



## **Sistema de Matricula**

Bases de Datos I

**Instituto Tecnológico de Costa Rica**

Grupo 20

### **Estudiantes:**

Araya Morera Adrián

Cordero Díaz Jesús Gabriel

López Sosa Nicole Alejandra

### **Profesor:**

Shum Chan Alberto

II Semestre 2022

## Contenido

Resumen .....	3
Introducción .....	3
Desarrollo .....	4
Creación del diagrama del modelo Entidad-Relación .....	4

## **Resumen**

Se propone un sistema de base de datos el cual al momento solo maneja la información de un periodo (semestre) y se solicita que se realicen los cambios y/o ajustes necesarios para conseguir manejar múltiples periodos y años, ejemplo: periodo 1-2018, periodo 2-2018, periodo 1-2019... Además, se solicita la incorporación de la posibilidad para el manejo de requisitos de las diferentes materias.

## **Introducción**

Se planteará un diseño con base en los conocimientos adquiridos en modelaje, procedimiento almacenados, funciones, cursores y tablas temporales. En dos áreas específicas:

- Proceso de diseño de bases de datos (entidades, atributos, asociaciones/relaciones, cardinalidades, restricciones de integridad)
- Implementación de procedimientos almacenados, funciones, cursores, etc. para resolver las peticiones solicitadas.

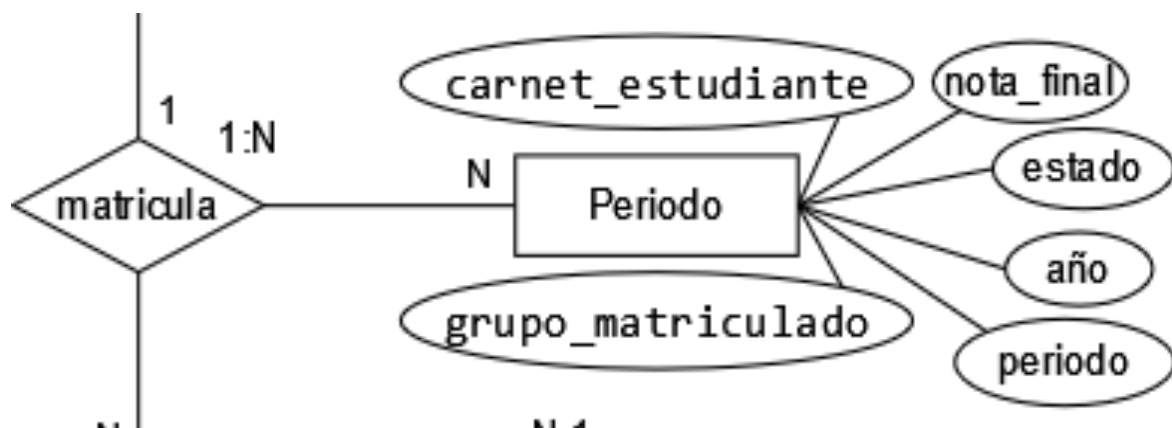
## Desarrollo

### Creación del diagrama del modelo Entidad-Relación

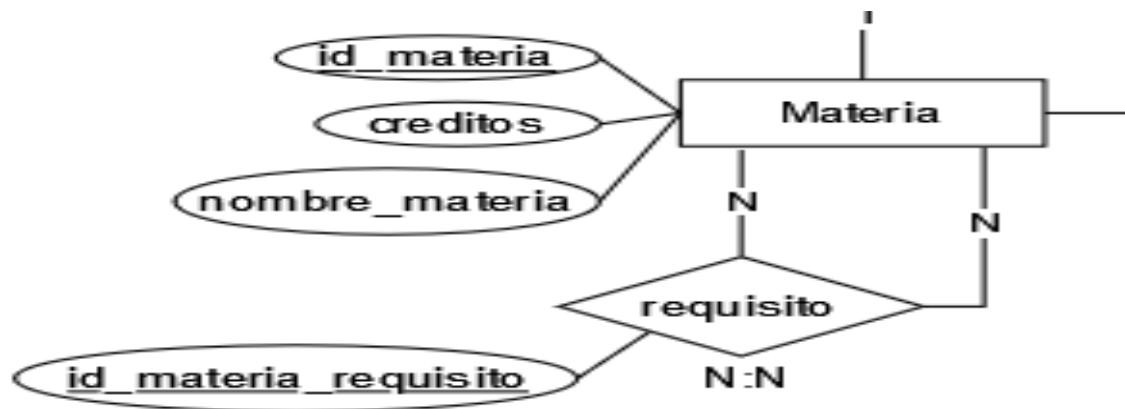
Para la realización del diseño se realizó un análisis de la información suministrada y se utilizó como base la descripción de las solicitudes proporcionados por el cliente, para efectos académicos se entenderá como el cliente la especificación del proyecto y el profesor. De esta forma se identificaron los elementos necesarios, así como aquellos que no estaban presentes en la descripción, sin embargo, fueron definidos en el proceso de análisis e interpretación de la especificación. Las solicitudes del cliente que deben de ser implementadas corresponden a las siguientes:

