



Universidad de Buenos Aires
Facultad de Ingeniería

Métodos y modelos de Ingeniería de software
1er Cuatrimestre 2024

Ejercicio 7
Diseño de base de datos relacional

ALUMNO: JORDA SANTIAGO - 102924 - sjorda@fi.uba.ar

Ejercicio 7

Diseño de base de datos

A partir del modelo de dominio ya desarrollado, **diseñar la base de datos relacional**. Revisar detalladamente el documento:

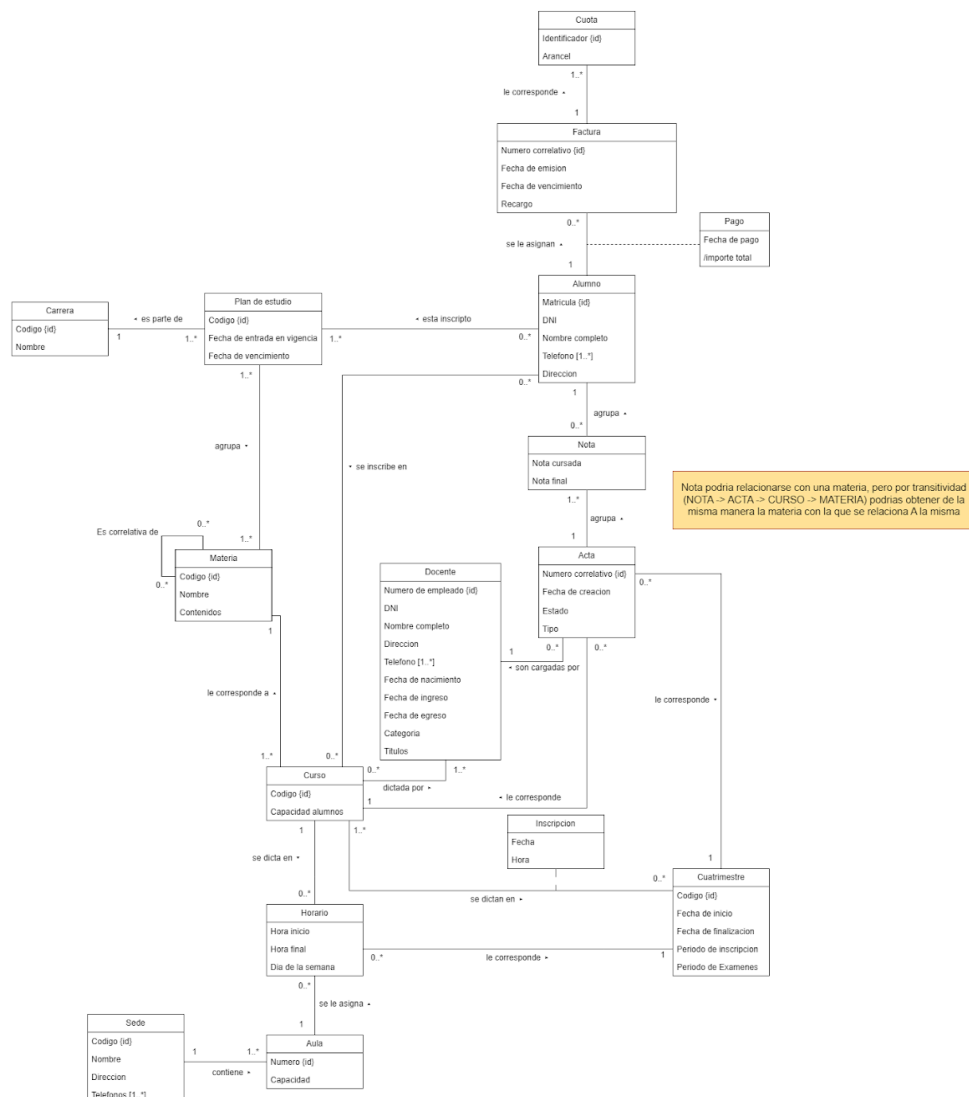
- Caso 01.02-Notas iniciales de Mariano

Emplear el subconjunto de UML propuesto en el capítulo 10 de la materia.

