



UNIVERSIDADE  
DE VIGO

ESCOLA SUPERIOR DE ENXEÑERÍA INFORMÁTICA

**PRÁCTICA 7: PLSQL**

**GRUPO: BDII4\_4**

**TEMÁTICA: AERO**

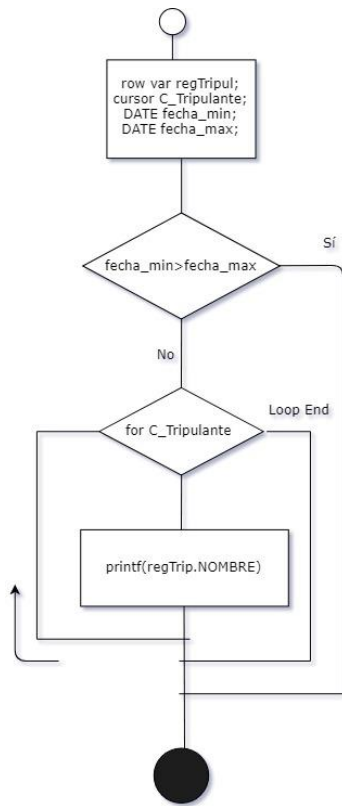
DNI: 44662548K NOMBRE: VÁZQUEZ AUGUSTO, MANUEL  
DNI: 45158726C NOMBRE: CURRÁS RODRÍGUEZ, ALEXANDRE  
DNI: 53194233V NOMBRE: OTERO CABALEIRO, VÍCTOR  
DNI: Y4237386H NOMBRE: LEONARD HERRRERA, ALEJANDRO



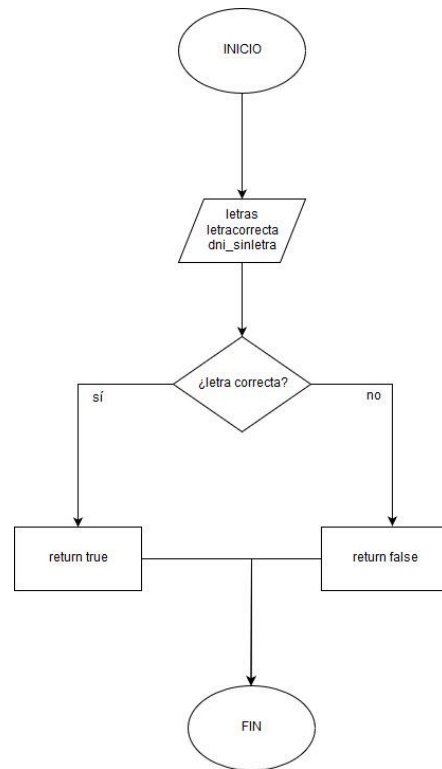
BASES DE DATOS II 2018-2019

CALIFICACIÓN

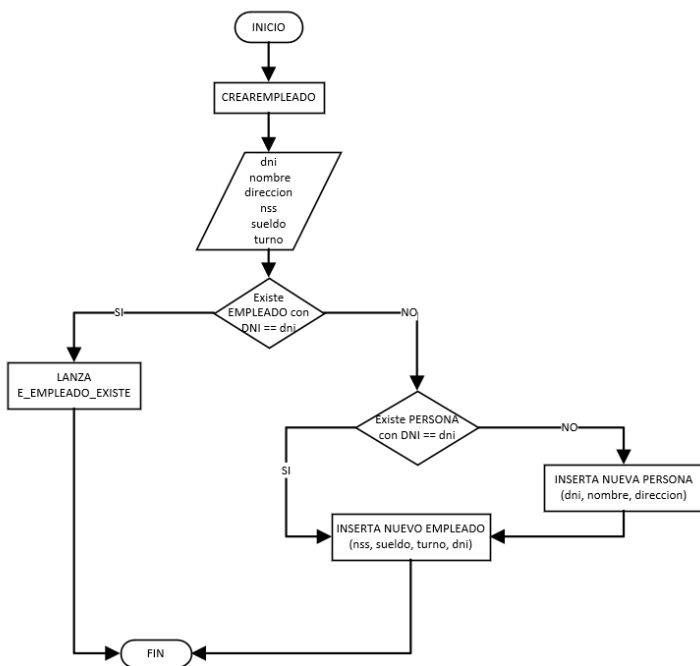
## DIAGRAMAS DE FLUJO DE PROCEDIMIENTOS Y FUNCIONES



