



UNIVERSIDADE
DE VIGO

ESCOLA SUPERIOR DE ENXEÑERÍA INFORMÁTICA

PRÁCTICA 7: PLSQL

GRUPO: BDII5_2

TEMÁTICA: FÚTBOL

DNI: 53489444T

DNI: 53819272P

DNI: 53819578S

DNI: 53798504D

NOMBRE: ARIAS CAMPAÑA, PABLO

NOMBRE: ÁLVAREZ IGLESIAS, DAVID

NOMBRE: MARTÍNEZ LÓPEZ, DIEGO

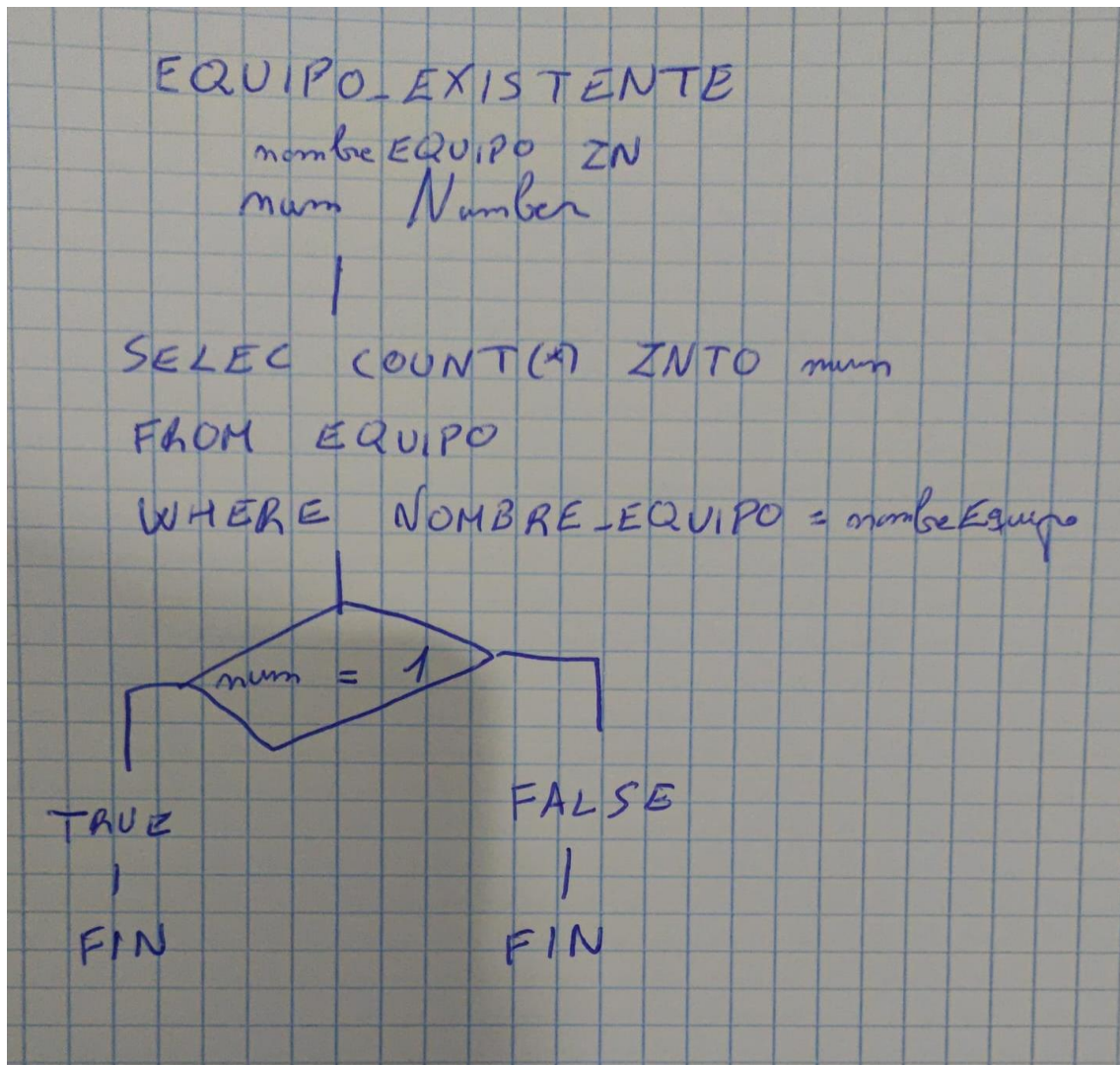
NOMBRE: OLIVEIRA SUÁREZ, RUBÉN



BASES DE DATOS II 2022-2023

CALIFICACIÓN

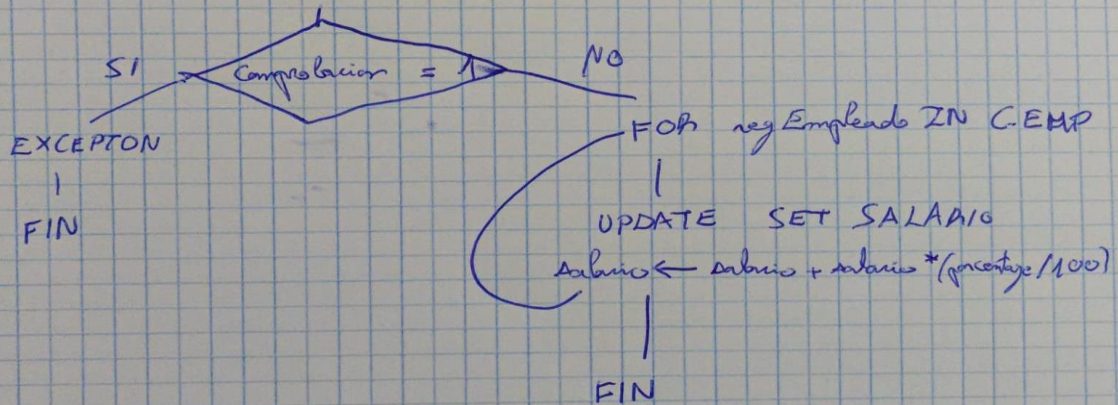
DIAGRAMAS DE FLUJO DE PROCEDIMIENTOS Y FUNCIONES



