

**Proyecto Programado que sustituye el Segundo Ordinario I
Cuatrimestre 2020*****Enunciado del Problema***

Como desarrollador se le ha contratado para elaborar un programa que permita realizar el registro de Notas de los alumnos de un centro educativo; para ello se le ha solicitado que la aplicación sea desarrollada en C# utilizando interfaz gráfica (GUI), así como, que la información sea almacenada en una base de datos SQL SERVER, que utilice colecciones y comunicación entre procesos mediante tecnología de sockets.

Por lo anterior, el programa debe manejar los conceptos de clientes y servidor y utilización de componente ADO.NET. La aplicación debe permitir el registro de alumnos, registro de cursos y registro de las notas de cada alumno por curso. El programa debe disponer de las siguientes funcionalidades y pantallas:

1. Al iniciar el programa debe desplegar una pantalla principal que permita ejecutar la aplicación "Servidor", la aplicación "Cliente" y "Salir".
2. Al llamar a la pantalla de servidor, esta debe desplegar las diferentes pantallas que permitan: Registrar Alumno, Registrar Curso, Registrar Notas de Alumno por Curso; así como, pantalla para consultar los usuarios (alumnos) pendientes de validar para tener acceso al sistema.
3. La pantalla Cliente al ser invocada debe solicitar usuario y contraseña para poder entrar al sistema; este usuario y contraseña debe estar previamente registrados y validados desde la aplicación servidor. Una vez que el usuario y contraseña sean correctos debe ingresar a la pantalla cliente y desplegar las notas por curso del usuario conectado.

Ahora bien, la parte servidor es la única que debe tener acceso a registrar, consultar y actualizar información en la base de datos de forma directa. Por lo que la parte cliente no debe tener acceso a la base de datos de forma directo, sino, para ellos debe realizarlo mediante comunicación con tecnología de sockets con el servidor.

Importante, indicar que para el desarrollo de este proyecto se debe utilizar el script de base de datos entregado con esta descripción del proyecto; así como, el cual debe ser utilizado tal y como es entregado, no se deben modificar.

Cálculos del Problema***Servidor***

1. Se debe tomar en consideración que el servidor debe permitir la comunicación de uno lo varios clientes; ya que el servidor debe permitir recibir y enviar mensajes desde y hacia los clientes que los soliciten.
2. Para la atención de clientes, el servidor se debe habilitar para escuchar clientes mediante el puerto 30000
3. Para el registro de Alumno se debe desarrollar pantalla de registro tomando en consideración:
 - Los campos por registrar de los alumnos son:
 - Identificación
 - Nombre
 - Primer Apellido
 - Segundo Apellido
 - Rol (el valor por defecto es "ESTUDIANTE")
 - Usuario
 - Contraseña
 - Desde esta pantalla todos los alumnos que se registren quedan automáticamente validados para poder utilizar la aplicación cliente, para ello debe ser registrado con el rol "ESTUDIANTE", por lo que no necesitan proceso de validación.
 - No es necesario que la pantalla despliegue los datos registrados
4. Para el registro de los Cursos se debe tomar en consideración:
 - Los campos por registrar son:
 - Número de Curso
 - Nombre del Curso
 - Nivel
 - Cantidad de Horas

- No es necesario que la pantalla despliegue los datos registrados
5. Para el registro de las notas por Alumno se debe tomar en consideración:
- Los campos por registrar son:
 - Identificación de Alumno
 - Número de Curso
 - Nota Ordinario 1
 - Nota Ordinario 2
 - Nota de Proyecto
 - Esta pantalla además de permitir el registro de la información, debe permitir consultar todos los registros almacenados para todos los estudiantes y cursos con sus respectivas notas
 - Las identificaciones de los alumnos y los números de cursos deben estar previamente registrados.
 - Esta pantalla al registrar los datos, automáticamente debe consultar la información de notas registradas. Adicionalmente, debe disponer de un mecanismo (botón) que permita únicamente consultar sin la necesidad de registrar información.
6. Debe existir una pantalla para validar o denegar los usuarios pendientes, los cuales fueron registrados desde el cliente mediante envío de mensajes entre el cliente y el servidor. Por lo anterior, esta pantalla debe permitir consultar los usuarios pendientes de validar, así como, disponer de dos opciones: "Validar" o "Denegar". Si un usuario es validado se debe cambiar su rol a "ESTUDIANTE" y si es denegado se debe cambiar el rol a "DENEGADO"

Cliente

1. Se debe tomar en consideración que el cliente NO tiene comunicación directa con la base de datos, para ello debe conectar con el servidor mediante tecnología socket para el envío y recepción de mensajes.

