

UNIVERSIDAD AUTÓNOMA DE CHIHUAHUA

FACULTAD DE INGENIERÍA

BASES DE DATOS AVANZADAS



2.44 PROYECTO 1A. EVALUACIÓN

ALUMNOS:

368068— LUISA FERNANDA HERNÁNDEZ HERNÁNDEZ

367760 — INGRID YULIANA PÉREZ RODRÍGUEZ

367785— MAURICIO ELÍAS NAVARRETE FLORES

PROFESOR:

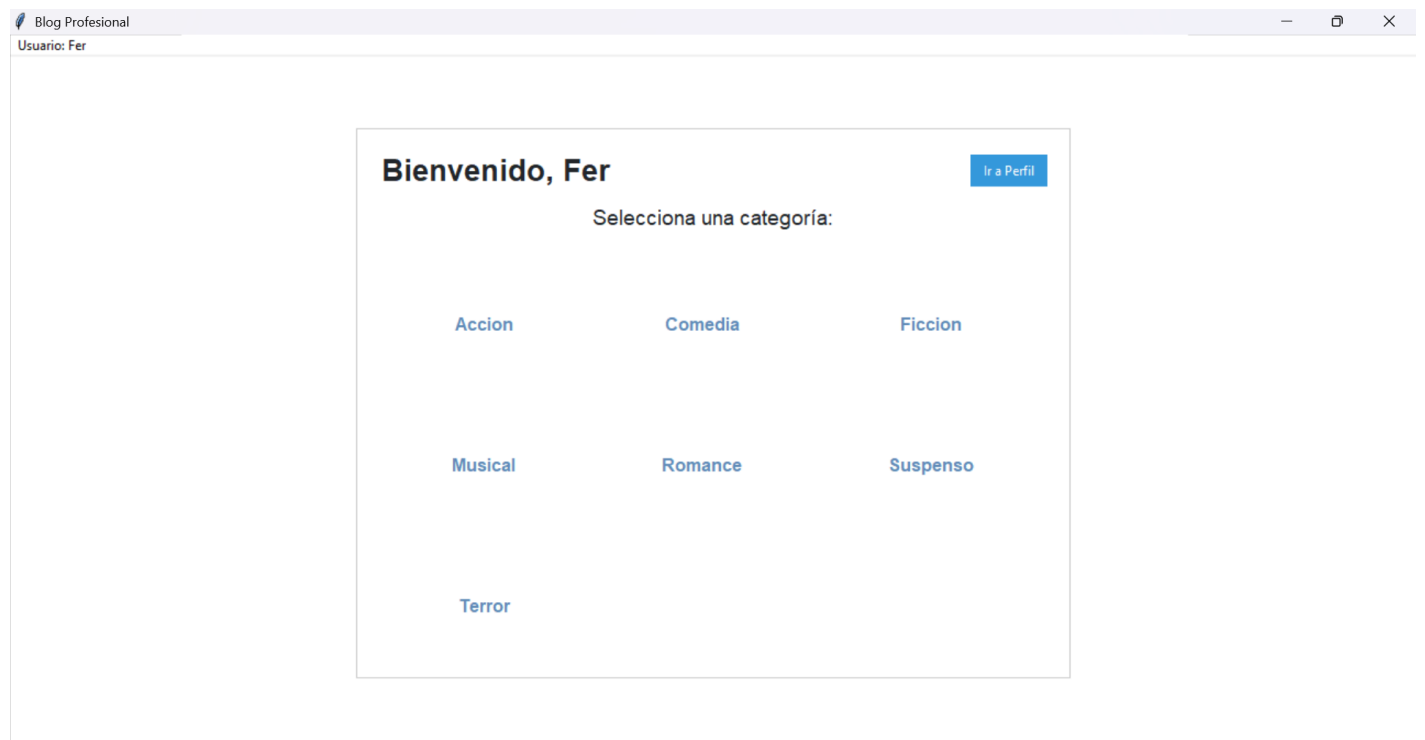
JOSÉ SAUL DE LIRA MIRAMONTES

CHIHUAHUA, CHIHUAHUA A 30 DE SEPTIEMBRE DEL 2025.

--Obtener todas las categorías

```
CREATE OR REPLACE FUNCTION get_all_categories
RETURN SYS_REFCURSOR
IS
    v_cursor SYS_REFCURSOR;
BEGIN
    OPEN v_cursor FOR
        SELECT cat_name
        FROM Categories
        ORDER BY cat_name;

    RETURN v_cursor;
END;
```



--Tags de un artículo

```
CREATE OR REPLACE FUNCTION get_tags_by_article(p_article_id IN INTEGER)
RETURN SYS_REFCURSOR
```

IS

```
v_cursor SYS_REFCURSOR;
```

BEGIN

```
OPEN v_cursor FOR
```

```
SELECT tag_name
```

```
FROM Article_tag
```

```
WHERE article_id = p_article_id
```

```
ORDER BY tag_name;
```

```
RETURN v_cursor;
```

END;

Blog Profesional

Usuario: Fer

F1: The Movie

Autor: Fer

Contenido:

Un piloto de Fórmula 1 que sufrió un terrible accidente años atrás, regresa a la pista para darle una última oportunidad a un equipo de leyenda pero en crisis, enfrentándose a un novato compañero de equipo que es una superestrella en ascenso.