CAMBIAR_SALARIO

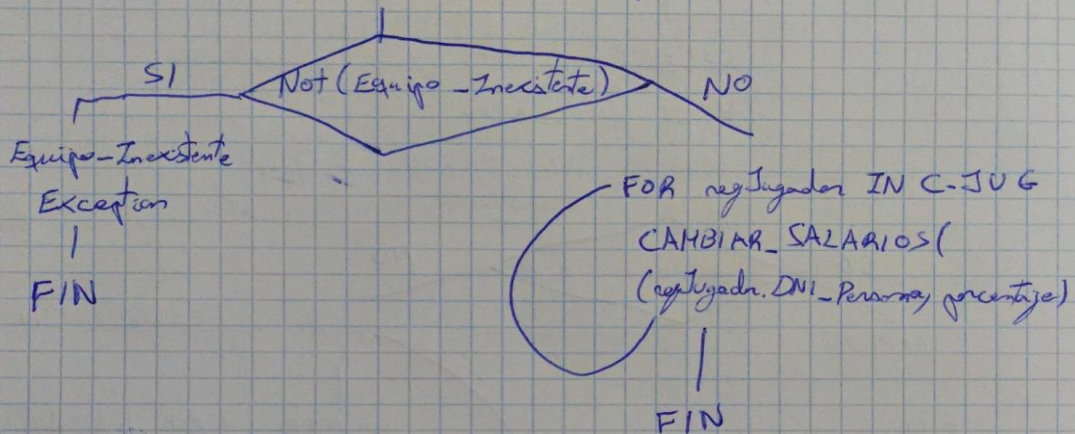
dni IN
Porcentaje IN

SELECT count(*) into comprobacion
FROM EMPLEADO
WHERE DNI_PERSONA = dni



CAMBIAR_SALARIOS_EQUIPO

nombreEquipo IN
~~Porcentaje~~ Porcentaje IN
Equip-Inexistente Exception

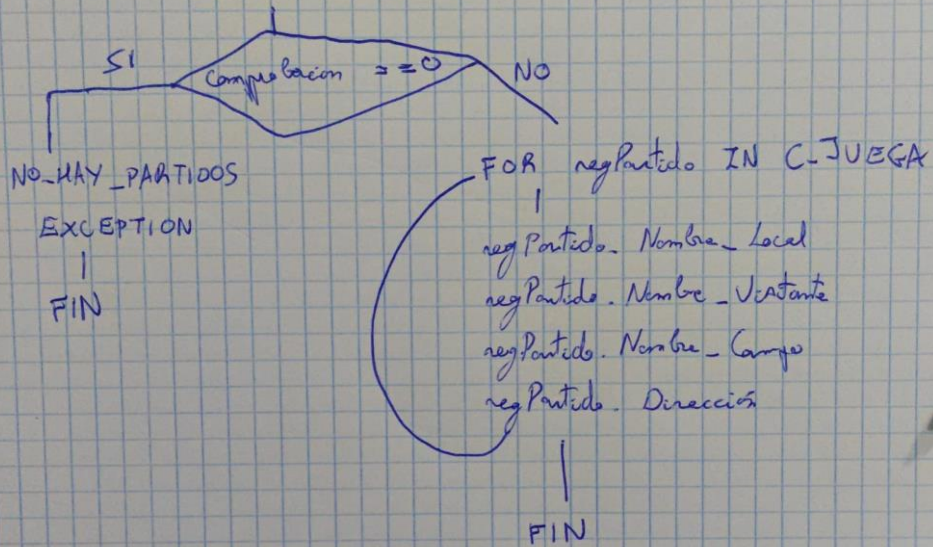


LISTAR - PARTIDOS - FECHA

fecha IN Date
comprobacion Number

NO-HAY-PARTIDOS Exception

SELECT COUNT(*) INTO comprobacion



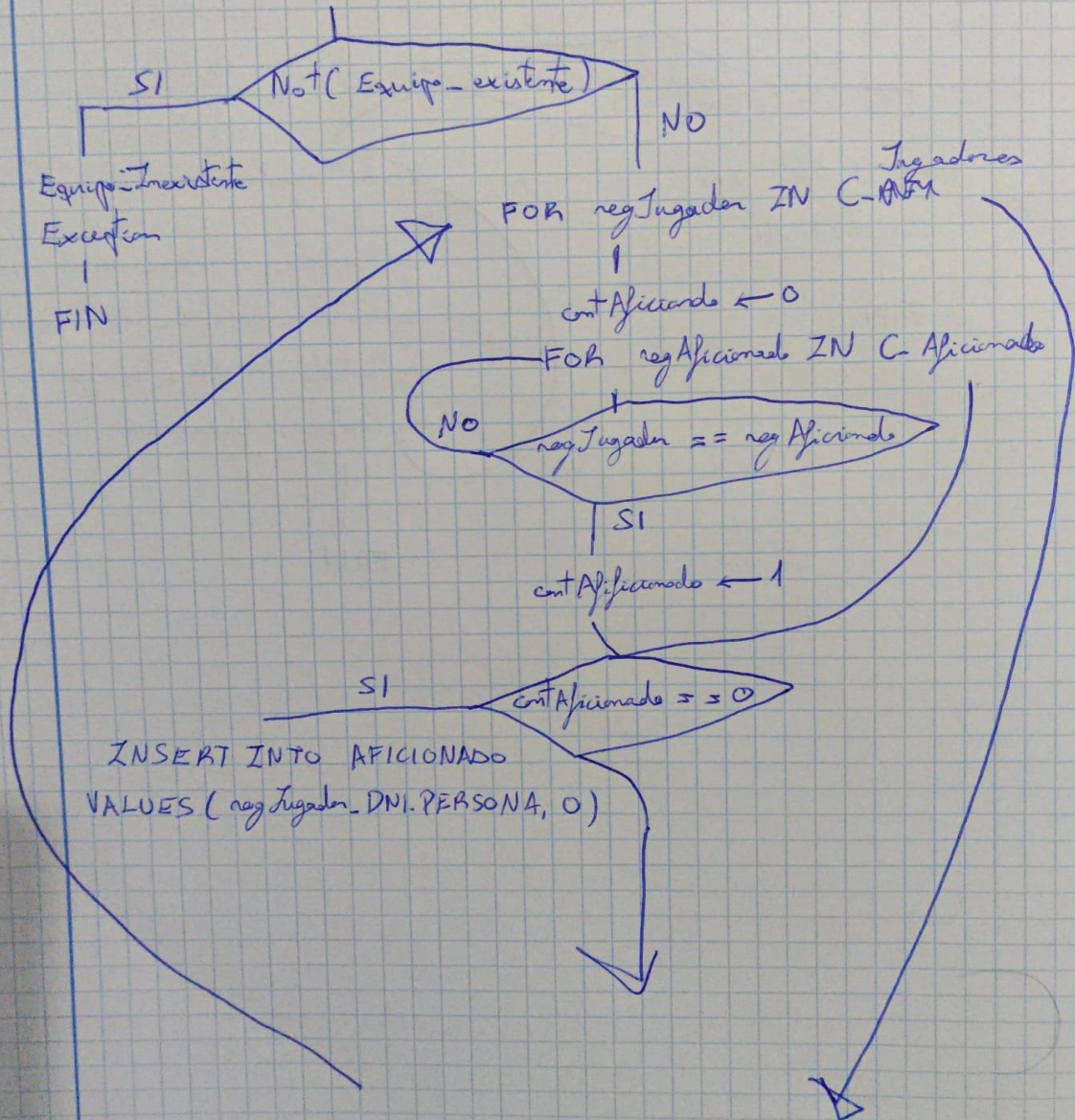
CREAR_AFICIONADOS_VIP-EQUIPO

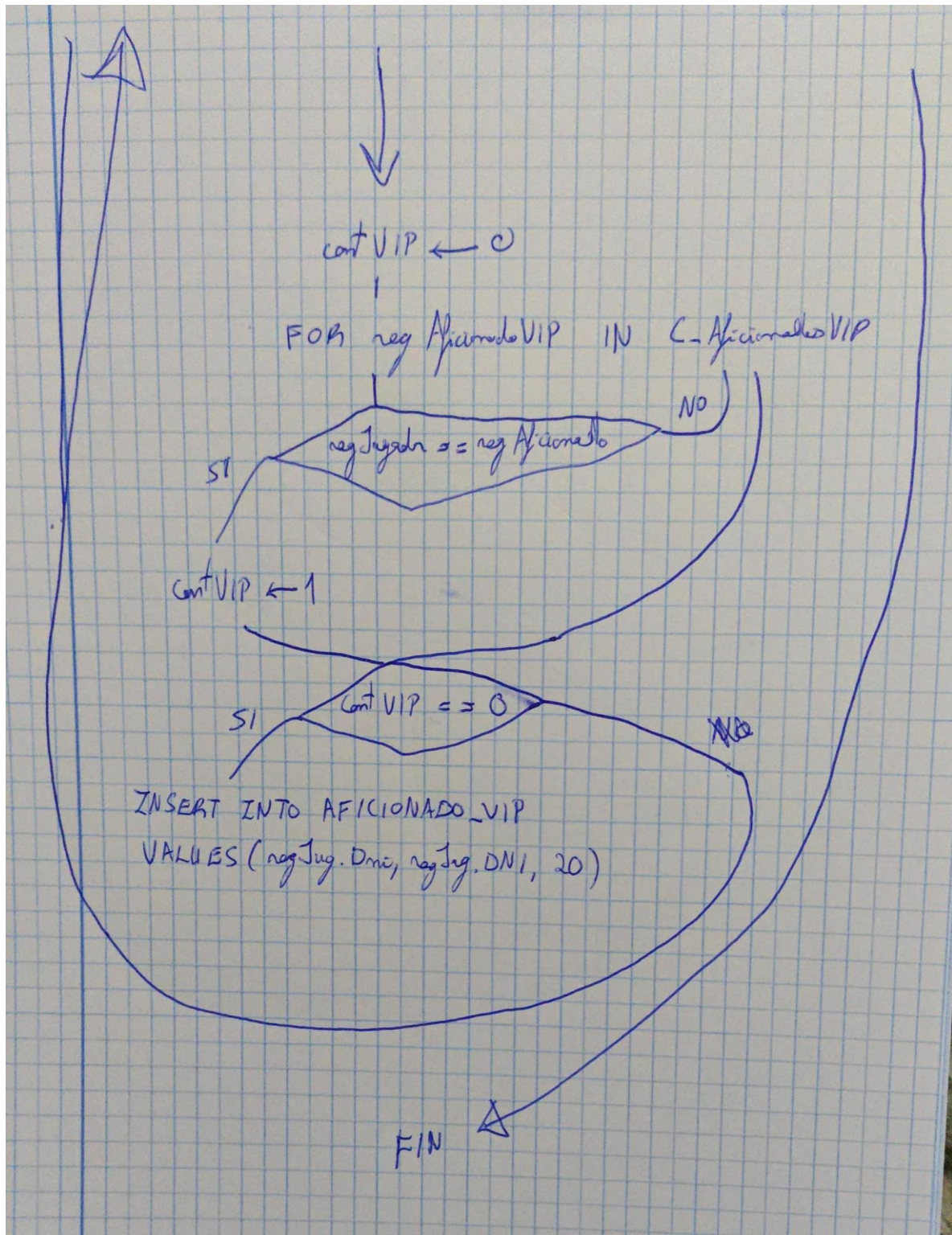
nombreEquipo IN

Equipo-Inexistente Exception

contAficionado Number

contVIP Number





SENTENCIAS DE DEFINICIÓN DE PROCEDIMIENTOS Y FUNCIONES

```

/*****
/* 7.- Procedimientos y Funciones PL/SQL */
*****/

DROP FUNCTION EQUIPO_EXISTENTE;
DROP PROCEDURE LISTA_JUGADORES_EQUIPO;
DROP PROCEDURE CAMBIAR_SALARIOS;
DROP PROCEDURE CAMBIAR_SALARIOS_EQUIPO;
