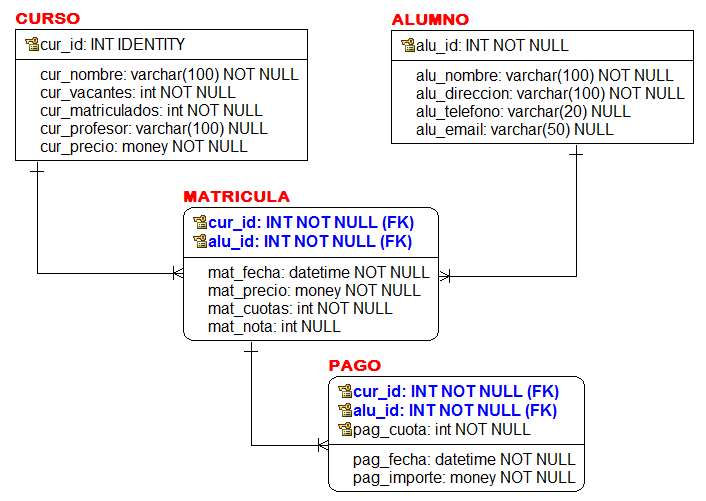
PROYECTO EDUCA

# PEQUEÑO ANALISIS

## Base de Datos



## Procesos

1. Registrar matricula (**El pago es en otro proceso**)
2. Registrar pago/cuota
3. Registrar nota
4. Cambio de curso (Definir condiciones)
5. Cambio de numero de cuotas (Definir condiciones)
6. Anular matricula (Definir condiciones)

# CASOS DE USO

## CU01: Registrar matricula

### Diagrama del caso de uso

### Especificación del caso de uso

### Prototipo

### Diagrama de Secuencia

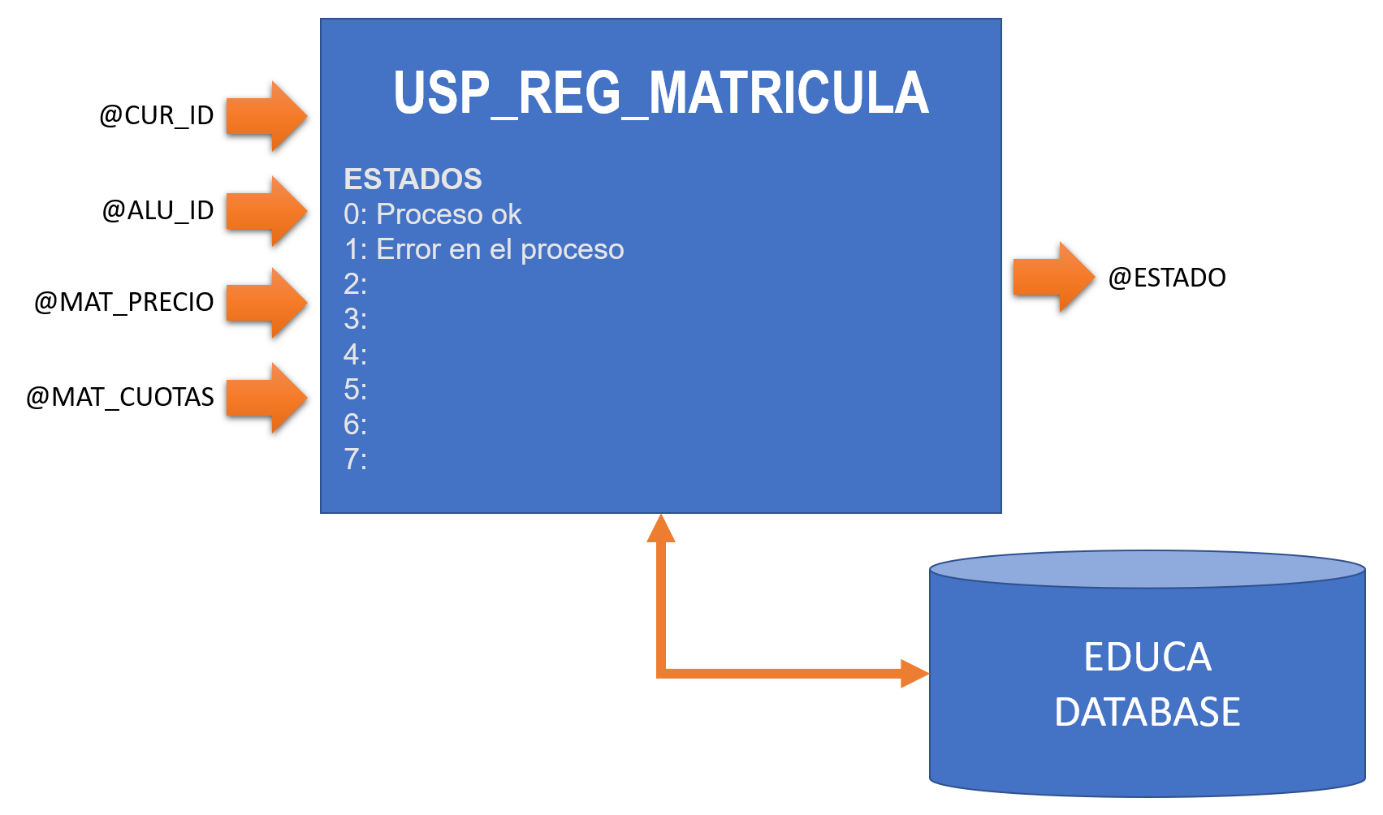
### Diagrama ER del caso de uso



Análisis:

|  |  |
| --- | --- |
| **DATO** | **DESCRIPCIÓN** |
| **CUR\_VACANTES** | Vacantes programadas. |
| **CUR\_MATRICULADOS** | Cantidad de matriculados.  Disminuye en uno con cada matricula. |
| **VACANTES DISPONIBLES** | CUR\_VACANTES – CUR\_MATRICULADOS |
| **MAT\_NOTA** | Cuando un alumno se matricula no tiene aún nota. |
| **MAT\_PRECIO** | Precio que el alumno pagara por el curso, solo si el pago es en 1 cuota el precio puede ser menor al precio de lista. |
| **MAT\_CUOTAS** | El pago puede ser en 1, 2 o 3 cuotas. |
| **MAT\_FECHA** | La fecha se debe tomar de la base de datos. |

### Análisis de la caja negra del SP



### Código del SP

ALTER PROCEDURE USP\_REG\_MATRICULA

(

@CUR\_ID INT,

@ALU\_ID INT,

@MAT\_PRECIO MONEY,

@MAT\_CUOTAS INT,

@ESTADO INT OUT

)

AS

BEGIN

-- VARIABLES

DECLARE @CONT INT, @CUR\_PRECIO INT, @DISPONIBLES INT;

-- PROCESO

SET @ESTADO = 0;

BEGIN TRY

-- Inicia la Tx

BEGIN TRANSACTION;

-- Validar curso

SELECT @CONT = COUNT(1) FROM CURSO WHERE CUR\_ID=@CUR\_ID;

if(@CONT=0)

begin

set @ESTADO= 2;

THROW 51000, 'No existe el curso.', 1;

end;

-- Validar alumno

SELECT @CONT = COUNT(1) FROM ALUMNO WHERE ALU\_ID=@ALU\_ID;

if(@CONT=0)

begin

set @ESTADO= 3;

THROW 51000, 'No existe el alumno.', 1;

end;

-- Matricula ya existe

SELECT @CONT = COUNT(1) FROM MATRICULA

WHERE cur\_id=@CUR\_ID AND ALU\_ID=@ALU\_ID;

if(@CONT=1)

begin

set @ESTADO= 4;

THROW 51000, 'Matricula ya existe.', 1;

end;

-- Validar cantidad de cuotas

if(@MAT\_CUOTAS<=0 OR @MAT\_CUOTAS>3)

begin

set @ESTADO= 5;

THROW 51000, 'Nro. de cuotas fuera de rango.', 1;

end;

-- Datos del curso

SELECT @CUR\_PRECIO=cur\_precio, @DISPONIBLES=cur\_vacantes-cur\_matriculados

FROM CURSO WHERE cur\_id = @CUR\_ID;

-- Verificar vacantes disponibles

if(@DISPONIBLES<=0)

begin

set @ESTADO= 6;

THROW 51000, 'No hay vacantes disponibles.', 1;

end;

-- Validar precio

IF(@MAT\_PRECIO<@CUR\_PRECIO\*0.5 OR @MAT\_PRECIO>@CUR\_PRECIO)

BEGIN

set @ESTADO= 7;

THROW 51000, 'Precio esta fuera de rango.', 1;

END

IF(@MAT\_CUOTAS=1 AND @MAT\_PRECIO<>@CUR\_PRECIO)

BEGIN

set @ESTADO= 7;

THROW 51000, 'El precio debe ser igual al precio de lista.', 1;

END;

-- Transacción

INSERT INTO MATRICULA(cur\_id,alu\_id,mat\_fecha,mat\_precio,mat\_cuotas)

VALUES(@CUR\_ID,@ALU\_ID,GETDATE(),@MAT\_PRECIO,@MAT\_CUOTAS);

UPDATE CURSO

SET cur\_matriculados = cur\_matriculados + 1

WHERE cur\_id = @CUR\_ID;

-- Confirma la Tx

COMMIT TRANSACTION;

END TRY

BEGIN CATCH

-- Cancela la Tx

print(ERROR\_MESSAGE());

ROLLBACK TRANSACTION;

if(@ESTADO=0) set @ESTADO=1;

END CATCH;

END;

GO

### Pruebas del SP

**Prueba 1: Curso no existe**

-- Curso no existe

DECLARE @ESTADO INT;

EXEC USP\_REG\_MATRICULA 999, 1, 99999, 100, @ESTADO OUT;

PRINT 'ESTADO: ' + CAST(@ESTADO AS VARCHAR(20));

GO

No existe el curso.

ESTADO: 2

**Prueba 2**

## CU02: Registrar un pago

### Diagrama del caso de uso

### Especificación del caso de uso

### Prototipo

### Diagrama de Secuencia

### Diagrama ER del caso de uso

### Análisis de la caja negra del SP

### Código del SP

### Prueba del SP

## CU007: Registrar un retiro

### Diagrama del caso de uso

### Especificación del caso de uso

### Prototipo

### Diagrama de Secuencia

### Diagrama ER del caso de uso

### Análisis de la caja negra del SP

### Código del SP

### Prueba del SP