- *El modelo de datos del sistema de matrícula que hemos desarrollado hasta el momento solo maneja la información de un periodo (semestre) por lo que se quiere hacer los ajustes necesarios para que permita el manejo de múltiples periodos de modo que se pueda mantener la información de varios años. Por ejemplo, periodo 1-2018, período 2-2018, periodo 1-2019, etc.*
- *La otra mejora que se desea incorporar en el modelo es la posibilidad de poder manejar el esquema de requisitos para las materias/cursos, pero solo se tomaran en cuenta los requisitos, no los correquisitos. De modo de que, si algún estudiante intenta matricular un curso, se debe validar si cumple con los requisitos.*

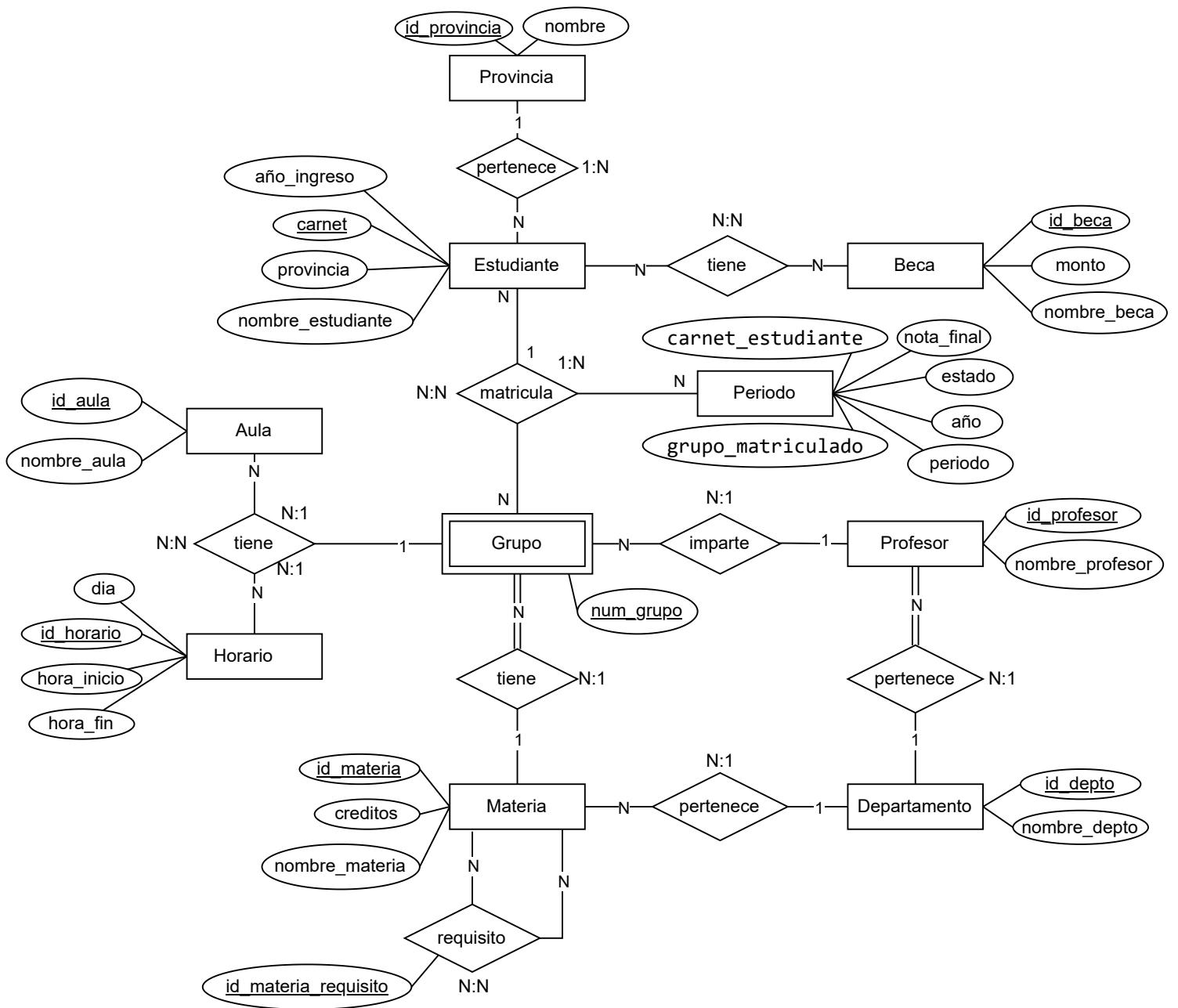
Para el manejo de diferentes periodos se creó una entidad periodo, en la cual se estarán acumulados dichos periodos, esta entidad almacenara el carnet del estudiante, el grupo que matriculo, la nota final (de dicho estudiante), el estado (Aprobado o Reprobado), el año en el cual realizo la matricula y el periodo matricula (1 para el primer semestre y 2 para el segundo).



