

Pontificia Universidad Católica Del Perú
Facultad de Ciencias e Ingeniería
Bases de Datos [INF246]



**MODELAMIENTO Y ANÁLISIS DE LA BASE DE
DATOS DE LA VETERINARIA “ABC”**

Flavio Roberto Pujay Angeles	20217327
Giss Araceli Castellón Saavedra	20201984
Luis Ángel Sánchez Valle	20216164
Kevin Martín Díaz Mendieta	20206113
Joseph Celso Aparicio Bohorquez	20192393

Profesor

César Augusto Aguilera Serpa

Jefe de práctica

Ronald Álvarez

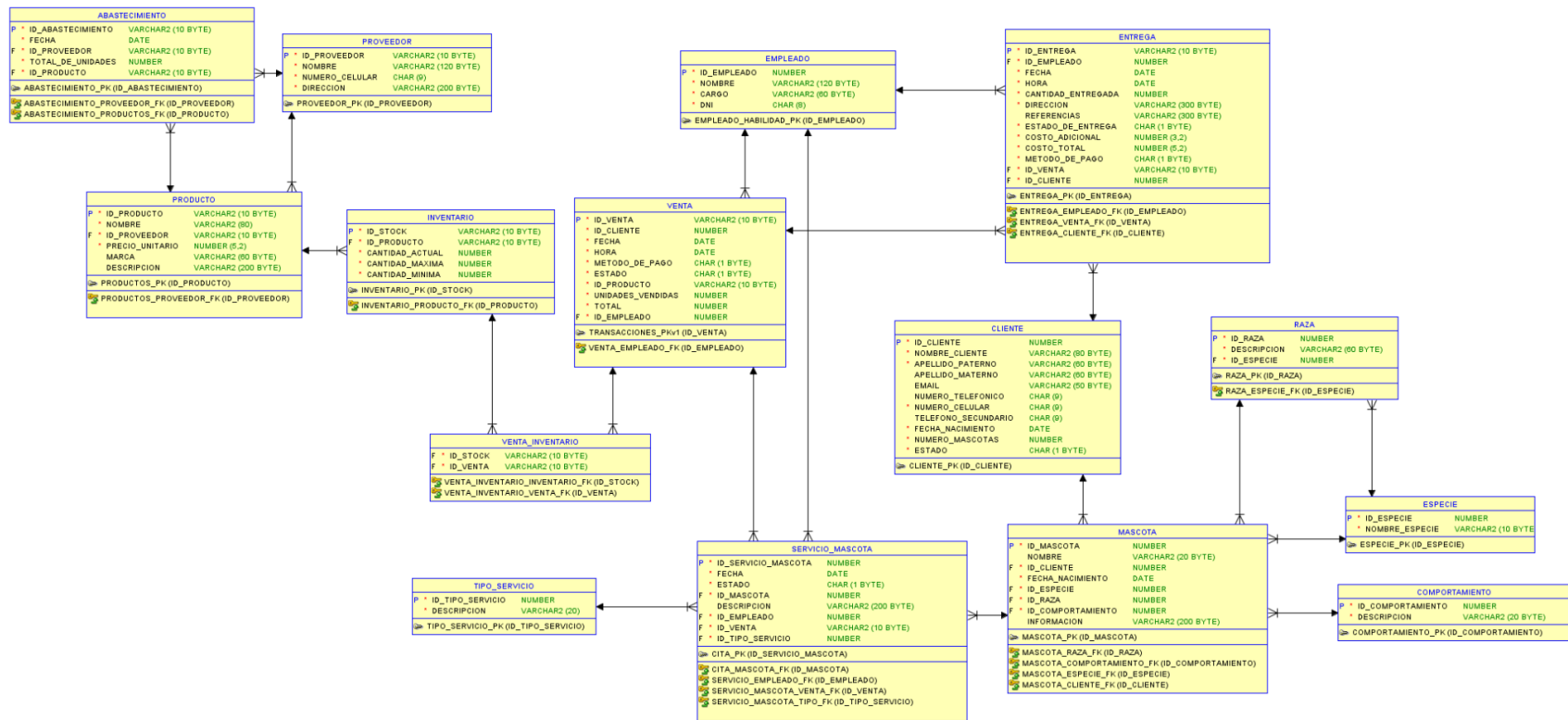
Lima – Perú

Diciembre del 2023

TABLA DE CONTENIDO

1. MODELO RELACIONAL

El modelo a continuación ilustra la estructura de la base de datos de una clínica veterinaria, destacando aspectos clave como el manejo del inventario, la prestación de servicios para las mascotas y la opción de entrega a domicilio de productos relacionados con el cuidado de los animales de compañía.