DROP PROCEDURE LISTAR_PARTIDOS_FECHA;
DROP PROCEDURE CREAR_AFICIONADOS_VIP_EQUIPO;

SET SERVEROUTPUT ON
/
--Funcion que devuelve true si el equipo existe
CREATE OR REPLACE FUNCTION EQUIPO_EXISTENTE(nombreEquipo IN
VARCHAR2)
RETURN BOOLEAN
IS
    num NUMBER;
BEGIN
    SELECT COUNT(*) INTO num
        FROM EQUIPO
        WHERE NOMBRE_EQUIPO = nombreEquipo;
    RETURN num = 1;
END EQUIPO_EXISTENTE;
/

--Procedimiento que lista los jugadores de un equipo
CREATE OR REPLACE PROCEDURE LISTA_JUGADORES_EQUIPO(nombreEquipo
IN VARCHAR2, numJug OUT NUMBER )
IS
    regJug JUGADOR%ROWTYPE;
    SIN_JUGADORES EXCEPTION;
    EQUIPO_INEXISTENTE EXCEPTION;
    CURSOR C_JUG IS
        SELECT DNI_PERSONA, NOMBRE_JUGADOR, NOMBRE_EQUIPO,
POSICION, NACIONALIDAD
        FROM JUGADOR
        WHERE JUGADOR.NOMBRE_EQUIPO = nombreEquipo;
BEGIN
    IF (NOT EQUIPO_EXISTENTE(nombreEquipo)) THEN
        RAISE EQUIPO_INEXISTENTE;