Para el manejo de los requisitos de las diferentes materias se implementó mediante una relación (N:N) consigo misma, la tabla materia, de la cual se crearía una tabla intermediara, la cual manejaría a la materia "x" y sus diferentes "materias requisito".



Como resultando de todas las implementaciones y solicitudes por parte del cliente se tiene el siguiente diagrama Entidad-Relación:



## Código utilizado para el desarrollo de las funcionalidades solicitadas

```
set serveroutput on;

CREATE TABLE provincia (
    id_provincia NUMBER NOT NULL,
    nombre        VARCHAR2(20) NOT NULL,
    CONSTRAINT id_provincia_pk PRIMARY KEY ( id_provincia )
);

CREATE TABLE estudiante (
    carnet        NUMBER NOT NULL,
    nombre        VARCHAR2(20) NOT NULL,
    provincia     NUMBER NOT NULL,
    anno_ingreso  VARCHAR2(20) NOT NULL,
    CONSTRAINT fk_provincia FOREIGN KEY ( provincia )
        REFERENCES provincia ( id_provincia ),
    CONSTRAINT carnet_pk PRIMARY KEY ( carnet )
);

CREATE TABLE beca (
    id_beca       NUMBER NOT NULL,
    monto         FLOAT,
    nombre_beca   VARCHAR2(20),
    CONSTRAINT id_beca_pk PRIMARY KEY ( id_beca )
);

CREATE TABLE aula (
    id_aula       NUMBER NOT NULL,
    nombre_aula   VARCHAR2(20) NOT NULL,
    CONSTRAINT id_aula_pk PRIMARY KEY ( id_aula )
);

CREATE TABLE horario (
    id_horario    NUMBER NOT NULL,
    dia           VARCHAR2(20),
    hora_inicio   VARCHAR2(20),
    hora_fin      VARCHAR2(20),
    CONSTRAINT id_horario_pk PRIMARY KEY ( id_horario )
);

CREATE TABLE materia (
    id_materia    NUMBER NOT NULL,
    creditos      NUMBER(1),
    nombre_materia VARCHAR2(50) NOT NULL,
    CONSTRAINT id_materia_pk PRIMARY KEY ( id_materia )
);

CREATE TABLE profesor (
    id_profesor   NUMBER NOT NULL,
    nombre_profesor VARCHAR2(20) NOT NULL,
    CONSTRAINT id_profesor_pk PRIMARY KEY ( id_profesor )
);

CREATE TABLE departamento (
    id_depto      NUMBER NOT NULL,
    nombre_depto  VARCHAR2(20) NOT NULL,
    CONSTRAINT id_depto_pk PRIMARY KEY ( id_depto )
);

CREATE TABLE grupo (
    num_grupo     NUMBER NOT NULL,
    id_profesor_grupo NUMBER NOT NULL,
    id_horario_grupo NUMBER NOT NULL,
    id_aula_grupo  NUMBER NOT NULL,
    id_materia_grupo NUMBER NOT NULL,
    CONSTRAINT fk_id_profesor_grupo FOREIGN KEY ( id_profesor_grupo )
        REFERENCES profesor ( id_profesor ),
    CONSTRAINT fk_id_horario_grupo FOREIGN KEY ( id_horario_grupo )
        REFERENCES horario ( id_horario ),
```

```

CONSTRAINT fk_id_aula_grupo FOREIGN KEY ( id_aula_grupo )
    REFERENCES aula ( id_aula ),
CONSTRAINT fk_id_materia_grupo FOREIGN KEY ( id_materia_grupo )
    REFERENCES materia ( id_materia ),
CONSTRAINT num_grupo_pk PRIMARY KEY ( num_grupo )
);

CREATE TABLE matricula (
    id_matricula          NUMBER
        GENERATED BY DEFAULT ON NULL AS IDENTITY,
    carnet_matricula      NUMBER NOT NULL,
    id_grupo_matriculado  NUMBER NOT NULL,
    CONSTRAINT fk_carnet_matricula FOREIGN KEY ( carnet_matricula )
        REFERENCES estudiante ( carnet ),
    CONSTRAINT fk_id_grupo_matriculado FOREIGN KEY ( id_grupo_matriculado )
        REFERENCES grupo ( num_grupo ),
    PRIMARY KEY ( id_matricula,
        carnet_matricula,
        id_grupo_matriculado )
);

-- La tabla periodo conservara el registro de los cursos ya finalizados por los
-- estudiantes. Similar a las actas utilizadas a en las entregas de actas oficiales.
-- Se utilizará para el cálculo de ponderados y la comprobación de requisitos.
CREATE TABLE periodo (
    periodo              NUMBER(1),
    carnet_estudiante    NUMBER NOT NULL,
    id_grupo_matriculado NUMBER NOT NULL,
    anno                 VARCHAR2(4),
    nota_final           NUMBER(3),
    estado               VARCHAR2(20),
    CONSTRAINT fk_carnet_estudiante_periodo FOREIGN KEY ( carnet_estudiante )
        REFERENCES estudiante ( carnet ),
    CONSTRAINT fk_id_grupo_matriculado_periodo FOREIGN KEY ( id_grupo_matriculado )
        REFERENCES grupo ( num_grupo ),
    PRIMARY KEY ( periodo,
        carnet_estudiante,
        id_grupo_matriculado )
);

--N:N
CREATE TABLE estudiante_beca (
    carnet_eb  NUMBER NOT NULL,
    id_beca_eb NUMBER NOT NULL,
    CONSTRAINT fk_carnet_eb FOREIGN KEY ( carnet_eb )
        REFERENCES estudiante ( carnet ),
    CONSTRAINT fk_id_beca_eb FOREIGN KEY ( id_beca_eb )
        REFERENCES beca ( id_beca ),
    PRIMARY KEY ( carnet_eb,
        id_beca_eb )
);

ALTER TABLE profesor ADD (
    id_departamento_profesor NUMBER NOT NULL
)
    ADD CONSTRAINT fk_id_departamento_profesor FOREIGN KEY ( id_departamento_profesor )
        REFERENCES departamento ( id_depto );

ALTER TABLE materia ADD (
    id_departamento_materia NUMBER NOT NULL
)
    ADD CONSTRAINT fk_id_departamento_materia FOREIGN KEY ( id_departamento_materia )
        REFERENCES departamento ( id_depto );

CREATE TABLE requisito_materia (
    id_materia          NUMBER NOT NULL,
    id_materia_requisito NUMBER NOT NULL,
    CONSTRAINT fk_id_materia FOREIGN KEY ( id_materia )
        REFERENCES materia ( id_materia ),
    CONSTRAINT fk_id_materia_requisito FOREIGN KEY ( id_materia_requisito )

