**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](#_Toc77093067)

[2. REQUERIMIENTOS 3](#_Toc77093068)

[2.1. REQUERIMIENTOS FUNCIONALES 3](#_Toc77093069)

[3. REFERENCIA 4](#_Toc77093070)

[3.1. DESCRIPCIÓN DE ESTÁNDAR DE BASE DE DATOS. 4](#_Toc77093071)

[4. SERVIDOR DE BASE DE DATOS 4](#_Toc77093087)

[5. ESQUEMA CONCEPTUAL DE BASE DE DATOS 5](#_Toc77093088)

[6. ESQUEMA LÓGICO DE BASE DE DATOS 5](#_Toc77093090)

[7. ESQUEMA FÍSICO DE BASE DE DATOS 5](#_Toc77093092)

[8. DISEÑO DE TABLAS 6](#_Toc77093094)

[8.1. TABLAS 6](#_Toc77093095)

[8.1.1. CLASE 6](#_Toc77093096)

[8.1.2. CURSO 6](#_Toc77093097)

[8.1.3. DOCENTE 7](#_Toc77093098)

[8.1.4. ESPECIALIZACION 8](#_Toc77093099)

[8.1.5. INSCRIPCION 8](#_Toc77093100)

[8.1.6. INSCRIPCIONCLASE 9](#_Toc77093101)

[8.1.7. MODULO 10](#_Toc77093102)

[8.1.8. PERSONA 11](#_Toc77093103)

[8.1.9. USUARIO 12](#_Toc77093104)

[9. PROCEDIMIENTOS ALMACENADOS 12](#_Toc77093105)

[10. FUNCIONES 29](#_Toc77093106)

[11. CONSULTAS 31](#_Toc77093107)

[12. VISTAS 32](#_Toc77093108)

1. INTRODUCCIÓN
2. REQUERIMIENTOS
   1. REQUERIMIENTOS FUNCIONALES

### 

|  |  |  |
| --- | --- | --- |
| **N°** | **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 ira 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 modulos que pertenecen a los cursos. |
| RN07 | Valoración de Curso | Una vez terminado un curso el usuario podrá dar una valoración de este. |

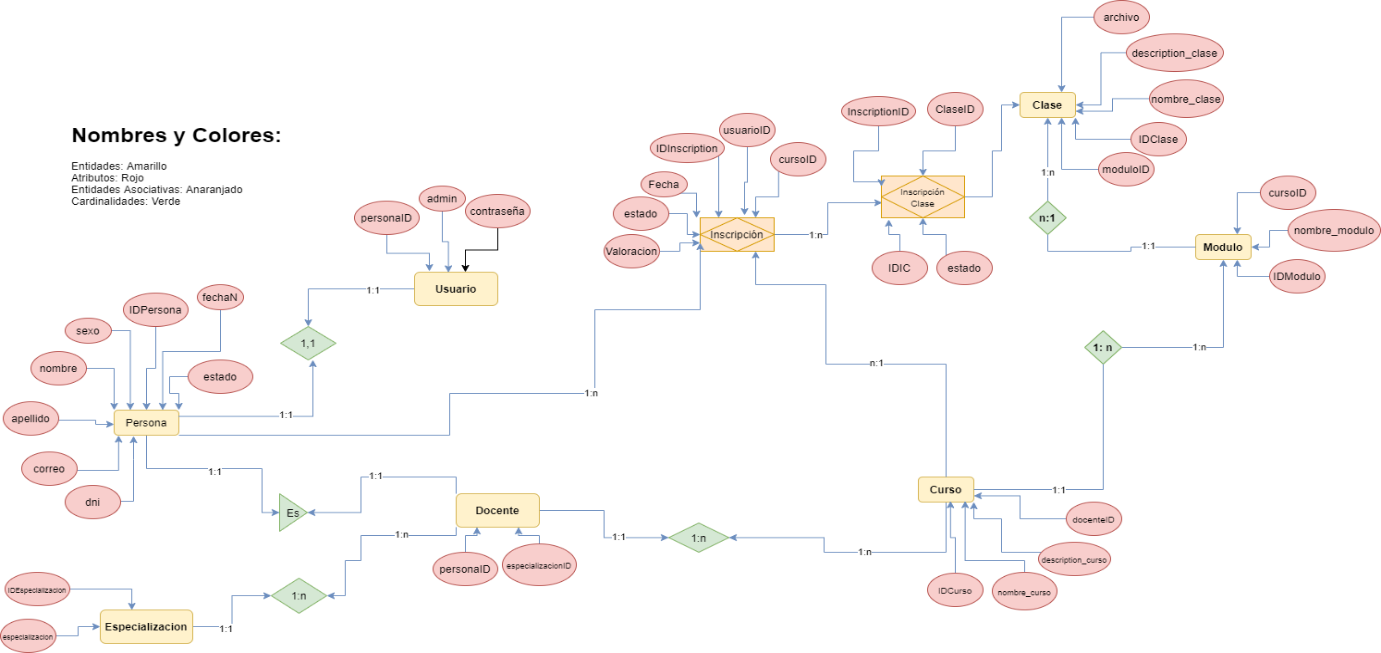
1. REFERENCIA
   1. DESCRIPCIÓN DE ESTÁNDAR DE BASE DE DATOS.
      * + Los nombres de la tabla iran con mayúscula; por ejemplo: USUARIO, CLASE, CURSO, etc.
        + Los nombres de las columas (atributos) también iran cn mayúscula.