Categorías:

[Acción](#) [Ficción](#)

Tags:

[Crecimiento Personal](#) [Formula 1](#) [Trabajo en Equipo](#)

Comentarios:

No hay comentarios aún.

Agregar comentario:

[Comentar](#)

[Volver](#)

--users

--Verificar contraseña y usuario

```
CREATE OR REPLACE FUNCTION get_user_name(p_email IN VARCHAR2, p_pass IN VARCHAR2)
```

```
RETURN VARCHAR2
```

IS

```
v_name_user VARCHAR2(30);

BEGIN

    SELECT NAME_USER

    INTO v_name_user

    FROM T_USERS

    WHERE EMAIL = p_email AND PASS = p_pass;

    RETURN v_name_user;

EXCEPTION

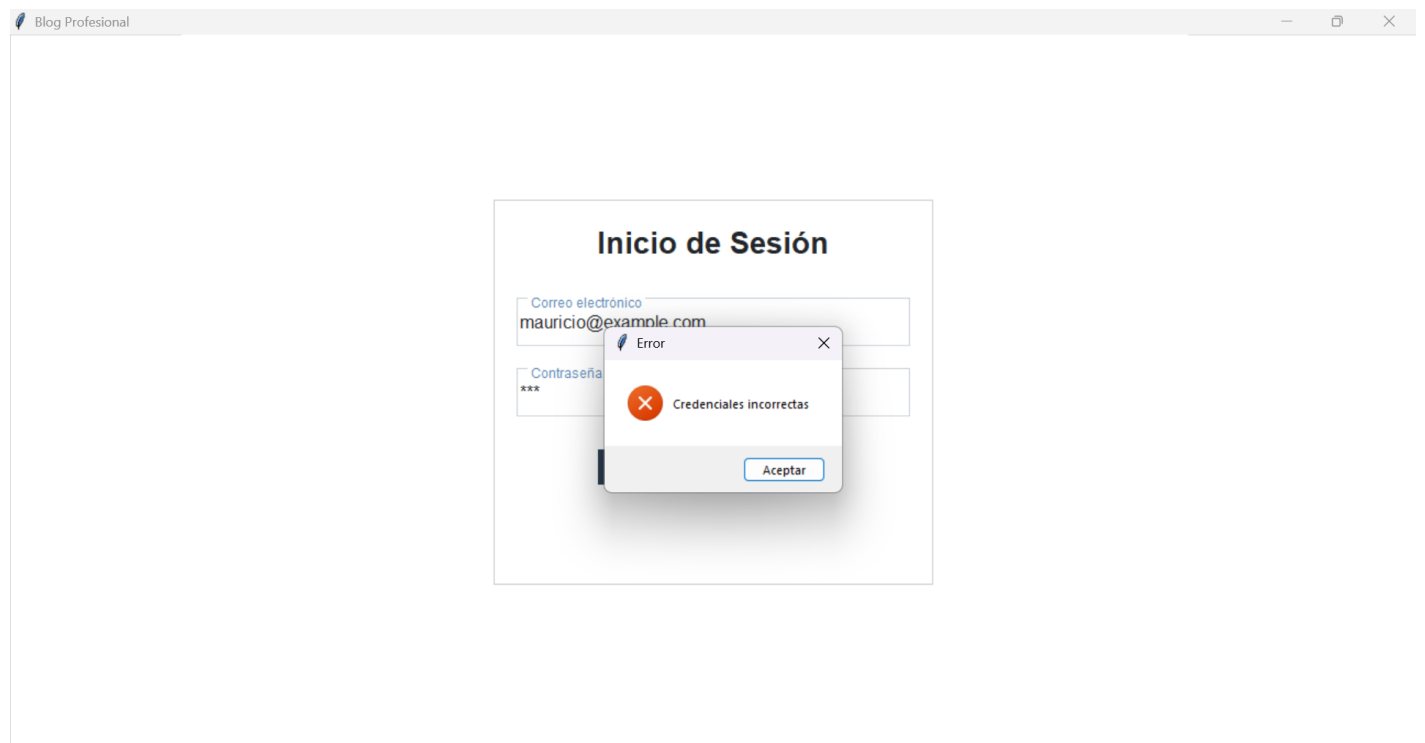
    WHEN NO_DATA_FOUND THEN

        RETURN NULL;

    WHEN OTHERS THEN

        RETURN NULL;

END;
```



--Función de apoyo para cambiar contraseña

create or replace FUNCTION get_user_name2(p_email IN VARCHAR2)

```

RETURN VARCHAR2

IS

    v_name_user VARCHAR2(30);

BEGIN

    SELECT NAME_USER

    INTO v_name_user

    FROM T_USERS

    WHERE EMAIL = p_email;


    RETURN v_name_user;

EXCEPTION

    WHEN NO_DATA_FOUND THEN

        RETURN NULL;

    WHEN OTHERS THEN

        RETURN NULL;

END;