Desarrollo

Adjunto el link para poder ver los diagramas: [Link a los diagramas en Draw.io](#)

Para el desarrollo de este ejercicio se hizo una actualización del **modelo de dominio** del **Área de Administración de aranceles** quedando de la siguiente manera:



```

classDiagram
    class Carrera {
        int id PK
        varchar nombre
    }
    class Plan_de_Estudio {
        int codigo PK
        int id Carrera FK
        varchar fecha_inicio_vigencia
        varchar fecha_vencimiento
    }
    class Materia {
        int id Materia PK
        int id Plan_de_Estudio FK
        int id Materia PK
        varchar nombre
        varchar contenido
    }
    class Curso {
        int id Curso PK
        int id Materia FK
        int id Curso FK
        int capacidad_alumnos
    }
    class Alumno {
        int matricula PK
        int dni
        varchar nombre_completo
        varchar direccion
        varchar telefonos
    }
    class Docente {
        int legajo PK
        int dni
        varchar nombre_completo
        varchar direccion
        varchar telefonos
        varchar fecha_nacimiento
        varchar fecha_ingreso
        varchar fecha_egreso
        varchar categoria
        varchar titulos
    }
    class Acta {
        int numero_de_acta PK
        int id Curso FK
        int id Cuatrimestre FK
        varchar fecha_creacion
        varchar estado
        varchar tipo
    }
    class Cuatrimestre {
        int id Cuatrimestre PK
        int id Curso FK
        varchar fecha_inicio
        varchar fecha_finalizacion
        varchar periodo_examenes
    }
    class Inscricion {
        int id Inscricion PK
        int id Curso FK
        int id Cuatrimestre FK
        varchar hora
    }
    class Sede {
        int id Sede PK
        varchar nombre
        varchar direccion
        varchar telefonos
    }
    class Aula {
        int numero_aula PK
        int id Sede FK
        int id Capacidad
    }
    class Horario {
        int id Horario PK
        int id Curso FK
        int id Aula FK
        int id Cuatrimestre FK
        varchar dia_semana
        varchar hora_inicio
        varchar hora_final
    }
    class Factura {
        int numero_comprobante PK
        int id Alumno FK
        varchar fecha_emision
        varchar fecha_vencimiento
        float recargo
    }
    class Pago {
        int numero_comprobante PK
        int id Alumno FK
        int id Matricula FK
        varchar fecha_pago
        float importe
    }
    Carrera "1" -- "1..*" Plan_de_Estudio
    Plan_de_Estudio "1..*" -- "1" Carrera
    Plan_de_Estudio "1..*" -- "0..*" Materia
    Materia "1..*" -- "1" Plan_de_Estudio
    Materia "1..*" -- "1" Curso
    Curso "1..*" -- "1" Materia
    Curso "1..*" -- "0..*" Alumno
    Alumno "1..*" -- "1" Curso
    Alumno "1..*" -- "0..*" Inscricion
    Docente "1..*" -- "0..*" Inscricion
    Inscricion "1..*" -- "1" Alumno
    Inscricion "1..*" -- "1" Docente
    Inscricion "1..*" -- "1" Acta
    Acta "1..*" -- "1" Inscricion
    Acta "1..*" -- "0..*" Cuatrimestre
    Cuatrimestre "1..*" -- "1" Acta
    Cuatrimestre "1..*" -- "1" Inscricion
    Inscricion "1..*" -- "1" Cuatrimestre
    Inscricion "1..*" -- "1" Horario
    Horario "1..*" -- "1" Inscricion
    Horario "1..*" -- "1" Aula
    Aula "1..*" -- "1" Horario
    Aula "1..*" -- "1" Sede
    Sede "1..*" -- "1" Aula
    Factura "1..*" -- "1" Alumno
    Pago "1..*" -- "1" Alumno
    Pago "1..*" -- "1" Factura
    
```