2. Para el ingreso a la aplicación cliente debe presentar pantalla de login o acceso, la cual solicita usuario y contraseña. Importante he de indicar que el usuario para ingresar debe estar registrado con un rol de "ESTUDIANTE".
3. Si al ejecutar la aplicación cliente no se tiene usuario registrado, desde esta pantalla se debe llamar a funcionalidad o pantalla que permita el registro del usuario.
4. Para el registro de Alumno desde el cliente se debe tomar en consideración:
 - Los campos por registrar son:
 - Identificación
 - Nombre
 - Primer Apellido
 - Segundo Apellido
 - Rol (el valor por defecto es "PENDIENTE")
 - Usuario
 - Contraseña
 - Desde esta pantalla todos los alumnos que se registren quedan como pendientes de validar, por lo que no podrán tener acceso a consultar las notas, hasta que sean validados desde la aplicación servidor. Por lo anterior, estos datos deben ser registrados con el rol "PENDIENTE".
 - Los datos deben quedar almacenados en la base de datos mediante comunicación con el servidor, no directamente con la base de datos.
 - No es necesario que la pantalla despliegue los datos registrados
5. Al digitar un usuario y contraseña valido, debe ingresar y desplegar pantalla que permita consultar automáticamente las notas del Alumno.

Validaciones requeridas en la Resolución del Problema

Para el desarrollo de este proyecto se deben tomar en consideración las siguientes validaciones o condiciones:

1. Se debe desarrollar las clases correspondientes a las entidades o tablas de la base de datos.
2. Para el registro de alumnos se debe desarrollar método que reciba como parámetro un objeto Alumno
3. Para el registro de Cursos se debe desarrollar método que reciba como parámetro un objeto Curso
4. Para el registro de Notas se debe desarrollar método que reciba como parámetro un objeto CursoAlumno
5. Se debe desarrollar método que permita consultar los alumnos "PENDIENTES" de validar, este método debe devolver una colección genérica.
6. Se deben desarrollar uno o más métodos que permitan consultar las notas por alumno, ya sea que retorne todos los registros de notas y alumnos, o bien, las notas de un alumno en específico; se debe tomar en consideración que este método o métodos debe retornar un DataTable.

Ejemplo de Pantallas de ingreso de Datos:

A continuación, se presentan ejemplos de pantallas para el registro, gestión y consulta de información, las mismas son solo una guía.

Servidor**Pantalla Registro Alumno**

Identificación	Usuario
<input type="text"/>	<input type="text"/>
Nombre	Contraseña
<input type="text"/>	<input type="text"/>
Primer Apellido	
<input type="text"/>	
Segundo Apellido	
<input type="text"/>	
Rol	<input type="button" value="Registrar"/>
<input type="text" value="ESTUDIANTE"/>	

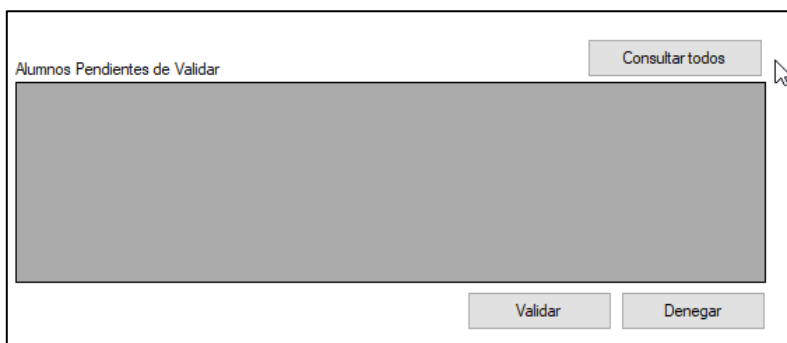
Pantalla Registro Curso

Número Curso	
<input type="text"/>	
Nombre Curso	
<input type="text"/>	
Nivel	<input type="button" value="Registrar"/>
<input type="text"/>	
Cantidad Horas	
<input type="text"/>	

Pantalla Registro Notas

Alumno	Curso	
<input type="text"/>	<input type="text"/>	<input type="button" value="Registrar"/>
Nota Ordinario 1	Nota Ordinario 2	Nota Proyecto
<input type="text"/>	<input type="text"/>	<input type="text"/>
		<input type="button" value="Consultar Notas"/>
Notas		
<div></div>		

