Ing. William González

PROYECTO FINAL DE PROGRAMACIÓN

🚀 **Título del Proyecto:** *Sistema de Gestión para [Colegio/Hospital/Clínica]*

📌 **Objetivo:**
Desarrollar un sistema de gestión que permita administrar información clave de una institución (*colegio, hospital o clínica*), utilizando **Java** para la programación del cliente, **C# para la creación de APIs**, y **Git** para el control de versiones.

El sistema contará con **cuatro módulos** principales, cada uno con funcionalidades de **CRUD** (Crear, Leer, Actualizar y Eliminar), garantizando una arquitectura basada en servicios mediante **APIs RESTful**.

👥 **Requisitos del Proyecto**

- **Trabajo en parejas** (máximo 2 integrantes).
- **Uso obligatorio de Git** para la colaboración y el versionado del código.
- **Backend en C# con APIs RESTful** para manejar la base de datos.
- **Frontend en Java** para consumir las APIs y mostrar la información.
- **Base de datos relacional** para almacenar los datos de la institución.
- **Implementación de POO en Java** con conceptos como **herencia, polimorfismo y encapsulamiento**.
- **Manejo de archivos** para almacenar logs o información adicional.
- **Uso de ciclos, condicionales, estructuras de datos y clases en Java**.

🛠️ **Módulos del Sistema**

El proyecto debe contar con **4 módulos** clave. Dependiendo del tipo de sistema elegido, los módulos pueden ser los siguientes:

📚 **Opción 1: Sistema de Gestión para un Colegio**

1. **Gestión de Estudiantes** (CRUD: Registrar, modificar, eliminar y consultar estudiantes).
2. **Gestión de Docentes** (CRUD de profesores, asignación de materias).
3. **Gestión de Asignaturas** (CRUD de cursos y horarios).
4. **Gestión de Notas** (CRUD de calificaciones por estudiante).

🏥 **Opción 2: Sistema de Gestión para un Hospital/Clínica**

1. **Gestión de Pacientes** (CRUD: Registro de pacientes, historial médico).
2. **Gestión de Médicos** (CRUD de médicos, especialidades, horarios).
3. **Gestión de Citas Médicas** (CRUD de citas con fechas y horarios).
4. **Gestión de Facturación** (CRUD de pagos y facturas de consultas).

Cada módulo debe estar estructurado en **clases en Java** que utilicen los principios de **POO (Herencia, Polimorfismo y Encapsulamiento)**.

🔗 Conectividad del Proyecto

- **Base de datos:** Usar **SQL Server** o **MySQL**.
- **APIs en C#:** Deben exponer endpoints RESTful con los métodos GET, POST, PUT, DELETE.
- **Consumo de APIs en Java:** Deben realizar peticiones HTTP desde Java para interactuar con el backend.

📝 Instrucciones de Desarrollo

1️⃣ Configuración Inicial

1. Crear un **repositorio en GitHub** y agregar a ambos integrantes como colaboradores.
2. Crear una rama principal (main) y ramas secundarias para cada módulo (modulo-estudiantes, modulo-docentes, etc.).

2️⃣ Desarrollo del Backend en C#

1. Crear un **proyecto ASP.NET Core** para las APIs.
2. Implementar los modelos y controladores para cada módulo.
3. Conectar con una base de datos relacional.
4. Probar los endpoints con **Postman o Swagger**.

3️⃣ Desarrollo del Frontend en Java

1. Crear una aplicación en **Java** que consuma las APIs con HttpURLConnection o una librería como **Retrofit**.
2. Implementar la **interfaz gráfica** con Java Swing o JavaFX (opcional).
3. Crear clases con **Herencia, Polimorfismo y Encapsulamiento**.

4️⃣ Integración y Pruebas

1. Asegurar que la API responde correctamente a las solicitudes.
2. Probar la integración entre el frontend en Java y el backend en C#.
3. Documentar los endpoints y las respuestas esperadas.

5️⃣ Manejo de Git y Entrega Final

1. Hacer commits y push regularmente con mensajes descriptivos.
2. Usar **Pull Requests** para fusionar las ramas en main.
3. Entregar el **código fuente completo en GitHub**.

🗓️ **Entregables del Proyecto**

✅ Código fuente en GitHub con un **README** que explique la instalación y uso.
✅ Base de datos con datos de prueba.
✅ Documentación de los endpoints de la API en un archivo .pdf o .md.
✅ Capturas de pantalla del sistema funcionando.
✅ Presentación final mostrando el sistema en acción.

📌 **Fecha de entrega:** *[14-06-2025]*

🚀 **¡Éxito en su proyecto!**