Proyecto de Sistemas de Bases de Datos I

*Sistema planificador de tareas*

**Sistemas de Bases de Datos I**

##### Primer Término 2021-2022

**ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL**

**Facultad de ingeniería en electricidad y computación**

**Índice**

[Integrantes 3](#_Toc73716941)

[Titulo del Proyecto 3](#_Toc73716942)

[Objetivo del Proyecto 3](#_Toc73716943)

[Objetivos Específicos 3](#_Toc73716945)

[Descripción General 3](#_Toc73716947)

[Descripciones Funcionales 3](#_Toc73716947)

[Modelo Conceptual 4](#_Toc73716949)

[Diccionario de Datos 4](#_Toc73716951)

[Modelo Lógico 4](#_Toc73716952)

[Flujo de Navegación 4](#_Toc73716954)

# Integrantes

* Choez Villacis Steven
* Maza Punine Isaac
* Sánchez Gómez Robert

# Titulo del Proyecto

Sistema planificador de tareas

**Etapa de Desarrollo: Análisis del Sistema**

# Objetivo del Proyecto

Crear un sistema junto a una aplicación que permita la gestión de tareas a los estudiantes, profesores y ayudantes en un ámbito universitario, organizando de la manera más efectiva posible sus actividades en el día a día para que los usuarios puedan revisar de forma clara y precisa tareas pendientes e incluso generar recordatorios.

### Etapa de Desarrollo: Análisis del Sistema

# Objetivos Específicos

- Ubicar en un calendario las tareas agregadas por el usuario.

- Organizar las tareas de acuerdo con cursos y tiempos de entrega.

- Establecer prioridades entre las tareas pendientes.

- Generar un recordatorio para que no se pasen por alto las tareas.

### Etapa de Desarrollo: Análisis del Sistema

# Descripción General

En el ámbito universitario, tanto estudiantes como profesores y ayudantes tienen una basta cantidad de tareas y actividades que, de no organizarse de la manera correcta, pueden pasarse por alto o presentarse con atrasos que puede llegar a costar una calificación o nota importante.

Esa es la razón por la que es necesario un sistema que organice todas las tareas y actividades del día a día de manera que no se olviden.

Describiendo el sistema que tenemos 4 actores que serán el administrador, el profesor, el ayudante y el estudiante, todos cumplen un rol fundamental para el correcto funcionamiento del organizador de tareas.

El administrador es el que se encarga, con ayuda e información del profesor, a crear los respectivos cursos y a la vez asignar un profesor, un ayudante y a los alumnos en caso de ser necesario, este también tendrá la potestad de eliminar los cursos a su término o a los estudiantes en el transcurso de este; todo esto mientras está disponible para la resolución de cualquier inconveniente en el sistema.

Para el profesor entre sus principales funciones están el asignar tareas a un curso, estas tareas tendrán fecha de envío, fecha de entrega, calificación y estado de la tarea, estado que puede consultar por cada estudiante.

El ayudante tiene el mismo acceso del profesor al estado de las tareas de un curso, puede asignar tareas al igual que el profesor y además asignar al calendario de un curso una “ayudantía” que tendrá una fecha, hora de inicio y hora de fin y tema.

El estudiante por su parte cuenta con la función de revisar estas tareas asignadas por el profesor y el ayudante además de las tareas propias que el haya agregado y revisar las tareas pendientes y para tener una visión más amplia puede revisar su calendario que es una tabla de actividades pendientes, tareas y citas que tendrá en un mes, también cambiar el estado de las tareas que serán: pendiente, entregado, con atraso y no entregado.

Especificaciones del Sistema:

AWS

Roles de usuario:

Estudiantes: Revisa y realiza las tareas que le asignan los profesores, tendrán el horario de sus actividades en un calendario.

Profesores: Asigna y observa estado de tareas de cada estudiante, verifica que los ayudantes cumplan sus asignaciones.

Ayudantes: Ayudan a profesores con la revisión, asignación de tareas, también generan horarios para ayudantías.