```



```

        REFERENCES materia ( id_materia ),
        PRIMARY KEY ( id_materia,
                      id_materia_requisito )
);

```

```

-- Datos de prueba -----

```

```

INSERT INTO provincia VALUES (
    1, -- id_provincia
    'Guanacaste' -- nombre
);

```

```

INSERT INTO provincia VALUES (
    2, -- id_provincia
    'San Jose' -- nombre
);

```

```

INSERT INTO provincia VALUES (
    3, -- id_provincia
    'Alajuela' -- nombre
);

```

```

INSERT INTO provincia VALUES (
    4, -- id_provincia
    'Cartago' -- nombre
);

```

```

--- 2017
INSERT INTO estudiante VALUES (
    201700000, -- carnet
    'Juan Perez', -- nombre
    1, -- provincia
    '2017' -- anno de ingreso
);

```

```

INSERT INTO estudiante VALUES (
    201700001, -- carnet
    'Maria Lopez', -- nombre
    2, -- provincia
    '2017' -- anno de ingreso
);

```

```

INSERT INTO estudiante VALUES (
    201700002, -- carnet
    'Pedro Sanchez', -- nombre
    3, -- provincia
    '2017' -- anno de ingreso
);

```

```

INSERT INTO estudiante VALUES (
    201700003, -- carnet
    'Jose Jimenez', -- nombre
    4, -- provincia
    '2017' -- anno de ingreso
);

```

```

INSERT INTO estudiante VALUES (
    201700004, -- carnet
    'Manuel Mora', -- nombre
    4, -- provincia
    '2017' -- anno de ingreso
);

```

```

INSERT INTO estudiante VALUES (
    201700005, -- carnet
    'Ignacio Mendez', -- nombre
    2, -- provincia
    '2017' -- anno de ingreso
);