```

--Agregar usuario

```

CREATE OR REPLACE PROCEDURE add_user (

    p_name_user IN VARCHAR2,

    p_email    IN VARCHAR2,

    p_pass in VARCHAR

) AS

BEGIN

    INSERT INTO T_Users (name_user, email, pass )

    VALUES (p_name_user, p_email, p_pass);

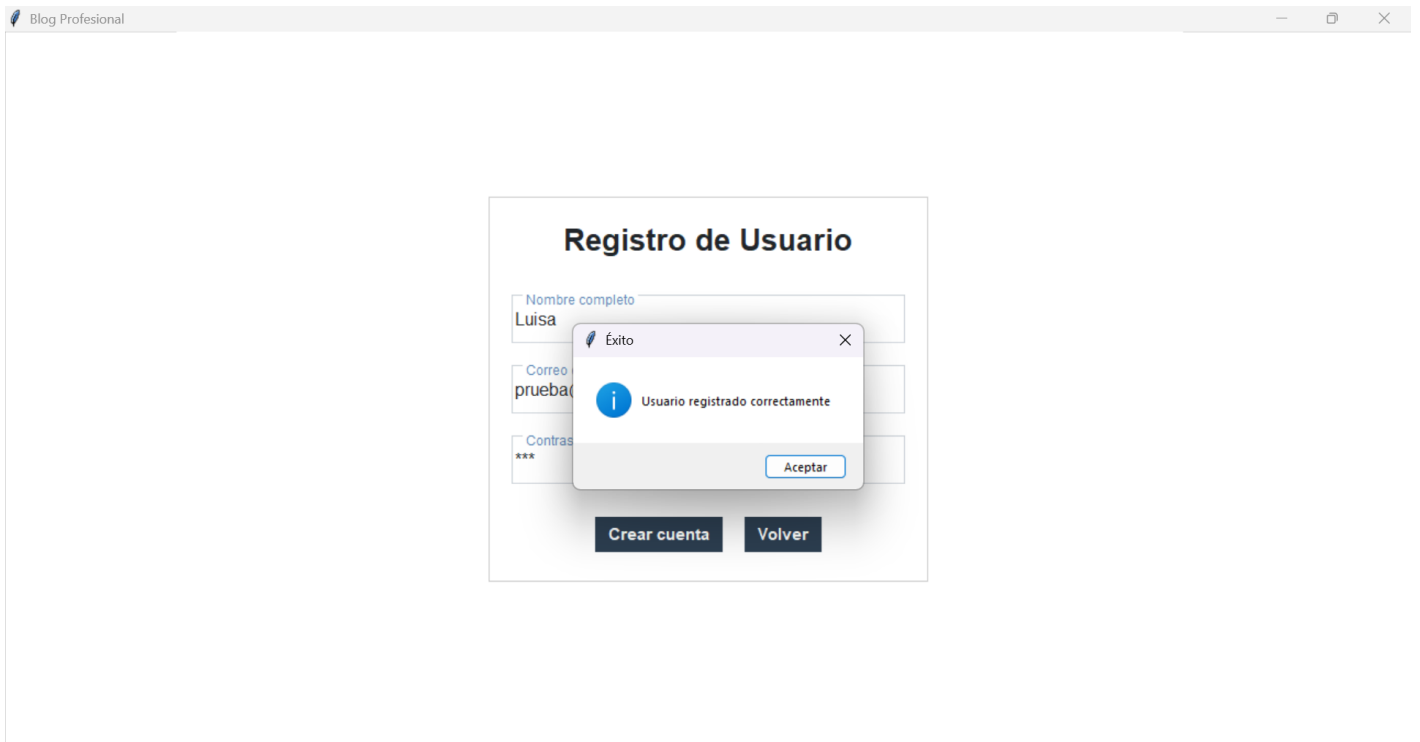
EXCEPTION

    WHEN DUP_VAL_ON_INDEX THEN