Administrador: Controla el sistema y verifica que no se den fallos, además recibe retroalimentación de estudiantes, profesores y ayudantes.

Sistemas externos: Base de datos de la universidad.

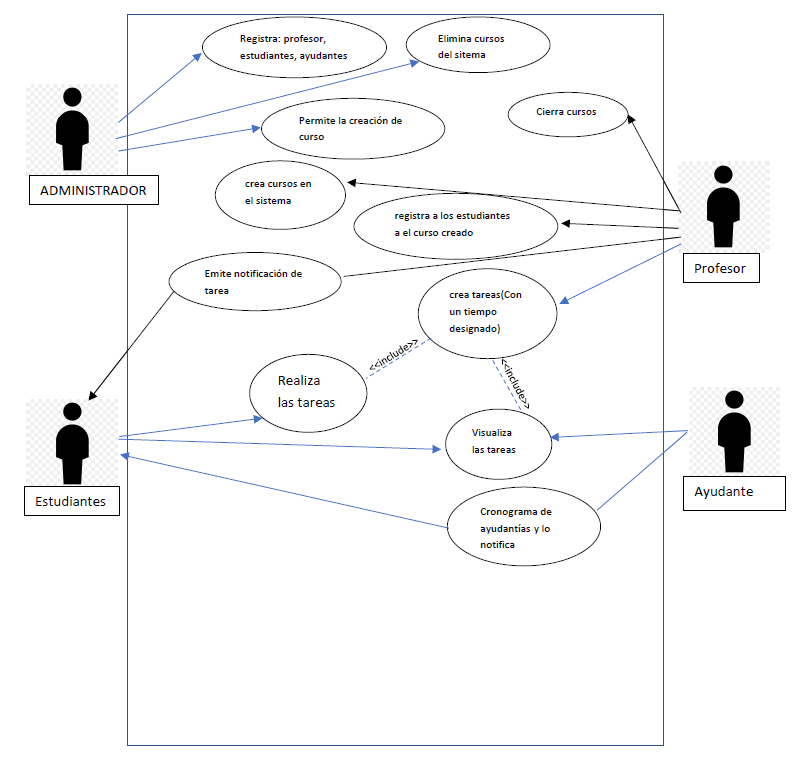
Tipos de seguridad (segundo parcial)

Transacciones críticas y frecuentes (segundo parcial)

### Etapa de Desarrollo: Análisis del Sistema

# 

# Descripciones Funcionales



**Procesos:**

**Administrador**:

**Nombre:** Agregar profesor

**Descripción:** Este proceso permite al administrador agregar un nuevo profesor al sistema.

**Nota:** Se validará que el profesor no esté registrado.

**Entrada:** IdProfesor, nombres, apellidos, dirección, teléfono, email.

**Salida:** Mensaje de éxito: “Se agregó profesor con éxito”

Mensaje de error: “Profesor ya está registrado”

“Falta información para el registro”

**Nombre:** Eliminar profesor

**Descripción:** Este proceso permite al administrador eliminar un profesor del sistema.

**Nota:** Se validará que el profesor esté registrado.

**Entrada:** IdProfesor.

**Salida:** Mensaje de éxito: “Se eliminó profesor con éxito”

Mensaje de error: “Profesor no estaba registrado”

Salida: []

**Nombre:** Agregar estudiante

**Descripción:** Este proceso permite a un administrador agregar un nuevo estudiante al sistema.

**Nota:** Se validará que el estudiante no esté registrado.

**Entrada:** IdEstudiante, IdCurso, nombres, apellidos, dirección, teléfono, email, carrera, matrícula.

**Salida:** Mensaje de éxito: “Se registró estudiante con éxito”

Mensaje de error: “Estudiante ya está registrado”

Mensaje de error: “Faltan información para el registro”

**Nombre:** Eliminar estudiante

**Descripción:** Este proceso permite a un administrador eliminar un estudiante del sistema.

**Nota:** Se validará que el estudiante esté registrado.

**Entrada:** IdEstudiante.

**Salida:** Mensaje de éxito: “Se eliminó estudiante con éxito”

Mensaje de error: “Estudiante no estaba registrado”