```

```

--- 2018
INSERT INTO estudiante VALUES (

```

```

    201800001, -- carnet
    'Andres Soto', -- nombre
    2, -- provincia
    '2018' -- anno de ingreso
);

INSERT INTO estudiante VALUES (
    201800002, -- carnet
    'Andrea Arias', -- nombre
    2, -- provincia
    '2018' -- anno de ingreso
);

INSERT INTO estudiante VALUES (
    201800003, -- carnet
    'Carlos Suarez', -- nombre
    2, -- provincia
    '2018' -- anno de ingreso
);

INSERT INTO estudiante VALUES (
    201800004, -- carnet
    'Nancy Oramas', -- nombre
    2, -- provincia
    '2018' -- anno de ingreso
);

INSERT INTO beca VALUES (
    1, -- id beca
    1000, -- monto
    'Beca 1' -- nombre beca
);

INSERT INTO beca VALUES (
    2, -- id beca
    2000, -- monto
    'Beca 2' -- nombre beca
);

INSERT INTO aula VALUES (
    1, -- id aula
    'Aula 1' -- nombre aula
);

INSERT INTO aula VALUES (
    2, -- id aula
    'Aula 2' -- nombre aula
);

INSERT INTO aula VALUES (
    3, -- id aula
    'Aula 3' -- nombre aula
);

INSERT INTO aula VALUES (
    4, -- id aula
    'Aula 4' -- nombre aula
);

INSERT INTO aula VALUES (
    5, -- id aula
    'Aula 5' -- nombre aula
);

INSERT INTO horario VALUES (
    1, -- id horario
    'Lunes', -- dia
    '9:00',
    '10:00'
);

```

```
INSERT INTO horario VALUES (
    2, -- id horario
    'Lunes', -- dia
    '10:00', -- inicio
    '11:00' -- fin
);

INSERT INTO horario VALUES (
    3, -- id horario
    'Lunes', -- dia
    '12:00', -- inicio
    '13:00' -- fin
);

INSERT INTO horario VALUES (
    4, -- id horario
    'Lunes', -- dia
    '14:00', -- inicio
    '15:00' -- fin
);

INSERT INTO horario VALUES (
    5, -- id horario
    'Lunes', -- dia
    '15:00', -- inicio
    '16:00' -- fin
);

INSERT INTO horario VALUES (
    6, -- id horario
    'Martes', -- dia
    '9:00', -- inicio
    '10:00' -- fin
);

INSERT INTO horario VALUES (
    7, -- id horario
    'Martes', -- dia
    '10:00', -- inicio
    '11:00' -- fin
);

INSERT INTO horario VALUES (
    8, -- id horario
    'Martes', -- dia
    '12:00', -- inicio
    '13:00' -- fin
);

INSERT INTO horario VALUES (
    9, -- id horario
    'Martes', -- dia
    '14:00', -- inicio
    '15:00' -- fin
);

INSERT INTO horario VALUES (
    10, -- id horario
    'Martes', -- dia
    '15:00', -- inicio
    '16:00' -- fin
);

INSERT INTO horario VALUES (
    11, -- id horario
    'Miercoles', -- dia
    '9:00', -- inicio
    '10:00' -- fin
);
```

```
INSERT INTO horario VALUES (  
    12, -- id horario  
    'Miercoles', -- dia  
    '10:00', -- inicio  
    '11:00' -- fin  
);  
  
INSERT INTO horario VALUES (  
    13, -- id horario  
    'Miercoles', -- dia  
    '12:00', -- inicio  
    '13:00' -- fin  
);  
  
INSERT INTO horario VALUES (  
    14, -- id horario  
    'Miercoles', -- dia  
    '14:00', -- inicio  
    '15:00' -- fin  
);  
  
INSERT INTO horario VALUES (  
    15, -- id horario  
    'Miercoles', -- dia  
    '15:00', -- inicio  
    '16:00' -- fin  
);  
  
INSERT INTO horario VALUES (  
    16, -- id horario  
    'Jueves', -- dia  
    '9:00', -- inicio  
    '10:00' -- fin  
);  
  
INSERT INTO horario VALUES (  
    17, -- id horario  
    'Jueves', -- dia  
    '10:00', -- inicio  
    '11:00' -- fin  
);  
  
INSERT INTO horario VALUES (  
    18, -- id horario  
    'Jueves', -- dia  
    '12:00', -- inicio  
    '13:00' -- fin  
);  
  
INSERT INTO horario VALUES (  
    19, -- id horario  
    'Jueves', -- dia  
    '14:00', -- inicio  
    '15:00' -- fin  
);  
  
INSERT INTO horario VALUES (  
    20, -- id horario  
    'Jueves', -- dia  
    '15:00', -- inicio  
    '16:00' -- fin  
);  
  
INSERT INTO departamento VALUES (  
    1, -- id  
    'Departamento 1' -- Nombre  
);  
  
INSERT INTO departamento VALUES (  
    2, -- id  
    'Departamento 2' -- Nombre  
);
```

```

    2, -- id
    'Departamento 2' -- Nombre
);

INSERT INTO departamento VALUES (
    3, -- id
    'Departamento 3' -- Nombre
);

INSERT INTO departamento VALUES (
    4, -- id
    'Departamento 4' -- Nombre
);

-- ID, creditos, nombre
INSERT INTO materia VALUES (
    1, -- id materia
    2, -- Creditos
    'MATEMATICA GENERAL', -- Nombre
    1 -- Departamento
);

INSERT INTO materia VALUES (
    2, -- id materia
    3, -- Creditos
    'FUNDAMENTOS DE ORGANIZACION DE COMPUTADORAS', -- Nombre
    2 -- Departamento
);

INSERT INTO materia VALUES (
    3, -- id materia
    4, -- Creditos
    'ESTRUCTURAS DE DATOS', -- Nombre
    2 -- Departamento
);

INSERT INTO materia VALUES (
    4, -- id materia
    3, -- Creditos
    'INTRODUCCION A LA PROGRAMACION', -- Nombre
    4 -- Departamento
);

INSERT INTO materia VALUES (
    5, -- id materia
    3, -- Creditos
    'TALLER DE PROGRAMACION', -- Nombre
    3 -- Departamento
);

INSERT INTO materia VALUES (
    6, -- id materia
    2, -- Creditos
    ' INGLÉS BASICO', -- Nombre
    2 -- Departamento
);

INSERT INTO materia VALUES (
    7, -- id materia
    2, -- Creditos
    ' INGLÉS 1', -- Nombre
    2 -- Departamento
);

INSERT INTO materia VALUES (
    8, -- id materia
    2, -- Creditos
    ' INGLÉS 2', -- Nombre
    2 -- Departamento
);