2. DICCIONARIO DE DATOS

Tabla N°1: Atributos de la tabla CLIENTE

NOMBRE	TIPO DE ATRIBUTO	DESCRIPCIÓN	TIPO DE LLAVE
ID_CLIENTE	NUMERIC	Identificador del cliente.	PK
NOMBRE_CLIENTE	VARCHAR2(80)	Nombre del cliente.	
APELLIDO_PATERNO	VARCHAR2(60)	Apellido paterno del cliente.	
APELLIDO_MATERNO	VARCHAR2(60)	Apellido materno del cliente.	
EMAIL	VARCHAR2(50)	Correo electrónico del cliente.	
NUMERO_TELEFONICO	CHAR (9)	Número telefónico del cliente.	
NUMERO_CELULAR	CHAR (9)	Número de celular del cliente.	
TELEFONO_SECUNDARIO	CHAR (9)	Número secundario del celular del cliente.	
FECHA_NACIMIENTO	DATE	Fecha de nacimiento del cliente.	
NUMERO_MASCOTAS	NUMERIC	Cantidad de mascotas del cliente.	
ESTADO	CHAR (1)	Estado del cliente: A: Activo I: Inactivo	

Tabla N°2: Atributos de la tabla MASCOTA

NOMBRE	TIPO DE ATRIBUTO	DESCRIPCIÓN	TIPO DE LLAVE
ID_MASCOTA	NUMERIC	Identificador de la mascota.	PK
NOMBRE	VARCHAR2(20)	Nombre de la mascota.	
ID_CLIENTE	NUMERIC	Identificador del dueño de la mascota.	FK
FECHA_NACIMIENTO	DATE	Fecha de nacimiento de la mascota.	
ID_ESPECIE	NUMERIC	Identificador de la especie de la mascota.	FK
ID_RAZA	NUMERIC	Identificador de la raza de la mascota.	FK
ID_COMPORTAMIENTO	NUMERIC	Identificador del comportamiento de la mascota.	FK
INFORMACION	VARCHAR2(200)	Información adicional de la mascota.	

Tabla N°3: Atributos de la tabla RAZA

NOMBRE	TIPO DE ATRIBUTO	DESCRIPCIÓN	TIPO DE LLAVE
ID_RAZA	NUMERIC	Identificador de la raza de la mascota.	PK
DESCRIPCION	VARCHAR2(60)	Descripción de la raza de la mascota.	
ID_ESPECIE	NUMERIC	Identificador de la especie de la mascota.	FK

Tabla N°4: Atributos de la tabla ESPECIE

NOMBRE	TIPO DE ATRIBUTO	DESCRIPCIÓN	TIPO DE LLAVE
ID_ESPECIE	NUMERIC	Identificador de la especie de la mascota.	PK
NOMBRE_ESPECIE	VARCHAR2(10)	Nombre de la especie de la mascota.	

Tabla N°5: Atributos de la tabla COMPORTAMIENTO

NOMBRE	TIPO DE ATRIBUTO	DESCRIPCIÓN	TIPO DE LLAVE
ID_COMPORTAMIENTO	NUMERIC	Identificador del comportamiento de la mascota.	PK
DESCRIPCION	VARCHAR2(20)	Descripción del comportamiento de la mascota: 1: Amigable 2: Agresivo 3: Pasivo	

Tabla N°6: Atributos de la tabla SERVICIO_MASCOTA

NOMBRE	TIPO DE ATRIBUTO	DESCRIPCIÓN	TIPO DE LLAVE
ID_SERVICIO_MASCOTA	NUMERIC	Identificador del cliente.	PK
FECHA	DATE	Fecha del servicio.	
ESTADO	CHAR (1)	Estado del servicio: A: Activo I: Inactivo	
ID_MASCOTA	NUMERIC	Identificador de la mascota.	
DESCRIPCION	VARCHAR2(200)	Descripción del servicio.	
ID_EMPLEADO	NUMERIC	Identificador del empleado.	
ID_VENTA	VARCHAR2(10)	Identificador de la venta.	
ID_TIPO_SERVICIO	NUMERIC	Identificador del tipo de servicio.	FK

