

UNIVERSIDAD PERUANA UNIÓN
FACULTAD DE INGENIERÍA Y ARQUITECTURA
E.P. DE INGENIERÍA DE SISTEMAS



**ENTREGABLE DE ADMINISTRACIÓN DE BASE DE
DATOS I**

KHENDRA

DOCENTE:
MG. CYNTHIA ACUÑA SALINAS

INTEGRANTES:

KATERIN ESTRELLITA MAS MENDOZA
HELEN OBLITAS DIAZ
DAVID PRADA LINAREZ
GERALD FRANCO SANDY CORAL

MORALES, JULIO, 2021

Contenido

| | |
|--|----|
| 1. INTRODUCCIÓN | 3 |
| 2. REQUERIMIENTOS | 3 |
| 2.1. REQUERIMIENTOS FUNCIONALES | 3 |
| 3. REFERENCIA..... | 4 |
| 3.1. DESCRIPCIÓN DE ESTÁNDAR DE BASE DE DATOS..... | 4 |
| 4. SERVIDOR DE BASE DE DATOS..... | 4 |
| 5. ESQUEMA CONCEPTUAL DE BASE DE DATOS | 5 |
| 6. ESQUEMA LÓGICO DE BASE DE DATOS..... | 5 |
| 7. ESQUEMA FÍSICO DE BASE DE DATOS..... | 5 |
| 8. DISEÑO DE TABLAS..... | 6 |
| 8.1. TABLAS..... | 6 |
| 8.1.1. CLASE..... | 6 |
| 8.1.2. CURSO..... | 6 |
| 8.1.3. DOCENTE | 7 |
| 8.1.4. ESPECIALIZACION..... | 8 |
| 8.1.5. INSCRIPCION | 8 |
| 8.1.6. INSCRIPCIONCLASE..... | 9 |
| 8.1.7. MODULO | 10 |
| 8.1.8. PERSONA | 11 |
| 8.1.9. USUARIO | 12 |
| 9. PROCEDIMIENTOS ALMACENADOS | 12 |
| 10. FUNCIONES | 29 |
| 11. CONSULTAS | 31 |
| 12. VISTAS | 32 |

1. INTRODUCCIÓN
2. REQUERIMIENTOS
 - 2.1. REQUERIMIENTOS FUNCIONALES

| Nº | Requisito funcional | Especificación |
|------|---------------------|--|
| RF01 | Registro de Usuario | Una vez ubicados en la página de inicio de sesión, el potencial usuario debe dirigirse al hipervínculo que indica la creación de una nueva cuenta, llenar el formulario con sus respectivos campos obligatorios y proceder a finalizar el proceso pulsando en el botón indicado. |
| RF02 | Logeo del Usuario | Una vez ubicados en la página de inicio de sesión, el usuario debe ingresar su correo y contraseña en los respectivos campos y para finalizar debe pulsar el botón de iniciar sesión o ingresar. Si este no se encuentra registrado, podrá hacerlo en la opción de Registrarse. |
| RN03 | Creación de Curso | El administrador podrá crear tantos cursos como quiera, una vez subido todo el curso podrá hacerlo público para todos puedan verlo. |
| RN04 | Inscripción a curso | Una vez ubicados en la página de inicio de sesión, el usuario irá a loguearse y así poder estar en la interfaz donde se muestra todos los cursos disponibles y cuales podría elegir. |
| RN05 | Creación de Módulos | Una vez creado el curso, el administrador tiene que dirigirse a la parte superior y pulsar el botón de crear nuevo módulo. |
| RN06 | Creación de clase | El administrador podrá crear clases dentro de los módulos que pertenecen a los cursos. |
| RN07 | Valoración de Curso | Una vez terminado un curso el usuario podrá dar una valoración de este. |

3. REFERENCIA

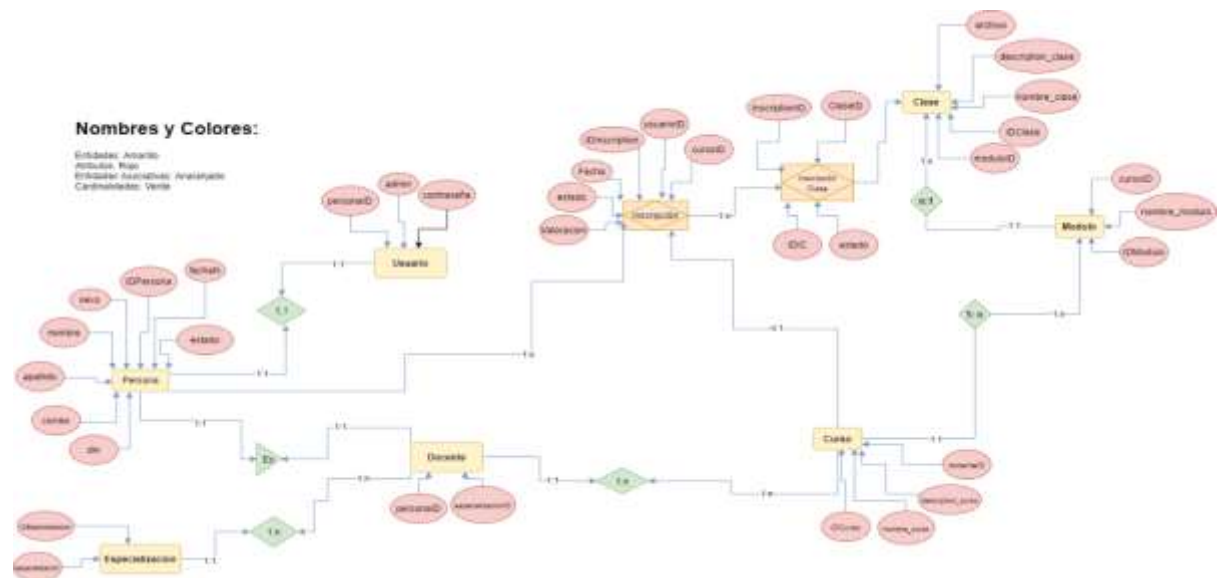
3.1. DESCRIPCIÓN DE ESTÁNDAR DE BASE DE DATOS.

- Los nombres de la tabla iran con mayúscula; por ejemplo: USUARIO, CLASE, CURSO, etc.
- Los nombres de las columnas (atributos) también iran en mayúscula.
- Para los CRUD
 - Secuencias: el nombre de la secuencia del CRUD ira KS_NOMBRETABLA_ID
 - Cabecera: la cabecera de cada CRUD tendra el nombre de KPK_CRUD_NOMBRETABLA
 - Cuerpo: tendrá el mismo nombre de la cabecera
 - Procedimientos:
 - ✓ Buscar: KSP_BUS_NOMBRETABLA
 - ✓ Insertar: KSP_INS_NOMBRETABLA
 - ✓ Actualizar: KSP_UP_NOMBRETABLA
 - ✓ Eliminar: KSP_DEL_NOMBRETABLA
 - Parámetros: cada parametro de cada procedimientos llevara el nombre de KP_NOMBRECOLUMNA (columna de la tabla que se esta trabajando)
- Funciones: KP_TRANSFORM_NOMBRETABLA
- Vistas: KV_NOMBRETABLA
- Consultas: irán con mayúsculas asi como todo

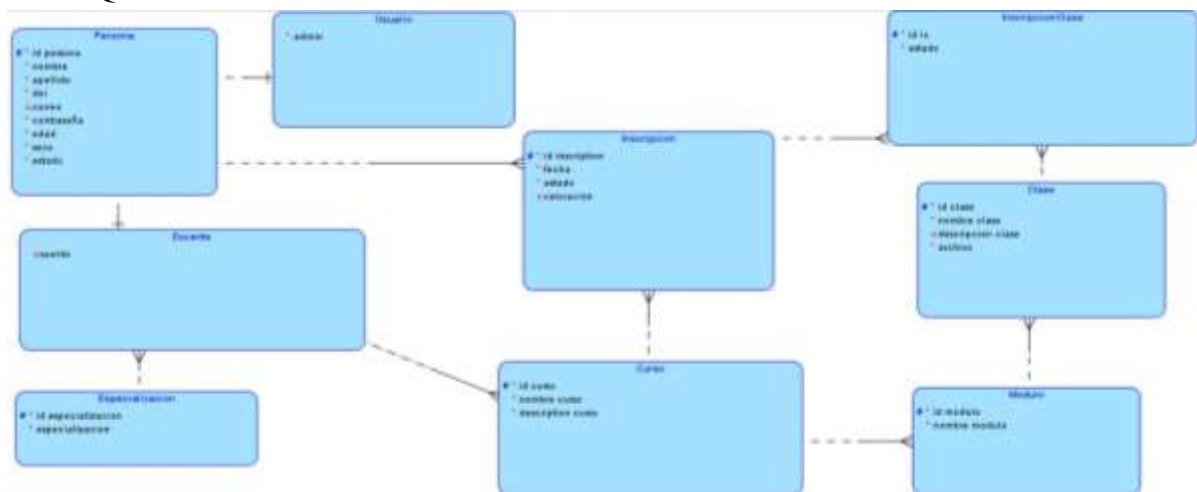
4. SERVIDOR DE BASE DE DATOS

| Servidor de Base de Datos | |
|---------------------------|---|
| Equipo: | El equipo WORLD SYSTEM esta conformado por 4 integrantes. |
| IP: | 3.234.177.170 |
| Sistema Operativo: | Windows |
| SGBD | Oracle |
| Usuarios: | KHENDRA |
| Claves: | 123456 |