```

```
RAISE_APPLICATION_ERROR(-20101, 'El usuario ya existe.');
```

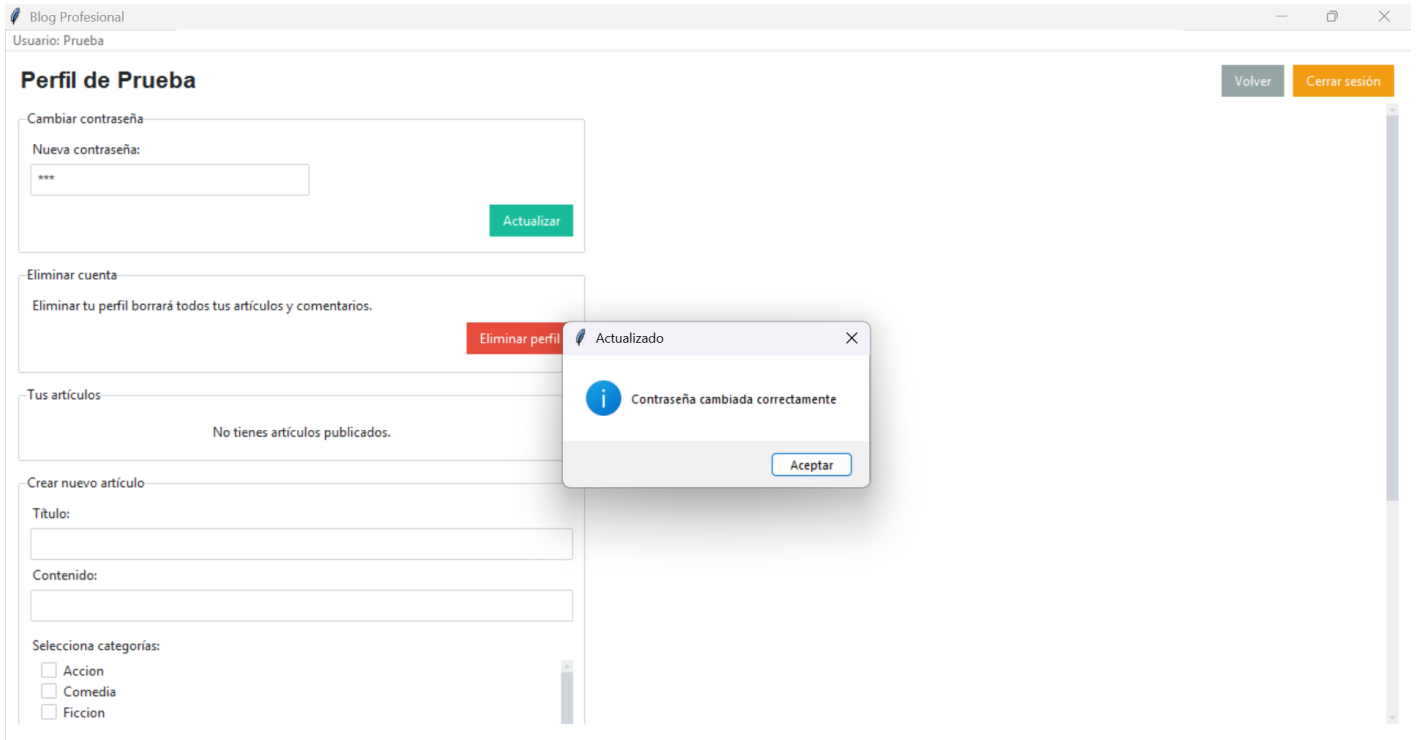
```
END;
```



--*Actualizar contraseña*

```
CREATE OR REPLACE PROCEDURE update_user_password (  
    p_name_user IN VARCHAR2,  
    p_pass IN VARCHAR2  
) AS  
BEGIN  
    UPDATE T_Users  
    SET pass = p_pass  
    WHERE name_user = p_name_user;  
  
    IF SQL%ROWCOUNT = 0 THEN  
        RAISE_APPLICATION_ERROR(-20104, 'Usuario no encontrado.');
```

```
    END IF;  
END;
```



--Obtener email

CREATE OR REPLACE FUNCTION get_user_email(p_name_user IN VARCHAR2)

RETURN VARCHAR2

IS

 v_email VARCHAR2(30);

BEGIN

 SELECT email

 INTO v_email

 FROM T_Users

 WHERE name_user = p_name_user;

 RETURN v_email;

EXCEPTION

 WHEN NO_DATA_FOUND THEN

 RETURN NULL;

 WHEN OTHERS THEN

```
        RETURN NULL;

END;
```

--Eliminar usuario

```
CREATE OR REPLACE PROCEDURE delete_user (

    p_name_user IN VARCHAR2

) AS

    v_user_exists NUMBER;

BEGIN

    SELECT COUNT(*) INTO v_user_exists

    FROM T_Users

    WHERE name_user = p_name_user;

    IF v_user_exists = 0 THEN

        RAISE_APPLICATION_ERROR(-20103, 'Usuario no encontrado.');
```

```
    END IF;

    DELETE FROM Comments

    WHERE article_id IN (

        SELECT article_id FROM Articles WHERE name_user = p_name_user

    );

    DELETE FROM Comments

    WHERE name_user = p_name_user;

    DELETE FROM Article_tag

    WHERE article_id IN (
```



```
SELECT article_id FROM Articles WHERE name_user = p_name_user  
);
```

```
DELETE FROM Article_Categorie  
WHERE article_id IN (  
    SELECT article_id FROM Articles WHERE name_user = p_name_user  
);
```

```
DELETE FROM Articles  
WHERE name_user = p_name_user;
```

```
DELETE FROM T_Users  
WHERE name_user = p_name_user;
```

```
COMMIT;
```