Tabla N°7: Atributos de la tabla ENTREGA

NOMBRE	TIPO DE ATRIBUTO	DESCRIPCIÓN	TIPO DE LLAVE
ID_ENTREGA	VARCHAR2(10)	Identificador de la entrega.	PK
ID_EMPLEADO	NUMERIC	Identificador del empleado.	FK
FECHA	DATE	Fecha de entrega.	
HORA	DATE	Hora de entrega.	
CANTIDAD_ENTREGADA	NUMERIC	Cantidad entregada en la entrega.	
DIRECCION	VARCHAR2(300)	Dirección de la entrega.	
REFERENCIAS	VARCHAR2(300)	Referencias de la dirección.	
ESTADO_DE_ENTREGA	CHAR (1)	Estado de la entrega: 1: Pendiente 2: En proceso 3: Entregado 4: Cancelado	
COSTO_ADICIONAL	NUMERIC (3,2)	Costo adicional de entrega.	
COSTO_TOTAL	NUMERIC (5,2)	Costo total de la entrega.	
ID_VENTA	VARCHAR2(10)	Identificador de venta.	FK
ID_CLIENTE	NUMERIC	Identificador del cliente.	FK

Tabla N°8: Atributos de la tabla EMPLEADO

NOMBRE	TIPO DE ATRIBUTO	DESCRIPCIÓN	TIPO DE LLAVE
ID_EMPLEADO	NUMERIC	Identificador del empleado.	PK
NOMBRE	VARCHAR2(120)	Nombre del empleado.	
CARGO	VARCHAR2(60)	Cargo del empleado: Repartidor Veterinario Cajero	
DNI	CHAR (8)	Documento de identidad del empleado.	

Tabla N°9: Atributos de la tabla VENTA_INVENTARIO

NOMBRE	TIPO DE ATRIBUTO	DESCRIPCIÓN	TIPO DE LLAVE
ID_STOCK	VARCHAR2(10)	Identificador del inventario.	FK
ID_VENTA	VARCHAR2(10)	Identificador de una venta.	FK

Tabla N°10: Atributos de la tabla INVENTARIO

NOMBRE	TIPO DE ATRIBUTO	DESCRIPCIÓN	TIPO DE LLAVE
ID_STOCK	VARCHAR2(10)	Identificador de un producto del inventario.	PK
ID_PRODUCTO	VARCHAR2(10)	Identificador del producto.	FK
CANTIDAD_ACTUAL	NUMERIC	Cantidad actual del producto.	
CANTIDAD_MAXIMA	NUMERIC	Cantidad máxima del producto.	
CANTIDAD_MINIMA	NUMERIC	Cantidad mínima del producto.	

Tabla N°11: Atributos de la tabla VENTA

NOMBRE	TIPO DE ATRIBUTO	DESCRIPCIÓN	TIPO DE LLAVE
ID_VENTA	VARCHAR2(10)	Identificador de la venta.	PK
ID_CLIENTE	NUMERIC	Identificador del cliente.	
FECHA	DATE	Fecha de la venta.	
HORA	DATE	Hora de la venta.	
METODO_DE_PAGO	CHAR (1)	Método de pago: 1: Efectivo 2: Tarjeta de débito 3: Yape 4: Plin	
ESTADO	CHAR (1)	Estado de la entrega: 1: Pendiente 2: En proceso 3: Entregado 4: Cancelado	

ID_PRODUCTO	VARCHAR2(10)	Identificador del producto.	
UNIDADES_VENDIDAS	NUMERIC	Unidades vendidas.	
TOTAL	NUMERIC	Costo total.	
ID_EMPLEADO	NUMERIC	Identificador del empleado.	FK

Tabla N°12: Atributos de la tabla PROVEEDOR

NOMBRE	TIPO DE ATRIBUTO	DESCRIPCIÓN	TIPO DE LLAVE
ID_PROVEEDOR	VARCHAR2(10)	Identificador del proveedor.	PK
NOMBRE	VARCHAR2(120)	Nombre del proveedor.	
NUMERO_CELULAR	CHAR (9)	Número de celular	
DIRECCION	VARCHAR2(200)	Dirección del proveedor.	