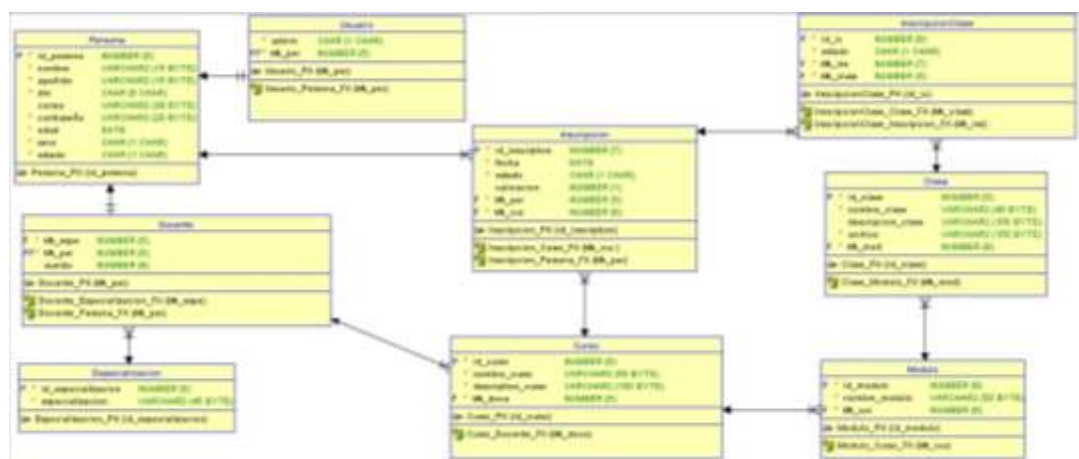
5. ESQUEMA CONCEPTUAL DE BASE DE DATOS



6. ESQUEMA LÓGICO DE BASE DE DATOS



7. ESQUEMA FÍSICO DE BASE DE DATOS



8. DISEÑO DE TABLAS

8.1. TABLAS

8.1.1. CLASE

| | | | |
|----------------------------|--|------------------------|-----------|
| Nombre tabla: | CLASE | Fecha creación: | 12-JUL-21 |
| Descripción: | En la tabla "CLASES" se encontraran los datos de la clase y los archivos determinados. | | |
| Número de columnas: | 5 | | |

| Columnas | | | | | | | |
|----------|-------------------|----|----|--------------|--------|------|--|
| Nro | Nombre de columna | PK | FK | Tipo de dato | Tamaño | Nulo | Comentario |
| 1 | KFK_MOD | | F | NUMBER | (8,0) | N | |
| 2 | ID_CLASE | P | | NUMBER | (5,0) | N | |
| 3 | ARCHIVO | | | VARCHAR2 | (150) | N | En esta columna ira la URL del video de cada clase, el cual sera jalado de YouTube |
| 4 | DESCRIPCION_CLASE | | | VARCHAR2 | (150) | Y | |
| 5 | NOMBRE_CLASE | | | VARCHAR2 | (40) | N | |

| Indices | | |
|---------------|----------------------|---------------|
| Nombre índice | Nombre de la columna | Tipo |
| CLASE_PK | ID_CLASE | NORMAL UNIQUE |

| Constraints | | |
|-------------|-----------------------|--------------|
| Tipo | Nombre de restricción | Columna |
| C | SYS_C007990 | ID_CLASE |
| C | SYS_C007991 | NOMBRE_CLASE |
| C | SYS_C007992 | ARCHIVO |
| C | SYS_C007993 | KFK_MOD |

| Llaves foráneas (REFERENCIA A) | | | |
|--------------------------------|--------------|-------------|-------------------|
| Nombre | Referencia a | Obligatorio | Nombre de columna |
| CLASE.CLASE_MODULO_FK | MODULO | N | KFK_MOD |

8.1.2. CURSO

| | | | |
|----------------------------|---|------------------------|-----------|
| Nombre tabla: | CURSO | Fecha creación: | 12-JUL-21 |
| Descripción: | En la tabla "CURSO" se encontraran datos del curso que el administrador colocara al crear un nuevo curso. | | |
| Número de columnas: | 4 | | |

| Columnas | | | | | | | |
|----------|-------------------|----|----|--------------|--------|------|------------|
| Nro | Nombre de columna | PK | FK | Tipo de dato | Tamaño | Nulo | Comentario |
| 1 | KFK_DOCE | | F | NUMBER | (5,0) | N | |
| 2 | ID_CURSO | P | | NUMBER | (6,0) | N | |
| 3 | NOMBRE_CURSO | | | VARCHAR2 | (50) | N | |
| 4 | DESCRIPTION_CURSO | | | VARCHAR2 | (150) | N | |

| Indices | | |
|---------------|----------------------|---------------|
| Nombre índice | Nombre de la columna | Tipo |
| CURSO_PK | ID_CURSO | NORMAL UNIQUE |

| Constraints | | |
|-------------|-----------------------|-------------------|
| Tipo | Nombre de restricción | Columna |
| C | SYS_C007995 | ID_CURSO |
| C | SYS_C007996 | NOMBRE_CURSO |
| C | SYS_C007997 | DESCRIPTION_CURSO |
| C | SYS_C007998 | KFK_DOCE |

| Llaves foráneas (REFERENCIA A) | | | |
|--------------------------------|--------------|-------------|-------------------|
| Nombre | Referencia a | Obligatorio | Nombre de columna |
| CURSO.CURSO_DOCENTE_FK | DOCENTE | N | KFK_DOCE |

8.1.3. DOCENTE

| | | | |
|----------------------------|---|------------------------|-----------|
| Nombre tabla: | DOCENTE | Fecha creación: | 12-JUL-21 |
| Descripción: | En la tabla "DOCENTE" se encontraran datos relacionados al docente. | | |
| Número de columnas: | 3 | | |

| Columnas | | | | | | | |
|----------|-------------------|----|----|--------------|--------|------|------------|
| Nro | Nombre de columna | PK | FK | Tipo de dato | Tamaño | Nulo | Comentario |
| 1 | KFK_ESPE | | F | NUMBER | (5,0) | N | |
| 2 | KFK_PER | | F | NUMBER | (5,0) | N | |
| 3 | KFK_PER | P | | NUMBER | (5,0) | N | |
| 4 | SUELDO | | | NUMBER | (8,0) | Y | |

| Indices | | |
|---------------|----------------------|---------------|
| Nombre índice | Nombre de la columna | Tipo |
| DOCENTE_PK | KFK_PER | NORMAL UNIQUE |

| Constraints | | |
|--------------------|-----------------------|----------|
| Tipo | Nombre de restricción | Columna |
| C | SYS_C008000 | KFK_ESPE |
| C | SYS_C008001 | KFK_PER |

| Llaves foráneas (REFERENCIA A) | | | |
|---------------------------------------|-----------------|-------------|-------------------|
| Nombre | Referencia a | Obligatorio | Nombre de columna |
| DOCENTE.DOCENTE_ESPECIALIZACION_FK | ESPECIALIZACION | N | KFK_ESPE |
| DOCENTE.DOCENTE_PERSONA_FK | PERSONA | N | KFK_PER |

8.1.4. ESPECIALIZACION

| | | | |
|----------------------------|---|------------------------|-----------|
| Nombre tabla: | ESPECIALIZACION | Fecha creación: | 12-JUL-21 |
| Descripción: | En la tabla "ESPECIALIZACION" se encontraran todas las especializaciones de cada docente que dictara una clase. | | |
| Número de columnas: | 2 | | |

| Columnas | | | | | | | |
|-----------------|--------------------|----|----|--------------|--------|------|------------|
| Nro | Nombre de columna | PK | FK | Tipo de dato | Tamaño | Nulo | Comentario |
| 1 | ID_ESPECIALIZACION | P | | NUMBER | (5,0) | N | |
| 2 | ESPECIALIZACION | | | VARCHAR2 | (45) | N | |

| Indices | | |
|--------------------|----------------------|---------------|
| Nombre índice | Nombre de la columna | Tipo |
| ESPECIALIZACION_PK | ID_ESPECIALIZACION | NORMAL UNIQUE |

| Constraints | | |
|--------------------|-----------------------|--------------------|
| Tipo | Nombre de restricción | Columna |
| C | SYS_C008003 | ID_ESPECIALIZACION |
| C | SYS_C008004 | ESPECIALIZACION |

8.1.5. INSCRIPCION

| | | | |
|----------------------------|--|------------------------|-----------|
| Nombre tabla: | INSCRIPCION | Fecha creación: | 12-JUL-21 |
| Descripción: | Esta es una tabla asociativa que une a las tablas USUARIO y CURSO. | | |
| Número de columnas: | 6 | | |

| Columnas | | | | | | | |
|-----------------|-------------------|----|----|--------------|--------|------|---|
| Nro | Nombre de columna | PK | FK | Tipo de dato | Tamaño | Nulo | Comentario |
| 1 | ID_INSCRIPTION | P | | NUMBER | (7,0) | N | |
| 2 | KFK_CUR | | F | NUMBER | (6,0) | N | |
| 3 | KFK_PER | | F | NUMBER | (5,0) | N | |
| 4 | ESTADO | | | CHAR | (,) | N | Esta columna especifica si es 1 = esta activa, 0 = no esta activa |
| 5 | FECHA | | | DATE | (,) | N | |
| 6 | VALORACION | | | NUMBER | (1,0) | Y | |

| Indices | | |
|----------------|----------------------|---------------|
| Nombre índice | Nombre de la columna | Tipo |
| INSCRIPCION_PK | ID_INSCRIPTION | NORMAL UNIQUE |

| Constraints | | |
|--------------------|-----------------------|----------------|
| Tipo | Nombre de restricción | Columna |
| C | SYS_C008006 | ID_INSCRIPTION |
| C | SYS_C008007 | FECHA |
| C | SYS_C008008 | ESTADO |
| C | SYS_C008009 | KFK_PER |
| C | SYS_C008010 | KFK_CUR |

| Llaves foráneas (REFERENCIA A) | | | |
|---------------------------------------|--------------|-------------|-------------------|
| Nombre | Referencia a | Obligatorio | Nombre de columna |
| INSCRIPCION.INSCRIPCION_CURSO_FK | CURSO | N | KFK_CUR |
| INSCRIPCION.INSCRIPCION_PERSONA_FK | PERSONA | N | KFK_PER |