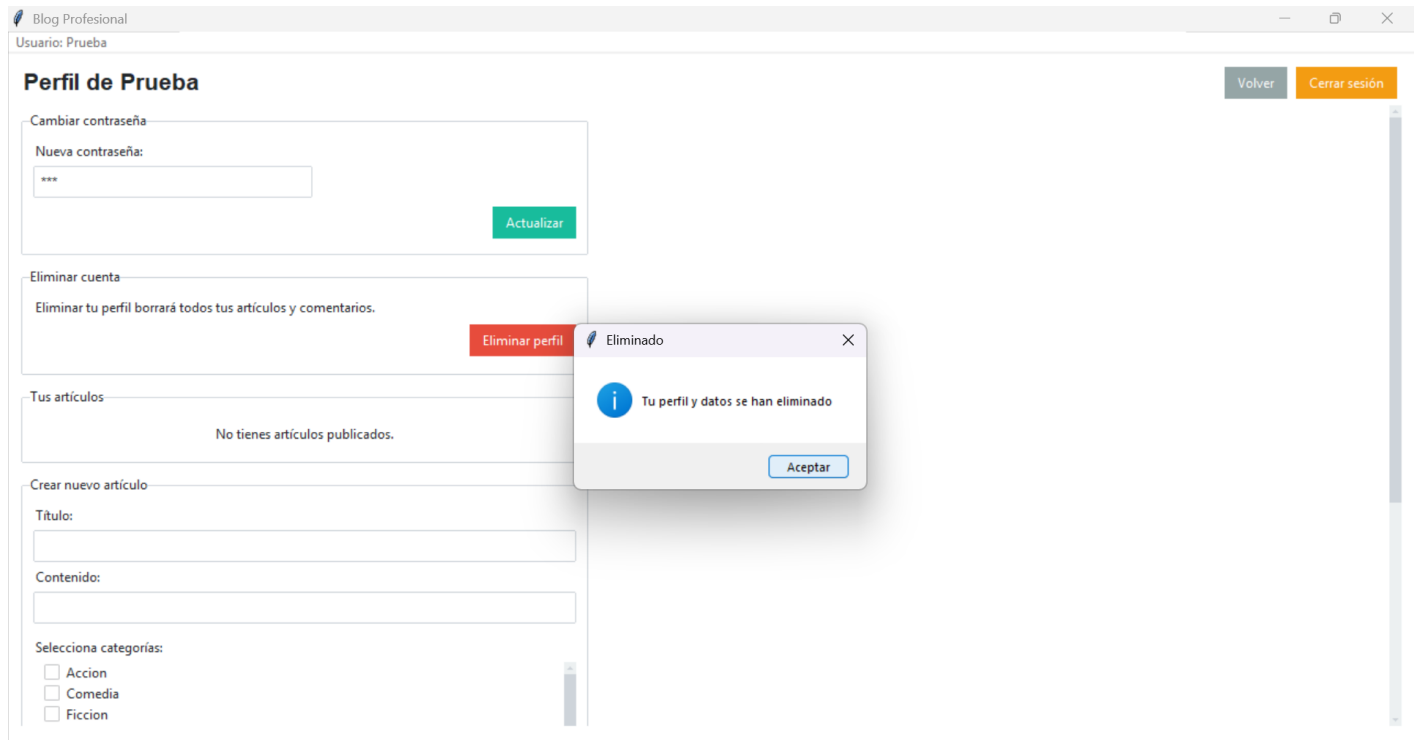
```
EXCEPTION
```

```
WHEN OTHERS THEN
```

```
    ROLLBACK;
```

```
    RAISE;
```

```
END;
```



-- *Obtener Artículo*

CREATE OR REPLACE FUNCTION get_article_by_id(p_article_id IN INTEGER)

RETURN SYS_REFCURSOR

IS

 v_cursor SYS_REFCURSOR;

BEGIN

 OPEN v_cursor FOR

 SELECT article_id, title, name_user, a_text

 FROM Articles

 WHERE article_id = p_article_id;

 RETURN v_cursor;

END;

-- *Artículos por categorías*

CREATE OR REPLACE FUNCTION get_articles_by_category(p_cat_name IN VARCHAR2)

```

RETURN SYS_REFCURSOR

IS

    v_cursor SYS_REFCURSOR;

BEGIN

    OPEN v_cursor FOR

        SELECT a.article_id, a.title, a.name_user, a.a_text

        FROM Articles a

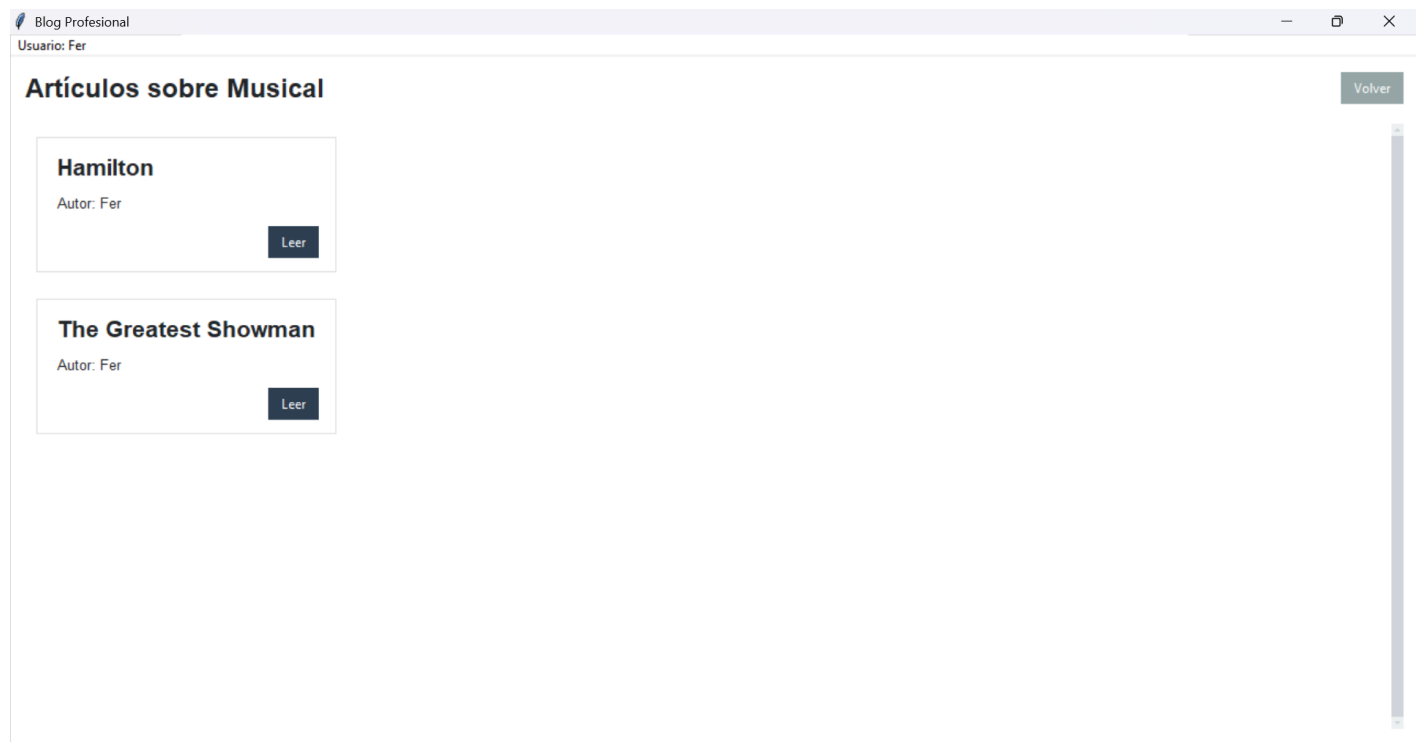
        JOIN Article_Categorie ac ON a.article_id = ac.article_id

        WHERE ac.cat_name = p_cat_name;

    RETURN v_cursor;

END;

