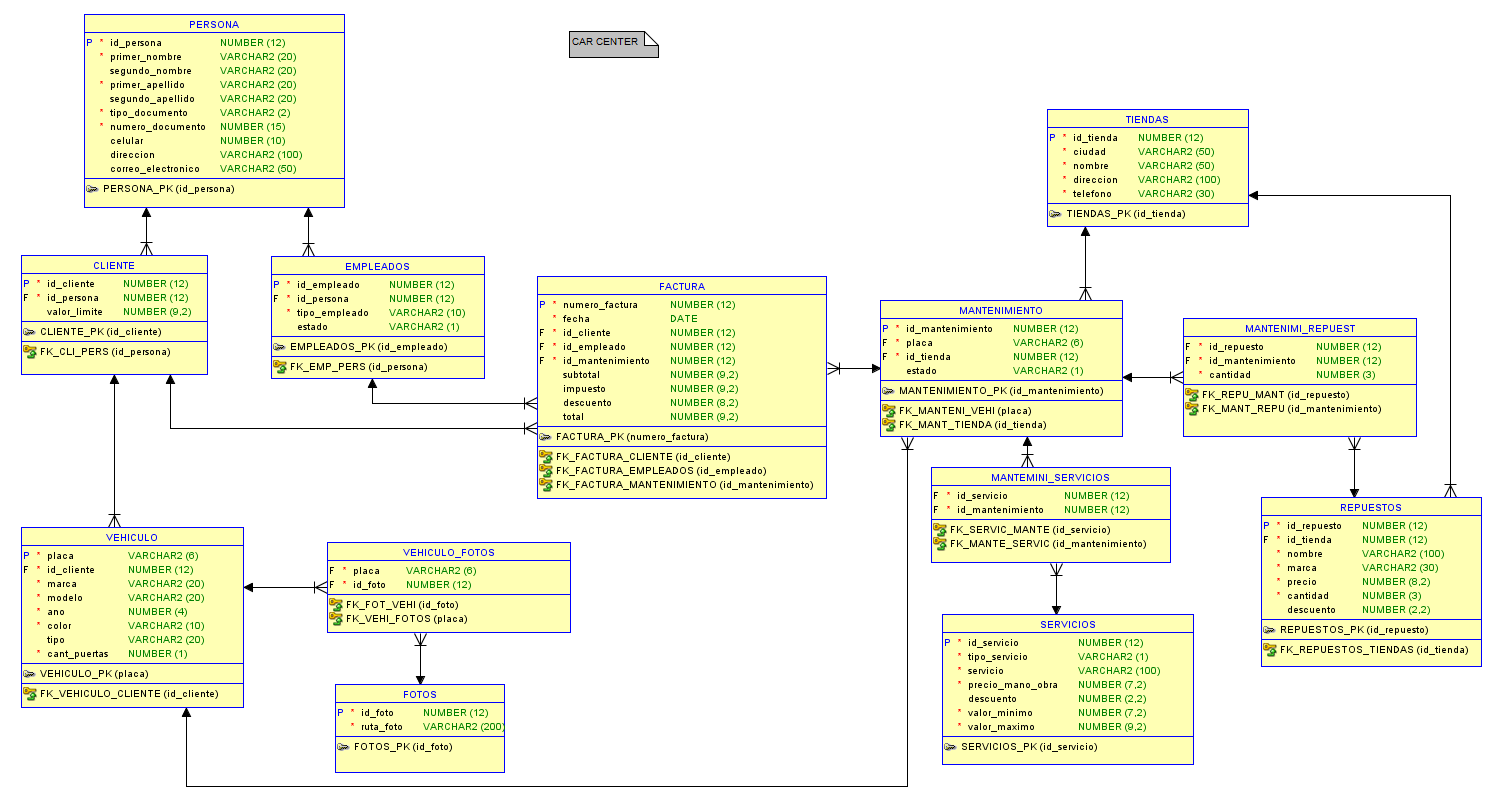
ALTER TABLE factura ADD CONSTRAINT fk\_factura\_cliente FOREIGN KEY ( id\_cliente ) REFERENCES cliente ( id\_cliente );

ALTER TABLE factura ADD CONSTRAINT fk\_factura\_empleados FOREIGN KEY ( id\_empleado ) REFERENCES empleado ( id\_empleado );

ALTER TABLE factura ADD CONSTRAINT fk\_factura\_mantenimiento FOREIGN KEY ( id\_mantenimiento ) REFERENCES mantenimiento ( id\_mantenimiento );

/

1. Modelo E/R



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

SELECT P.tipo\_documento,

P.numero\_documento,

P.primer\_nombre||' '||P.segundo\_nombre||' '||P.primer\_apellido||' '||P.segundo\_apellido AS nombre,

p.celular,

P.direccion,

P.correo\_electronico

FROM persona p,

cliente c

WHERE p.id\_persona = c.id\_persona

AND c.id\_cliente IN (SELECT id\_cliente

FROM (SELECT c.id\_cliente,

SUM(f.total)

FROM cliente c,

factura f

WHERE c.ID\_CLIENTE = f.ID\_CLIENTE

AND TRUNC(f.fecha) > sysdate-60

GROUP BY c.id\_cliente

HAVING SUM(total) > 100000))

;

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

SELECT r.\*

FROM repuestos r

WHERE r.id\_repuesto IN (SELECT id\_repuesto

FROM (SELECT mr.id\_repuesto,

SUM(cantidad)