Pantalla Consulta Alumnos Pendientes de Validar



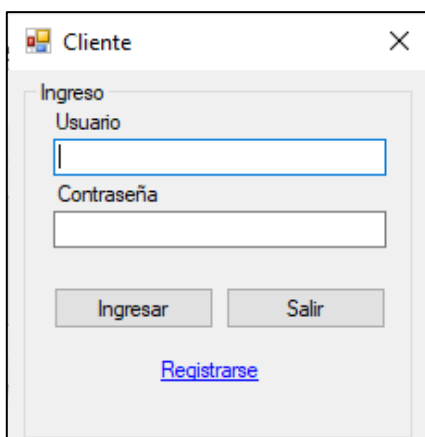
Alumnos Pendientes de Validar

Consultar todos

Validar Denegar

Cliente:

Login



Cliente

Ingreso

Usuario

Contraseña

Ingresar Salir

[Registrarse](#)

Registrar Alumno



Cliente

Datos Alumno

Identificación

Nombre

Primer Apellidos

Segundo Apellido

Rol

PENDIENTE

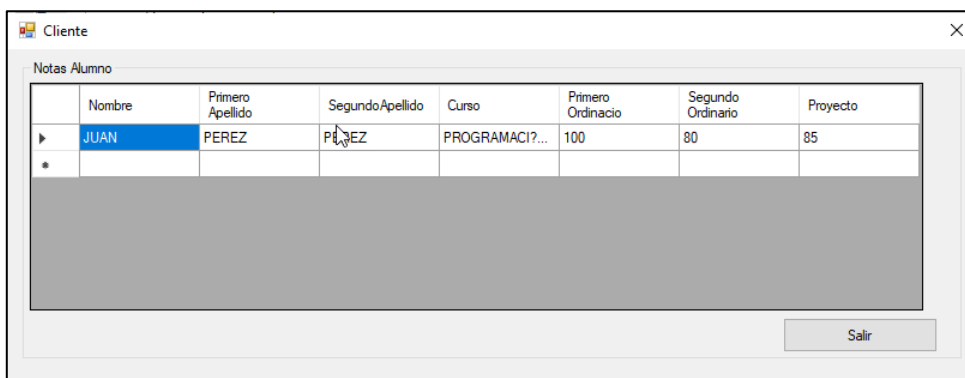
Usuario

Contraseña

Registrar

Salir

Consultar Notas

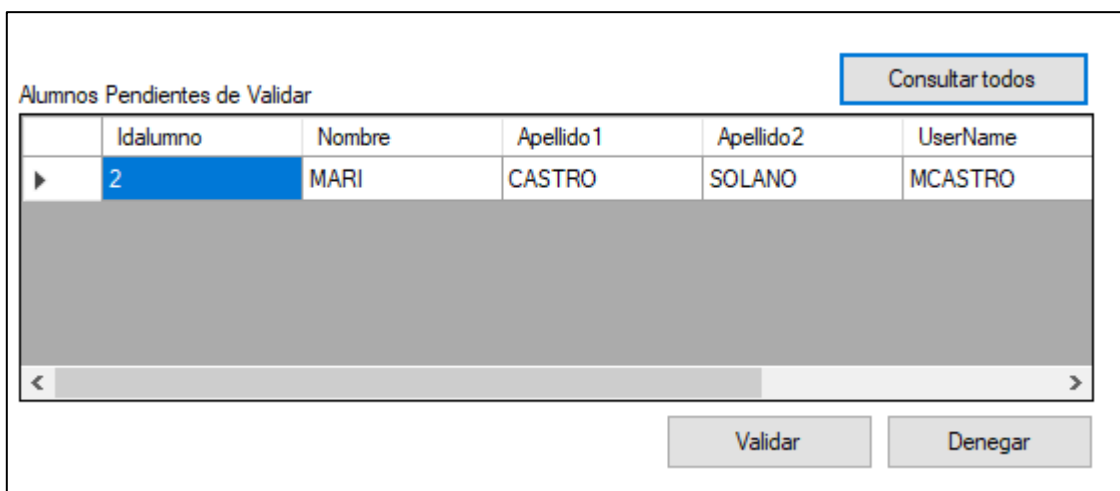


	Nombre	Primero Apellido	SegundoApellido	Curso	Primero Ordinacio	Segundo Ordinario	Proyecto
▶	JUAN	PEREZ	PEREZ	PROGRAMACI?...	100	80	85
•							

Salir

Ejemplo de Salida de Datos:

Usuarios Pendientes de validar (Pantalla desde servidor)



	Idalumno	Nombre	Apellido1	Apellido2	UserName
▶	2	MARI	CASTRO	SOLANO	MCASTRO

Consultar todos

Validar Denegar

Consulta Notas por Alumno y Curso (Desde Servidor)