Tabla N°13: Atributos de la tabla ABASTECIMIENTO

NOMBRE	TIPO DE ATRIBUTO	DESCRIPCIÓN	TIPO DE LLAVE
ID_ABASTECIMIENTO	VARCHAR2(10)	Identificador del abastecimiento.	PK
FECHA	DATE	Fecha de abastecimiento.	
ID_PROVEEDOR	VARCHAR2(10)	Identificador del proveedor.	FK
TOTAL_DE_UNIDADES	NUMERIC	Total, de unidades por producto.	
ID_PRODUCTO	VARCHAR2(10)	Identificador del producto.	FK

Tabla N°14: Atributos de la tabla PRODUCTO

NOMBRE	TIPO DE ATRIBUTO	DESCRIPCIÓN	TIPO DE LLAVE
ID_PRODUCTO	VARCHAR2(10)	Identificador del producto.	PK
NOMBRE	VARCHAR2(80)	Nombre del producto.	
ID_PROVEEDOR	VARCHAR2(10)	Identificador de un proveedor.	FK
PRECIO_UNITARIO	NUMERIC (5,2)	Precio unitario del producto.	
MARCA	VARCHAR2(60)	Marca del producto.	
DESCRIPCION	VARCHAR2(200)	Descripción del producto.	

Tabla N°15: Atributos de la tabla TIPO_SERVICIO

NOMBRE	TIPO DE ATRIBUTO	DESCRIPCIÓN	TIPO DE LLAVE
ID_TIPO_SERVICIO	NUMERIC	Identificador del tipo de servicio	PK
DESCRIPCION	VARCHAR2(20)	Descripción del tipo de servicio: 1: Consulta médica 2: Baño regular 3: Baño medicado 4: Peluquería	

3. DESCRIPCIÓN DE LOS PROCEDIMIENTOS Y FUNCIONES

El procedimiento **AUMENTAR_INVENTARIO** recibe como parámetros el identificador del almacén, el identificador del producto y una cantidad de ese producto. Si encontramos el producto, aumentamos su stock, en caso contrario se mostrará un mensaje de error.

La función **OBTENERMONTOMAXIMO** recibe como parámetros el nombre del cliente, su apellido paterno y materno, retornando el monto máximo gastado en la compra de un producto o servicio.

Los **BLOQUES ANÓNIMOS 1 y 2** nos permiten hallar el producto con mayor y menor cantidad de ventas de la tabla **VENTAS**, en su forma base, pero se podría modificar para calcularlo para el mes o el año del sistema (Se agrega una línea que valide que la fecha de la venta sea correspondiente al mes o al año del sistema).

El trigger **ACTUALIZAR_ESTADO_VENTA** se dispara después de insertar o actualizar en la tabla **ENTREGA**. Verifica el estado de la entrega (**ESTADO_DE_ENTREGA**) y actualiza el estado correspondiente en la tabla **VENTA (ESTADO)** basándose en la identificación de la venta (**ID_VENTA**).

El trigger **ACTUALIZAR_STOCK_DESPUES_VENTA** se ejecuta después de insertar o actualizar en la tabla **VENTA**. Calcula la cantidad vendida **UNIDADES_VENDIDAS** y actualiza la cantidad actual en la tabla **INVENTARIO (CANTIDAD_ACTUAL)** para el producto vendido (**ID_PRODUCTO**).

El trigger **VALIDAR_VENTA** se dispara solo una vez luego de insertar en la tabla **VENTA**. Se verifica que la cantidad solicitada en la última venta se encuentre dentro del rango de stock disponible del producto seleccionado. Por un lado, actualiza el stock en la tabla **INVENTARIO** con la condición de que exista la cantidad solicitada. Caso contrario, se elimina la venta ya que sin stock disponible no se podría continuar con el proceso.

El procedimiento: **listar_servicios_mascota**, nos permite saber las cantidad de servicios recibidos en un año, la descripción de cada servicio ofrecido y el total de ingresos generados por estos servicios. Por otro lado, el procedimiento **listar_ventas_del_mes**, nos ayuda a saber en un determinado mes que ventas se han realizado, con sus descripciones correspondientes a cada una.