```



-- *Artículos por tags*

```

CREATE OR REPLACE FUNCTION get_articles_by_tag(p_tag_name IN VARCHAR2)

RETURN SYS_REFCURSOR

IS

```

```

v_cursor SYS_REFCURSOR;

BEGIN

OPEN v_cursor FOR

    SELECT a.article_id, a.title, a.name_user, a.a_text

    FROM Articles a

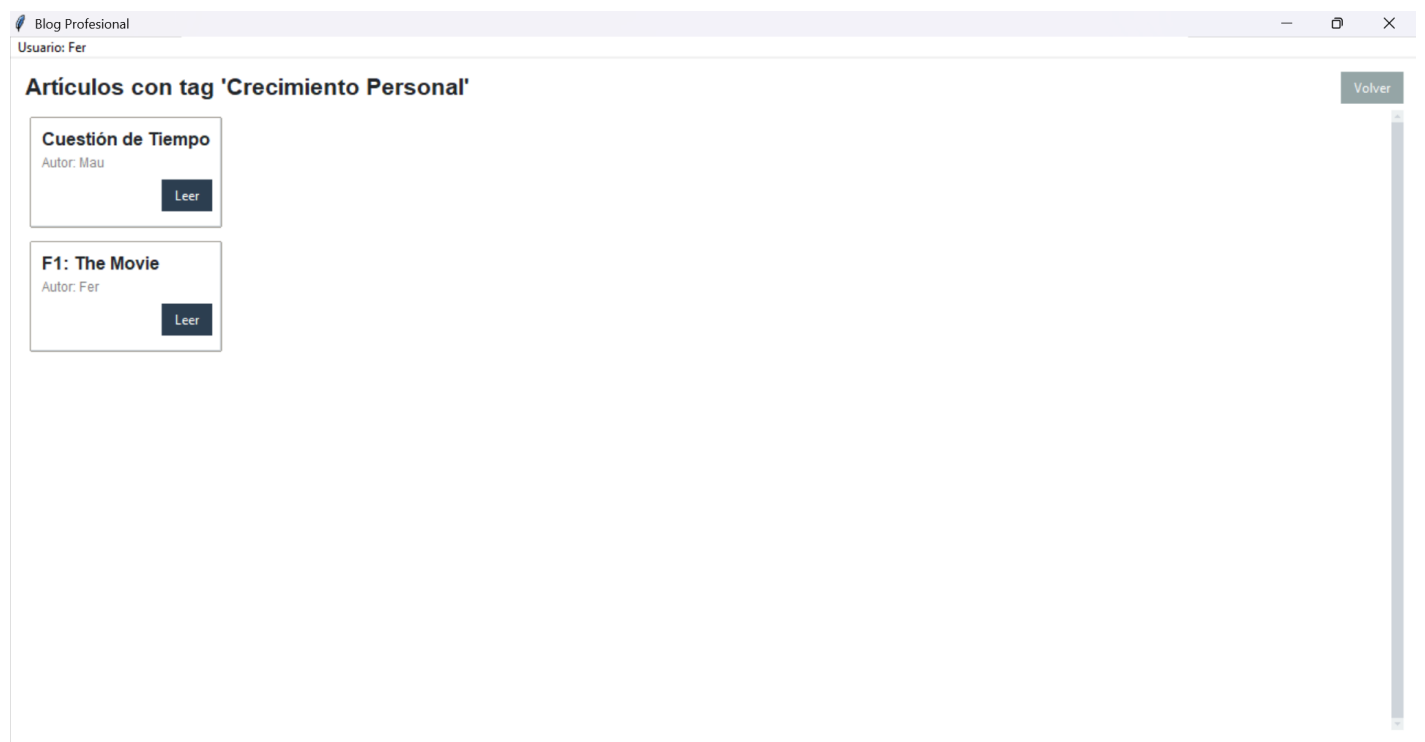
    JOIN Article_tag at ON a.article_id = at.article_id

    WHERE at.tag_name = p_tag_name;

RETURN v_cursor;

END;

```



--*Artículos por autor*

```

CREATE OR REPLACE FUNCTION get_articles_by_author(p_name_user IN VARCHAR2)

RETURN SYS_REFCURSOR

IS

    v_cursor SYS_REFCURSOR;

BEGIN