FROM mantenimiento m,

mantenimi\_repuest mr,

factura f

WHERE m.id\_mantenimiento = mr.id\_mantenimiento

AND m.id\_mantenimiento = f.id\_mantenimiento

AND TRUNC(f.fecha) > SYSDATE-30

GROUP BY mr.id\_repuesto

ORDER BY 2 DESC ))

AND ROWNUM < 101

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

SELECT s.\*

FROM tiendas s

WHERE s.id\_tienda IN (SELECT id\_tienda

FROM (SELECT t.id\_tienda,

SUM(cantidad)

FROM mantenimiento m,

mantenimi\_repuest mr,

factura f,

tiendas t

WHERE m.id\_mantenimiento = mr.id\_mantenimiento

AND m.id\_mantenimiento = f.id\_mantenimiento

AND m.id\_tienda = t.id\_tienda

AND mr.id\_repuesto = 100

AND TRUNC(f.fecha) > SYSDATE-60

GROUP BY t.id\_tienda

HAVING SUM(cantidad) > 100));

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

SELECT P.tipo\_documento,

P.numero\_documento,

P.primer\_nombre||' '||P.segundo\_nombre||' '||P.primer\_apellido||' '||P.segundo\_apellido AS nombre,

p.celular,

P.direccion,

P.correo\_electronico

FROM persona p,

cliente c

WHERE p.id\_persona = c.id\_persona

AND c.id\_cliente IN (SELECT id\_cliente

FROM (SELECT cli.id\_cliente,

m.id\_mantenimiento,

COUNT(\*)

FROM factura f,

mantenimiento m,

cliente cli

WHERE f.id\_mantenimiento = m.id\_mantenimiento

AND f.id\_cliente = cli.id\_cliente

AND f.fecha > SYSDATE-30

GROUP BY cli.id\_cliente,

m.id\_mantenimiento

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 p\_resta\_cant\_invent( p\_id\_tienda tiendas.id\_tienda%TYPE,

p\_id\_repuesto repuestos.id\_repuesto%TYPE,

p\_cantidad NUMBER) IS

BEGIN

UPDATE repuestos

SET cantidad = cantidad - p\_cantidad

WHERE ID\_REPUESTO = p\_id\_repuesto

AND ID\_TIENDA = p\_id\_tienda;

COMMIT;

EXCEPTION

WHEN OTHERS THEN

raise\_application\_error(-20055,'Fallo en p\_resta\_cant\_invent. ');

END p\_resta\_cant\_invent;

**Convenciones**

* Lenguaje : Oracle PL/SQL ó Transact-SQL

**Por favor tener en cuenta:**

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

**Conceptos**

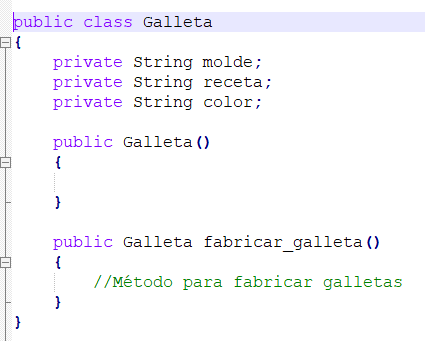
1. ¿Cuál es Componente básico para la ejecución de un programa en .Net?
2. Eclipse
3. Framework
4. **Compilador**
5. Ninguna de las anteriores
6. Describa de manera sintética:
7. Que es una clase

La clase es donde definimos las particularidades o características de un objeto, allí se definen y/o desarrollan los atributos y métodos de un objeto.

1. Que es un objeto

Un objeto es una representación de lago del mundo real, además de que se le llama objeto a una clase cuando dicha clase se instancia.

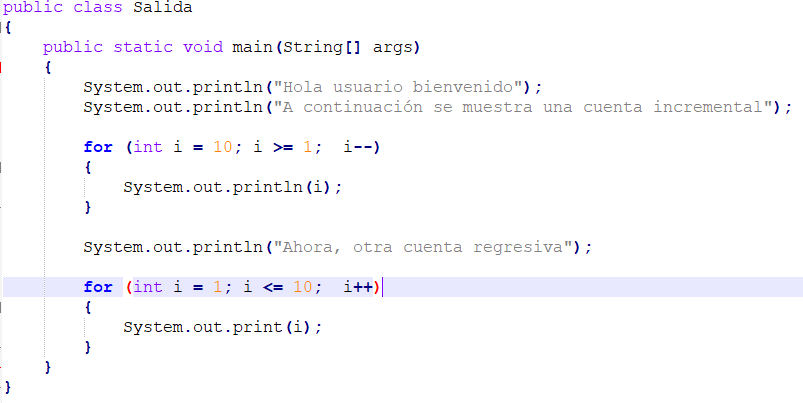
1. En el siguiente ejemplo de código:



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
6. ¿Qué es un API?

Un API es un método para integrar aplicaciones de software, en donde se debe presentar una solicitud y una respuesta dadas por un protocolo definido previamente por ambas partes cliente y servidor.

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

Es una interfaz de comunicación entre dos sistemas dicha comunicación se hace por medio de un protocolo HTTP.

No estoy tan seguro que sea la única diferencia, pero me parece que el SOAP no se comunica por medio de protocolo de HTTP y la comunicación es por medio de XMLs, mientras que el REST es por medio de JSON.