Salida: []

**Nombre:** Agregar ayudante

**Descripción:** Este proceso permite a un administrador agregar un nuevo ayudante al sistema.

**Nota:** Se validará que el ayudante no esté registrado en otro curso.

**Entrada:** IdAyudante, IdCurso, nombres, apellidos, dirección, teléfono, email, carrera, matrícula.

**Salida:** Mensaje de éxito: “Se registró ayudante con éxito”

Mensaje de error: “Ayudante ya está registrado en un curso”

Mensaje de error: “Faltan información para el registro”

**Nombre:** Eliminar ayudante

**Descripción:** Este proceso permite a un administrador eliminar un ayudante del sistema.

**Nota:** Se validará que el estudiante esté registrado.

**Entrada:** IdAyudante.

**Salida:** Mensaje de éxito: “Se eliminó ayudante con éxito”

Mensaje de error: “Ayudante no estaba registrado”

Salida: []

**Nombre:** Crear curso

**Descripción:** Este proceso permite al administrador agregar un nuevo curso en su sistema.

**Nota:** Se validará que el nombre del curso no exista.

**Entrada:** IdCurso, IdProfesor, IdAyudante, IdEstudiante, materia, horario de clases.

**Salida:** Mensaje de éxito: “Se creó curso con éxito”

Mensaje de error: “Curso ya existe”

Mensaje de error: “Falta información para el registro”

**Nombre:** Eliminar curso

**Descripción:** Este proceso permite al administrador eliminar un curso del sistema.

**Nota:** Se validará que el nombre del curso exista.

**Entrada:** IdCurso.

**Salida:** Mensaje de éxito: “Se eliminó curso con éxito”

Mensaje de error: “Curso no existe”

Salida: []

**Profesor:**

**Nombre:** Asignar tareas a un curso

**Descripción:** Este proceso permite a un profesor asignar una nueva tarea a un determinado curso en su sistema.

**Nota:** Esta se agregará al calendario de los estudiantes y se validará que la fecha de entrega sea en el futuro.

**Entrada:**  IdCurso, nombreTarea, fechaEnvio, fechaEntrega y estado de la tarea,

**Salida:** Mensaje de éxito: “Se agregó tarea con éxito”

Mensaje de error: “Fecha de tarea debe ser en el futuro”

**Nombre:** Revisar estado de tareas

**Descripción:** Este proceso permite a un profesor revisar si los estudiantes han completado la tarea.

**Nota:** La tarea tendrá tres estados: pendiente, entregada, con atraso y no entregada.

**Entrada:** IdCurso, IdEstudiante.

**Salida:** Mensaje de éxito:

|  |  |  |
| --- | --- | --- |
| IdEstudiante | nombreTarea | estado |

Mensaje de error: No se encontró estudiante

**Estudiantes:**

**Nombre:** Agregar tarea

**Descripción:** Este proceso permite a un estudiante agregar una tarea a su cronograma

**Nota:** La tarea tendrá una fecha límite de entrega.

**Entrada:** IdCurso, nombreTarea, fechaLimite.

**Salida:** Mensaje de éxito: “Tarea agregada con éxito”

|  |  |  |
| --- | --- | --- |
| IdCurso | nombreTarea | estado |

**Nombre:** Revisar tareas pendientes

**Descripción:** Este proceso permite a un estudiante revisar las tareas pendientes en orden cronológico.

**Nota:** Se notificará si se acumulan muchas tareas.

**Entrada:** Fecha

**Salida:** Mensaje de éxito: “Tareas pendientes: ”

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| IdCurso | nombreTarea | fechaEnvio | fechaEntrega | estado |

Mensaje de error: “Las tareas no se pueden mostrar”