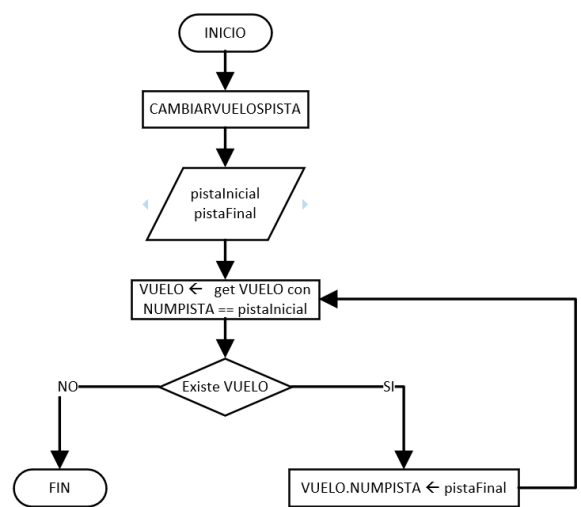
**imprimeTripulantes**



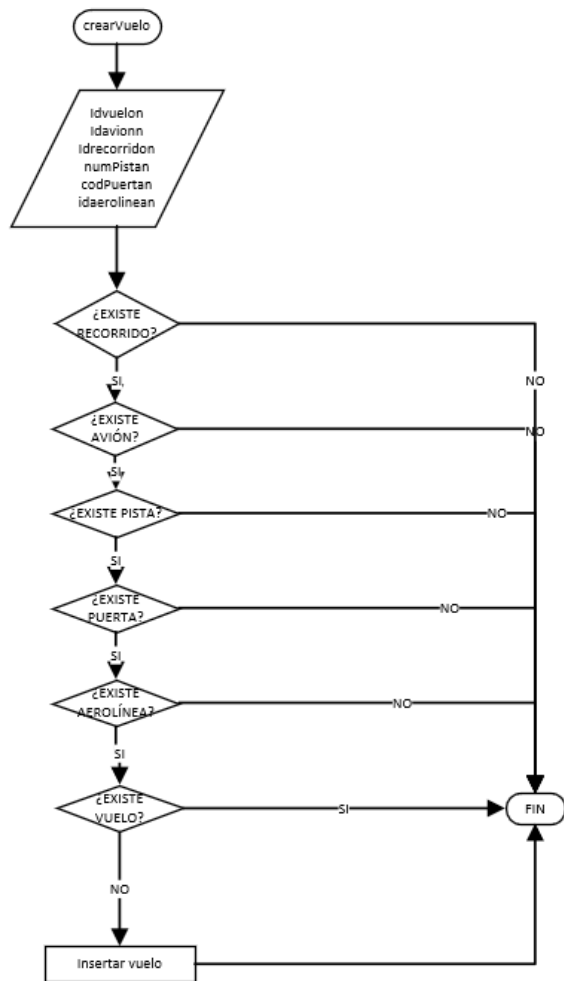
**validadni**



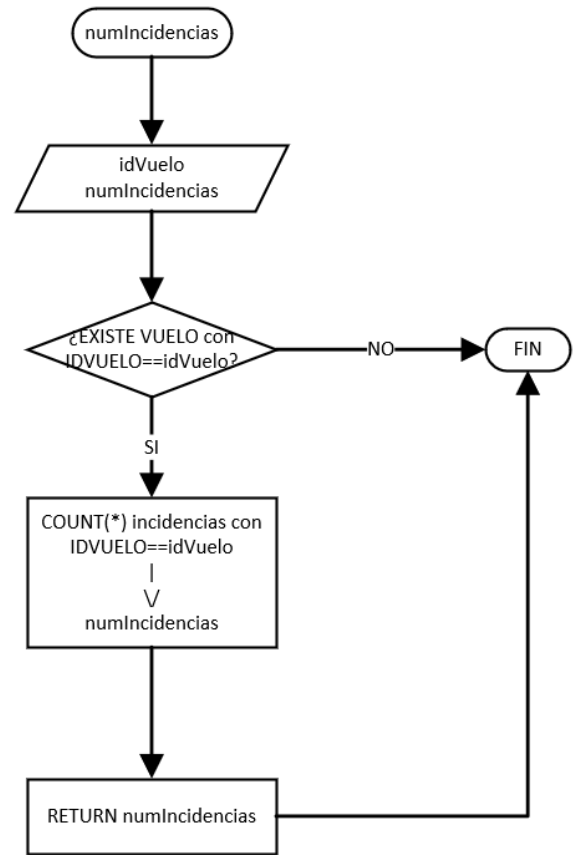
**crearEmpleado**



**cambiarVuelosPista**



**crearVuelo**



**numIncidentes**

## SENTENCIAS DE DEFINICIÓN DE PROCEDIMIENTOS Y FUNCIONES

```
CREATE OR REPLACE PROCEDURE imprimeTripulantes (fecha_inicio IN
VUELO.FECHA%TYPE, fecha_fin IN VUELO.FECHA%TYPE)
IS
    regTripul TRIPULANTE%ROWTYPE;

    E_TIEMPO_NO_VALIDO EXCEPTION;
    PRAGMA EXCEPTION_INIT(E_TIEMPO_NO_VALIDO, -10900);

    CURSOR C_TRIPULANTE IS
        SELECT DISTINCT NOMBRE FROM PERSONA,TRIPULANTE,VUELO,
TRABAJA_EN
        WHERE PERSONA.DNI=TRIPULANTE.DNI_TRIPULANTE
        AND TRIPULANTE.DNI_TRIPULANTE=TRABAJA_EN.DNI_TRIPULANTE
        AND TRABAJA_EN.IDVUELO=VUELO.IDVUELO
        AND VUELO.FECHA >= fecha_inicio AND VUELO.FECHA <=
fecha_fin;

BEGIN

    IF fecha_inicio > fecha_fin THEN
        RAISE E_TIEMPO_NO_VALIDO;
    ELSE
        DBMS_OUTPUT.PUT_LINE('Las siguientes personas fueron
tripulantes entre: '|| fecha_inicio || ' y '|| fecha_fin);
        FOR regTripul IN C_TRIPULANTE LOOP
            DBMS_OUTPUT.PUT_LINE(regTripul.NOMBRE);
        END LOOP;

    END IF;
END imprimeTripulantes;
/

CREATE OR REPLACE PROCEDURE crearEmpleado (dnin IN
PERSONA.DNI%TYPE, nombre IN PERSONA.NOMBRE%TYPE, direccion IN
PERSONA.DIRECCION%TYPE, nss IN EMPLEADO.NSS%TYPE, sueldo IN
EMPLEADO.SUELDO%TYPE, turno IN EMPLEADO.TURNO%TYPE)
IS
    pers PERSONA%ROWTYPE;
    emp EMPLEADO%ROWTYPE;

    E_EMPLEADO_EXISTE EXCEPTION;