Alumno

Curso

Registrar

Nota Ordinario 1

Nota Ordinario 2

Nota Proyecto

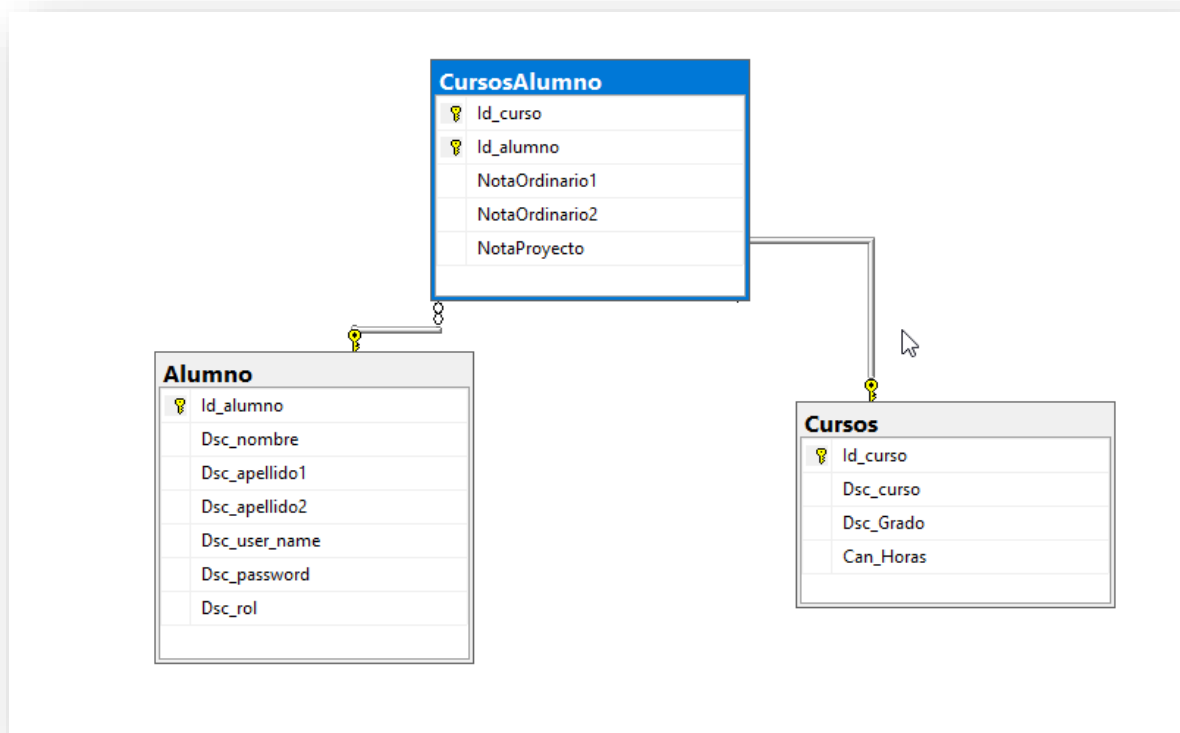
Consultar Notas

Notas

	Nombre	Apellido	Apellido2	Curso	Ordinario1
▶	JUAN	PEREZ	PEREZ	PROGRAMACIÓ...	100
	MARI	CASTRO	SOLANO	PROGRAMACIÓ...	70
*					

< >

Diagrama de base de datos.



Rúbrica de calificación

Criterio	Cumple a satisfacción lo indicado en la evaluación.	Cumple medianamente en lo indicado en la evaluación	Cumple en contenido y formato pero los aportes no son significantes	No cumple o no presenta lo solicitado
Redacción y Ortografía. Comentarios en código: No comete errores gramaticales u ortográficos dentro de los comentarios del código	2	1	0.5	0
Presentación del código: Correcto uso de variables, nombres de métodos significativos, no presenta métodos vacíos, programación orientada a objetos. Cumple con lo solicitado en la definición de los métodos	10	5	2	0
Conexión de la aplicación cliente con el servidor mediante TCP	20	10	5	0
Registro de Alumnos, cursos y notas en la base de datos desde la aplicación servidor. Validación de	20	10	5	0

Alumnos				
Validación de acceso en aplicación cliente	10	5	2	0
Registro de Alumnos desde Cliente	15	8	4	0
Consulta de Notas desde Aplicación Cliente	10	5	2	0
Correcto uso del manejo de excepciones	8	4	2	0
Interfaz de usuario. La aplicación es fácil de usar e intuitiva.	5	2	1	0
Total	100			