**Nombre:** Cambiar estado de tarea

**Descripción:** Este proceso permite a un estudiante pasar el estado de su tarea a completado.

**Nota:** Se preguntará a estudiante si terminó su tarea.

**Entrada:** Click en botón: presentar tarea.

**Salida:** Mensaje de éxito: “Tarea terminada con éxito”

Mensaje de error: “No se pudo registrar la tarea como completa o se registró como atrasada”

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| IdCurso | nombreTarea | fechaEnvio | fechaEntrega | Estado:  “Entregada”  “Pendiente”  “Atrasada”  “No entregada” |

**Nombre:** Revisar calendario

**Descripción:** Este proceso permite a un estudiante revisar el calendario con todas sus asignaciones pendientes.

**Nota:** Se presentará calendario del mes ingresado por el estudiante

**Entrada:** Mes a consultar.

**Salida:** Mensaje de éxito: “Calendario: ”

Mensaje de error: “Ingresar mes válido”

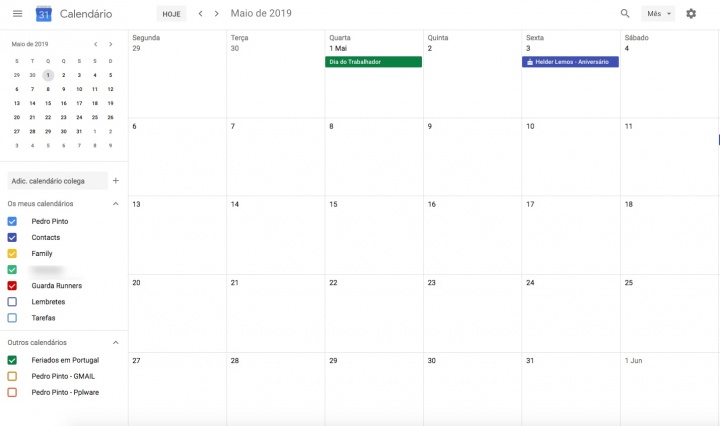


Imagen relacionada

**Ayudantes:**

**Nombre:** Asignar tareas a un curso

**Descripción:** Este proceso permite a un ayudante asignar una nueva tarea a un determinado curso en su sistema.

**Nota:** Esta se agregará al calendario de los estudiantes y se validará que la fecha de entrega sea en el futuro.

**Entrada:**  IdCurso, nombreTarea, fechaEnvio, fechaEntrega y estado de la tarea,

**Salida:** Mensaje de éxito: “Se agregó tarea con éxito”

Mensaje de error: “Fecha de tarea debe ser en el futuro”

|  |  |  |
| --- | --- | --- |
| IdCurso | nombreTarea | estado |

**Nombre:** Revisar estado de tareas

**Descripción:** Este proceso permite a un ayudante revisar si los estudiantes han completado la tarea.

**Nota:** La tarea tendrá tres estados: pendiente, entregada, con atraso y no entregada.

**Entrada:** IdCurso, IdEstudiante.

**Salida:** Mensaje de éxito:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| IdCurso | nombreTarea | fechaEnvio | fechaEntrega | estado |

Mensaje de error: No se encontró estudiante

**Nombre:** Generar horario de ayudantía.

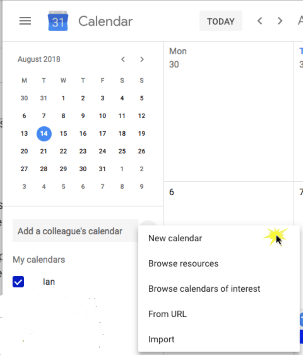
**Descripción:** Este proceso permite a un ayudante asignar el un horario de ayudantía en el calendario de todos los estudiantes en sus cursos.

**Nota:** Se agregará al calendario de sus alumnos y se verificará que no coincida con el horario de los otros ayudantes.