* Para los CRUD
* Secuencias: el nombre de la secuencia del CRUD ira KS\_NOMBRETABLA\_ID
* Cabecera: la cabecedera 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 mayúculas asi como todo

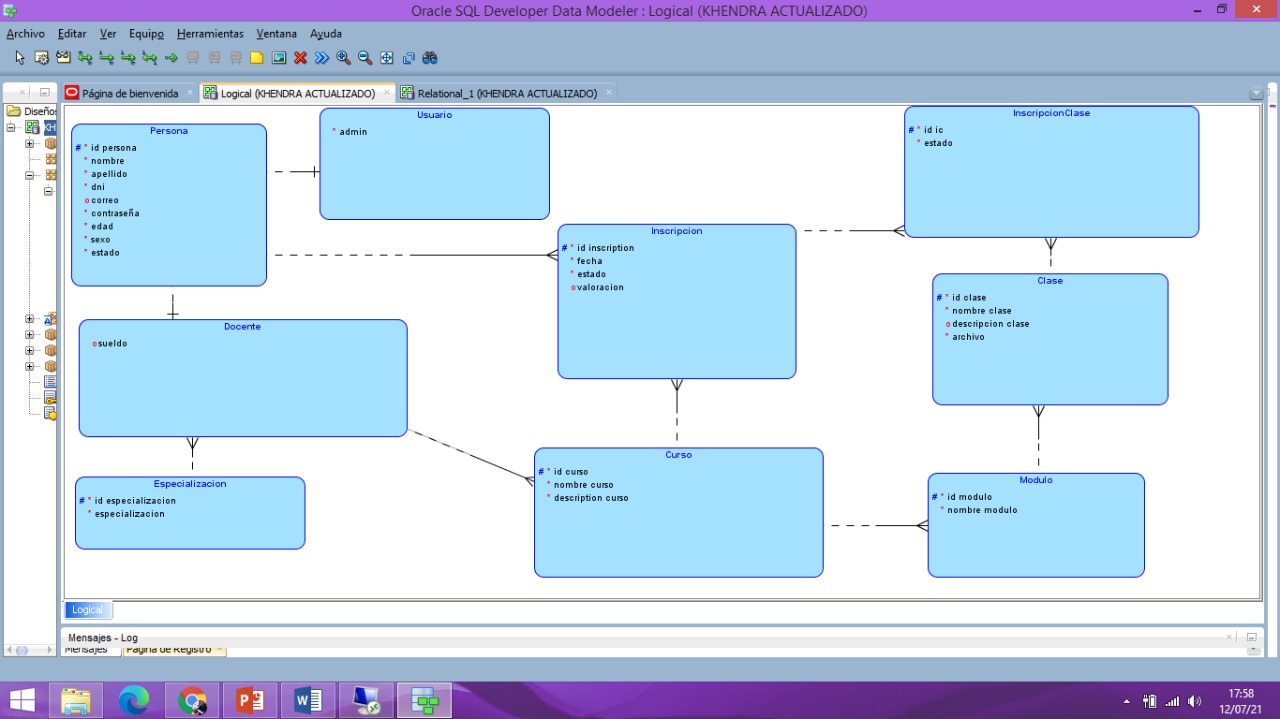
1. SERVIDOR DE BASE DE DATOS

|  |  |
| --- | --- |
| **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 |

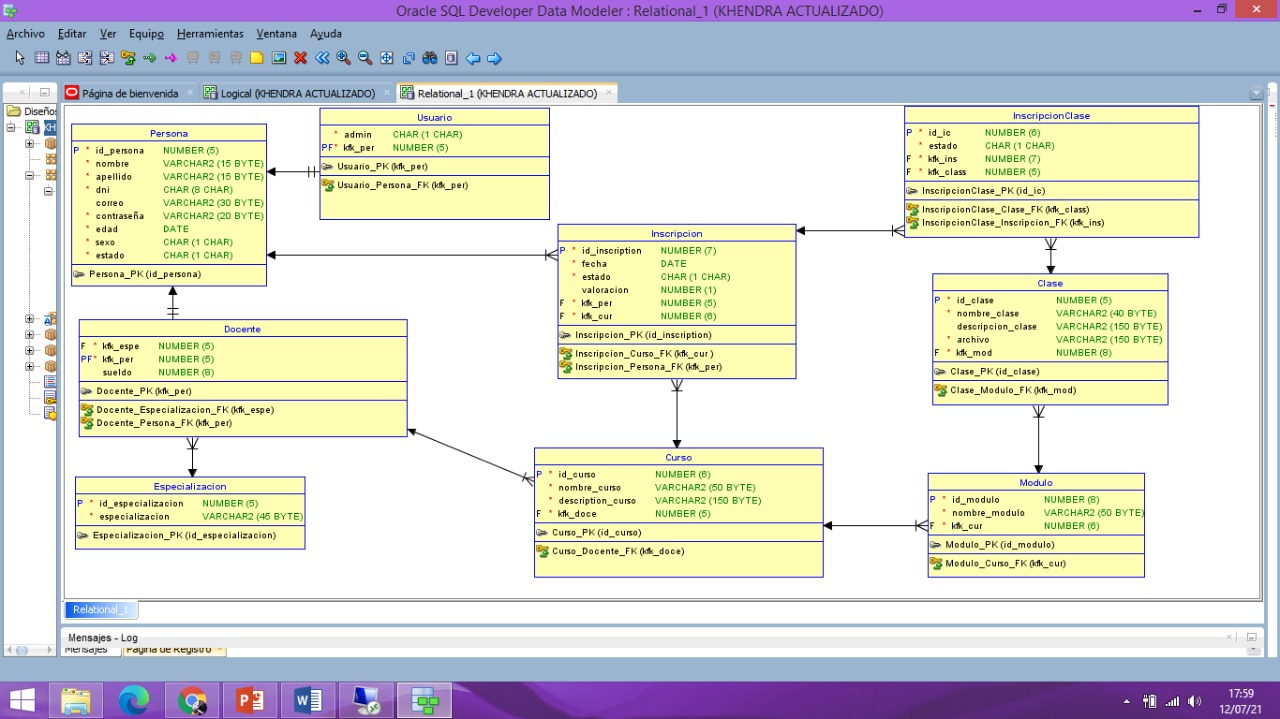
1. ESQUEMA CONCEPTUAL DE BASE DE DATOS



1. ESQUEMA LÓGICO DE BASE DE DATOS



1. ESQUEMA FÍSICO DE BASE DE DATOS



1. DISEÑO DE TABLAS
   1. TABLAS
      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 | | | |

* + 1. 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 | | | |

* + 1. 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 | | |

* + 1. 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 |

* + 1. 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 | |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ***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 |