    END IF;
    OPEN C_JUG;
    DBMS_OUTPUT.PUT_LINE('Jugadores del equipo: ' ||
nombreEquipo);
    DBMS_OUTPUT.NEW_LINE;

    LOOP
        FETCH C_JUG INTO regJug;
        EXIT WHEN C_JUG%NOTFOUND;

```

```

        DBMS_OUTPUT.PUT_LINE(regJug.NOMBRE_JUGADOR || ' ' ||
regJug.POSICION);
    END LOOP;

    numJug := C_JUG%ROWCOUNT;

    IF (numJug = 0) THEN
        RAISE SIN_JUGADORES;
    END IF;

    CLOSE C_JUG;

EXCEPTION
    WHEN EQUIPO_INEXISTENTE THEN
        DBMS_OUTPUT.PUT_LINE('El equipo no existe');
    WHEN SIN_JUGADORES THEN
        DBMS_OUTPUT.PUT_LINE('El equipo no tiene ningún
jugador');
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('Código: ' || SQLCODE);
END LISTA_JUGADORES_EQUIPO;
/

--Cambiar salario a empleados segun dni
CREATE OR REPLACE PROCEDURE CAMBIAR_SALARIOS(dni IN VARCHAR2 ,
PORCENTAJE IN NUMBER)
IS
    DNI_INEXISTENTE EXCEPTION;
    comprobacion NUMBER;