4. PROGRAMAS FUENTE

AUMENTAR_INVENTARIO

```
create or replace procedure aumentar_inventario(id_inventario varchar2, id_producto
varchar2,
stock number)
is
    resultado number;
begin
    select count(*) into resultado
    from INVENTARIO
    where upper(id_stock) = upper(id_inventario)
        and upper(id_producto) = upper(id_producto);
    if resultado = 0 then
        dbms_output.put_line('No se encontró el producto');
    else
        update INVENTARIO
        set cantidad_actual = cantidad_actual + stock
        where upper(id_stock) = upper(id_inventario)
            and upper(id_producto) = upper(id_producto);
    end if;
exception
    when NO_DATA_FOUND then
        dbms_output.put_line('Error inesperado');
end;
```

OBTENERMONTOMAXIMO

```
CREATE OR REPLACE FUNCTION obtenerMontoMaximo(p_nombreCliente IN
VARCHAR2,p_apellidoPaterno IN VARCHAR2,p_apellidoMaterno IN VARCHAR2)
RETURN NUMBER
IS
    montoMaximo NUMBER;
```

```

BEGIN

    SELECT MAX(TOTAL) INTO montoMaximo

FROM VENTA

WHERE UPPER(ID_CLIENTE) IN (

    SELECT ID_CLIENTE

FROM CLIENTE

WHERE

    UPPER(NOMBRE_CLIENTE) = UPPER(p_nombreCliente) AND

    UPPER(APELLIDO_PATERNO) = UPPER(p_apellidoPaterno) AND

    (UPPER(APELLIDO_MATERNO) = UPPER(p_apellidoMaterno) OR

APELLIDO_MATERNO IS NULL)

);

IF montoMaximo IS NULL THEN

    DBMS_OUTPUT.PUT_LINE('No se encontraron ventas para el cliente
especificado');

END IF;

RETURN NVL(montoMaximo, 0);

EXCEPTION

    WHEN OTHERS THEN

        DBMS_OUTPUT.PUT_LINE('Error inesperado');

END;

```

BLOQUE ANÓNIMO 1

```

DECLARE

    CURSOR ventas_cursor IS

```

```

        SELECT id_producto
        FROM venta;

CURSOR productos_cursor IS

        SELECT id_producto
        FROM producto;

id_producto_max varchar(20);

cant_vendidos NUMBER :=0;

cant_vendidos_max NUMBER :=0;

producto_max varchar(80);

BEGIN

FOR producto IN productos_cursor LOOP

        cant_vendidos:=0;

        FOR ventas IN ventas_cursor LOOP

                IF(ventas.ID_PRODUCTO = producto.ID_PRODUCTO) THEN

                        cant_vendidos:=cant_vendidos+1;

                END IF;

        END LOOP;

        IF(cant_vendidos>cant_vendidos_max)THEN

                cant_vendidos_max:=cant_vendidos;

                id_producto_max:=producto.ID_PRODUCTO;

        END IF;

END LOOP;

SELECT P.NOMBRE INTO producto_max

FROM PRODUCTO P

```

```

WHERE P.id_producto=id_producto_max;

DBMS_OUTPUT.PUT_LINE('El producto mas vendido es: '||producto_max||' con '||
cant_vendidos_max||' de productos vendidos.');
```

END;

BLOQUE ANÓNIMO 2

```

DECLARE

CURSOR ventas_cursor IS

    SELECT id_producto

    FROM venta;

CURSOR productos_cursor IS

    SELECT id_producto

    FROM producto;

id_producto_min varchar(20);

cant_vendidos NUMBER :=0;

cant_vendidos_min NUMBER :=200;

producto_min varchar(80);

BEGIN

FOR producto IN productos_cursor LOOP

    cant_vendidos:=0;

    FOR ventas IN ventas_cursor LOOP

        IF(ventas.ID_PRODUCTO = producto.ID_PRODUCTO) THEN

            cant_vendidos:=cant_vendidos+1;
```



```

        END IF;

    END LOOP;

    IF(cant_vendidos<cant_vendidos_min)THEN

        cant_vendidos_min:=cant_vendidos;

        id_producto_min:=producto.ID_PRODUCTO;

    END IF;

END LOOP;

SELECT P.NOMBRE INTO producto_min

FROM PRODUCTO P

WHERE P.id_producto=id_producto_min;


DBMS_OUTPUT.PUT_LINE('El producto menos vendido es: '||producto_min||' con
'||

cant_vendidos_min||' de productos vendidos.');
```

