

Curso: SOF-S-MA-6-1



Docente: Ing. Gary Reyes

Proyecto Final

Grupo F

Edison Eduardo Vera Romero

Arellano Chiliquinga Jeremy Anthony

Chicaiza Fienco Lizbeth Dayana

Monserate Totoy Jair Michael

Theurer Iñiguez Erick Arturo

Vera García Odalis Del Carmen

Fecha: 12 de marzo del 2023

PORCENTAJES DE PARTICIPACIÓN INDIVIDUAL

	NOMBRES	REUNION 1	PROMEDIO
Estudiante 1 (Líder)	Vera Romero Edison Eduardo	100	100
Estudiante 2	Arellano Chiliquinga Jeremy Anthony	100	100
Estudiante 3	Chicaiza Fienco Lizbeth Dayana	100	100
Estudiante 4	Monserate Totoy Jair Michael	100	100
Estudiante 5	Theurer Iñiguez Erick Arturo	100	100
Estudiante 6	Vera García Odalis Del Carmen	100	100

Link del video : <https://youtu.be/eK4i9ofHtPQ>

Código Fuente:

```
--Punto numero 1

--Procedimiento para agregar un nuevo proyecto:
/*
Nombre: agregar_proyecto
Parámetros:
- p_nombre (IN): Nombre del proyecto (VARCHAR2)
- p_descripcion (IN): Descripción del proyecto (VARCHAR2)
- p_objetivo (IN): Objetivo de financiamiento del proyecto (NUMBER)
Descripción: Agrega un nuevo proyecto a la tabla de proyectos.
*/
CREATE OR REPLACE PROCEDURE agregar_proyecto(p_nombre IN VARCHAR2, p_descripcion IN VARCHAR2, p_objetivo IN NUMBER) AS
BEGIN
    INSERT INTO proyectos (nombre, descripcion, objetivo, total_recaudado, estado) VALUES (p_nombre, p_descripcion, p_objetivo, 0, 'activo');
    COMMIT;
END;

/*
Nombre: desactivar_proyecto
Parámetros:
- p_id (IN): ID del proyecto a desactivar (NUMBER)
Descripción: Actualiza el estado de un proyecto a 'inactivo' en la tabla de proyectos.
*/
CREATE OR REPLACE PROCEDURE desactivar_proyecto(p_id IN NUMBER) AS
BEGIN
    UPDATE proyectos SET estado = 'inactivo' WHERE proyecto_id = p_id;
    COMMIT;
END;

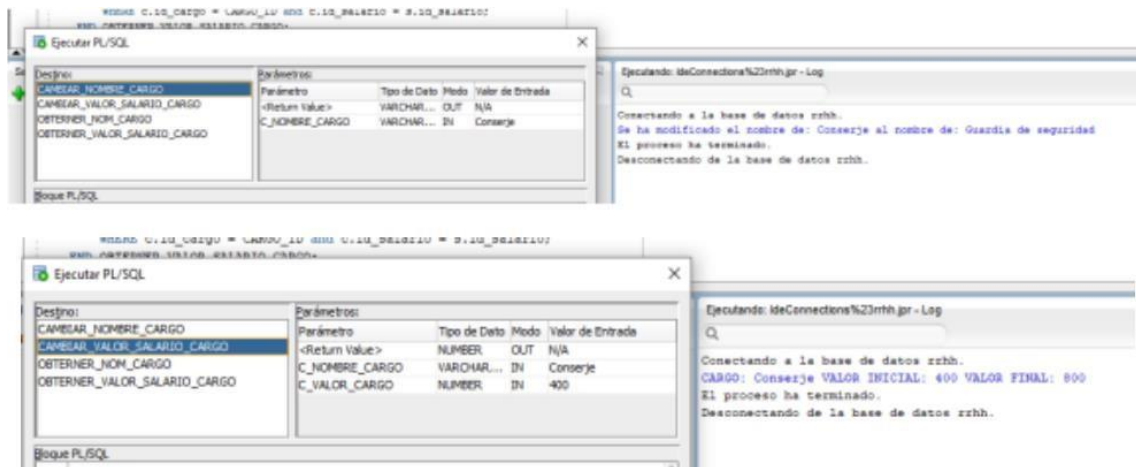
/*
Nombre: obtener_informacion_proyecto
Parámetros:
```

```
--Funciones

/*
Nombre: obtener_total_proyectos_activos
Parámetros: Ninguno
Descripción: Devuelve el número total de proyectos que tienen el estado 'activo' en la tabla de proyectos.
*/
CREATE OR REPLACE FUNCTION obtener_total_proyectos_activos RETURN NUMBER AS
    total_proyectos NUMBER;
BEGIN
    SELECT COUNT(*) INTO total_proyectos FROM proyectos WHERE estado = 'activo';
    RETURN total_proyectos;
END;

/*
Nombre: obtener_total_recaudado_proyecto
Parámetros:
- p_id (IN): ID del proyecto a obtener el monto total recaudado (NUMBER)
Descripción: Devuelve el monto total recaudado por un proyecto específico en la tabla de proyectos.
*/
CREATE OR REPLACE FUNCTION obtener_total_recaudado_proyecto(p_id IN NUMBER) RETURN NUMBER AS
    total_recaudado NUMBER;
BEGIN
    SELECT total_recaudado INTO total_recaudado FROM proyectos WHERE proyecto_id = p_id;
    RETURN total_recaudado;
END;
```