    PRAGMA EXCEPTION_INIT(E_EMPLEADO_EXISTE, -10100);

    CURSOR C_PERS IS
        SELECT * FROM PERSONA WHERE PERSONA.DNI LIKE dnin;
    CURSOR C_EMPL IS
        SELECT * FROM EMPLEADO WHERE EMPLEADO.DNI_EMPLEADO=dnin;

BEGIN
```

```

OPEN C_EMPL;
OPEN C_PERS;

FETCH C_EMPL INTO emp;
FETCH C_PERS INTO pers;

IF C_EMPL%FOUND THEN
    RAISE E_EMPLEADO_EXISTE;
ELSIF C_PERS%NOTFOUND THEN
    INSERT INTO PERSONA(DNI, NOMBRE, DIRECCION) VALUES
(dnin, nombre, direccion);
END IF;

INSERT INTO EMPLEADO (DNI_EMPLEADO,NSS,SUELDO,TURNO) VALUES
(dnin, nss, sueldo, turno);

CLOSE C_EMPL;
CLOSE C_PERS;

END crearEmpleado;
/

CREATE OR REPLACE PROCEDURE cambiarVuelosPista (pistaInicial IN
VUELO.NUMPISTA%TYPE, pistaFinal IN VUELO.NUMPISTA%TYPE)
IS

    regVuelo VUELO%ROWTYPE;

    CURSOR C_VUELO IS
        SELECT * FROM VUELO WHERE NUMPISTA=pistaInicial
        FOR UPDATE;
BEGIN

    FOR regVuelo IN C_VUELO LOOP
        DBMS_OUTPUT.PUT_LINE('El vuelo ' || regVuelo.IDVUELO ||
' se cambia a la pista ' || pistaFinal);
        UPDATE VUELO SET NUMPISTA = pistaFinal WHERE CURRENT OF
C_VUELO;
    END LOOP;

END cambiarVuelosPista;
/

CREATE OR REPLACE PROCEDURE crearvuelo (
    idvuelon      IN vuelo.idvuelo%TYPE,
    idavionn      IN vuelo.idavion%TYPE,
    idrecorridon  IN vuelo.idrecorrido%TYPE,
    fechan        IN vuelo.fecha%TYPE,
    numpistan     IN vuelo.numpista%TYPE,
    codpuertan    IN vuelo.codpuerta%TYPE,
    idaeron       IN vuelo.idaerolinea%TYPE
) IS

    vueloC vuelo%rowtype;
    recorridoC recorrido%rowtype;