**Entrada:** IdCurso, fecha, hora, nombreAyudantia.

**Salida:** Mensaje de éxito: “Horario creado con éxito”

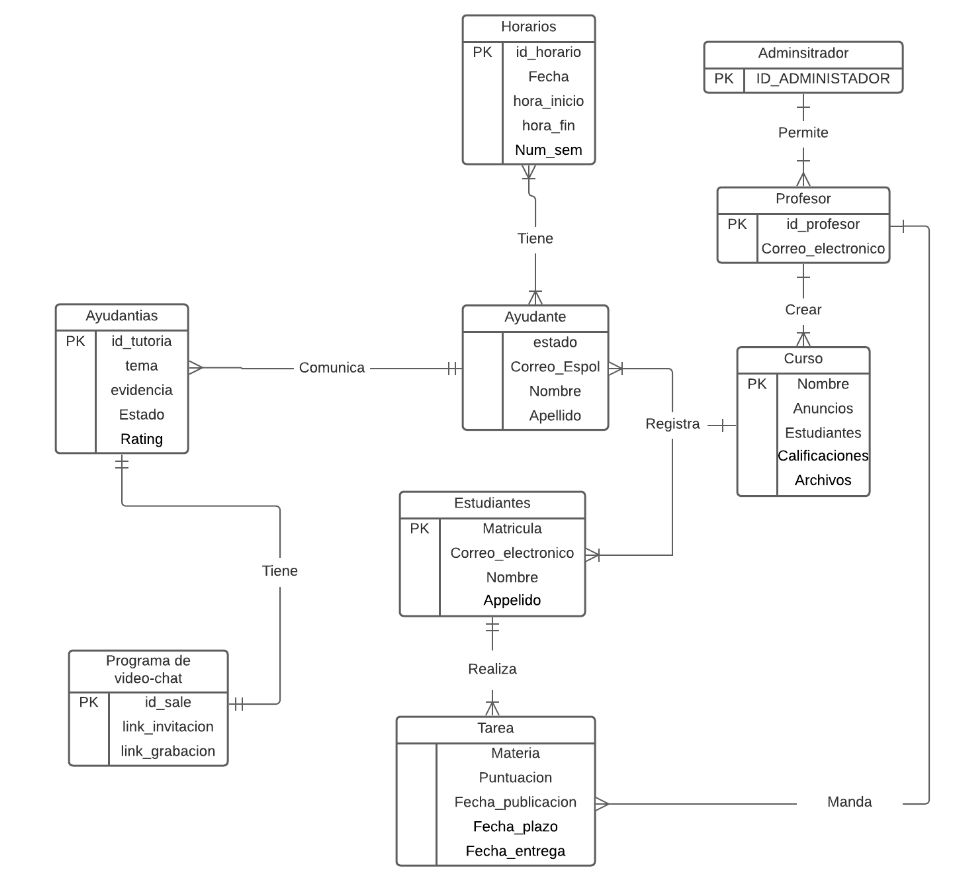
Mensaje de error: “Horario coincide con otro”



### Etapa de Desarrollo: Análisis del Sistema

# Modelo Conceptual

Presentación del Modelo Conceptual(diagrama)



### Etapa de Desarrollo: Diseño del Sistema

# Diccionario de Datos

Tabla: **Profesor**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nombre** | **Descripción** | **Tipo de datos** | **Dominio** | **Null** |
| ID\_Admin | Clave del administrador | numérico | Int(15) | NO |

Tabla: **Profesor**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nombre** | **Descripción** | **Tipo de datos** | **Dominio** | **Null** |
| Id\_curso | Id del curso | numérico | Int(4) | NO |
| Cedula | Cedula del profesor | numérico | Int(10) | NO |

Tabla: **Tarea**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nombre** | **Descripción** | **Tipo de datos** | **Dominio** | **Null** |
| **ID\_Tarea** | Id de la tarea | Numerico | Int(4) | NO |
| **Materia** | Nombre de la materia | Texto | String(20) | NO |
| **Puntuacion** | Verifica si la materia ha sido Entregada”  “Pendiente”  “Atrasada”  “No entregada | Texto | Varchar(20) | NO |
| **Fecha\_publicada** | Fecha que publicada | Fecha y hora | Datetime | NO |
| **Fecha\_entregada** | Fecha que ha sido entregada | Fecha y hora | Datetime | NO |
| **Fecha\_plazo** | Fecha máxima a entregar | Fecha y hora | Datetime | NO |
| **Id\_curso** | Clave del curso | numérico | Int(4) | NO |
| **Id\_profesor** | Clave del profesor | Numérico | Int(4) | NO |