    CURSOR C_EMP IS
        SELECT EMPLEADO.DNI_PERSONA, PERSONA.NOMBRE, SALARIO
        FROM EMPLEADO, PERSONA
        WHERE EMPLEADO.DNI_PERSONA = dni AND
EMPLEADO.DNI_PERSONA = PERSONA.DNI_PERSONA
        FOR UPDATE;
BEGIN
    SELECT count(*) INTO comprobacion
    FROM EMPLEADO
    WHERE EMPLEADO.DNI_PERSONA = dni;
    IF comprobacion = 0 THEN
        RAISE DNI_INEXISTENTE;
    END IF;
    FOR regEmpleado IN C_EMP LOOP
        DBMS_OUTPUT.PUT_LINE('Nombre: ' || regEmpleado.NOMBRE ||
' Empleado: ' || regEmpleado.DNI_PERSONA || ' Sueldo antes :
' || regEmpleado.SALARIO);
        UPDATE EMPLEADO SET SALARIO = SALARIO +
SALARIO*(PORCENTAJE)/100
        WHERE CURRENT OF C_EMP;
    END LOOP;
EXCEPTION
    WHEN DNI_INEXISTENTE THEN
        DBMS_OUTPUT.PUT_LINE('No existe ningun empleado con este
dni. ');
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('Código: ' || SQLCODE);