8.1.6. INSCRIPCIONCLASE

| | | | |
|----------------------------|---|------------------------|-----------|
| Nombre tabla: | INSCRIPCIONCLASE | Fecha creación: | 12-JUL-21 |
| Descripción: | Esta es una tabla ayudará a llevar el control entre las clase de los cursos que el usuario haya tomado. | | |
| Número de columnas: | 4 | | |

| Columnas | | | | | | | |
|-----------------|-------------------|----|----|--------------|--------|------|---|
| Nro | Nombre de columna | PK | FK | Tipo de dato | Tamaño | Nulo | Comentario |
| 1 | ID_IC | P | | NUMBER | (6,0) | N | |
| 2 | KFK_CLASS | | F | NUMBER | (5,0) | N | |
| 3 | KFK_INS | | F | NUMBER | (7,0) | N | |
| 4 | ESTADO | | | CHAR | (,) | N | Esta columna especifica si es 1 = esta activa, 0 = no esta activa |

| <i>Indices</i> | | |
|---------------------|----------------------|---------------|
| Nombre índice | Nombre de la columna | Tipo |
| INSCRIPCIONCLASE_PK | ID_IC | NORMAL UNIQUE |

| <i>Constraints</i> | | |
|--------------------|-----------------------|-----------|
| Tipo | Nombre de restricción | Columna |
| C | SYS_C008012 | ID_IC |
| C | SYS_C008013 | ESTADO |
| C | SYS_C008014 | KFK_INS |
| C | SYS_C008015 | KFK_CLASS |

| <i>Llaves foráneas (REFERENCIA A)</i> | | | |
|--|--------------|-------------|-------------------|
| Nombre | Referencia a | Obligatorio | Nombre de columna |
| INSCRIPCIONCLASE.INSCRIPCIONCLASE_CLASE_FK | CLASE | N | KFK_CLASS |
| INSCRIPCIONCLASE.INSCRIPCIONCLASE_INSCRIPCION_FK | INSCRIPCION | N | KFK_INS |

8.1.7. MODULO

| | | | |
|----------------------------|---|------------------------|-----------|
| <i>Nombre tabla:</i> | MODULO | <i>Fecha creación:</i> | 12-JUL-21 |
| <i>Descripción:</i> | Esta es una tabla se podra llevar un orden de las clases de cada curso. | | |
| <i>Número de columnas:</i> | 3 | | |

| <i>Columnas</i> | | | | | | | |
|-----------------|-------------------|----|----|--------------|--------|------|------------|
| Nro | Nombre de columna | PK | FK | Tipo de dato | Tamaño | Nulo | Comentario |
| 1 | ID_MODULO | P | | NUMBER | (8,0) | N | |
| 2 | KFK_CUR | | F | NUMBER | (6,0) | N | |
| 3 | NOMBRE_MODULO | | | VARCHAR2 | (50) | N | |

| <i>Indices</i> | | |
|----------------|----------------------|---------------|
| Nombre índice | Nombre de la columna | Tipo |
| MODULO_PK | ID_MODULO | NORMAL UNIQUE |

| <i>Constraints</i> | | |
|--------------------|-----------------------|---------------|
| Tipo | Nombre de restricción | Columna |
| C | SYS_C008017 | ID_MODULO |
| C | SYS_C008018 | NOMBRE_MODULO |
| C | SYS_C008019 | KFK_CUR |

Llaves foráneas (REFERENCIA A)

| Nombre | Referencia a | Obligatorio | Nombre de columna |
|------------------------|--------------|-------------|-------------------|
| MODULO.MODULO_CURSO_FK | CURSO | N | KFK_CUR |

8.1.8. PERSONA

| | | | |
|----------------------------|--|------------------------|-----------|
| Nombre tabla: | PERSONA | Fecha creación: | 12-JUL-21 |
| Descripción: | Esta es la tabla padre de USUARIO y DOCENTE. | | |
| Número de columnas: | 9 | | |

| Columnas | | | | | | | |
|-----------------|-------------------|----|----|--------------|--------|------|---|
| Nro | Nombre de columna | PK | FK | Tipo de dato | Tamaño | Nulo | Comentario |
| 1 | ID_PERSONA | P | | NUMBER | (5,0) | N | |
| 2 | EDAD | | | DATE | (,) | N | |
| 3 | CORREO | | | VARCHAR2 | (30) | Y | |
| 4 | APELLIDO | | | VARCHAR2 | (15) | N | |
| 5 | NOMBRE | | | VARCHAR2 | (15) | N | |
| 6 | ESTADO | | | CHAR | (,) | N | Esta columna especifica si es 1 = esta activa, 0 = no esta activa |
| 7 | CONTRASEÑA | | | VARCHAR2 | (20) | N | |
| 8 | SEXO | | | CHAR | (,) | N | |
| 9 | DNI | | | CHAR | (,) | N | |

| Indices | | |
|----------------|----------------------|---------------|
| Nombre índice | Nombre de la columna | Tipo |
| PERSONA_PK | ID_PERSONA | NORMAL UNIQUE |

| Constraints | | |
|--------------------|-----------------------|------------|
| Tipo | Nombre de restricción | Columna |
| C | SYS_C008021 | ID_PERSONA |
| C | SYS_C008022 | NOMBRE |
| C | SYS_C008023 | APELLIDO |
| C | SYS_C008024 | DNI |
| C | SYS_C008025 | CONTRASEÑA |
| C | SYS_C008026 | EDAD |
| C | SYS_C008027 | SEXO |
| C | SYS_C008028 | ESTADO |

8.1.9. USUARIO

| | | | |
|----------------------------|---|------------------------|-----------|
| Nombre tabla: | USUARIO | Fecha creación: | 12-JUL-21 |
| Descripción: | Esta es la tabla encontraremos todos los usuarios del sistema sea o no administrador. | | |
| Número de columnas: | 2 | | |

| Columnas | | | | | | | |
|----------|-------------------|----|----|--------------|--------|------|---|
| Nro | Nombre de columna | PK | FK | Tipo de dato | Tamaño | Nulo | Comentario |
| 1 | KFK_PER | P | | NUMBER | (5,0) | N | |
| 2 | KFK_PER | | F | NUMBER | (5,0) | N | |
| 3 | ADMIN | | | CHAR | (,) | N | Esta columna especifica si es 1 = admin, 0 = no admin |

| Indices | | |
|---------------|----------------------|---------------|
| Nombre índice | Nombre de la columna | Tipo |
| USUARIO_PK | KFK_PER | NORMAL UNIQUE |

| Constraints | | |
|-------------|-----------------------|---------|
| Tipo | Nombre de restricción | Columna |
| C | SYS_C008030 | ADMIN |
| C | SYS_C008031 | KFK_PER |

| Llaves foráneas (REFERENCIA A) | | | |
|--------------------------------|--------------|-------------|-------------------|
| Nombre | Referencia a | Obligatorio | Nombre de columna |
| USUARIO.USUARIO_PERSONA_FK | PERSONA | N | KFK_PER |

9. PROCEDIMIENTOS ALMACENADOS

| |
|--|
| Nombre del procedimiento: KPK_CRUD_CLASE |
| Descripción: Los procedimientos almacenados dentro del paquete KPK_CRUD_CLASE tienen como funcionalidad hacer las acciones respectivas de nuestra tabla CLASE (CRUD). |
| Referencia: Requisito Funcional número 06 |
| --CRUD PARA LA TABLA CLASE CREATE SEQUENCE KS_CLASE_ID START WITH 16 INCREMENT BY 1; |

```

CREATE OR REPLACE PACKAGE KPK_CRUD_CLASE IS
    PROCEDURE KSP_BUS_CLASE(KP_ID_CLASE CLASE.ID_CLASE%TYPE,
        KP_CLASE OUT CLASE%ROWTYPE);
    PROCEDURE KSP_INS_CLASE(KP_NOMBRE_CLASE CLASE.NOMBRE_CLASE%TYPE,
        KP_DESCRIPCION_CLASE CLASE.DESCRIPCION_CLASE%TYPE,
        KP_ARCHIVO CLASE.ARCHIVO%TYPE, KP_KFK_MOD
        CLASE.KFK_MOD%TYPE);
    PROCEDURE KSP_UP_CLASE(KP_ID_CLASE CLASE.ID_CLASE%TYPE,
        KP_NOMBRE_CLASE CLASE.NOMBRE_CLASE%TYPE,
        KP_DESCRIPCION_CLASE CLASE.DESCRIPCION_CLASE%TYPE,
        KP_ARCHIVO CLASE.ARCHIVO%TYPE, KP_KFK_MOD
        CLASE.KFK_MOD%TYPE);
    PROCEDURE KSP_DEL_CLASE(KP_ID_CLASE CLASE.ID_CLASE%TYPE);
END KPK_CRUD_CLASE;