Tabla: **Curso**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nombre** | **Descripción** | **Tipo de datos** | **Dominio** | **Null** |
| **ID\_Curso** | Id del curso | Numérico | Int(4) | NO |
| **Anuncios** | Los anuncios que se han publicado |  |  | Si |
| **N°\_Estudiante** | Numero de estudiante en el curso | Numérico | Int(4) | NO |
| **Calificaciones** | Calificaciones del curso | Numérico | Double(4) | No |
| **Archivos** | Archivos dados en el curso | Texto | String(1000000) | NO |

Tabla: **Estudiante**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nombre** | **Descripción** | **Tipo de datos** | **Dominio** | **Null** |
| **Id\_Estudiante** | Clave del estudiante | numérico | Int(4) | NO |
| **Matricula** | Matricula del estudiante | numérico | Int(10) | NO |
| **Ayudante** | Determina si es o no un estudiante | Lógico | Boolean | NO |
| **Estado** | Verifica si está o no en línea | Lógico | Boolean | NO |
| **ID\_Curso** | Clave del curso donde recibe la materia | numérico | Int(5) | NO |

Tabla: **Usuario**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nombre** | **Descripción** | **Tipo de datos** | **Dominio** | **Null** |
| Id\_usuario | Clave del usuario del 1111 a 9999 | Numerico | Int(4) | NO |
| Tipo | Describe si es ayudante o maestro o profesor | texto | Varchar(7) | NO |
| Nombre\_completo | String(150) | Texto | String(1500) | NO |
| Fecha\_Nacimiento | Fecha de nacimiento del usuario | Fecha | Date | NO |
| Teléfono | Numérico | Numerico | Int(10) | SI |
| Correo | Cadena | Texto | String(50) | NO |
| Contraseña | Cadena | Texto | String | NO |
| Id\_Admin | Clave del administrador del 1111 a 9999 | Numerico | Int(4) | NO |

Tabla: **Ayudantías**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nombre** | **Descripción** | **Tipo de datos** | **Dominio** | **Null** |
| ID\_Tutoria | Clave del horario del 1111 a 9999 | numérico | Int(4) | No |
| ID\_Ayudante | Clave que identifica al ayudante de la sesión | numérico | Int(4) | No |
| Tema | Tema para abordar en la ayudantía | Texto | String(150) | No |
| Evidencia | Lista de asistencia de espectadores | lista | ArrayList | Si |
| Estado | Verifica si dio o no la ayudantía | lógico | Boolean(True or False) | No |
| Rating | Valor numérico que hace referencia a la cantidad de espectadores | numérico | Int(4) | Si |
| ID\_Horario | Clave del horario del 1111 a 9999 | Numérico | Int(4) | NO |
| Id\_Estuidante | Clave del Estudiante del 1111 a 9999 | Numérico | Int(4) | No |
| Zoom | Incluye el id de la reunión así | Numerico | Int(10) | Si |
| Link\_grabacion | Link donde se hospeda la grabación de la ayudantía dada | Texto | String(1500) | Si |

Tabla:**Horarios**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nombre** | **Descripcion** | **Tipo de datos** | **Dominio** | **Null** |
| ID\_Horario | Clave del horario del 1111 a 9999 | numérico | Int(4) | No |
| Fecha | Van de un intervalo de 1900-01-01 a 2079-06-06 | Fecha | Date | No |
| H\_Inicio | Dato que refleja la hora la hora de inicio de una ayudantía | Hora | Time | No |
| H\_Fin | Dato que refleja la hora la hora final de una ayudantía | Hora | Time | Sí |
| N°\_sem | Refleja en que semana transcurre un horario que | Numérico | Int(2) | No |
| ID-Ayudante | Clave del ayudante del 1111 a 9999 | numérico | Int(4) | No |