```

```
OPEN v_cursor FOR

    SELECT article_id, title, a_date, a_text

    FROM Articles

    WHERE name_user = p_name_user

    ORDER BY a_date DESC;

RETURN v_cursor;

END;
```

--Comentarios en artículos

```
CREATE OR REPLACE FUNCTION get_comments_by_article(p_article_id IN INTEGER)
RETURN SYS_REFCURSOR
IS
    v_cursor SYS_REFCURSOR;
BEGIN
    OPEN v_cursor FOR

        SELECT name_user, c_date, c_text

        FROM Comments

        WHERE article_id = p_article_id

        ORDER BY c_date;

    RETURN v_cursor;

END;
```

Blog Profesional

Usuario: Fer

Conjuro 4

Autor: Ingrid

Contenido:

El Conjuro 4" Será la cuarta entrega de la saga principal y probablemente la última película de la serie central, concluyendo la historia de Ed y Lorraine Warren. Patrick Wilson y Vera Farmiga volverán como los investigadores paranormales.

Categorías:

SuspenseTerror

Tags:

Conjuro

Comentarios:

Mau dice:

Comentario 1

Fer dice:

Comentario 2

Agregar comentario:

Volver

--Secuencia para article_id

```
CREATE SEQUENCE article_seq
```

```
START WITH 1
```

```
INCREMENT BY 1
```

```
NOCACHE
```

```
NOCYCLE;
```

--Agregar artículos

```
CREATE OR REPLACE FUNCTION add_article(
```

```
  p_title  IN VARCHAR2,
```

```
  p_name_user IN VARCHAR2,
```

```
  p_a_text  IN VARCHAR2
```

```
) RETURN NUMBER
```

```
IS
```

```
  v_article_id NUMBER;
```

```
BEGIN
```

```
  v_article_id := article_seq.NEXTVAL;
```

```

INSERT INTO Articles (article_id, title, name_user, a_text, a_date)

VALUES (v_article_id, p_title, p_name_user, p_a_text, SYSDATE);

RETURN v_article_id;

EXCEPTION

WHEN OTHERS THEN

RETURN NULL;

END;

```

--Agregar un article-tag

```

CREATE OR REPLACE FUNCTION add_article_tag(

    p_article_id IN INTEGER,

    p_tag_name  IN VARCHAR2

) RETURN VARCHAR2

IS

BEGIN

```

```

INSERT INTO Article_tag (article_id, tag_name)
VALUES (p_article_id, p_tag_name);

RETURN 'Relación tag agregada';

EXCEPTION

WHEN OTHERS THEN

    RETURN 'Error al agregar tag: ' || SQLERRM;

END;

```

--Agregar article_category

```

create or replace FUNCTION add_article_category(

    p_article_id IN INTEGER,

    p_cat_name  IN VARCHAR2

) RETURN VARCHAR2

IS

BEGIN

    INSERT INTO Article_Categorie (article_id, cat_name)

    VALUES (p_article_id, p_cat_name);

    RETURN 'Relación categoría agregada';

EXCEPTION

WHEN OTHERS THEN

    RETURN 'Error al agregar categoría: ' || SQLERRM;

END;

```

--Eliminar artículos

```

CREATE OR REPLACE PROCEDURE delete_article (

```



```
p_article_id IN NUMBER
) AS
BEGIN
    DELETE FROM Comments
    WHERE article_id = p_article_id;

    DELETE FROM Article_Categorie
    WHERE article_id = p_article_id;

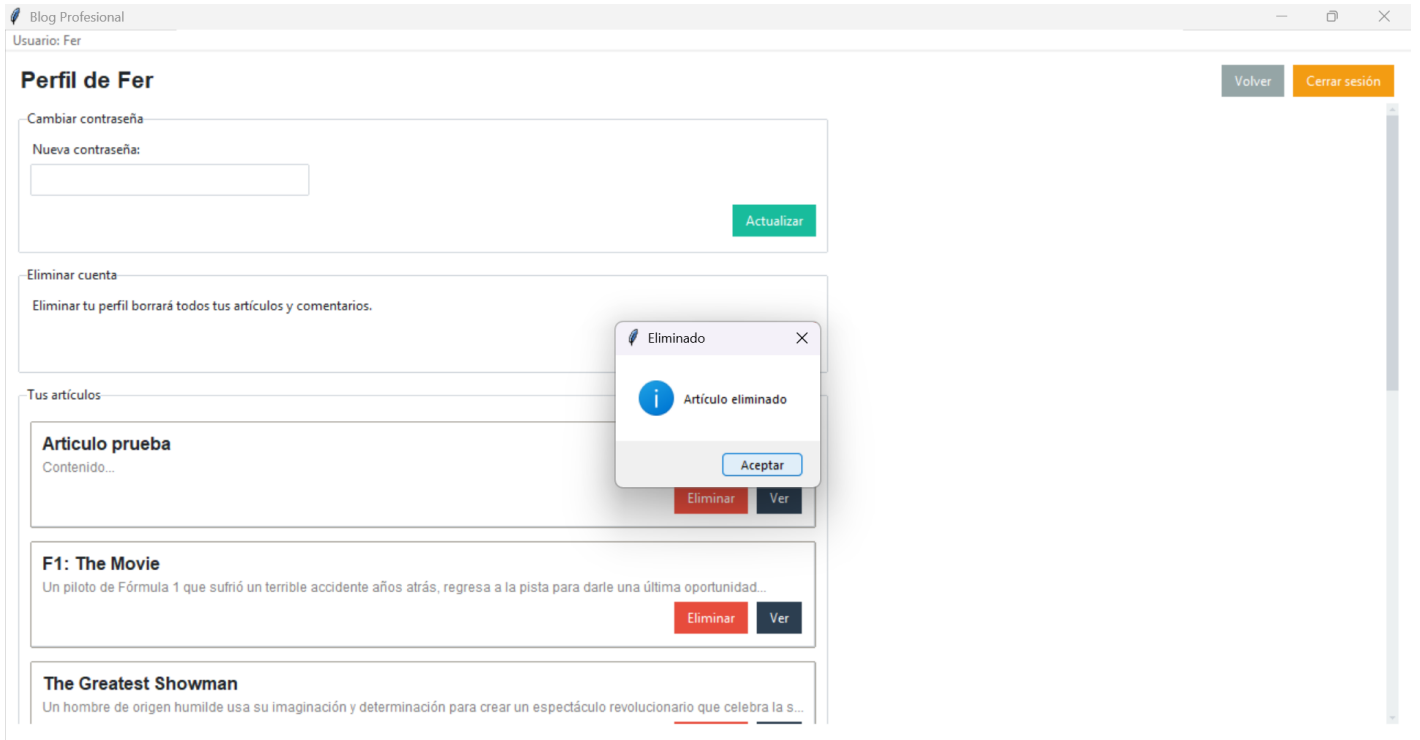
    DELETE FROM Article_tag
    WHERE article_id = p_article_id;

    DELETE FROM Articles
    WHERE article_id = p_article_id;

    IF SQL%ROWCOUNT = 0 THEN
        RAISE_APPLICATION_ERROR(-20203, 'Artículo no encontrado.');
```