```

```

END CAMBIAR_SALARIOS;
/

--Cambiar salario a jugadores de un equipo
CREATE OR REPLACE PROCEDURE CAMBIAR_SALARIOS_EQUIPO(nombreEquipo
IN VARCHAR2 , PORCENTAJE IN NUMBER)
IS
    EQUIPO_INEXISTENTE EXCEPTION;
    CURSOR C_JUG IS
        SELECT DNI_PERSONA
        FROM JUGADOR
        WHERE NOMBRE_EQUIPO = nombreEquipo
        FOR UPDATE;
BEGIN
    IF (NOT EQUIPO_EXISTENTE(nombreEquipo)) THEN
        RAISE EQUIPO_INEXISTENTE;
    END IF;
    DBMS_OUTPUT.PUT_LINE('A los jugadores de ' || nombreEquipo
|| ' se le cambiaran los sueldos en un : ' || PORCENTAJE || '%');
    FOR regJugador IN C_JUG LOOP
        CAMBIAR_SALARIOS(regJugador.DNI_PERSONA , PORCENTAJE);
    END LOOP;

EXCEPTION
    WHEN EQUIPO_INEXISTENTE THEN
        DBMS_OUTPUT.PUT_LINE('El equipo no existe');
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('Código: ' || SQLCODE);
END CAMBIAR_SALARIOS_EQUIPO;
/

--Procedimiento que lista los partidos de una fecha con el campo
y la direccion del campo
CREATE OR REPLACE PROCEDURE LISTAR_PARTIDOS_FECHA(fecha IN DATE)
IS
    comprobacion NUMBER;
    NO_HAY_PARTIDOS EXCEPTION;
    CURSOR C_JUEGA IS
        SELECT
NOMBRE_EQUIPO_LOCAL,NOMBRE_EQUIPO_VISITANTE,JUEGA.NOMBRE_CAMPO,D
IRECCION
        FROM JUEGA,CAMPO
        WHERE FECHA_PARTIDO = fecha AND JUEGA.NOMBRE_CAMPO =
CAMPO.NOMBRE_CAMPO;
BEGIN
    SELECT count(*) INTO comprobacion
    FROM JUEGA
    WHERE FECHA_PARTIDO = fecha;
    IF (comprobacion = 0) THEN
        RAISE NO_HAY_PARTIDOS;
    END IF;
    DBMS_OUTPUT.PUT_LINE('En la fecha ' || fecha || ' hay los
siguientes partidos :');
    DBMS_OUTPUT.NEW_LINE;
    FOR regPartido IN C_JUEGA LOOP

```

```

        DBMS_OUTPUT.PUT_LINE(regPartido.NOMBRE_EQUIPO_LOCAL || '
vs ' || regPartido.NOMBRE_EQUIPO_VISITANTE ||
        ' se jugará en el estadio ' || regPartido.NOMBRE_CAMPO
|| ' Dirección: ' || regPartido.DIRECCION);
    END LOOP;
EXCEPTION
    WHEN NO_HAY_PARTIDOS THEN
        DBMS_OUTPUT.PUT_LINE('No hay partidos en esa fecha');
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('Código: ' || SQLCODE);
END LISTAR_PARTIDOS_FECHA;
/