```

```

CREATE OR REPLACE PACKAGE BODY KPK_CRUD_CLASE IS
    --BUSCAR
    PROCEDURE KSP_BUS_CLASE(KP_ID_CLASE CLASE.ID_CLASE%TYPE,
        KP_CLASE OUT CLASE%ROWTYPE)
    IS
        BEGIN
            SELECT * INTO KP_CLASE
            FROM CLASE
            WHERE ID_CLASE=KP_ID_CLASE;
        EXCEPTION
            WHEN OTHERS THEN
                DBMS_OUTPUT.PUT_LINE('NO HAY DATA');
        END KSP_BUS_CLASE;

    PROCEDURE KSP_INS_CLASE(KP_NOMBRE_CLASE CLASE.NOMBRE_CLASE%TYPE,
        KP_DESCRIPCION_CLASE CLASE.DESCRIPCION_CLASE%TYPE,
        KP_ARCHIVO CLASE.ARCHIVO%TYPE, KP_KFK_MOD
        CLASE.KFK_MOD%TYPE)
    IS
        BEGIN
            INSERT INTO CLASE VALUES (KS_CLASE_ID.NEXTVAL, KP_NOMBRE_CLASE,
            KP_DESCRIPCION_CLASE, KP_ARCHIVO, KP_KFK_MOD);
            COMMIT;
        EXCEPTION
            WHEN OTHERS THEN
                DBMS_OUTPUT.PUT_LINE('NO HAY DATA');
        END KSP_INS_CLASE;

    PROCEDURE KSP_UP_CLASE(KP_ID_CLASE CLASE.ID_CLASE%TYPE,
        KP_NOMBRE_CLASE CLASE.NOMBRE_CLASE%TYPE,
        KP_DESCRIPCION_CLASE CLASE.DESCRIPCION_CLASE%TYPE,
        KP_ARCHIVO CLASE.ARCHIVO%TYPE, KP_KFK_MOD
        CLASE.KFK_MOD%TYPE)

```

```

IS
BEGIN
    UPDATE CLASE
    SET NOMBRE_CLASE=KP_NOMBRE_CLASE,
DESCRIPCION_CLASE=KP_DESCRIPCION_CLASE, ARCHIVO=KP_ARCHIVO,
KFK_MOD=KP_KFK_MOD
    WHERE ID_CLASE=KP_ID_CLASE;
    COMMIT;
    EXCEPTION
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('NO SE ACTUALIZÓ');
END KSP_UP_CLASE;

PROCEDURE KSP_DEL_CLASE(KP_ID_CLASE CLASE.ID_CLASE%TYPE)

IS
BEGIN
    DELETE FROM CLASE
    WHERE ID_CLASE=KP_ID_CLASE;
    COMMIT;
    EXCEPTION
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('NO ELIMINÓ');
END KSP_DEL_CLASE;

END KPK_CRUD_CLASE;

```

Nombre del procedimiento: KPK_CRUD_CURSO

Descripción: Los procedimientos almacenados dentro del paquete KPK_CRUD_CURSO tienen como funcionalidad hacer las acciones respectivas de nuestra tabla CURSO (CRUD).

Referencia: Requisito Funcional número 07

```

CREATE SEQUENCE KS_CURSO_ID
START WITH 5
INCREMENT BY 1;

CREATE OR REPLACE PACKAGE KPK_CRUD_CURSO IS
    PROCEDURE KSP_BUS_CURSO(KP_ID_CURSO CURSO.ID_CURSO%TYPE,
        KP_CURSO OUT CURSO%ROWTYPE);
    PROCEDURE KSP_INS_CURSO(KP_NOMBRE_CURSO CURSO.NOMBRE_CURSO%TYPE,
KP_DESCRIPTION_CURSO CURSO.DESCRIPCION_CURSO%TYPE,
        KP_KFK_DOCE CURSO.KFK_DOCE%TYPE);
    PROCEDURE KSP_UP_CURSO(KP_ID_CURSO CURSO.ID_CURSO%TYPE,
KP_NOMBRE_CURSO CURSO.NOMBRE_CURSO%TYPE,
        KP_DESCRIPTION_CURSO CURSO.DESCRIPCION_CURSO%TYPE,

```

```

        KP_KFK_DOCE CURSO.KFK_DOCE%TYPE);
PROCEDURE KSP_DEL_CURSO(KP_ID_CURSO CURSO.ID_CURSO%TYPE);

END KPK_CRUD_CURSO;

CREATE OR REPLACE PACKAGE BODY KPK_CRUD_CURSO IS
    --BUSCAR
    PROCEDURE KSP_BUS_CURSO(KP_ID_CURSO CURSO.ID_CURSO%TYPE,
        KP_CURSO OUT CURSO%ROWTYPE)
    IS
        BEGIN
            SELECT * INTO KP_CURSO
            FROM CURSO
            WHERE ID_CURSO=KP_ID_CURSO;
        EXCEPTION
            WHEN OTHERS THEN
                DBMS_OUTPUT.PUT_LINE('NO HAY DATA');
        END KSP_BUS_CURSO;

        PROCEDURE KSP_INS_CURSO(KP_NOMBRE_CURSO CURSO.NOMBRE_CURSO%TYPE,
            KP_DESCRIPTION_CURSO CURSO.DESCRPTION_CURSO%TYPE,
            KP_KFK_DOCE CURSO.KFK_DOCE%TYPE)
        IS
            BEGIN
                INSERT INTO CURSO VALUES (KS_CURSO_ID.NEXTVAL, KP_NOMBRE_CURSO,
                KP_DESCRIPTION_CURSO, KP_KFK_DOCE);
                COMMIT;
            EXCEPTION
                WHEN OTHERS THEN
                    DBMS_OUTPUT.PUT_LINE('NO HAY DATA');
            END KSP_INS_CURSO;

            PROCEDURE KSP_UP_CURSO(KP_ID_CURSO CURSO.ID_CURSO%TYPE,
                KP_NOMBRE_CURSO CURSO.NOMBRE_CURSO%TYPE,
                KP_DESCRIPTION_CURSO CURSO.DESCRPTION_CURSO%TYPE,
                KP_KFK_DOCE CURSO.KFK_DOCE%TYPE)
            IS
                BEGIN
                    UPDATE CURSO
                    SET NOMBRE_CURSO=KP_NOMBRE_CURSO,
                    DESCRIPTION_CURSO=KP_DESCRIPTION_CURSO, KFK_DOCE=KP_KFK_DOCE
                    WHERE ID_CURSO=KP_ID_CURSO;
                    COMMIT;
                EXCEPTION
                    WHEN OTHERS THEN
                        DBMS_OUTPUT.PUT_LINE('NO SE ACTUALIZÓ');
                END KSP_UP_CURSO;

            PROCEDURE KSP_DEL_CURSO(KP_ID_CURSO CURSO.ID_CURSO%TYPE)

```

```

IS
BEGIN
    DELETE FROM CURSO
    WHERE ID_CURSO=KP_ID_CURSO;
    COMMIT;
    EXCEPTION
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('NO ELIMINÓ');
    END KSP_DEL_CURSO;

END KPK_CRUD_CURSO;

```

Nombre del procedimiento: KPK_CRUD_DOCENTE

Descripción: Los procedimientos almacenados dentro del paquete KPK_CRUD_DOCENTE tienen como funcionalidad hacer las acciones respectivas de nuestra tabla DOCENTE (CRUD).

Referencia: Requisito Funcional número 01

```

CREATE SEQUENCE KS_CURSO_ID
START WITH 4
INCREMENT BY 1;

```

CREATE OR REPLACE PACKAGE KPK_CRUD_DOCENTE IS

```

    PROCEDURE KSP_BUS_DOCENTE(KP_ID_DOCENTE DOCENTE.KFK_PER%TYPE,
                               KP_DOCENTE OUT DOCENTE%ROWTYPE);
    PROCEDURE KSP_INS_DOCENTE(KP_ESPECIALIDAD DOCENTE.KFK_ESPE%TYPE,
                               KP_ID_DOCENTE DOCENTE.KFK_PER%TYPE,
                               KP_SUELDO DOCENTE.SUELDO%TYPE);
    PROCEDURE KSP_UP_DOCENTE(KP_ID_DOCENTE DOCENTE.KFK_PER%TYPE,
                              KP_ESPECIALIDAD DOCENTE.KFK_ESPE%TYPE,
                              KP_SUELDO DOCENTE.SUELDO%TYPE);
    PROCEDURE KSP_DEL_DOCENTE(KP_ID_DOCENTE DOCENTE.KFK_PER%TYPE);
END KPK_CRUD_DOCENTE;

```

CREATE OR REPLACE PACKAGE BODY KPK_CRUD_DOCENTE IS

--BUSCAR DOCENTE

```

    PROCEDURE KSP_BUS_DOCENTE(KP_ID_DOCENTE DOCENTE.KFK_PER%TYPE,
                               KP_DOCENTE OUT DOCENTE%ROWTYPE)

```

IS

BEGIN

```

    SELECT * INTO KP_DOCENTE

```

```

    FROM DOCENTE

```

```

    WHERE KFK_PER = KP_ID_DOCENTE;