END IF;

END;



--Sequencia para id_comment

```
CREATE SEQUENCE comment_seq
```

```
START WITH 1
```

```
INCREMENT BY 1
```

```
NOCACHE
```

```
NOCYCLE;
```

--Agregar comentario

```
CREATE OR REPLACE FUNCTION add_comment(
```

```
    p_article_id IN INTEGER,
```

```
    p_name_user  IN VARCHAR2,
```

```
    p_c_text    IN VARCHAR2
```

```
) RETURN VARCHAR2
```

```
IS
```

```
    v_id_comment NUMBER;
```

```
BEGIN
```

```
v_id_comment := comment_seq.NEXTVAL;
```

```
INSERT INTO Comments (id_comment, article_id, name_user, c_date, c_text)
```

```
VALUES (v_id_comment, p_article_id, p_name_user, SYSDATE, p_c_text);
```

```
RETURN 'Comentario agregado con ID: ' || v_id_comment;
```

```
EXCEPTION
```

```
WHEN OTHERS THEN
```

```
RETURN 'Error al agregar comentario: ' || SQLERRM;
```

```
END;
```

Blog Profesional

Usuario: prueba

Crepúsculo

Autor: Ingrid

Contenido:

Una joven adolescente se muda a un pequeño y lluvioso pueblo y se enamora de un compañero de clase atractivo y misterioso, quien resulta ser un vampiro de más de 100 años que lucha desesperadamente por controlar su instinto de sangre hacia ella.

Categorías:

Comedia Romance

Tags:

Amor Sobrenatural Hombre Lobo Vampiros

Comentarios:

prueba dice:
Comentario Nuevo

Agregar comentario:

Comentar

--Tags

--Agregar tag

```
CREATE OR REPLACE FUNCTION insert_tag_if_not_exists(p_tag_name IN VARCHAR2)
```

```
RETURN VARCHAR2
```

```
IS
```

```
v_exists NUMBER := 0;
```

```

v_url VARCHAR2(4000);

BEGIN

SELECT COUNT(*) INTO v_exists

FROM tags

WHERE LOWER(tag_name) = LOWER(p_tag_name);

IF v_exists = 0 THEN

v_url := 'https://example.com/tags/' || LOWER(p_tag_name);

INSERT INTO tags (tag_name, t_url)

VALUES (p_tag_name, v_url);

RETURN 'Tag insertado: ' || p_tag_name;

ELSE

RETURN 'Tag ya existe: ' || p_tag_name;

END IF;

EXCEPTION

WHEN OTHERS THEN

RETURN 'Error: ' || SQLERRM;

END;

```

--Categorías por artículo

```

create or replace FUNCTION get_cat_by_article(p_article_id IN INTEGER)

RETURN SYS_REFCURSOR

IS

v_cursor SYS_REFCURSOR;

BEGIN

OPEN v_cursor FOR

```

```
SELECT cat_name  
  
FROM article_categorie  
  
WHERE article_id = p_article_id  
  
ORDER BY cat_name;
```

```
RETURN v_cursor;
```

```
END;
```

Blog Profesional

Usuario: prueba

Crepúsculo

Autor: Ingrid

Contenido:

Una joven adolescente se muda a un pequeño y lluvioso pueblo y se enamora de un compañero de clase atractivo y misterioso, quien resulta ser un vampiro de más de 100 años que lucha desesperadamente por controlar su instinto de sangre hacia ella.

Categorías:

ComediaRomance

Tags:

Amor SobrenaturalHombre LoboVampiros

Comentarios:

prueba dice:

Comentario Nuevo

Agregar comentario:

Comentar