END;

TRIGGER ACTUALIZA EL ESTADO DE VENTA

```

CREATE OR REPLACE TRIGGER actualizar_estado_venta

AFTER INSERT OR UPDATE ON ENTREGA

FOR EACH ROW

DECLARE

    v_estado_venta CHAR(1);

BEGIN

    SELECT ESTADO INTO v_estado_venta

    FROM VENTA
```

```

WHERE ID_VENTA = :NEW.ID_VENTA;

IF :NEW.ESTADO_DE_ENTREGA = 'E' THEN

    UPDATE VENTA

    SET ESTADO = 'C'

    WHERE ID_VENTA = :NEW.ID_VENTA;

ELSIF :NEW.ESTADO_DE_ENTREGA = 'A' THEN

    UPDATE VENTA

    SET ESTADO = 'A'

    WHERE ID_VENTA = :NEW.ID_VENTA;

ELSIF :NEW.ESTADO_DE_ENTREGA = 'P' THEN

    UPDATE VENTA

    SET ESTADO = 'P'

    WHERE ID_VENTA = :NEW.ID_VENTA;

END IF;

END;

```

TRIGGER ACTUALIZA EL STOCK EN EL INVENTARIO

```

CREATE OR REPLACE TRIGGER actualizar_stock_despues_venta

AFTER INSERT OR UPDATE ON VENTA

FOR EACH ROW

```

DECLARE

v_id_producto VARCHAR2(10);

v_cantidad_vendida NUMBER;

BEGIN

v_id_producto := :new.ID_PRODUCTO;

v_cantidad_vendida := :new.UNIDADES_VENDIDAS;

UPDATE INVENTARIO inv

SET inv.CANTIDAD_ACTUAL = inv.CANTIDAD_ACTUAL -
v_cantidad_vendida

WHERE inv.ID_PRODUCTO = v_id_producto ;

EXCEPTION

WHEN NO_DATA_FOUND THEN

DBMS_OUTPUT.PUT_LINE('No se encontraron datos para la venta con ID: ' ||
:new.ID_VENTA);

NULL;

END;

Procedimiento para ver las ventas de cada mes.

CREATE OR REPLACE PROCEDURE listar_ventas_del_mes (mes NUMBER) IS

CURSOR ventas_cursor IS

SELECT id_venta, id_cliente, fecha, id_producto, total

FROM venta

WHERE TO_NUMBER(TO_CHAR(fecha, 'MM')) = mes;

```

CURSOR cliente_cursor IS

    SELECT id_cliente,nombre_cliente, apellido_paterno, apellido_materno

    FROM cliente;

    producto_adquirido varchar(80);

BEGIN

    FOR venta_rec IN ventas_cursor LOOP

        SELECT P.NOMBRE INTO producto_adquirido

        FROM PRODUCTO P

        WHERE P.id_producto = venta_rec.id_producto;

        FOR cliente_c IN cliente_cursor LOOP

            IF( cliente_c.id_cliente = venta_rec.id_cliente) THEN

                DBMS_OUTPUT.PUT_LINE('Venta ID: ' || venta_rec.id_venta);

                DBMS_OUTPUT.PUT_LINE('Cliente ID: ' || venta_rec.id_cliente);

                DBMS_OUTPUT.PUT_LINE('Nombre y apellidos del cliente: ' ||

                    cliente_c.nombre_cliente||' '||cliente_c.apellido_paterno||' '||

                    cliente_c.apellido_materno);

                DBMS_OUTPUT.PUT_LINE('Producto: ' || producto_adquirido);

                DBMS_OUTPUT.PUT_LINE('Fecha: ' || venta_rec.fecha);

                DBMS_OUTPUT.PUT_LINE('Total: ' || venta_rec.total);

            END IF;

        END LOOP;

    END LOOP;

    DBMS_OUTPUT.PUT_LINE('Procedimiento completado.');
```