```



```

EXCEPTION
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('NO SE ENCONTRO AL DOCENTE');

END KSP_BUS_DOCENTE;

--INSERTAR DOCENTE
PROCEDURE KSP_INS_DOCENTE(KP_ESPECIALIDAD
DOCENTE.KFK_ESPE%TYPE,
                        KP_ID_DOCENTE DOCENTE.KFK_PER%TYPE,
                        KP_SUELDO DOCENTE.SUELDO%TYPE)
IS
BEGIN
    INSERT INTO DOCENTE
    VALUES(KP_ESPECIALIDAD, KP_ID_DOCENTE, KP_SUELDO);
    COMMIT;
EXCEPTION
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('NO SE PUDO INSERTAR EL DOCENTE');
END KSP_INS_DOCENTE;

--ACTUALIZAR DOCENTE
PROCEDURE KSP_UP_DOCENTE(KP_ID_DOCENTE DOCENTE.KFK_PER%TYPE,
                        KP_ESPECIALIDAD DOCENTE.KFK_ESPE%TYPE,
                        KP_SUELDO DOCENTE.SUELDO%TYPE)
IS
BEGIN
    UPDATE DOCENTE
    SET KFK_ESPE=KP_ESPECIALIDAD, SUELDO=KP_SUELDO
    WHERE KFK_PER=KP_ID_DOCENTE;
    COMMIT;
EXCEPTION
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('NO SE PUDO ACTUALIZAR DOCENTE');

END KSP_UP_DOCENTE;

--ELIMINAR DOCENTE
PROCEDURE KSP_DEL_DOCENTE(KP_ID_DOCENTE DOCENTE.KFK_PER%TYPE)
IS
BEGIN
    UPDATE PERSONA
    SET ESTADO = '0'
    WHERE ID_PERSONA=KP_ID_DOCENTE;

DELETE DOCENTE
WHERE KFK_PER=KP_ID_DOCENTE;
    COMMIT;
EXCEPTION

```

```

        WHEN OTHERS THEN
            DBMS_OUTPUT.PUT_LINE('NO SE PUDO ELIMINAR DOCENTE');
        END KSP_DEL_DOCENTE;

END KPK_CRUD_DOCENTE;

```

Nombre del procedimiento: KPK_CRUD_ESPECIALIZACION

Descripción: Los procedimientos almacenados dentro del paquete KPK_CRUD_ESPECIALIZACION tienen como funcionalidad hacer las acciones respectivas de nuestra tabla ESPECIALIZACION (CRUD).

Referencia: --

```

CREATE SEQUENCE KS_ESPECIALIZACION_ID
START WITH 6
INCREMENT BY 1;

CREATE OR REPLACE PACKAGE KPK_CRUD_ESPECIALIZACION IS

    PROCEDURE KSP_BUS_ESPECIALIZACION(KP_ID_ESPECIALIZACION
ESPECIALIZACION.ID_ESPECIALIZACION%TYPE,
                                KP_ESPECIALIZACION OUT
ESPECIALIZACION%ROWTYPE);
    PROCEDURE KSP_INS_ESPECIALIZACION(KP_ESPECIALIZACION
ESPECIALIZACION.ESPECIALIZACION%TYPE);
    PROCEDURE KSP_UP_ESPECIALIZACION (KP_ID_ESPECIALIZACION
ESPECIALIZACION.ID_ESPECIALIZACION%TYPE,
                                KP_ESPECIALIZACION
ESPECIALIZACION.ESPECIALIZACION%TYPE);
    PROCEDURE KSP_DEL_ESPECIALIZACION (KP_ID_ESPECIALIZACION
ESPECIALIZACION.ID_ESPECIALIZACION%TYPE);

END KPK_CRUD_ESPECIALIZACION;

CREATE OR REPLACE PACKAGE BODY KPK_CRUD_ESPECIALIZACION IS
    --BUSCAR ESTUDIANTE
    PROCEDURE KSP_BUS_ESPECIALIZACION(KP_ID_ESPECIALIZACION
ESPECIALIZACION.ID_ESPECIALIZACION%TYPE,
                                KP_ESPECIALIZACION OUT
ESPECIALIZACION%ROWTYPE)
    IS
    BEGIN
        SELECT * INTO KP_ESPECIALIZACION
        FROM ESPECIALIZACION
        WHERE ID_ESPECIALIZACION = KP_ID_ESPECIALIZACION;

```

```

EXCEPTION
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('NO SE ENCONTRO ESPECIALIZACIONES');

END KSP_BUS_ESPECIALIZACION;

--INSERTAR
PROCEDURE KSP_INS_ESPECIALIZACION(KP_ESPECIALIZACION
ESPECIALIZACION.ESPECIALIZACION%TYPE)

IS
BEGIN
    INSERT INTO ESPECIALIZACION
    VALUES(KS_ESPECIALIZACION_ID.NEXTVAL, KP_ESPECIALIZACION);
    COMMIT;
EXCEPTION
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('NO SE PUDO INSERTAR ESPECIALIZACION');
END KSP_INS_ESPECIALIZACION;

--ACTUALIZAR
PROCEDURE KSP_UP_ESPECIALIZACION (KP_ID_ESPECIALIZACION
ESPECIALIZACION.ID_ESPECIALIZACION%TYPE,
                                KP_ESPECIALIZACION
ESPECIALIZACION.ESPECIALIZACION%TYPE)
IS
BEGIN
    UPDATE ESPECIALIZACION
    SET ESPECIALIZACION=KP_ESPECIALIZACION
    WHERE ID_ESPECIALIZACION=KP_ID_ESPECIALIZACION;
    COMMIT;
EXCEPTION
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('NO SE PUDO ACTUALIZAR ESPECIALIZACION');

END KSP_UP_ESPECIALIZACION;

--ELIMINAR
PROCEDURE KSP_DEL_ESPECIALIZACION (KP_ID_ESPECIALIZACION
ESPECIALIZACION.ID_ESPECIALIZACION%TYPE)
IS
BEGIN
    DELETE ESPECIALIZACION
    WHERE ID_ESPECIALIZACION=KP_ID_ESPECIALIZACION;
    COMMIT;
EXCEPTION
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('NO SE PUDO ELIMINAR ESPECIALIZACION');
END KSP_DEL_ESPECIALIZACION;

```

END KPK_CRUD_ESPECIALIZACION;

Nombre del procedimiento: KPK_CRUD_INSCRIPCION

Descripción: Los procedimientos almacenados dentro del paquete KPK_CRUD_INSCRIPCION tienen como funcionalidad hacer las acciones respectivas de nuestra tabla INSCRIPCION (CRUD).

Referencia: Requisito Funcional número 04

```
CREATE SEQUENCE KS_INSCRIPCION_ID
START WITH 11
INCREMENT BY 1;

CREATE OR REPLACE PACKAGE KPK_CRUD_INSCRIPCION IS
  PROCEDURE KSP_BUS_INSCRIPCION(KP_ID_INSCRIPTION
    INSCRIPCION.ID_INSCRIPTION%TYPE,
                                KP_INSCRIPCION OUT INSCRIPCION%ROWTYPE);
  PROCEDURE KSP_INS_INSCRIPCION(KP_FECHA INSCRIPCION.FECHA%TYPE,
    KP_ESTADO INSCRIPCION.ESTADO%TYPE,
                                KP_VALORACION INSCRIPCION.VALORACION%TYPE,
                                KP_KFK_PER INSCRIPCION.KFK_PER%TYPE, KP_KFK_CUR
    INSCRIPCION.KFK_CUR%TYPE);
  PROCEDURE KSP_UP_INSCRIPCION(KP_ID_INSCRIPTION
    INSCRIPCION.ID_INSCRIPTION%TYPE,
                                KP_FECHA INSCRIPCION.FECHA%TYPE, KP_ESTADO
    INSCRIPCION.ESTADO%TYPE,
                                KP_VALORACION INSCRIPCION.VALORACION%TYPE,
                                KP_KFK_PER INSCRIPCION.KFK_PER%TYPE, KP_KFK_CUR
    INSCRIPCION.KFK_CUR%TYPE);
  PROCEDURE KSP_DEL_INSCRIPCION(KP_ID_INSCRIPTION
    INSCRIPCION.ID_INSCRIPTION%TYPE);
END KPK_CRUD_INSCRIPCION;

CREATE OR REPLACE PACKAGE BODY KPK_CRUD_INSCRIPCION IS
  --BUSCAR
  PROCEDURE KSP_BUS_INSCRIPCION(KP_ID_INSCRIPTION
    INSCRIPCION.ID_INSCRIPTION%TYPE,
                                KP_INSCRIPCION OUT INSCRIPCION%ROWTYPE)
  IS
  BEGIN
    SELECT * INTO KP_INSCRIPCION
    FROM INSCRIPCION
    WHERE ID_INSCRIPTION=KP_ID_INSCRIPTION;
  EXCEPTION
    WHEN OTHERS THEN
      DBMS_OUTPUT.PUT_LINE('NO HAY DATA');
```

```

        END KSP_BUS_INSCRIPCION;
    PROCEDURE KSP_INS_INSCRIPCION(KP_FECHA INSCRIPCION.FECHA%TYPE,
    KP_ESTADO INSCRIPCION.ESTADO%TYPE,
        KP_VALORACION INSCRIPCION.VALORACION%TYPE,
        KP_KFK_PER INSCRIPCION.KFK_PER%TYPE, KP_KFK_CUR
    INSCRIPCION.KFK_CUR%TYPE)
    IS
    BEGIN
        INSERT INTO INSCRIPCION
        VALUES
    (KS_INSCRIPCION_ID.NEXTVAL,KP_FECHA,KP_ESTADO,KP_VALORACION,KP_KFK_PE
    R,KP_KFK_CUR);
        COMMIT;
        EXCEPTION
            WHEN OTHERS THEN
                DBMS_OUTPUT.PUT_LINE('NO SE INSERTÓ');
    END KSP_INS_INSCRIPCION;
    PROCEDURE KSP_UP_INSCRIPCION(KP_ID_INSCRIPTION
    INSCRIPCION.ID_INSCRIPTION%TYPE,
        KP_FECHA INSCRIPCION.FECHA%TYPE, KP_ESTADO
    INSCRIPCION.ESTADO%TYPE,
        KP_VALORACION INSCRIPCION.VALORACION%TYPE,
        KP_KFK_PER INSCRIPCION.KFK_PER%TYPE, KP_KFK_CUR
    INSCRIPCION.KFK_CUR%TYPE)
    IS
    BEGIN
        UPDATE
        INSCRIPCION
        SET FECHA=KP_FECHA, ESTADO=KP_ESTADO,VALORACION=KP_VALORACION,
    KFK_PER=KP_KFK_PER,KFK_CUR=KP_KFK_CUR
        WHERE ID_INSCRIPTION=KP_ID_INSCRIPTION;
        COMMIT;
        EXCEPTION
            WHEN OTHERS THEN
                DBMS_OUTPUT.PUT_LINE('NO SE ACTUALIZÓ');
    END KSP_UP_INSCRIPCION;
    PROCEDURE KSP_DEL_INSCRIPCION(KP_ID_INSCRIPTION
    INSCRIPCION.ID_INSCRIPTION%TYPE)
    IS
    BEGIN
        DELETE FROM INSCRIPCION
        WHERE ID_INSCRIPTION=KP_ID_INSCRIPTION;
        COMMIT;
        EXCEPTION
            WHEN OTHERS THEN
                DBMS_OUTPUT.PUT_LINE('NO ELIMINÓ');
    END KSP_DEL_INSCRIPCION;

END KPK_CRUD_INSCRIPCION;