* + 1. 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 | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***Indices*** |  |  | |  |
| **Nombre índice** | **Nombre de la columna** | | **Tipo** | | |
| INSCRIPCIONCLASE\_PK | ID\_IC | | NORMAL UNIQUE | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***Constraints*** | |  |  |  |  |
| **Tipo** | **Nombre de restricción** | | | | **Columna** |
| C | SYS\_C008012 | | | | ID\_IC |
| C | SYS\_C008013 | | | | ESTADO |
| C | SYS\_C008014 | | | | KFK\_INS |
| C | SYS\_C008015 | | | | KFK\_CLASS |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ***Llaves foráneas (REFERENCIA A)*** | |  |  |  |  |  |  |
| **Nombre** | **Referencia a** | | **Obligatorio** | **Nombre de columna** | | |
| INSCRIPCIONCLASE.INSCRIPCIONCLASE\_CLASE\_FK | CLASE | | N | KFK\_CLASS | | |
| INSCRIPCIONCLASE.INSCRIPCIONCLASE\_INSCRIPCION\_FK | INSCRIPCION | | N | KFK\_INS | | |

* + 1. MODULO

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

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ***Columnas*** | |  |  |  |  |  |  |
| **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 |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***Indices*** |  |  | |  |
| **Nombre índice** | **Nombre de la columna** | | **Tipo** | | |
| MODULO\_PK | ID\_MODULO | | NORMAL UNIQUE | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***Constraints*** | |  |  |  |  |
| **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 | | |

* + 1. 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 |

* + 1. 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 | | |

1. PROCEDIMIENTOS ALMACENADOS

|  |
| --- |
| Nombre del procedimiento: KPK\_CRUD\_CLASE |
| Descripción: Los procedimientos almacenados dentro del paquete KPK\_CRUD\_CLASE tienen como funcionalisdad 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 funcionalisdad 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.DESCRIPTION\_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.DESCRIPTION\_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.DESCRIPTION\_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.DESCRIPTION\_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 funcionalisdad 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 funcionalisdad 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 funcionalisdad 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\_PER,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\_INCRIPCIONCLASE |
| Descripción: Los procedimientos almacenados dentro del paquete KPK\_CRUD\_INCRIPCIONCLASE tienen como funcionalisdad 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 funcionalisdad 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 funcionalisdad 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 funcionalisdad 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; |

1. 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 |
| 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;  RETURN NOMBRE\_DOCENTE;  END; |

|  |
| --- |
| Nombre del 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 del 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 del la función: FK\_TRANSFORM\_VALORACION |
| Nombre Paquete: -- |
| Descripción: La función FK\_TRANSFORM\_VALORACION tiene como objetivo: **tranformar la valoración 2 que un usuario dio a un curso en letras.** |
| Referencia: tabla inscrpción |
| 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; |

1. CONSULTAS

|  |
| --- |
| Reporte 01 |
| Nombre Reporte: INFO\_DOCENTE |
| Descripción: Esta consulta INFO\_DOCENTE tiene como objetivo: **mostrar la informacion relevante del docente junto con la especializacion 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) administador(es)** |
| Referencia: tabla persona y usuario |
| 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; |

|  |
| --- |
| 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 |
| 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 |

|  |
| --- |
| 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 |
| SELECT P.NOMBRE ,P.ID\_PERSONA,P.SEXO  FROM PERSONA P JOIN INSCRIPCION I ON P.ID\_PERSONA=I.KFK\_PER  WHERE SEXO='M'; |

1. 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; |