```

--Procedimiento Para insertar a los jugadores de un equipo en Aficionado_VIP (Primero en aficionado si no están)

```

CREATE OR REPLACE PROCEDURE
CREAR_AFICIONADOS_VIP_EQUIPO(nombreEquipo IN VARCHAR2)
IS
    EQUIPO_INEXISTENTE EXCEPTION;
    contAficionado NUMBER;
    contVIP NUMBER;

    CURSOR C_JUGADORES IS
        SELECT JUGADOR.DNI_PERSONA
        FROM EQUIPO, JUGADOR
        WHERE EQUIPO.NOMBRE_EQUIPO = nombreEquipo AND
EQUIPO.NOMBRE_EQUIPO = JUGADOR.NOMBRE_EQUIPO;

    CURSOR C_AFICIONADOS IS
        SELECT AFICIONADO.DNI_PERSONA
        FROM AFICIONADO;

    CURSOR C_AFICIONADOS_VIP IS
        SELECT E_DNI_VIP
        FROM AFICIONADO_VIP;

BEGIN
    IF NOT EQUIPO_EXISTENTE(nombreEquipo) THEN
        RAISE EQUIPO_INEXISTENTE;
    END IF;

    FOR regJugador IN C_JUGADORES LOOP
        contAficionado := 0;

        FOR regAficionado IN C_AFICIONADOS LOOP
            IF regJugador.DNI_PERSONA =
regAficionado.DNI_PERSONA THEN
                contAficionado := 1;
                EXIT;
            END IF;
        END LOOP;

        IF contAficionado = 0 THEN
            INSERT INTO AFICIONADO(DNI_PERSONA, TELEFONO)
            VALUES(regJugador.DNI_PERSONA, 000000000);
        END IF;
    END LOOP;

```

```

        contVIP := 0;

        FOR regAficionadoVip IN C_AFICIONADOS_VIP LOOP
            IF regJugador.DNI_PERSONA =
regAficionadoVip.E_DNI_VIP THEN
                contVIP := 1;
                EXIT;
            END IF;
        END LOOP;

        IF contVIP = 0 THEN
            INSERT INTO AFICIONADO_VIP(A_DNI_VIP, E_DNI_VIP,
DESCUENTO_VIP)
                VALUES (regJugador.DNI_PERSONA,
regJugador.DNI_PERSONA, 20);
            END IF;
        END LOOP;

        DBMS_OUTPUT.PUT_LINE('Los jugadores del equipo ' ||
nombreEquipo || ' fueron introducidos como aficionados y
aficionados VIP.');
```

EXCEPTION

```

    WHEN EQUIPO_INEXISTENTE THEN
        DBMS_OUTPUT.PUT_LINE('El equipo no existe');
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('Error: ' || SQLERRM);
END CREAR_AFICIONADOS_VIP_EQUIPO;
/
```