```

Nombre del procedimiento: KPK_CRUD_INSCRIPCIONCLASE

Descripción: Los procedimientos almacenados dentro del paquete KPK_CRUD_INSCRIPCIONCLASE tienen como funcionalidad hacer las acciones respectivas de nuestra tabla INSCRIPCIONCLASE (CRUD).

Referencia: Requisito Funcional número 04

```
CREATE SEQUENCE KS_IC_ID
START WITH 37
INCREMENT BY 1;

CREATE OR REPLACE PACKAGE KPK_CRUD_INSCRIPCIONCLASE IS
  PROCEDURE KSP_BUS_INSCRIPCIONCLASE(KP_ID_IC
    INSCRIPCIONCLASE.ID_IC%TYPE,
                                     KP_INSCRIPCIONCLASE OUT
    INSCRIPCIONCLASE%ROWTYPE);
  PROCEDURE KSP_INS_INSCRIPCIONCLASE(KP_ESTADO
    INSCRIPCIONCLASE.ESTADO%TYPE,
                                     KP_KFK_INS INSCRIPCIONCLASE.KFK_INS%TYPE, KP_KFK_CLASS
    INSCRIPCIONCLASE.KFK_CLASS%TYPE);
  PROCEDURE KSP_UP_INSCRIPCIONCLASE(KP_ID_IC
    INSCRIPCIONCLASE.ID_IC%TYPE,
                                     KP_ESTADO INSCRIPCIONCLASE.ESTADO%TYPE,
                                     KP_KFK_INS INSCRIPCIONCLASE.KFK_INS%TYPE, KP_KFK_CLASS
    INSCRIPCIONCLASE.KFK_CLASS%TYPE);
  PROCEDURE KSP_DEL_INSCRIPCIONCLASE(KP_ID_IC
    INSCRIPCIONCLASE.ID_IC%TYPE);
END KPK_CRUD_INSCRIPCIONCLASE;

CREATE OR REPLACE PACKAGE BODY KPK_CRUD_INSCRIPCIONCLASE IS
  --BUSCAR
  PROCEDURE KSP_BUS_INSCRIPCIONCLASE(KP_ID_IC
    INSCRIPCIONCLASE.ID_IC%TYPE,
                                     KP_INSCRIPCIONCLASE OUT
    INSCRIPCIONCLASE%ROWTYPE)
  IS
  BEGIN
    SELECT * INTO KP_INSCRIPCIONCLASE
    FROM INSCRIPCIONCLASE
    WHERE ID_IC=KP_ID_IC;
  EXCEPTION
    WHEN OTHERS THEN
      DBMS_OUTPUT.PUT_LINE('NO HAY DATA');
  END KSP_BUS_INSCRIPCIONCLASE;
  PROCEDURE KSP_INS_INSCRIPCIONCLASE(KP_ESTADO
    INSCRIPCIONCLASE.ESTADO%TYPE,
```

```

                KP_KFK_INS INSCRIPCIONCLASE.KFK_INS%TYPE, KP_KFK_CLASS
INSCRIPCIONCLASE.KFK_CLASS%TYPE)
IS
BEGIN
    INSERT INTO INSCRIPCIONCLASE
    VALUES (KS_IC_ID.NEXTVAL,KP_ESTADO,KP_KFK_INS,KP_KFK_CLASS);
    COMMIT;
    EXCEPTION
        WHEN OTHERS THEN
            DBMS_OUTPUT.PUT_LINE('NO SE INSERTÓ');
END KSP_INS_INSCRIPCIONCLASE;
PROCEDURE KSP_UP_INSCRIPCIONCLASE(KP_ID_IC
INSCRIPCIONCLASE.ID_IC%TYPE,
                KP_ESTADO INSCRIPCIONCLASE.ESTADO%TYPE,
                KP_KFK_INS INSCRIPCIONCLASE.KFK_INS%TYPE, KP_KFK_CLASS
INSCRIPCIONCLASE.KFK_CLASS%TYPE)
IS
BEGIN
    UPDATE
    INSCRIPCIONCLASE
    SET ESTADO=KP_ESTADO,KFK_INS=KP_KFK_INS,KFK_CLASS=KP_KFK_CLASS
    WHERE ID_IC=KP_ID_IC;
    COMMIT;
    EXCEPTION
        WHEN OTHERS THEN
            DBMS_OUTPUT.PUT_LINE('NO SE ACTUALIZÓ');
END KSP_UP_INSCRIPCIONCLASE;
PROCEDURE KSP_DEL_INSCRIPCIONCLASE(KP_ID_IC
INSCRIPCIONCLASE.ID_IC%TYPE)
IS
BEGIN
    DELETE FROM INSCRIPCIONCLASE
    WHERE ID_IC=KP_ID_IC;
    COMMIT;
    EXCEPTION
        WHEN OTHERS THEN
            DBMS_OUTPUT.PUT_LINE('NO ELIMINÓ');
END KSP_DEL_INSCRIPCIONCLASE;

END KPK_CRUD_INSCRIPCIONCLASE;

```

Nombre del procedimiento: KPK_CRUD_MODULO

Descripción: Los procedimientos almacenados dentro del paquete KPK_CRUD_MODULO tienen como funcionalidad hacer las acciones respectivas de nuestra tabla MODULO (CRUD).

Referencia: Requisito Funcional número 05

```
CREATE SEQUENCE KS_MODULO_ID
START WITH 13
INCREMENT BY 1;
```

```
CREATE OR REPLACE PACKAGE KPK_CRUD_MODULO IS
```

```
    PROCEDURE KSP_BUS_MODULO(KP_ID_MODULO MODULO.ID_MODULO%TYPE,
                             KP_MODULO OUT MODULO%ROWTYPE);
    PROCEDURE KSP_INS_MODULO(KP_NOMBRE_MODULO
MODULO.NOMBRE_MODULO%TYPE,
                             KP_PK_CUR MODULO.KFK_CUR%TYPE);
    PROCEDURE KSP_UP_MODULO (KP_ID_MODULO MODULO.ID_MODULO%TYPE,
                             KP_NOMBRE_MODULO MODULO.NOMBRE_MODULO%TYPE,
                             KP_PK_CUR MODULO.KFK_CUR%TYPE);
    PROCEDURE KSP_DEL_MODULO(KP_ID_MODULO MODULO.ID_MODULO%TYPE);

END KPK_CRUD_MODULO;
```

```
CREATE OR REPLACE PACKAGE BODY KPK_CRUD_MODULO IS
```

```
--BUSCAR ESTUDIANTE
    PROCEDURE KSP_BUS_MODULO(KP_ID_MODULO MODULO.ID_MODULO%TYPE,
                             KP_MODULO OUT MODULO%ROWTYPE)
    IS
    BEGIN
        SELECT * INTO KP_MODULO
        FROM MODULO
        WHERE ID_MODULO = KP_ID_MODULO;

    EXCEPTION
        WHEN OTHERS THEN
            DBMS_OUTPUT.PUT_LINE('NO SE ENCONTRO DATOS');

    END KSP_BUS_MODULO;

--INSERTAR ESTUDIANTE
    PROCEDURE KSP_INS_MODULO(KP_NOMBRE_MODULO
MODULO.NOMBRE_MODULO%TYPE,
                             KP_PK_CUR MODULO.KFK_CUR%TYPE)
    IS
    BEGIN
        INSERT INTO MODULO
        VALUES(KS_MODULO_ID.NEXTVAL, KP_NOMBRE_MODULO, KP_PK_CUR);
        COMMIT;
    EXCEPTION
        WHEN OTHERS THEN
            DBMS_OUTPUT.PUT_LINE('NO SE PUDO INSERTAR MODULO');
    END KSP_INS_MODULO;

--ACTUALIZAR
```



```

PROCEDURE KSP_UP_MODULO (KP_ID_MODULO MODULO.ID_MODULO%TYPE,
                        KP_NOMBRE_MODULO MODULO.NOMBRE_MODULO%TYPE,
                        KP_PK_CUR MODULO.KFK_CUR%TYPE)

```

```

IS

```

```

BEGIN

```

```

    UPDATE MODULO

```

```

    SET NOMBRE_MODULO=KP_NOMBRE_MODULO, KFK_CUR=KP_PK_CUR

```

```

    WHERE ID_MODULO=KP_ID_MODULO;

```

```

    COMMIT;

```

```

EXCEPTION

```

```

    WHEN OTHERS THEN

```

```

        DBMS_OUTPUT.PUT_LINE('NO SE PUDO ACTUALIZAR MODULO');

```

```

END KSP_UP_MODULO;

```

```

--ELIMINAR

```

```

PROCEDURE KSP_DEL_MODULO(KP_ID_MODULO MODULO.ID_MODULO%TYPE)

```

```

IS

```

```

BEGIN

```

```

    DELETE

```

```

FROM MODULO

```

```

    WHERE ID_MODULO=KP_ID_MODULO;

```

```

    COMMIT;

```

```

EXCEPTION

```

```

    WHEN OTHERS THEN

```

```

        DBMS_OUTPUT.PUT_LINE('NO SE PUDO ELIMINAR MODULO');

```

```

END KSP_DEL_MODULO;

```

```

END KPK_CRUD_MODULO;

```

Nombre del procedimiento: KPK_CRUD_PERSONA

Descripción: Los procedimientos almacenados dentro del paquete KPK_CRUD_PERSONA tienen como funcionalidad hacer las acciones respectivas de nuestra tabla PERSONA (CRUD).