```

```

INSERT INTO materia VALUES (
    9, -- id materia
    4, -- Creditos
    'ANALISIS DE ALGORITMOS', -- Nombre
    2 -- Departamento
);

INSERT INTO materia VALUES (
    10, -- id materia
    4, -- Creditos
    'CALCULO DIFERENCIAL E INTEGRAL', -- Nombre
    2 -- Departamento
);

INSERT INTO profesor VALUES (
    1, -- id
    'Profesor 1', -- Nombre
    1 -- Departamento
);

INSERT INTO profesor VALUES (
    2,
    'Profesor 2',
    2
);

INSERT INTO profesor VALUES (
    3,
    'Profesor 3',
    3
);

INSERT INTO grupo VALUES (
    1, -- Numero de grupo
    1, -- Id profesor asignado
    1, -- Id horario asignado
    1, -- Id aula asignada
    1 -- Id materia del grupo
);

INSERT INTO grupo VALUES (
    2,
    2,
    2,
    2,
    2
);

INSERT INTO grupo VALUES (
    3,
    3,
    3,
    3,
    3
);

INSERT INTO grupo VALUES (
    4,
    3,
    4,
    4,
    4
);

INSERT INTO grupo VALUES (
    5,
    3,
    4,
    4,
    4
);

```

```

5
);

INSERT INTO grupo VALUES (
6,
3,
4,
4,
6
);

INSERT INTO grupo VALUES (
7,
3,
4,
4,
7
);

INSERT INTO grupo VALUES (
8,
3,
4,
4,
8
);

INSERT INTO grupo VALUES (
9,
3,
4,
4,
9
);

INSERT INTO grupo VALUES (
10,
3,
4,
4,
10
);

-- Datos de prueba para periodo (periodo,carnet_estudiante,id_grupo_matriculado, anno,
nota_final, estado)
-- Estudiantes carnet 2017
INSERT INTO periodo VALUES (
2, -- Numero de periodo
201700001, -- Carnet estudiante
1, -- Id grupo matriculado
'2018', -- Anno del curso
65, -- Nota final
'Reproado' -- Condicion
);

INSERT INTO periodo VALUES (
1,
201700002,
2,
'2019',
90,
'Aprobado'
);

INSERT INTO periodo VALUES (
2,
201700002,
1,
'2019',
60,

```

```

        'Reproado'
    );

INSERT INTO periodo VALUES (
    1,
    201700003,
    2,
    '2020',
    5,
    'Reproado'
);

INSERT INTO periodo VALUES (
    1,
    201700004,
    1,
    '2020',
    80,
    'Aprobado'
);

INSERT INTO periodo VALUES (
    1,
    201700005,
    3,
    '2020',
    75,
    'Aprobado'
);

INSERT INTO periodo VALUES (
    2,
    201700003,
    2,
    '2020',
    95,
    'Aprobado'
);

INSERT INTO periodo VALUES (
    2,
    201700004,
    2,
    '2020',
    80,
    'Aprobado'
);

INSERT INTO periodo VALUES (
    2,
    201700005,
    2,
    '2020',
    60,
    'Reproado'
);

INSERT INTO periodo VALUES (
    1,
    201700005,
    2,
    '2021',
    70,
    'Aprobado'
);

INSERT INTO periodo VALUES (
    1,
    201700005,
    6,