```

```

avionC avion%rowtype;
pistaC pista%rowtype;
puertaC puertaembarque%rowtype;
aeroC aerolinea%rowtype;

E_VUELO_EXISTE EXCEPTION;
e_avion_no_existe EXCEPTION;
e_aero_no_existe EXCEPTION;
e_recorrido_no_existe EXCEPTION;
    e_puerta_no_existe EXCEPTION;
    e_pista_no_existe EXCEPTION;

PRAGMA EXCEPTION_INIT(E_VUELO_EXISTE, -10200);
PRAGMA EXCEPTION_INIT(e_avion_no_existe, -10300);
PRAGMA EXCEPTION_INIT(e_aero_no_existe, -10400);
PRAGMA EXCEPTION_INIT(e_recorrido_no_existe, -10500);
PRAGMA EXCEPTION_INIT(e_puerta_no_existe, -10600);
PRAGMA EXCEPTION_INIT(e_pista_no_existe, -10700);

CURSOR c_vuelo IS SELECT * FROM vuelo WHERE vuelo.idvuelo
LIKE idvuelon;
CURSOR c_recorrido IS SELECT * FROM recorrido WHERE
recorrido.idrecorrido LIKE idrecorridon;
CURSOR c_avion IS SELECT * FROM avion WHERE avion.idavion
LIKE idavionn;
CURSOR c_pista IS SELECT * FROM pista WHERE pista.numPista
LIKE numpistan;
CURSOR c_puerta IS SELECT * FROM puertaembarque WHERE
puertaembarque.codpuerta LIKE codpuertan;
CURSOR c_aero IS SELECT * FROM aerolinea WHERE
aerolinea.idaerolinea LIKE idaeron;

BEGIN

    OPEN c_recorrido;
    FETCH c_recorrido INTO recorridoC;
    IF c_recorrido%notfound
        THEN
    RAISE e_recorrido_no_existe;
    END IF;
    CLOSE c_recorrido;

    OPEN c_avion;
    FETCH c_avion INTO avionC;
    IF c_avion%notfound
        THEN
    RAISE e_avion_no_existe;
    END IF;
    CLOSE c_avion;

    OPEN c_pista;
    FETCH c_pista INTO pistaC;
    IF c_pista%notfound
        THEN
    RAISE e_pista_no_existe;
    END IF;

```

```

CLOSE c_pista;

OPEN c_puerta;
FETCH c_puerta INTO puertaC;
IF c_puerta%notfound
    THEN
RAISE e_puerta_no_existe;
END IF;
CLOSE c_puerta;

OPEN c_aero;
FETCH c_aero INTO aeroC;
IF c_aero%notfound
    THEN
RAISE e_aero_no_existe;
END IF;
CLOSE c_aero;

OPEN c_vuelo;
FETCH c_vuelo INTO vueloC;
IF c_vuelo%found

THEN
    RAISE E_VUELO_EXISTE;
ELSE
    INSERT INTO vuelo (
        idvuelo,
        idavion,
        idrecorrido,
        fecha,
        numpista,
        codpuerta,
        idaerolinea)
VALUES (
        idvuelon,
        idavionn,
        idrecorridon,
        fechan,
        numpistan,
        codpuertan,
        idaeron);
END IF;

CLOSE c_vuelo;

END crearvuelo;
/

CREATE OR REPLACE FUNCTION numIncidencias(idvuelon IN
vuelo.idvuelo%TYPE)
RETURN NUMBER
IS
    numIncidencias NUMBER;
    vueloC vuelo%rowtype;

    E_VUELO_NO_EXISTE EXCEPTION;