END;

Procedimiento para ver las ventas totales de los servicios ofrecidos en un año

```
CREATE OR REPLACE PROCEDURE listar_servicios_mascota(yearh NUMBER) IS

    CURSOR servicios_cursor IS

        SELECT id_servicio_mascota,id_mascota, fecha, estado, id_empleado,id_venta

        FROM servicio_mascota

        WHERE TO_NUMBER(TO_CHAR(fecha,'YYYY')) = yearh;

    nombre_mascota VARCHAR2(60);

    nombre VARCHAR2(60);

    apellidoP VARCHAR2(60);

    apellidoM VARCHAR2(60);

    nombre_empleado VARCHAR2(60);

    costo_del_servicio NUMBER;

    total_de_costo NUMBER := 0;

    total_de_servicio_ofrecidos NUMBER := 0;

BEGIN

    FOR servicio_rec IN servicios_cursor

    LOOP

        SELECT nombre INTO nombre_empleado

        FROM empleado

        where id_empleado = servicio_rec.id_empleado;

        SELECT c.nombre_cliente, c.apellido_paterno, c.apellido_materno INTO

        nombre, apellidoP, apellidoM

        FROM cliente c, mascota m
```

```

where m.id_mascota = servicio_rec.id_mascota AND m.id_cliente = c.id_cliente;

SELECT nombre into nombre_mascota

FROM mascota

where id_mascota = servicio_rec.id_mascota;

SELECT total INTO costo_del_servicio

FROM venta

WHERE id_venta = servicio_rec.id_venta;

total_de_servicio_ofrecidos := total_de_servicio_ofrecidos +1;

total_de_costo := total_de_costo + costo_del_servicio;

DBMS_OUTPUT.PUT_LINE('Servicio ID: ' || servicio_rec.id_servicio_mascota);

DBMS_OUTPUT.PUT_LINE('Nombre de la mascota: '||nombre_mascota);

DBMS_OUTPUT.PUT_LINE('Nombre del dueño de la mascota: '||nombre||' '||
apellidoP||' '||apellidoM);

DBMS_OUTPUT.PUT_LINE('Fecha: ' || servicio_rec.fecha);

DBMS_OUTPUT.PUT_LINE('Estado: ' || servicio_rec.estado);

DBMS_OUTPUT.PUT_LINE('Empleado: ' || nombre_empleado);

DBMS_OUTPUT.PUT_LINE('Costo del servicio: '||costo_del_servicio);

DBMS_OUTPUT.PUT_LINE('-----');

END LOOP;

DBMS_OUTPUT.PUT_LINE('Procedimiento completado.');
```

```

DBMS_OUTPUT.PUT_LINE('Total de servicios ofrecidos:
'||total_de_servicio_ofrecidos);

DBMS_OUTPUT.PUT_LINE('Total de ingresos por servicios: '||total_de_costo);

END;
```

TRIGGER DE SENTENCIA PARA VALIDAR VENTAS

CREATE OR REPLACE TRIGGER VALIDAR_VENTA

AFTER INSERT ON VENTA

DECLARE

VENTA_ANALIZADA VARCHAR2(6);

PRODUCTO_ESCOGIDO VARCHAR2(6);

CANTIDAD_UNIDADES_VENTA NUMBER;

CANTIDAD_DISPONIBLE_ACTUAL NUMBER;

IDSTOCK_ESCOGIDO VARCHAR2(6);

BEGIN

SELECT MAX(ID_VENTA)

INTO VENTA_ANALIZADA

FROM VENTA;

SELECT ID_PRODUCTO,UNIDADES_VENDIDAS

INTO PRODUCTO_ESCOGIDO,CANTIDAD_UNIDADES_VENTA

FROM VENTA

WHERE ID_VENTA=VENTA_ANALIZADA;