#### Etapa de Desarrollo: Diseño del Sistema

# Modelo Lógico

Presentación del modelo Lógico(diagrama)

### Etapa de Desarrollo: Diseño del Sistema

# 

# AVANCE 3

# Realizar consultas

**1. Esta consulta permite a un profesor revisar si los estudiantes han completado la tarea.**

π IdEstudiante, nombreTarea, Tarea.estado (σ Tarea.estado = 'entregado'(Estudiantes ⨝ ESTUDIANTE.ID\_curso = Curso.ID\_curso Curso ⨝ Curso.ID\_curso = Tarea.ID\_Curso Tarea))

**2. Esta consulta permite a un estudiante revisar las tareas pendientes en orden cronológico.**

π Curso.IdCurso, Tarea.nombreTarea, Tarea.fechaEnvio, Tarea.fechaEntrega, estado (σ Tarea.estado = 'pendiente' (Tarea⨝ Tarea.ID\_curso = Curso.ID\_Curso Curso ⨝ Curso.ID\_curso = ESTUDIANTE.ID\_curso))

**3. Esta consulta permite a un profesor revisar la tarea de un estudiante mediante su matrícula**

π IdCurso, nombreTarea, fechaEnvio, fechaEntrega (σ Estudiantes.Matricula = 'aquivalamatricula' (Curso⨝Tarea⨝Estudiante))

**4. Esta consulta permite a un ayudante revisar si los estudiantes han completado la tarea**

π Curso.IdCurso, Tarea.nombreTarea, Tarea.fechaEnvio, Tarea.fechaEntrega, estado (σ Tarea.estado = 'entregado' (Tarea⨝ Tarea.ID\_curso = Curso.ID\_Curso Curso ⨝ Curso.ID\_curso = ESTUDIANTE.ID\_curso))

**5.Esta consulta permite a un estudiante revisar el calendario con todas sus asignaciones pendientes.**

π Curso.IdCurso, Tarea.nombreTarea, Tarea.fechaEntrega (σ Tarea.estado = 'pendiente' ^ Matricula = 'matriculaEstudiante' ^ fecha\_entrega = 'Mesdeentrega' (Tarea⨝ Tarea.ID\_curso = Curso.ID\_Curso Curso ⨝ Curso.ID\_curso = ESTUDIANTE.ID\_curso))

**6. Esta consulta cuenta cuantos estudiantes tiene un profesor en un determinado curso**

π Profesor.nombre, γ Count (σ Curso.Id\_curso = ‘iddelcurso' (Estudiantes⨝Curso⨝Profesor) as CANTIDAD

**Calificación (24/25)**

Descripción general **(2/3)**

Procesos **(5/5)**

Modelo conceptual **(4/4)**

Modelo lógico **(7/7)**

Diccionario de datos **(3/3)**

Redacción y ortografía **(3/3)**

Por favor, tener en cuenta estos cambios del Modelo lógico. Los necesitarán para crear correctamente su base de datos.

**Modelo conceptual (10/10)**

**Tablas (5/5)**

**Relaciones (3/3)**

**Cardinalidades (2/2)**

Materia debería ser un atributo de la tabla Profesor. Y debería ser multivaluado, ya que es posible que un profesor de más de una materia. Por ejemplo la miss da Bases de datos y Fundamentos.

**Modelo lógico (14.5/15)**

**Tablas (8/8)**

**Relaciones (3/3)**

**Cardinalidades (3.5/4)**

Sería más conveniente que entre Usuario y Administrador la FK se encuentre en administrador como FK id\_usuario. Porque si dejan ese id\_admin en Usuario tanto profesor como Estudiante heredarían ese atributo innecesariamente.

**Corregir cardinalidades**

Este diagrama hice para el grupo anterior que tiene un tema similar al suyo. Ahí pueden guiarse en cuanto a las cardinalidades de Profesor-Curso-Estudiante-Tareas. Y ver la nueva tabla Materia. Analícenlo y agreguen lo que les falta en el suyo.

