

```
CREATE OR REPLACE PACKAGE foundicu AS
  PROCEDURE InsertarPrestamo(p_signature CHAR);
END foundicu;
/
```

```
CREATE OR REPLACE PACKAGE BODY foundicu AS
  PROCEDURE InsertarPrestamo(p_signature CHAR) IS
    user CHAR(10);
    num_prestamos NUMBER;
    prestamos_activos NUMBER;
  BEGIN
    -- Obtener el ID del usuario actual
    SELECT user_id INTO user
    FROM users
    WHERE email = SYS_CONTEXT('USERENV', 'SESSION_USER');

    -- Verificar si el usuario existe
    IF user IS NULL THEN
      DBMS_OUTPUT.PUT_LINE('Error de verificación 1: Usuario no encontrado.');
```

```
      RETURN;
    END IF;

    -- Contar el número de préstamos activos del usuario
    SELECT COUNT(*) INTO num_prestamos
    FROM loans
    WHERE user_id = user AND return IS NULL;

    -- Verificar si el usuario ha alcanzado el límite de préstamos
    IF num_prestamos >= 5 THEN
      DBMS_OUTPUT.PUT_LINE('Error de verificación 2: Límite de préstamos
alcanzado.');
```

```
      RETURN;
    END IF;

    -- Verificar si el ejemplar está disponible (sin préstamos activos)
    SELECT COUNT(*) INTO prestamos_activos
    FROM loans
    WHERE signature = p_signature AND return IS NULL;

    IF prestamos_activos > 0 THEN
      DBMS_OUTPUT.PUT_LINE('Error de verificación 3: El ejemplar ya está prestado.');
```

```
      RETURN;
    END IF;

    -- Registrar el préstamo
    INSERT INTO loans (signature, user_id, stopdate, town, province, type, time, return)
    VALUES (p_signature, user, SYSDATE, 'Madrid', 'Madrid', 'L', 0, NULL);

    DBMS_OUTPUT.PUT_LINE('El préstamo se ha registrado correctamente.');
```

```
  END InsertarPrestamo;
END foundicu;
/
```

```
CREATE OR REPLACE PACKAGE foundicu AS
  PROCEDURE InsertarReserva(p_isbn VARCHAR2, p_date DATE);
END foundicu;
/
```

```
CREATE OR REPLACE PACKAGE BODY foundicu AS
```

```
  PROCEDURE InsertarReserva(p_isbn VARCHAR2, p_date DATE) IS
    user CHAR(10);
    num_prestamos NUMBER;
    id_prestamo CHAR(5);
  BEGIN
    -- Obtener el ID del usuario actual
    BEGIN
      SELECT user_id INTO user
      FROM users
      WHERE email = SYS_CONTEXT('USERENV', 'SESSION_USER');
    EXCEPTION
      WHEN NO_DATA_FOUND THEN
        DBMS_OUTPUT.PUT_LINE('Error de verificación 1: Usuario no encontrado.');
```

```
        RETURN;
    END;

    -- Verificar el número de préstamos actuales del usuario
    SELECT COUNT(*) INTO num_prestamos
    FROM loans
    WHERE user_id = user;

    IF num_prestamos >= 5 THEN
      DBMS_OUTPUT.PUT_LINE('Error de verificación 2: Límite de préstamos alcanzado.');
```

```
      RETURN;
    END IF;

    -- Verificamos la disponibilidad de un ejemplar con el ISBN especificado
    BEGIN
      SELECT c.signature INTO id_prestamo
      FROM copies c
      WHERE c.isbn = p_isbn
      AND NOT EXISTS (
        SELECT 1 FROM loans l
        WHERE l.signature = c.signature
        AND l.return IS NULL
      )
      FETCH FIRST 1 ROW ONLY;
    EXCEPTION
      WHEN NO_DATA_FOUND THEN
        DBMS_OUTPUT.PUT_LINE('Error de verificación 3: No hay ejemplares disponibles para
reservar.');
```

```
        RETURN;
    END;
```

```
-- Registramos la reserva
INSERT INTO loans (signature, user_id, stopdate, type, town, province, time)
VALUES (id_prestamo, user, p_date, 'R', 'Desconocido', 'Desconocido', 0);

DBMS_OUTPUT.PUT_LINE('La reserva se ha registrado correctamente..');

END InsertarReserva;

END foundicu;
/
```

```
CREATE OR REPLACE PACKAGE foundicu AS
  PROCEDURE RegistrarDevolucion(p_signature VARCHAR2);
END foundicu;
/
```

```
CREATE OR REPLACE PACKAGE BODY foundicu AS
```

```
  PROCEDURE RegistrarDevolucion(p_signature VARCHAR2) IS
    user NUMBER;
    num_prestamos NUMBER;

  BEGIN
    -- Obtener el ID del usuario actual
    SELECT user_id INTO user
    FROM users
    WHERE email = SYS_CONTEXT('USERENV', 'SESSION_USER');

    -- Verificar si el usuario existe
    IF user IS NULL THEN
      DBMS_OUTPUT.PUT_LINE('Error de verificación 1: Usuario no encontrado.');
```

```
      RETURN;
    END IF;

    -- Verificar que el usuario tiene el préstamo del ejemplar
    SELECT COUNT(*) INTO num_prestamos
    FROM loans
    WHERE signature = p_signature
      AND user_id = user
      AND return IS NULL; -- Verifica si el préstamo está activo

    IF num_prestamos = 0 THEN -- Si no hay préstamos activos, error
      DBMS_OUTPUT.PUT_LINE('Error de verificación 2: No existe un préstamo
registrado para este usuario y ejemplar.');
```

```
      RETURN;
    END IF;

    -- Eliminar el préstamo (devolución)
    DELETE FROM loans
    WHERE signature = p_signature
      AND user_id = user
      AND return IS NULL; -- Se asegura que solo borre los préstamos no devueltos

    DBMS_OUTPUT.PUT_LINE('La devolución se ha registrado correctamente.');
```

```
  END RegistrarDevolucion;

END foundicu;
/
```