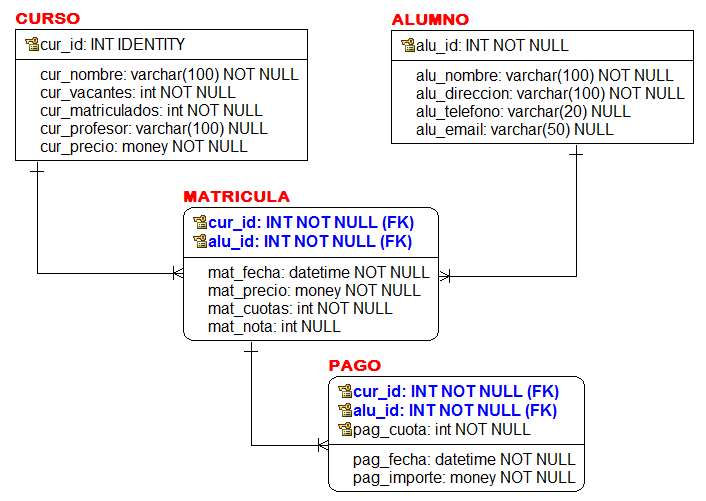
PROYECTO EDUCA

# MODELO DE DATOS



# CASOS DE USO

## CU001: Registrar Matricula

### Diagrama del caso de uso

### Especificación del caso de uso

### Prototipo

### Diagrama de Secuencia

### Diagrama ER del caso de uso



Análisis:

|  |  |
| --- | --- |
| DATO | DESCRIPCION |
| CUR\_VACANTES | Vacantes proramadas |
| CUR\_MATRICULADOS | Cantidad de alumnos matriculados |
| VACANTES DISPONIBLES | cur\_vacantes - cur\_matriculados |
| MAT\_FECHA | Se toma de la base de datos. |
| MATRICULA | En esta tabla se debe hacer un INSERT |
| CURSO | Se debe actualizar la columna CUR\_MATRICULADOS.  Cada vez que hay un nuevo matriculado se incrementa en 1. |

### Análisis de la caja negra del SP



### Código del SP

CREATE PROCEDURE USP\_MATRICULA

(

@CUR\_ID INT,

@ALU\_ID INT,

@MAT\_PRECIO MONEY,

@MAT\_CUOTAS INT,

@ESTADO INT OUT

)

AS

BEGIN

-- VARIABLES

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

-- Validar cuotas

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

BEGIN

set @ESTADO = 4;

THROW 51000, 'Cuotas fuera de rango.', 1;

END;

-- Verificar si ya existe la matricula

SELECT @CONT=COUNT(1) FROM MATRICULA WHERE alu\_id=@ALU\_ID AND cur\_id=@CUR\_ID;

IF(@CONT=1)

BEGIN

set @ESTADO = 5;

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

END;

-- Datos del curso

SELECT

@CUR\_PRECIO = cur\_precio,

@DISPONIBLES = cur\_vacantes - cur\_matriculados

FROM CURSO

WHERE cur\_id = @CUR\_ID;

-- Validar vacates disponibles

IF(@DISPONIBLES<=0)

BEGIN

set @ESTADO = 6;

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

END;

-- Validar precio

IF(@MAT\_PRECIO<(@CUR\_PRECIO/2) OR @MAT\_PRECIO>@CUR\_PRECIO)

BEGIN

set @ESTADO = 7;

THROW 51000, 'El precio no es correcto.', 1;

END;

-- Registrar Tx

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

### Prueba del SP

**Prueba 1 Curso no existe**

-- Curso no existe

DECLARE @ESTADO INT;

EXEC USP\_MATRICULA 9999, 1,10000, 10, @ESTADO OUT;

PRINT 'ESTADO: ' + CAST(@ESTADO AS VARCHAR(100));

GO

No existe el curso.

ESTADO: 2

**Prueba 2**

## CU002:

### 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:

### 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