```



```

        '2021',
        90,
        'Aprobado'
    );

INSERT INTO periodo VALUES (
    1,
    201700005,
    7,
    '2021',
    70,
    'Aprobado'
);

INSERT INTO periodo VALUES (
    1,
    201700005,
    8,
    '2021',
    70,
    'Aprobado'
);

INSERT INTO periodo VALUES (
    1,
    201700005,
    10,
    '2021',
    70,
    'Aprobado'
);

-- Estudiantes carnet 2018
-- Andrea Arias
INSERT INTO periodo VALUES (
    1,
    201800002,
    2,
    '2022',
    75,
    'Aprobado'
);

INSERT INTO periodo VALUES (
    2,
    201800002,
    6,
    '2021',
    90,
    'Aprobado'
);

INSERT INTO periodo VALUES (
    2,
    201800002,
    7,
    '2022',
    80,
    'Aprobado'
);

INSERT INTO periodo VALUES (
    1,
    201800002,
    8,
    '2021',
    85,
    'Aprobado'
);

```

```

INSERT INTO periodo VALUES (
    2,
    201800002,
    10,
    '2022',
    90,
    'Aprobado'
);

-- Carlos Suarez
INSERT INTO periodo VALUES (
    1,
    201800003,
    2,
    '2021',
    70,
    'Aprobado'
);

INSERT INTO periodo VALUES (
    2,
    201800003,
    6,
    '2022',
    80,
    'Aprobado'
);

INSERT INTO periodo VALUES (
    1,
    201800003,
    7,
    '2021',
    95,
    'Aprobado'
);

INSERT INTO periodo VALUES (
    1,
    201800003,
    8,
    '2022',
    70,
    'Aprobado'
);

INSERT INTO periodo VALUES (
    2,
    201800003,
    10,
    '2021',
    75,
    'Aprobado'
);

INSERT INTO estudiante_beca VALUES (
    201700000, -- Carnet de estudiante
    1 -- Id de la beca asignada
);

INSERT INTO estudiante_beca VALUES (
    201700001,
    2
);

INSERT INTO estudiante_beca VALUES (
    201700002,
    1
);

```

```

INSERT INTO requisito_materia VALUES (
    7, -- Materia principal
    6 -- Materia requisito que se quiere asignar
);

INSERT INTO requisito_materia VALUES (
    8,
    7
);

INSERT INTO requisito_materia VALUES (
    9,
    4
);

INSERT INTO requisito_materia VALUES (
    9,
    10
);

INSERT INTO requisito_materia VALUES (
    10,
    1
);

-- PARTE 2 -----

-- Funcion para obtener el requisito de una materia
CREATE OR REPLACE FUNCTION obtener_requisito (
    n_id_materia NUMBER
) RETURN NUMBER IS
    id_requisito NUMBER := 0;
BEGIN
    SELECT
        id_materia_requisito
    INTO id_requisito
    FROM
        requisito_materia
    WHERE
        id_materia = n_id_materia;
    RETURN id_requisito;
EXCEPTION
    WHEN no_data_found THEN
        id_requisito := 0;
    RETURN id_requisito;
END;
/

-- Funcion para retornar id_materia de un grupo especifico
CREATE OR REPLACE FUNCTION obtener_materia (
    n_grupo NUMBER
) RETURN NUMBER IS
    id_ma NUMBER;
BEGIN
    SELECT
        id_materia_grupo
    INTO id_ma
    FROM
        grupo
    WHERE
        num_grupo = n_grupo;

    RETURN id_ma;
END;
/

-- Funcion para verificar aprobacion de materia
CREATE OR REPLACE FUNCTION verificar_aprobacion (
    n_carnet      NUMBER,

```

```

        n_id_materia NUMBER
    ) RETURN NUMBER IS

        resultado NUMBER;
        nota        NUMBER;
        prueba       NUMBER;
        id_grupo     NUMBER;
        CURSOR cur_historial IS
        SELECT
            id_grupo_matriculado,
            nota_final
        FROM
            periodo
        WHERE
            carnet_estudiante = n_carnet;

BEGIN
    prueba := 0;
    id_grupo := 0;
    FOR reg IN cur_historial LOOP
        prueba := obtener_materia(reg.id_grupo_matriculado);
        IF prueba = n_id_materia THEN
            id_grupo := reg.id_grupo_matriculado;
        END IF;
    END LOOP;

    SELECT
        MAX(nota_final)
    INTO nota
    FROM
        periodo
    WHERE
        id_grupo_matriculado = id_grupo
        AND carnet_estudiante = n_carnet;

    IF nota >= 70 THEN
        resultado := 0;
    ELSE
        resultado := 1;
    END IF;

    RETURN resultado;
END;
/

-- Procedimiento para matricular en un grupo
CREATE OR REPLACE PROCEDURE matricular_estudiante (
    n_carnet NUMBER,
    n_grupo  NUMBER
) IS
    id_materia NUMBER;
    requisito  NUMBER;
    aprobacion NUMBER;
BEGIN
    id_materia := obtener_materia(n_grupo);
    requisito  := obtener_requisito(id_materia);
    aprobacion := verificar_aprobacion(n_carnet, requisito);
    IF aprobacion = 0 OR requisito = 0 THEN
        INSERT INTO matricula VALUES (
            NULL,
            n_carnet,
            n_grupo
        );
    ELSE
        dbms_output.put_line('No puede matricular el grupo '
            || n_grupo
            || ' porque no cumple con el requisito');
    END IF;