Compilado:



Código Fuente:

```
--Paquete para manejar las donaciones
/*
Este paquete se encarga de manejar las donaciones realizadas a los proyectos.
Contiene dos procedimientos y dos funciones que utilizan un parámetro tipo cursor.
*/

CREATE OR REPLACE PACKAGE manejo_donaciones AS

    TYPE cur_t IS REF CURSOR;

    PROCEDURE agregar_donacion(p_id_proyecto IN NUMBER, p_cantidad IN NUMBER);
    PROCEDURE eliminar_donacion(p_id_donacion IN NUMBER);
    FUNCTION obtener_total_donado_proyecto(p_id_proyecto IN NUMBER) RETURN NUMBER;
    FUNCTION obtener_donaciones_proyecto(p_id_proyecto IN NUMBER) RETURN cur_t;

END manejo_donaciones;

--El primer procedimiento, agregar_donacion,
--agrega una nueva donación a la tabla de donaciones. Toma como entrada el ID del proyecto y la cantidad donada.

PROCEDURE agregar_donacion(p_id_proyecto IN NUMBER, p_cantidad IN NUMBER) IS
BEGIN
    UPDATE proyectos
    SET total_donaciones = total_donaciones + p_cantidad
    WHERE id_proyecto = p_id_proyecto;

    COMMIT;
END agregar_donacion;
```

```

    COMMIT;
END eliminar_donacion;

--La primera función, obtener_total_donado_proyecto,
--devuelve el monto total donado a un proyecto específico en la tabla de donaciones. Toma como entrada el ID del proyecto.
FUNCTION obtener_total_donado_proyecto(p_id_proyecto IN NUMBER) RETURN NUMBER IS
    v_total_donado NUMBER;
BEGIN
    SELECT total_donaciones
    INTO v_total_donado
    FROM proyectos
    WHERE id_proyecto = p_id_proyecto;

    RETURN v_total_donado;
END obtener_total_donado_proyecto;

--La segunda función, obtener_donaciones_proyecto, devuelve un cursor que contiene todas las donaciones
--realizadas a un proyecto específico en la tabla de donaciones. Toma como entrada el ID del proyecto.
FUNCTION obtener_donaciones_proyecto(p_id_proyecto IN NUMBER) RETURN cur_t IS
    v_cursor cur_t;
BEGIN
    OPEN v_cursor FOR
        SELECT *
        FROM donaciones
        WHERE id_proyecto = p_id_proyecto;

    RETURN v_cursor;
END obtener_donaciones_proyecto;

```

Compilado:

1	4800
2	5100
3	34800
4	18000
5	14400
6	10800

Código Fuente:

```
--Paquete para manejar los usuarios

/*
Este paquete se encarga de manejar los usuarios que se registran en la plataforma de crowdfunding.
Contiene dos procedimientos y dos funciones que utilizan un parámetro tipo cursor.
*/

CREATE OR REPLACE PACKAGE manejo_usuarios AS

    TYPE cur_t IS REF CURSOR;

    PROCEDURE registrar_usuario(p_nombre IN VARCHAR2, p_correo IN VARCHAR2, p_contrasena IN VARCHAR2);
    PROCEDURE eliminar_usuario(p_id_usuario IN NUMBER);
    FUNCTION obtener_informacion_usuario(p_id_usuario IN NUMBER) RETURN cur_t;
    FUNCTION validar_credenciales(p_correo IN VARCHAR2, p_contrasena IN VARCHAR2) RETURN NUMBER;

END manejo_usuarios;

--El primer procedimiento, registrar_usuario, agrega un nuevo usuario a la tabla de usuarios.
--Toma como entrada el nombre, correo y contraseña del usuario.

PROCEDURE registrar_usuario(p_nombre IN VARCHAR2, p_correo IN VARCHAR2, p_contrasena IN VARCHAR2) IS
BEGIN
    INSERT INTO usuarios(nombre, correo, contrasena)
    VALUES(p_nombre, p_correo, p_contrasena);

    COMMIT;
END registrar_usuario;

--El segundo procedimiento, eliminar_usuario, elimina un usuario de la tabla de usuarios. Toma como entrada el ID del usuario.

PROCEDURE eliminar_usuario(p_id_usuario IN NUMBER) IS
BEGIN
    DELETE FROM usuarios
```

```
FUNCTION obtener_informacion_usuario(p_id_usuario IN NUMBER) RETURN cur_t IS
    v_cursor cur_t;
BEGIN
    OPEN v_cursor FOR
        SELECT *
        FROM usuarios
        WHERE id_usuario = p_id_usuario;

    RETURN v_cursor;
END obtener_informacion_usuario;

--Esta función toma como entrada el correo y la contraseña de un usuario, y devuelve el ID del usuario si las credenciales son válidas, o 0 si no lo son.

FUNCTION validar_credenciales(p_correo IN VARCHAR2, p_contrasena IN VARCHAR2) RETURN NUMBER IS
    v_id_usuario NUMBER;
BEGIN
    SELECT id_usuario INTO v_id_usuario
    FROM usuarios
    WHERE correo = p_correo AND contrasena = p_contrasena;

    RETURN v_id_usuario;
EXCEPTION
    WHEN NO_DATA_FOUND THEN
        RETURN 0;
END validar_credenciales;
```

Compilado:

Salida de Script x					
Tarea terminada en 0,531 segundos					
ResultSet #1					
EMPLOYEE_ID	FIRST_NAME	LAST_NAME	SALARY	EMAIL	PHONE_NUMBER
174	Ellen	Abel	11000	EABEL	011.44.1644.429267
166	Sundar	Ande	6400	SANDE	011.44.1346.629268
130	Mozhe	Atkinson	2800	MATKINSO	650.124.6234
105	David	Austin	4800	DAUSTIN	590.423.4569
204	Hermann	Baer	10000	HBAER	515.123.8888
116	Shelli	Baida	2900	SBAIDA	515.127.4563
167	Amit	Banda	6200	ABANDA	011.44.1346.729268
172	Elizabeth	Bates	7300	EBATES	011.44.1343.529268
192	Sarah	Bell	4000	SBELL	650.501.1876
151	David	Bernstein	9500	DBERNSTE	011.44.1344.345268
129	Laura	Bissot	3300	LBISSOT	650.124.5234
EMPLOYEE_ID	FIRST_NAME	LAST_NAME	SALARY	EMAIL	PHONE_NUMBER
169	Harrison	Bloom	10000	HBLROOM	011.44.1343.829268
185	Alexis	Bull	4100	ABULL	650.509.2876
187	Anthony	Cabrio	3000	ACABRIO	650.509.4876
148	Gerald	Cambrault	11000	GCAMBRAU	011.44.1344.619268
154	Nanette	Cambrault	7500	NCAMBRAU	011.44.1344.987668
110	John	Chen	8200	JCHEN	515.124.4269
188	Kelly	Chung	3800	KCHUNG	650.505.1876