Referencia: Requisito Funcional número 01

```

CREATE SEQUENCE KS_PERSONA_ID

```

```

START WITH 9

```

```

INCREMENT BY 1;

```

```

CREATE OR REPLACE PACKAGE KPK_CRUD_PERSONA IS

```

```

    PROCEDURE KSP_BUS_PERSONA(KP_ID_PERSONA
PERSONA.ID_PERSONA%TYPE,

```

```

        KP_PERSONA OUT PERSONA%ROWTYPE);

```

```

    PROCEDURE KSP_INS_PERSONA(KP_NOMBRE PERSONA.NOMBRE%TYPE,

```

```

        KP_APELLIDO PERSONA.APELLIDO%TYPE,

```

```

        KP_DNI PERSONA.DNI%TYPE,
        KP_CORREO PERSONA.CORREO%TYPE,
        KP_CONTRASEÑA PERSONA.CONTRASEÑA%TYPE,
        KP_EDAD PERSONA.EDAD%TYPE,
        KP_SEXO PERSONA.SEXO%TYPE);
PROCEDURE KSP_UP_PERSONA (KP_ID_PERSONA PERSONA.ID_PERSONA%TYPE,
        KP_NOMBRE PERSONA.NOMBRE%TYPE,
        KP_APELLIDO PERSONA.APELLIDO%TYPE,
        KP_DNI PERSONA.DNI%TYPE,
        KP_CORREO PERSONA.CORREO%TYPE,
        KP_CONTRASEÑA PERSONA.CONTRASEÑA%TYPE,
        KP_EDAD PERSONA.EDAD%TYPE,
        KP_SEXO PERSONA.SEXO%TYPE);
PROCEDURE KSP_DEL_PERSONA (KP_ID_PERSONA PERSONA.ID_PERSONA%TYPE);
END KPK_CRUD_PERSONA;

CREATE OR REPLACE PACKAGE BODY KPK_CRUD_PERSONA IS
    --BUSCAR PERSONA
    PROCEDURE KSP_BUS_PERSONA(KP_ID_PERSONA
PERSONA.ID_PERSONA%TYPE,
                                KP_PERSONA OUT PERSONA%ROWTYPE)
    IS
    BEGIN
        SELECT * INTO KP_PERSONA
        FROM PERSONA
        WHERE ID_PERSONA = KP_ID_PERSONA;

    EXCEPTION
        WHEN OTHERS THEN
            DBMS_OUTPUT.PUT_LINE('NO SE ENCONTRO A LA PERSONA');

    END KSP_BUS_PERSONA;

    --INSERTAR PERSONA
    PROCEDURE KSP_INS_PERSONA(KP_NOMBRE PERSONA.NOMBRE%TYPE,
        KP_APELLIDO PERSONA.APELLIDO%TYPE,
        KP_DNI PERSONA.DNI%TYPE,
        KP_CORREO PERSONA.CORREO%TYPE,
        KP_CONTRASEÑA PERSONA.CONTRASEÑA%TYPE,
        KP_EDAD PERSONA.EDAD%TYPE,
        KP_SEXO PERSONA.SEXO%TYPE)
    IS
    BEGIN
        INSERT INTO PERSONA
        VALUES(KS_PERSONA_ID.NEXTVAL, KP_NOMBRE, KP_APELLIDO, KP_DNI,
KP_CORREO, KP_CONTRASEÑA, KP_EDAD, KP_SEXO, '1');
        COMMIT;
    EXCEPTION
        WHEN OTHERS THEN

```

```
DBMS_OUTPUT.PUT_LINE('NO SE PUDO INSERTAR PERSONA');  
END KSP_INS_PERSONA;
```

```
--ACTUALIZAR PERSONA
```

```
PROCEDURE KSP_UP_PERSONA (KP_ID_PERSONA  
PERSONA.ID_PERSONA%TYPE,  
KP_NOMBRE PERSONA.NOMBRE%TYPE,  
KP_APELLIDO PERSONA.APELLIDO%TYPE,  
KP_DNI PERSONA.DNI%TYPE,  
KP_CORREO PERSONA.CORREO%TYPE,  
KP_CONTRASEÑA PERSONA.CONTRASEÑA%TYPE,  
KP_EDAD PERSONA.EDAD%TYPE,  
KP_SEXO PERSONA.SEXO%TYPE)  
  
IS  
BEGIN  
    UPDATE PERSONA  
    SET NOMBRE=KP_NOMBRE, APELLIDO=KP_APELLIDO, DNI=KP_DNI,  
CORREO=KP_CORREO, CONTRASEÑA=KP_CONTRASEÑA, EDAD=KP_EDAD,  
SEXO=KP_SEXO  
    WHERE ID_PERSONA=KP_ID_PERSONA;  
    COMMIT;  
EXCEPTION  
    WHEN OTHERS THEN  
        DBMS_OUTPUT.PUT_LINE('NO SE PUDO ACTUALIZAR PERSONA');  
  
END KSP_UP_PERSONA;
```

```
--ELIMINAR PERSONA
```

```
PROCEDURE KSP_DEL_PERSONA (KP_ID_PERSONA  
PERSONA.ID_PERSONA%TYPE)  
IS  
BEGIN  
    UPDATE PERSONA  
SET ESTADO = '0'  
    WHERE ID_PERSONA=KP_ID_PERSONA;  
    COMMIT;  
EXCEPTION  
    WHEN OTHERS THEN  
        DBMS_OUTPUT.PUT_LINE('NO SE PUDO ELIMINAR PERSONA');  
END KSP_DEL_PERSONA;  
  
END KPK_CRUD_PERSONA;
```

Nombre del procedimiento: KPK_CRUD_USUARIO

Descripción: Los procedimientos almacenados dentro del paquete KPK_CRUD_USUARIO tienen como funcionalidad hacer las acciones respectivas de nuestra tabla USUARIO (CRUD).

Referencia: Requisito Funcional número 01

CREATE OR REPLACE PACKAGE KPK_CRUD_USUARIO IS

```
    PROCEDURE KSP_BUS_USUARIO(KP_ID_USUARIO USUARIO.KFK_PER%TYPE,
                               KP_USUARIO OUT USUARIO%ROWTYPE);
    PROCEDURE KSP_INS_USUARIO(KP_ADMIN USUARIO.ADMIN%TYPE,
                               KP_KFK_PER USUARIO.KFK_PER%TYPE);
    PROCEDURE KSP_UP_USUARIO(KP_ID_USUARIO USUARIO.KFK_PER%TYPE,
                              KP_ADMIN USUARIO.ADMIN%TYPE);
    PROCEDURE KSP_DEL_USUARIO(KP_ID_USUARIO USUARIO.KFK_PER%TYPE);
```

END KPK_CRUD_USUARIO;

CREATE OR REPLACE PACKAGE BODY KPK_CRUD_USUARIO IS

--BUSCAR USUARIO

```
    PROCEDURE KSP_BUS_USUARIO(KP_ID_USUARIO USUARIO.KFK_PER%TYPE,
                               KP_USUARIO OUT USUARIO%ROWTYPE)
```

IS

BEGIN

```
    SELECT * INTO KP_USUARIO
    FROM USUARIO
    WHERE KFK_PER = KP_ID_USUARIO;
```

EXCEPTION

WHEN OTHERS THEN

```
    DBMS_OUTPUT.PUT_LINE('NO SE ENCONTRO AL USUARIO');
```

END KSP_BUS_USUARIO;

--INSERTAR USUARIO

```
    PROCEDURE KSP_INS_USUARIO(KP_ADMIN USUARIO.ADMIN%TYPE,
                               KP_KFK_PER USUARIO.KFK_PER%TYPE)
```

IS

BEGIN

```
    INSERT INTO USUARIO
    VALUES(KP_ADMIN,KP_KFK_PER);
    COMMIT;
```

EXCEPTION

WHEN OTHERS THEN

```
    DBMS_OUTPUT.PUT_LINE('NO SE PUDO INSERTAR EL USUARIO');
```

END KSP_INS_USUARIO;

--ACTUALIZAR USUARIO

```
    PROCEDURE KSP_UP_USUARIO(KP_ID_USUARIO USUARIO.KFK_PER%TYPE,
                              KP_ADMIN USUARIO.ADMIN%TYPE)
```