```

```

END;
/

--PARTE 3 -----

--- Funcion para calcular los creditos por grupo
CREATE OR REPLACE FUNCTION obtener_creditos (
    n_grupo NUMBER
) RETURN NUMBER IS
    total_creditos NUMBER;
    id_ma          NUMBER;
BEGIN
    SELECT
        id_materia_grupo
    INTO id_ma
    FROM
        grupo
    WHERE
        num_grupo = n_grupo;

    SELECT
        creditos
    INTO total_creditos
    FROM
        materia
    WHERE
        id_materia = id_ma;

    RETURN total_creditos;
END;
/

/*
Calcula el ponderado de todos los estudiantes en un anno determinado.
Solo el ultimo periodo matriculado con dos decimales de precision.
*/
CREATE OR REPLACE FUNCTION promedio_ponderado (
    n_carnet NUMBER
) RETURN NUMBER IS

    total_creditos NUMBER;
    sumatoria      NUMBER;
    ponderado      NUMBER(4, 2);
    creditos       NUMBER;
    ultimo_anno    NUMBER(4);
    ultimo_periodo NUMBER(4);
    id_grupo       NUMBER;
    nota           NUMBER;
    CURSOR cur_periodos_estd IS
    SELECT
        periodo,
        id_grupo_matriculado,
        nota_final,
        anno
    FROM
        periodo
    WHERE
        carnet_estudiante = n_carnet
    ORDER BY
        anno DESC,
        periodo DESC;

BEGIN
    total_creditos := 0;
    sumatoria := 0;

-- Se cargan datos de muestreo para obtener el ultimo periodo matriculado por el estudiante
    OPEN cur_periodos_estd;
    FETCH cur_periodos_estd INTO

```

```

        ultimo_periodo,
        id_grupo,
        nota,
        ultimo_anno;
CLOSE cur_periodos_estd;
FOR reg IN cur_periodos_estd LOOP
    IF
        reg.anno = ultimo_anno
        AND reg.periodo = ultimo_periodo
    THEN -- se comprueba que se este avaluando el mismo periodo
        creditos := obtener_creditos(reg.id_grupo_matriculado);
        total_creditos := total_creditos + creditos;
        sumatoria := ( reg.nota_final * creditos ) + sumatoria;
    ELSE
        EXIT;
    END IF;

    ultimo_anno := reg.anno;
    ultimo_periodo := reg.periodo;
END LOOP;

IF total_creditos = 0 THEN
    total_creditos := 1;
END IF;
ponderado := sumatoria / total_creditos;
RETURN ponderado;
END;
/

-- Tabla temporal global
CREATE GLOBAL TEMPORARY TABLE temp_estudiantes_anno (
    carnet    NUMBER NOT NULL,
    nombre    VARCHAR2(20) NOT NULL,
    ponderado FLOAT NOT NULL
);

-- Procedimiento para insertar en tabla temporal
/*
Calcula el ponderado de todos los estudiantes en un anno determinado.
Solo el ultimo periodo matriculado con dos decimales de precision.
*/
CREATE OR REPLACE PROCEDURE agregar_estudiante_anno (
    v_anno IN VARCHAR2
) IS
    CURSOR cur_agregar_estudiante IS
    SELECT
        carnet,
        nombre
    FROM
        estudiante
    WHERE
        anno_ingreso = v_anno;
BEGIN
    FOR reg IN cur_agregar_estudiante LOOP
        INSERT INTO temp_estudiantes_anno VALUES (
            reg.carnet,
            reg.nombre,
            promedio_ponderado(reg.carnet)
        );
    END LOOP;
END;
/

-- eliminar en cascada las tablas y sus datos en onder inverso de como se crearon las tablas
drop table requisito_materia cascade constraints;
drop table estudiante_beca cascade constraints;
drop table periodo cascade constraints;
drop table matricula cascade constraints;

```

```
drop table grupo cascade constraints;  
drop table departamento cascade constraints;  
drop table profesor cascade constraints;  
drop table materia cascade constraints;  
drop table horario cascade constraints;  
drop table aula cascade constraints;  
drop table beca cascade constraints;  
drop table estudiante cascade constraints;  
drop table provincia cascade constraints;  
drop table temp_estudiantes_anno cascade constraints;
```