Las fechas se que se pueden representar mediante un entero en milisegundos o como texto, queda a implementacion del sistema en parsear ese tipo de dato a su modo, por ende puedo elegir cualquier forma

Como en el sistema no voy a estar trabajando profundamente con los **teléfonos** de alumnos, sedes y docentes, a estos los voy a representar como un string, después el backend podría ser el encargado de parsearlo a un array, pero no se cree necesario crear una tabla aparte para cada uno.

En alguna futura iteración estas implementaciones podrían refinarse y trabajarlas más profundamente.

Se adjuntan las correcciones del ejercicio 5:

Historias de Usuario (Modificado)

Se agregaron más historias de usuario

1. Creación de cuotas

Como administrador de cobranzas, **quiero** poder crear nuevas cuotas, **para** que los alumnos paguen la cuota mensual

Escenario 1: Creación exitosa

Dado que se elige un mes futuro el cual no tiene una cuota asociada y se ingresa el monto valido de arancel

Cuando se quiere crear una nueva cuota

Entonces se crea la cuota con el monto ingresado correspondiente al mes seleccionado

Escenario 2: Creación no exitosa, ingreso de un valor de arancel no válido

Dado que se ingresa un valor no válido de arancel

Cuando se quiere crear una nueva cuota

Entonces el sistema informa que el valor de arancel ingresado no es válido

Escenario 3: Creación no exitosa, selección de un mes pasado

Dado que se elige un mes pasado

Cuando se quiere crear una nueva cuota

Entonces el sistema informa que ya existe su respectiva cuota

2. Actualización de cuotas

Como administrador de cobranzas, **quiero** poder actualizar los precios de las cuotas mensuales, **para** no tener aranceles desactualizados

Escenario 1: Actualización exitosa

Dado que se elige una cuota de un mes futuro

Cuando se quiere actualizar la cuota

Entonces se actualiza el precio de las cuota mensual seleccionada

Escenario 2: Actualización no exitosa, selección de un mes pasado

Dado que se elige una cuota de un mes pasado

Cuando se quiere actualizar la cuota

Entonces se le informa que no es posible actualizar el precio de la cuota.

Escenario 3: Actualización no exitosa, ingreso de un valor no válido

Dado que se ingresa un valor no válido para un monto de una cuota

Cuando se quiere actualizar la cuota

Entonces se le informa que no es posible actualizar el precio de la cuota.

3. Creación de factura

Como administrador de cobranzas, **quiero** poder crear facturas **para** que los alumnos paguen

Escenario 1: Creación exitosa

Dado se ha seleccionado una cuota que existe, la cual no fue generada su facturación

Cuando se quiere generar las facturas para todos los alumnos

Entonces estas se crean exitosamente las facturas correspondientes a la cuota con todos los detalles necesarios, como la fecha de emisión, fecha de vencimiento, datos del alumno, datos de la cuota, en caso de poseer recargo por pagos previos fuera de término su respectiva deuda y el importe total.

Escenario 2: Creación no exitosa, facturación duplicada

Dado se ha seleccionado una cuota que existe, la cual ya fue generada su facturación

Cuando se quiere generar las facturas

Entonces el sistema informa que la factura ya fue generada previamente.

4. Facturas alumnos

Como alumno, **quiero** recibir automáticamente una factura quince días antes del vencimiento de cada cuota, con todos los detalles necesarios **para** abonar la próxima cuota.

Escenario 1: Factura sin recargo

Dado un alumno regular con cuotas al día

Cuando faltan 15 días para el vencimiento de una cuota.

Entonces recibe una factura de dicha cuota con todos los detalles necesarios, como la fecha de emisión, fecha de vencimiento, datos del alumno, datos de la cuota y el importe total.

