

## **Proyecto a entregar**

**Nombre del proyecto:** Gestión de Notas.

**Tiempo:** 3 o 4 Semanas.

**Objetivo:** Desarrollar una aplicación de escritorio que permita registrar y gestionar los estudiantes, los grados, las materias, los docentes y las notas de un colegio para mejorar el seguimiento académico de los estudiantes.

**Lenguaje:** C# - .NET Framework.

**Arquitectura:** 3 CAPAS [BLL, DAL, Entity, Presentacion]

**La aplicación debe contener [REQUERIMIENTOS]**

### **Gestión de Estudiantes:**

- RF1: Registro de estudiantes con los siguientes datos:
  - Nombre completo
  - Identificación (ID)
  - Fecha de nacimiento
  - Dirección
  - Teléfono de contacto
- RF2: Modificación de los datos de estudiantes.
- RF3: Eliminación de estudiantes del sistema.
- RF4: Búsqueda de estudiantes por nombre y/o ID.

### **Gestión de Grados:**

- RF5: Registro de grados escolares (Sexto, Séptimo, Octavo, Noveno, Décimo, Once).
- RF6: Asignación de estudiantes a grados específicos.
- RF7: Visualización de la lista de estudiantes por grado.

### **Gestión de Materias y Docentes:**

- RF8: Registro de materias con los siguientes datos:
  - Nombre de la materia
  - Docente encargado [Filtrar por el tipo de materia y su especialidad]
- RF9: Asociación de materias a un grado específico.

- RF10: Modificación y eliminación de materias.

### **Registro de Notas:**

- RF11: Ingreso de notas por materia para cada estudiante (Cada nota debe obtener una observación para el estudiante, esta debe ser tenida en cuenta la momento de imprimir el boletín).
- RF12: Modificación y eliminación de notas (Solo aplica si NO se registro un periodo después, es decir si no se ha registrado nota del PERIODO 2 se puede modificar el PERIODO 1, una vez se registre nota en el PERIODO 2 ya no se puede modificar la nota del PERIODO 1, si quiere hacerlo solo lo puede hacer el superusuario).
- RF13: Consulta de notas por estudiante y materia.

### **Reportes:**

- RF14: Generación de reportes de notas por estudiante, materia y grado.
- RF15: Reportes de rendimiento por grado y materia [En Graficas y en PDF debe decir el mejor estudiante por salón y ordenarlos del mejor promedio al peor para seleccionar los puestos].

### **Gestión de Usuarios:**

- RF16: Registro de usuarios para docentes con capacidad de ingresar y modificar notas de las materias que imparten.
- RF17: Registro de un superusuario (directivos) con permisos para administrar todo el sistema y acceso a todos los reportes y funcionalidades.
- RF18: Login seguro para docentes y directivos (Deseable utilizar JWT para encriptar el inicio, si no es posible hacerlo normal).

### **Generación de Boletines:**

- RF19: Los estudiantes pueden ingresar al sistema utilizando su cédula para generar un PDF de su boletín académico, el pdf debe ser envía via CORREO ELECTRONICO.
- RF20: Los boletines incluirán la lista de materias, las notas obtenidas en cada periodo académico (Periodo 1, Periodo 2, Periodo 3, Periodo 4) y un promedio final, es decir el estudiante puede seleccionar entre imprimir el Primer, segundo, tercer o cuarto OJO: ESTO DEPENDE DEL

PERIODO EL CUAL TENGA REGISTRADO NOTAS, SI NO SE HA REGISTRADO NOTAS PARA EL PERIODO X EN SU TOTALIDAD NO SE PUEDE IMPRIMIR, TODAS LAS MATERIAS DEBEN TENER SU NOTA PARA PODER IMPRIMIR.

- RF21: Funcionalidad para imprimir, descargar y enviar el boletín en formato PDF.

### **Gestión de Docentes:**

- RF22: Registro de estudiantes con los siguientes datos:
  - Nombre completo
  - Identificación (ID)
  - Fecha de nacimiento
  - Dirección
  - Especialidad [Si es de naturales, física, matemáticas].
  - Teléfono de contacto
- RF23: Modificación de los datos de docente.
- RF24: Eliminación de docente del sistema [NO SE PUEDE ELIMINAR SI TIENE UNA MATERIA ASIGNADA].
- RF25: Búsqueda de DOCENTE por nombre y/o ID.

### **Entregables**

- Código Fuente:
  - Aplicación completamente funcional con el código fuente comentado.
  - Desde la creación del proyecto este debe estar subido en GITHUB, durante el desarrollo se debe evidenciar los commit y push.

### **Documentación:**

- Documento de especificaciones de software.
- Guía de usuario y administrador [Como usar la aplicación].

Diagramas de diseño (Modelo de base de datos, diagramas de clases).  
[Aprender cómo hacerlo y lo implementan].

**Deseable:**

- Código limpio.
- No Trabajar con sentencias directas si no con procedimientos almacenados [Preguntar a William como implementarlos en caso de que no sepan o que les regale una muestra de la creación del SP].
- Buen diseño y amigable.
- Fácil de usar y entender.
- Implementar cosas nuevas adicionales a los requerimientos.
- Innovación.

**RETO:**

Desarrollar este software utilizando los conocimientos del primer curso y el actual, adicional aprender a buscar en internet nuevos conocimientos e intentar implementarlos en la aplicación.

**Premios:**

- La primera persona en terminar gana una salchipapa para dos personas de su restaurante favorito.
- La segunda persona en terminar gana una salchipapa personal.
- La tercera persona en terminar se gana un perro caliente.