```

```

PRAGMA EXCEPTION_INIT(E_VUELO_NO_EXISTE, -10800);

CURSOR c_vuelo IS SELECT * FROM vuelo WHERE vuelo.idvuelo
LIKE idvuelon;
BEGIN
    OPEN c_vuelo;
    FETCH c_vuelo INTO vueloC;
    IF c_vuelo%notfound
    THEN
        RAISE E_VUELO_NO_EXISTE;
    END IF;
    CLOSE c_vuelo;
    SELECT COUNT(*) INTO numIncidencias FROM INCIDENCIA
WHERE INCIDENCIA.idvuelo LIKE idvuelon;
    RETURN numIncidencias;

END numIncidencias;
/

CREATE OR REPLACE FUNCTION validadni (
    dni    IN persona.dni%TYPE
) RETURN BOOLEAN AS
    letras          CHAR(23) := 'TRWAGMYFPDXBNJZSQVHLCKE';
    letracorrecta   CHAR(1);
    dni_sinletra    INTEGER;
BEGIN
    dni_sinletra := substr(dni,0,8);
    letracorrecta := substr(letras, (dni_sinletra MOD 23) +
1,1);
    IF
        ( letracorrecta = upper(substr(dni,9,1) ) )
    THEN
        RETURN true;
    ELSE
        RETURN false;
    END IF;

END validadni;
/

```



## SENTENCIAS DE DEFINICIÓN DE BLOQUES DE PRUEBAS

```
set serveroutput on;
```

```
DECLARE
```

```
    fechamin VUELO.FECHA%TYPE;
```

```
    fechamax VUELO.FECHA%TYPE;
```

```
    E_TIEMPO_NO_VALIDO EXCEPTION;
```

```
    PRAGMA EXCEPTION_INIT(E_TIEMPO_NO_VALIDO, -10900);
```

```
BEGIN
```

```
    fechamin := DATE '1999-03-4';
```

```
    fechamax := DATE '1999-03-6';
```

```
    dbms_output.put_line('-----
```

```
>imprimeTripulantes<-----');;
```

```
    imprimeTripulantes(fechamin, fechamax);
```

```
    EXCEPTION
```

```
    WHEN E_TIEMPO_NO_VALIDO THEN
```

```
        DBMS_OUTPUT.PUT_LINE('Intervalo de tiempo no válido!');
```

```
    WHEN OTHERS THEN
```

```
        DBMS_OUTPUT.PUT_LINE('Error inesperado');
```

```
END;
```

```
/
```

```
DECLARE
```

```
    idvuelo vuelo.idvuelo%TYPE;
```

```
    num NUMBER;
```

```
    E_VUELO_NO_EXISTE EXCEPTION;
```

```
    PRAGMA EXCEPTION_INIT(E_VUELO_NO_EXISTE, -10800);
```

```
BEGIN
```

```
    idvuelo := 'VUELO6';
```

```
    dbms_output.put_line('-----
```

```
>numIncidencias<-----');
```

```
    dbms_output.put_line('El numero de incidencias es: ' ||
```

```
numIncidencias(idvuelo));
```

```
    EXCEPTION
```

```
    WHEN E_VUELO_NO_EXISTE THEN
```

```
        dbms_output.put_line('El vuelo no existe');
```

```
    WHEN OTHERS THEN
```

```
        dbms_output.put_line('Error inesperado al calcular el  
numero de incidencias');
```

```
END;
```

```
/
```

```
DECLARE
```

```
    dni PERSONA.DNI%TYPE;
```

```
    nombre PERSONA.NOMBRE%TYPE;
```

```
    direccion PERSONA.DIRECCION%TYPE;
```

```
    nss EMPLEADO.NSS%TYPE;
```

```
    sueldo EMPLEADO.SUELDO%TYPE;
```

```
    turno EMPLEADO.TURNO%TYPE;
```

```

E_EMPLEADO_EXISTE EXCEPTION;

PRAGMA EXCEPTION_INIT(E_EMPLEADO_EXISTE, -10100);

BEGIN
    dni := '44662548K';
    nombre := 'Paco';
    direccion := 'Ourense, Rua Nova, nº 10';
    nss := 'GTHD222233337';
    sueldo := '1200';
    turno := 'DIA';

    dbms_output.put_line('-----');
>crearEmpleado<-----');
    crearEmpleado(dni, nombre, direccion, nss, sueldo, turno);

EXCEPTION
    WHEN E_EMPLEADO_EXISTE THEN
        DBMS_OUTPUT.PUT_LINE('El empleado ya existe');
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('Error inesperado');
END;
/

DECLARE
    dniComprobar PERSONA.DNI%TYPE;
BEGIN
    dbms_output.put_line('-----');
>validadni<-----');

    dniComprobar := '446625p8P';
    IF validadni(dniComprobar) THEN
        DBMS_OUTPUT.PUT_LINE('DNI BIEN');
    ELSE
        DBMS_OUTPUT.PUT_LINE('DNI MAL');
    END IF;

EXCEPTION
    WHEN value_error THEN
        dbms_output.put_line('El dni introducido no tiene el
formato correcto, 8 digitos + letra');
    WHEN OTHERS THEN
        dbms_output.put_line('Código: ' || sqlcode);
        dbms_output.put_line('[Mensaje]:'
                                || substr(sqlerrm,11,100) );
END;
/

DECLARE
    pistaInicial VUELO.NUMPISTA%TYPE;
    pistaFinal VUELO.NUMPISTA%TYPE;
BEGIN
    dbms_output.put_line('-----');
>cambiarVuelosPista<-----');
    cambiarVuelosPista(2,3);