SENTENCIAS DE DEFINICIÓN DE BLOQUES DE PRUEBAS

```

/*****
/* 8.- Bloque para prueba de Procedimientos y Funciones */
*****/

SET SERVEROUTPUT ON
DECLARE
    numJugadores NUMBER;
    VoF boolean;
BEGIN
    DBMS_OUTPUT.NEW_LINE;

    BEGIN
-- FUNCTION EQUIPO_EXISTENTE
        DBMS_OUTPUT.PUT_LINE('=====>FUNCION: EQUIPO_EXISTENTE');
        Vof := EQUIPO_EXISTENTE('España');
        DBMS_OUTPUT.PUT_LINE('=====>FIN PROCEDIMIENTO:
EQUIPO_EXISTENTE');
        DBMS_OUTPUT.NEW_LINE;
    EXCEPTION
        WHEN OTHERS THEN
            DBMS_OUTPUT.PUT_LINE('[EXCEPCIÓN]');
            DBMS_OUTPUT.PUT_LINE('[Código]: ' || SQLCODE);
            DBMS_OUTPUT.PUT_LINE('[Mensaje]: ' || SQLERRM);
    END;

    BEGIN
-- PROCEDIMIENTO LISTA_JUGADORES_EQUIPO
        DBMS_OUTPUT.PUT_LINE('=====>INICIO PROCEDIMIENTO:
LISTA_JUGADORES_EQUIPO');
        DBMS_OUTPUT.NEW_LINE;
        LISTA_JUGADORES_EQUIPO('España', numJugadores);
        DBMS_OUTPUT.NEW_LINE;
        DBMS_OUTPUT.PUT_LINE('El equipo tiene ' || numJugadores ||
' jugadores.' );
        DBMS_OUTPUT.PUT_LINE('=====>FIN PROCEDMIENTO:
LISTA_JUGADORES_EQUIPO');
        DBMS_OUTPUT.NEW_LINE;
    EXCEPTION
        WHEN OTHERS THEN
            DBMS_OUTPUT.PUT_LINE('[EXCEPCIÓN]');
            DBMS_OUTPUT.PUT_LINE('[Código]: ' || SQLCODE);
            DBMS_OUTPUT.PUT_LINE('[Mensaje]: ' || SQLERRM);
    END;

    BEGIN
-- PROCEDIMIENTO CAMBIAR_SALARIOS
        DBMS_OUTPUT.PUT_LINE('=====>INICIO PROCEDIMIENTO:
CAMBIAR_SALARIOS');
        DBMS_OUTPUT.NEW_LINE;
        CAMBIAR_SALARIOS('53798504D', 5);
        DBMS_OUTPUT.NEW_LINE;
    END;
END;
```

```

        DBMS_OUTPUT.PUT_LINE('=====>FIN PROCEDIMIENTO:
CAMBIAR_SALARIOS');
        DBMS_OUTPUT.NEW_LINE;
    EXCEPTION
        WHEN OTHERS THEN
            DBMS_OUTPUT.PUT_LINE('[EXCEPCIÓN]');
            DBMS_OUTPUT.PUT_LINE('[Código]: ' || SQLCODE);
            DBMS_OUTPUT.PUT_LINE('[Mensaje]: ' || SQLERRM);
    END;

BEGIN
-- PROCEDIMIENTO CAMBIAR_SALARIOS_EQUIPO
    DBMS_OUTPUT.PUT_LINE('=====>INICIO PROCEDIMIENTO:
CAMBIAR_SALARIOS_EQUIPO');
    DBMS_OUTPUT.NEW_LINE;
    CAMBIAR_SALARIOS_EQUIPO('España' , 5);
    DBMS_OUTPUT.NEW_LINE;
    DBMS_OUTPUT.PUT_LINE('=====>FIN PROCEDIMIENTO:
CAMBIAR_SALARIOS_EQUIPO');
    DBMS_OUTPUT.NEW_LINE;
    EXCEPTION
        WHEN OTHERS THEN
            DBMS_OUTPUT.PUT_LINE('[EXCEPCIÓN]');
            DBMS_OUTPUT.PUT_LINE('[Código]: ' || SQLCODE);
            DBMS_OUTPUT.PUT_LINE('[Mensaje]: ' || SQLERRM);
    END;

BEGIN
-- PROCEDIMIENTO LISTAR_PARTIDOS_FECHA
    DBMS_OUTPUT.PUT_LINE('=====>INICIO PROCEDIMIENTO:
LISTAR_PARTIDOS_FECHA');
    DBMS_OUTPUT.NEW_LINE;
    LISTAR_PARTIDOS_FECHA('21-03-2018');
    DBMS_OUTPUT.NEW_LINE;
    DBMS_OUTPUT.PUT_LINE('=====>FIN PROCEDIMIENTO:
LISTAR_PARTIDOS_FECHA');
    DBMS_OUTPUT.NEW_LINE;
    EXCEPTION
        WHEN OTHERS THEN
            DBMS_OUTPUT.PUT_LINE('[EXCEPCIÓN]');
            DBMS_OUTPUT.PUT_LINE('[Código]: ' || SQLCODE);
            DBMS_OUTPUT.PUT_LINE('[Mensaje]: ' || SQLERRM);
    END;

BEGIN
-- PROCEDIMIENTO CREAR_AFICIONADOS_VIP_EQUIPO
    DBMS_OUTPUT.PUT_LINE('=====>INICIO PROCEDIMIENTO:
CREAR_AFICIONADOS_VIP_EQUIPO');
    DBMS_OUTPUT.NEW_LINE;
    CREAR_AFICIONADOS_VIP_EQUIPO('España');
    DBMS_OUTPUT.NEW_LINE;
    DBMS_OUTPUT.PUT_LINE('=====>FIN PROCEDIMIENTO:
CREAR_AFICIONADOS_VIP_EQUIPO');
    DBMS_OUTPUT.NEW_LINE;
    EXCEPTION
        WHEN OTHERS THEN

```

```
        DBMS_OUTPUT.PUT_LINE('[EXCEPCIÓN]');  
        DBMS_OUTPUT.PUT_LINE('[Código]: ' || SQLCODE);  
        DBMS_OUTPUT.PUT_LINE('[Mensaje]: ' || SQLERRM);  
    END;  
END;
```

HOJA DE FIRMAS

DNI: 53489444T

NOMBRE: ARIAS CAMPAÑA, PABLO

DNI: 53819272P

NOMBRE: ÁLVAREZ IGLESIAS, DAVID

DNI: 53819578S

NOMBRE: MARTÍNEZ LÓPEZ, DIEGO

DNI: 53798504D

NOMBRE: OLIVEIRA SUÁREZ, RUBÉN