Escenario 2: Factura con recargo

Dado un alumno regular que posee pagos previos fuera de término

Cuando faltan 15 días para el vencimiento de una cuota.

Entonces recibe una factura de dicha cuota con todos los detalles necesarios, como la fecha de emisión, fecha de vencimiento, datos del alumno, datos de la cuota, recargo por pagos previos fuera de término y el importe total.

5. Administración de regularidad de los alumnos

Como administrador de cobranzas, **quiero** recibir notificaciones cuando un alumno esté cerca de perder su estado regular, **para** tomar medidas administrativas de reincorporación.

Escenario 1: Administrador recibe notificación de alumno a punto de perder regularidad

Dado un alumno que posee

Cuando esté cerca de de perder su regularidad

Entonces recibir notificación de que dicho alumno está por perder su estado regular

La historias de usuario de la sección **Administración de regularidad de alumnos** se encuentra incompleta:

Se debería definir **“Cuando este cerca de de perder su regularidad”**

Se debería preguntar al cliente en una próxima reunión que se refiere la palabra “cerca”

¿A qué periodo hace referencia? 1 mes? 2 meses? otro?

Y en caso de no tenerlo definido, deberíamos tratar de analizar el dominio del problema para acercar al cliente a una solución y así poder definir esa variable.

6. Tramite de alumno regular

Como alumno no regular **quiero** iniciar un trámite de reincorporación, **para** volver a ser alumno regular

Escenario 1: Reincorporacion con exito

Dado un alumno en condicion no regular que ha pagado todas las cuotas que adeuda

Cuando inicia el trámite de reincorporación

Entonces vuelve a ser alumno regular

Escenario 2: Reincorporacion sin exito, posee deuda

Dado un alumno en condicion no regular que posee cuotas sin pagar

Cuando inicia el trámite de reincorporación

Entonces se le informa que todavía debe cuotas

7. Gestión de transacciones

Como administrador de cobranzas, **quiero** recibir diariamente el detalle de las transacciones de pago realizadas por los alumnos **para** mantener registro actualizado de los pagos recibidos

Escenario 1: Se reciben las transacciones correctamente

Dado que se recibieron las transacciones de pago

Cuando se accede a ver los detalles de transacciones de pago del día

Entonces se visualizan los detalles de todas las transacciones, como la fecha de transacción, número de matrícula asociada al alumno, número correlativo a la facturas, cuotas asociadas a las mismas y el importe abonado.

Escenario 2: No recibieron aún las transacciones del día

Dado que no se recibieron las transacciones de pago

Cuando se accede a ver los detalles de transacciones de pago del día

Entonces se muestra un mensaje diciendo que todavía no aún no se han cargado las transacciones del día.

8. Carga de transacciones de pago

Como empleado de Tangerine, **quiero** cargar las transacciones realizadas por alumnos **para** mantener registro actualizado de los pagos recibidos

Escenario 1: Carga con éxito

Dado que un empleado de Tangerine se encuentra en el sistema con transacciones con el formato válido

Cuando quiere cargar las transacciones de pago

Entonces se guardan todos los detalles de las transacciones tales como fecha de pago, monto, el número de identificación de la factura, y el mes-año de la factura asociada y el sistema debe mostrar un mensaje que la carga fue exitosa.

Escenario 2: Carga sin éxito,

Dado que un empleado de Tangerine se encuentra en el sistema con transacciones con el formato no válido

Cuando quiere cargar las transacciones de pago

Entonces el sistema mostrar un mensaje diciendo que la carga no fue exitosa, que

Escenario 3: Carga sin éxito, transacciones no tienen

Dado que un empleado de Tangerine y ya hubieron sido cargadas las transacciones del día

Cuando quiere cargar las transacciones de pago

Entonces el sistema mostrar un mensaje diciendo que las transacciones ya fueron cargadas ese día