

Universidad Gerardo Barrios  
Facultad de Ciencia y Tecnología



Integrantes:

Fabiola Alejandra Benítez Osorto – SMSS111223  
Andrea Patricia Ramos Hernández – SMSS101123  
Ivette Azucena Mendiola Requeno – SMSS063723

Docente:

William Alexis Montes Girón

Asignatura:

Programación Computacional II

Fecha de entrega:

Domingo 17 de noviembre de 2024

### **Establecimiento de prioridades.**

Para nuestro proyecto, es fundamental identificar cuáles de las funcionalidades pendientes son prioritarias para garantizar que la aplicación cumpla con su objetivo principal de facilitar la gestión académica. Actualmente, el proyecto necesita mejoras que permitan almacenar datos de forma permanente, simplificar la búsqueda y edición de notas, así como generar reportes útiles para los docentes. Por ello, hemos clasificado las funcionalidades restantes según su nivel de prioridad, asegurando que primero se implementen aquellas que son esenciales.

<b>Prioridades</b>	<b>Descripción</b>	<b>Razón</b>
<b>Alta</b>	<b>Base de datos (MySQL o SQLite):</b> Es fundamental para almacenar las notas de manera permanente, ya que actualmente los datos solo se guardan temporalmente en la interfaz gráfica.	Sin esta funcionalidad, los datos no estarán disponibles en futuras sesiones.
<b>Media</b>	<b>Modificar las notas:</b> Permitir modificar las notas ya registradas es crucial para corregir errores o actualizar información.	La falta de esta función podría generar dificultades en la gestión de las calificaciones.
	<b>Búsqueda de notas:</b> Implementar una función de	Es esencial para mejorar la experiencia del

<b>Baja</b>	búsqueda por criterios (estudiante, asignatura, periodo, etc.) agilizará el acceso a datos específicos.	usuario, especialmente cuando se maneja un gran volumen de datos.
-------------	---	---

### **Resolución del plan de trabajo detallado en el segundo avance**

Este código tiene dos clases principales que representan diferentes ventanas en una aplicación PyQt5. A continuación, se describen sus funcionalidades:

#### **1. VentanaPrincipal**

Esta clase es una ventana inicial donde el usuario puede ingresar información sobre un docente y su asignatura. Sus funcionalidades incluyen:

**Ingreso de datos del docente:** como el campo para ingresar el nombre del docente, la selección de la materia que imparte mediante un QComboBox con opciones predefinidas y también selección del curso en el que imparte la materia.

- Validación de datos:

Se valida que todos los campos (nombre del docente, materia y curso) estén llenos antes de proceder. Si algún campo está vacío, se muestra un mensaje de advertencia (QMessageBox.warning).

**Navegación:** Al hacer clic en el botón "Ingresar", se cierra esta ventana y se abre una nueva ventana para registrar notas, que es la primera ventana que creamos en nuestro proyecto

Si los datos son inválidos, no permite avanzar.

#### **2. RegistroNotas**

Esta ventana permite registrar notas de estudiantes en una tabla. Sus funcionalidades son:

- Ingreso de información sobre notas como campo para ingresar el nombre del estudiante, la selección del periodo evaluativo, Selección del tipo de evaluación, así como el campo para ingresar la nota obtenida.
- Validación de notas: Se verifica que la nota ingresada sea un número entero entre 0 y 10. Si la nota es inválida:
  - Muestra un mensaje de advertencia
  - Si la nota es válida, se agrega a una tabla.
- Registro en la tabla: La tabla muestra un resumen de las notas ingresadas, con columnas:
  - Estudiante
  - Periodo
  - Evaluación
  - Nota
- Permite agregar múltiples registros de notas.
- Limpieza de campos: Después de guardar una nota, los campos de entrada se limpian automáticamente para facilitar el ingreso de nuevos registros.

#### Características generales:

- Las dos ventanas son independientes y están conectadas mediante un flujo lógico.
- Utiliza elementos visuales como QLabel, QLineEdit, QComboBox, y QPushButton para crear interfaces de usuario interactivas.
- La navegación entre ventanas es fluida y controlada por eventos.
- Es una solución básica para registrar información de docentes y las notas de los estudiantes.

Hasta el momento nuestro proyecto se estaría viendo de la siguiente manera habiendo implementado dos ventanas por el momento, recalcando que haría falta agregar css o personalizarlo para que se vea más estético.

Ventana Principal

Información del ...

Nombre del Docente:

mario

Materia:

Matemáticas

Curso:

7°

Ingresar

Ventana del registro de notas

Registro de Notas

Estudiante:

Periodo:

Trimestre 1

Tipo de Evaluación:

Examen

Nota:

Guardar Nota

	Estudiante	Periodo	Evaluación	Nota
1	andrea	Trimestre 1	Examen	9

## Objetivos Faltantes y Plan de Desarrollo

Para el proyecto nos faltarían aun bastantes puntos como lo son:

### Funcionalidades Pendientes:

#### 1. Base de Datos para Almacenamiento Permanente:

- **Descripción:** Implementar una base de datos MySQL para almacenar de forma permanente las notas, datos de los estudiantes y configuraciones de los docentes.
- **Desarrollo:**
  - Crear una base de datos con tablas como *estudiantes*, *notas*, y *asignaturas*.
  - Utilizar la biblioteca *PyMySQL* para conectar la aplicación PyQt5 con MySQL.
  - Configurar operaciones CRUD (Crear, Leer, Actualizar, Eliminar) para manejar la información.
- **Estado Anterior:** Actualmente, los datos se almacenan temporalmente en una tabla visual dentro de la aplicación, pero no persisten después de cerrar el programa.
- **Cambio:** Con esta mejora, la información será recuperable en futuras sesiones.

#### 2. Función de Búsqueda de Notas:

- **Descripción:** Permitir buscar registros específicos por criterios como nombre del estudiante, asignatura o periodo.
- **Desarrollo:**
  - Añadir una barra de búsqueda y botones de filtro en la interfaz *RegistroNotas*.
  - Implementar consultas SQL dinámicas en la base de datos para devolver resultados específicos.
  - Mostrar los resultados en la tabla existente.

- **Estado Anterior:** No existía una opción para buscar notas específicas, lo que dificultaba el manejo de grandes cantidades de datos.

### 3. Modificar las Notas:

- **Descripción:** Habilitar la modificación de registros existentes directamente desde la tabla o mediante un formulario adicional.
- **Desarrollo:**
  - Añadir un botón "Editar" en la interfaz.
  - Implementar funciones para cargar datos seleccionados en los campos de entrada, permitir modificaciones y guardar los cambios en la base de datos.
  - Validar los datos antes de actualizar el registro.
- **Estado Anterior:** No se podían modificar notas una vez registradas.

### 4. Mejora en el Diseño de la Interfaz:

- **Descripción:** Mejorar la estética y usabilidad de las ventanas existentes.
- **Desarrollo:**
  - Utilizar estilos personalizados con hojas de estilo CSS integradas en PyQt5.
  - Añadir iconos y colores para mejorar la experiencia del usuario.
  - Optimizar la disposición de los elementos para que sea más intuitiva.