IS

BEGIN

```
    UPDATE USUARIO
```

```

        SET admin=KP_ADMIN
        WHERE KFK_PER=KP_ID_USUARIO;
        COMMIT;
    EXCEPTION
        WHEN OTHERS THEN
            DBMS_OUTPUT.PUT_LINE('NO SE PUDO ACTUALIZAR USUARIO');

    END KSP_UP_USUARIO;

--ELIMINAR USUARIO
    PROCEDURE KSP_DEL_USUARIO(KP_ID_USUARIO USUARIO.KFK_PER%TYPE)
    IS
    BEGIN
        UPDATE PERSONA
        SET ESTADO = '0'
        WHERE ID_PERSONA=KP_ID_USUARIO;

    DELETE USUARIO
    WHERE KFK_PER=KP_ID_USUARIO;
        COMMIT;
    EXCEPTION
        WHEN OTHERS THEN
            DBMS_OUTPUT.PUT_LINE('NO SE PUDO ELIMINAR USUARIO');
    END KSP_DEL_USUARIO;

END KPK_CRUD_USUARIO;

```

10. FUNCIONES

| |
|---|
| Nombre de la función: FK_VER_DOCENTE |
| Nombre Paquete: -- |
| Descripción: La función FK_VER_DOCENTE tiene como objetivo: transformar el ID pasado por un parámetro para traer el nombre y el apellido del docente. |
| Referencia: tabla persona y docente |
| <pre> CREATE OR REPLACE FUNCTION FK_VER_DOCENTE(P_ID_DOCENTE NUMBER) RETURN VARCHAR2 IS NOMBRE_DOCENTE VARCHAR2(65); BEGIN SELECT NOMBRE ' ' APELLIDO INTO NOMBRE_DOCENTE FROM PERSONA WHERE ID_PERSONA=P_ID_DOCENTE; </pre> |

```
RETURN NOMBRE_DOCENTE;  
END;
```

Nombre de la función: FK_TRANSFORM_EDAD

Nombre Paquete: --

Descripción: La función FK_TRANSFORM_EDAD tiene como objetivo: **transformar la fecha de nacimiento del usuario en años contables.**

Referencia: tabla persona

```
CREATE OR REPLACE FUNCTION FK_TRANSFORM_EDAD(P_NACIMIENTO  
DATE)  
RETURN NUMBER  
IS  
EDAD NUMBER(3);  
BEGIN  
    EDAD := ROUND((SYSDATE-P_NACIMIENTO)/360);  
    RETURN EDAD;  
END;
```

Nombre de la función: FK_TRANSFORM_SEXO

Nombre Paquete:

Descripción: La función FK_TRANSFORM_SEXO tiene como objetivo: transformar el sexo, donde si es F se transformará en Femenino y si es M en Masculino.

Referencia: tabla persona

```
CREATE OR REPLACE FUNCTION FK_TRANSFORM_SEXO(P_ID_PERSONA  
NUMBER)  
RETURN VARCHAR2  
IS  
SEXO VARCHAR2(20);  
SEXO_RETURN VARCHAR2(20);  
BEGIN  
    SELECT SEXO INTO SEXO  
    FROM PERSONA  
    WHERE ID_PERSONA=P_ID_PERSONA;  
    IF SEXO = 'F' THEN  
        SEXO_RETURN := 'FEMENINO';  
    ELSE  
        SEXO_RETURN := 'MASCULINO';  
    END IF;  
    RETURN SEXO_RETURN;
```

END;

Nombre de la función: FK_TRANSFORM_VALORACION

Nombre Paquete: --

Descripción: La función FK_TRANSFORM_VALORACION tiene como objetivo: **transformar la valoración 2 que un usuario dio a un curso en letras.**

Referencia: tabla inscripción

```
CREATE OR REPLACE FUNCTION FK_TRANSFORM_VALORACION(P_ID_VAL
NUMBER)
RETURN VARCHAR2
IS
VALORACION VARCHAR2(5);
VALORACION_RETURN VARCHAR2(5);
BEGIN
    SELECT VALORACION INTO VALORACION
    FROM INSCRIPCION
    WHERE ID_INSCRIPTION=P_ID_VAL;
    IF VALORACION = 2 THEN
        VALORACION_RETURN := 'Dos';
    END IF;
    RETURN VALORACION_RETURN;
END;
```

11. CONSULTAS

Reporte 01

Nombre Reporte: INFO_DOCENTE

Descripción: Esta consulta INFO_DOCENTE tiene como objetivo: **mostrar la información relevante del docente junto con la especialización y los cursos que este instruye.**

Referencia: tabla persona, docente, especialización y curso

```
SELECT P.ID_PERSONA, P.NOMBRE, P.APELLIDO, P.DNI,P.CORREO,
P.CONTRASEÑA,
P.EDAD, P.SEXO, P.ESTADO, D.SUELDO, E.ESPECIALIZACION,
C.NOMBRE_CURSO
FROM PERSONA P JOIN DOCENTE D
ON P.ID_PERSONA=D.KFK_PER JOIN ESPECIALIZACION E
ON E.ID_ESPECIALIZACION=D.KFK_ESPE JOIN CURSO C
ON C.KFK_DOCE=D.KFK_PER
WHERE P.ESTADO=1;
```

| Reporte 02 |
|---|
| Nombre Reporte: ADMIN_USUARIO |
| Descripción: Esta consulta ADMIN_USUARIO tiene como objetivo: mostrar los datos (ID, NOMBRE, APELLIDO, DNI, CORREO, CONTRASEÑA EDAD, SEXO, ESTADO) del usuario(s) administrador(es) |
| Referencia: tabla persona y usuario |
| <pre>SELECT P.ID_PERSONA, P.NOMBRE, P.APELLIDO, P.DNI,P.CORREO, P.CONTRASEÑA, P.EDAD, P.SEXO, P.ESTADO, U.ADMIN FROM PERSONA P JOIN USUARIO U ON P.ID_PERSONA=U.KFK_PER WHERE P.ESTADO=1 AND U.ADMIN=1;</pre> |

| Reporte 03 |
|--|
| Nombre Reporte: CLASS_MOD |
| Descripción: Esta consulta CLASS_MOD tiene como objetivo: mostrar las clases del módulo cuando su PK es igual a 1 |
| Referencia: tabla clase y módulo |
| <pre>SELECT C.NOMBRE_CLASE, M.NOMBRE_MODULO FROM CLASE C JOIN MODULO M ON C.ID_CLASE=M.KFK_CUR WHERE M.ID_MODULO=1</pre> |

| Reporte 04 |
|---|
| Nombre Reporte: M_USUS |
| Descripción: Esta consulta M_USUS tiene como objetivo: mostrar el nombre, ID de todos los usuarios que su sexo=M |
| Referencia: tabla persona e inscripción |
| <pre>SELECT P.NOMBRE ,P.ID_PERSONA,P.SEXO FROM PERSONA P JOIN INSCRIPCION I ON P.ID_PERSONA=I.KFK_PER WHERE SEXO='M';</pre> |

12. VISTAS

Vista 01

Nombre Vista: KV_DOCENTE

Descripción: Esta vista KV_DOCENTE tiene como objetivo: **mostrar todos los usuarios docentes con estado 1**

Referencia: Tabla persona y especialización

```
CREATE OR REPLACE VIEW KV_DOCENTE AS
SELECT P.ID_PERSONA, P.NOMBRE, P.APELLIDO, P.DNI,P.CORREO,
P.CONTRASEÑA,
P.EDAD, P.SEXO, P.ESTADO, D.SUELDO, E.ESPECIALIZACION,
C.NOMBRE_CURSO
FROM PERSONA P JOIN DOCENTE D
ON P.ID_PERSONA=D.KFK_PER JOIN ESPECIALIZACION E
ON E.ID_ESPECIALIZACION=D.KFK_ESPE JOIN CURSO C
ON C.KFK_DOCE=D.KFK_PER
WHERE P.ESTADO=1;

--Consultar VIEW
SELECT * FROM KV_DOCENTE;
```

Vista 02

Nombre Vista: KV_USUARIO

Descripción: Esta vista KV_USUARIO tiene como objetivo: **mostrar si el usuario es administrador o no, sabiendo que 1=admin y 0=no admin**

Referencia: tabla persona y usuario

```
CREATE OR REPLACE VIEW KV_USUARIO AS
SELECT P.ID_PERSONA, P.NOMBRE, P.APELLIDO, P.DNI,P.CORREO,
P.CONTRASEÑA,
P.EDAD, P.SEXO, P.ESTADO, U.ADMIN
FROM PERSONA P JOIN USUARIO U
ON P.ID_PERSONA=U.KFK_PER
WHERE P.ESTADO=1;

--Consultando vista
SELECT * FROM KV_USUARIO;
```

Vista 03

Nombre Reporte: KV_CURSO

Descripción: Esta vista KV_CURSO tiene como objetivo: **mostrar usuarios y cursos a los**

que se han inscrito

Referencia: Tabla persona y curso

```
CREATE OR REPLACE VIEW KV_CURSO AS
SELECT P.ID_PERSONA, P.NOMBRE, P.APELLIDO, P.DNI, P.CORREO,
P.CONTRASEÑA, P.EDAD, P.SEXO, P.ESTADO, C.NOMBRE_CURSO
FROM PERSONA P JOIN CURSO C
ON P.ID_PERSONA=C.ID_CURSO
WHERE P.ESTADO=1;
```

--Consultando vista

```
SELECT * FROM KV_CURSO;
```

Vista 04

Nombre Reporte: KV_INSCRIPCION

Descripción: Esta vista KV_INSCRIPCION tiene como objetivo: **mostrar todos los usuarios incritos en cursos y estan desactivos.**

Referencia: Tabla inscripción y persona

```
CREATE VIEW KV_INSCRIPCION AS
SELECT P.NOMBRE NOMBRE_USER ,P.ID_PERSONA ID_PERSONA,I.ESTADO
ESTADO_ACTUAL
FROM PERSONA P JOIN INSCRIPCION I ON P.ID_PERSONA=I.KFK_PER
WHERE I.ESTADO=0;
```

--Consultando vista

```
SELECT * FROM KV_INSCRIPCION;
```