```

```

EXCEPTION
    WHEN NO_DATA_FOUND THEN
        DBMS_OUTPUT.PUT_LINE('No hay ningún vuelo en la pista
        ||pistaInicial);
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('Error inesperado');
END;
/

DECLARE
    idvuelo          vuelo.idvuelo%TYPE;
    idavion          vuelo.idavion%TYPE;
    idrecorrido      vuelo.idrecorrido%TYPE;
    fecha            vuelo.fecha%TYPE;
    numpista         vuelo.numpista%TYPE;
    codpuerta        vuelo.codpuerta%TYPE;
    idaero           vuelo.idaerolinea%TYPE;

    E_VUELO_EXISTE EXCEPTION;
    e_avion_no_existe EXCEPTION;
    e_aero_no_existe EXCEPTION;
    e_recorrido_no_existe EXCEPTION;
    e_puerta_no_existe EXCEPTION;
    e_pista_no_existe EXCEPTION;

    PRAGMA EXCEPTION_INIT(E_VUELO_EXISTE, -10200);
    PRAGMA EXCEPTION_INIT(e_avion_no_existe, -10300);
    PRAGMA EXCEPTION_INIT(e_aero_no_existe, -10400);
    PRAGMA EXCEPTION_INIT(e_recorrido_no_existe, -10500);
    PRAGMA EXCEPTION_INIT(e_puerta_no_existe, -10600);
    PRAGMA EXCEPTION_INIT(e_pista_no_existe, -10700);
BEGIN
    idvuelo := 'VUELO5';
    idavion := 'AX4555';
    idrecorrido := '5666523147';
    fecha := '11/05/95';
    numpista := '2';
    codpuerta := 'PL23';
    idaero := 'AEU';
    dbms_output.put_line('-----
>crearVuelo<-----');

    crearvuelo(idvuelo,idavion,idrecorrido,fecha,numpista,codpuerta,
    idaero);
EXCEPTION
    WHEN E_VUELO_EXISTE THEN
        dbms_output.put_line('El vuelo ya existe');
    WHEN e_avion_no_existe THEN
        dbms_output.put_line('El avion no existe');
    WHEN e_aero_no_existe THEN
        dbms_output.put_line('La aerolinea no existe');
    WHEN e_recorrido_no_existe THEN
        dbms_output.put_line('El recorrido no existe');
    WHEN e_pista_no_existe THEN
        dbms_output.put_line('La pista no existe');
    WHEN e_puerta_no_existe THEN

```

```
        dbms_output.put_line('La puerta no existe');  
    WHEN OTHERS THEN  
        dbms_output.put_line('Error inesperado al insertar  
vuelo');  
END;  
/
```

## **HOJA DE FIRMAS**

**DNI: 44662548K NOMBRE: VÁZQUEZ AUGUSTO, MANUEL**

**DNI: 53194233V NOMBRE: OTERO CABALEIRO, VÍCTOR**

**DNI: 45158726C NOMBRE: CURRÁS RODRÍGUEZ, ALEXANDRE**

**DNI: Y4237386H NOMBRE: LEONARD HERRRERA, ALEJANDRO**