SELECT CANTIDAD_ACTUAL

INTO CANTIDAD_DISPONIBLE_ACTUAL

FROM INVENTARIO

WHERE ID_PRODUCTO=PRODUCTO_ESCOGIDO;

```
IF CANTIDAD_UNIDADES_VENTA>CANTIDAD_DISPONIBLE_ACTUAL
THEN

    DELETE FROM VENTA

    WHERE ID_VENTA=VENTA_ANALIZADA;

    DBMS_OUTPUT.PUT_LINE('No hay suficiente stock para realizar la venta');

ELSE

    UPDATE INVENTARIO

    SET CANTIDAD_ACTUAL=
CANTIDAD_ACTUAL-CANTIDAD_UNIDADES_VENTA

    WHERE ID_PRODUCTO=PRODUCTO_ESCOGIDO;

    DBMS_OUTPUT.PUT_LINE('Existe stock suficiente para realizar la venta');

END IF;

EXCEPTION


    WHEN NO_DATA_FOUND THEN

        DBMS_OUTPUT.PUT_LINE('No se ha ingresado correctamente los datos');


END;
```


5. REPORTES DE GESTIÓN

Reporte sobre las ventas realizadas al 31 de octubre del 2023

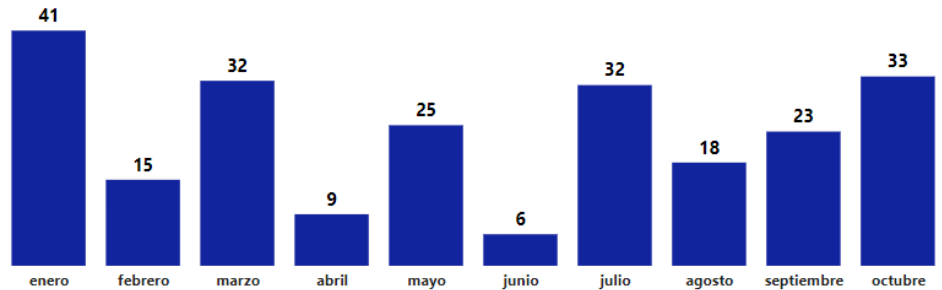


50% pagos en diferido (Ventas por tarjeta de crédito o débito)



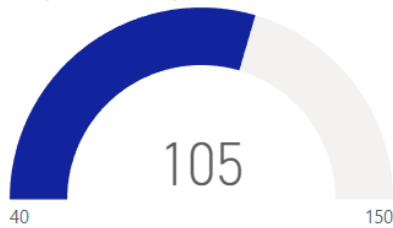
3 trabajadores a cargo de las ventas

Ventas mensuales (S/)

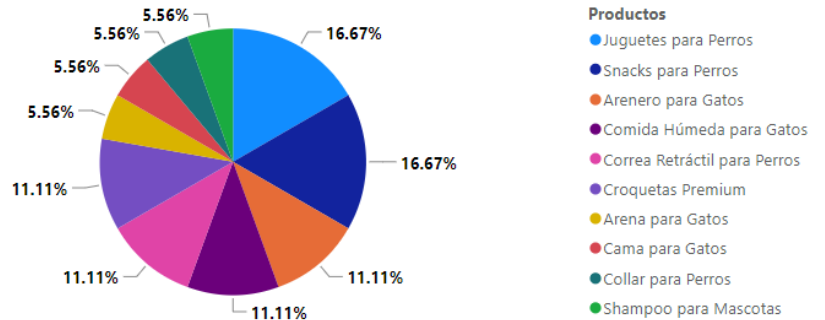


Seleccionar código de producto
INV010

Capacidades del producto en el almacén



Productos vendidos



Registro de atención animal al 31 de octubre del 2023

Registro de atención animal

Al 31 de octubre del 2023

Cientes

Todas

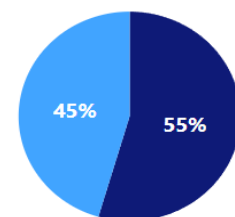
S/ 233.14 Ingresos por los servicios

Ciente # de mascotas

Ana	3
Carmen	3
Juan	2
Laura	2
Sergio	2
Jorge	1
María	1
Pedro	1
Carlos	0
Isabel	0
Total	15

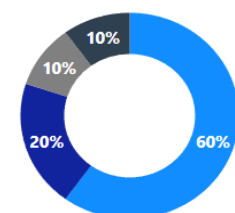
Tipo de mascotas atendidas

Perro Gato



Servicios brindados

Consulta medica Peluquería Baño medicado Baño regular



Contribución de los empleados



10 Servicios brindados

10 Clientes atendidos

