



UNIVERSIDAD NACIONAL JORGE BASADRE GROHMANN

FACULTAD DE INGENIERÍA

Escuela Profesional de Ingeniería en Informática y Sistemas

Informe final

Proyecto Sistema de Notas de Colegio

Curso: Programación avanzada

Docente: MSc. Ing. Hugo Manuel Barraza Vizcarra

Integrantes:

Russbell Juan Pablo Arratia Paz 2025-119009

Alexander Efrain Contreras Rodriguez 2025-119003

Tacna – Perú
2025

ÍNDICE GENERAL

1. Resumen	3
2. Antecedentes	3
3. Planteamiento del problema	3
a. Problema	3
b. Justificación	3
c. Alcance	4
4. Objetivos	4
5. Marco teórico	4
6. Desarrollo de la solución	5
a. Tecnología de Desarrollo	5
b. Metodología de implementación	5
(VISIÓN, SRS, SAD)	
7. Conclusiones y recomendaciones	7
8. Bibliografía	8
9. Anexos	9

1. Resumen

El presente proyecto, titulado "Proyecto Sistema de Notas de Colegio", consiste en el diseño y desarrollo de una aplicación ejecutable en consola usando el lenguaje de programación C++. El objetivo principal es simular la gestión básica de notas de una institución educativa, permitiendo el uso diferenciada entre dos tipos de usuarios: estudiantes y docentes.

Para la implementación se utilizó el paradigma de programación de la Programación Orientada a Objetos (POO), utilizando herencia, el encapsulamiento y el polimorfismo. El sistema permite registrar y visualizar datos personales, así como asignar notas, registrar estudiantes, y ver el registro de notas. El resultado es un prototipo funcional que aprovecha la reutilización de código mediante una clase base llamada Persona y sus derivadas, facilitando la escalabilidad futura del software.

2. Antecedentes

La gestión de la información en las instituciones educativas se realizaba de manera manual, utilizando registros físicos en papel que eran propensos al deterioro, pérdida de información y errores en la transcripción.

Algunos colegios aún no poseen un sistema de notas para los estudiantes y profesores, desaprovechando las ventajas que la tecnología actual ofrece, ya sea en conservación de información así como eficiencia.

3. Planteamiento del problema

a. Problema

El manejo manual de datos académicos genera inconsistencias y redundancia. Sin un sistema centralizado que utilice una arquitectura de objetos, el mantenimiento del código para nuevas funcionalidades se vuelve desorganizado. Existen instituciones educativas que aún no manejan un sistema de notas y sus usuarios son quienes dificultan su interacción mientras se mantenga así.

b. Justificación

El desarrollo de este sistema se da por la necesidad de aplicar conceptos avanzados de programación para optimizar el manejo de datos para aquellas instituciones que por diversos motivos no pueden implementarlas.

c. Alcance

El prototipo presentado abarca la consulta de notas por parte de los estudiantes, así como la subida de estos por parte de los profesores, además de generar reportes y contar con búsqueda y ordenación.

4. Objetivos

Objetivo General

- Desarrollar el prototipo de un sistema de notas para un colegio usando el lenguaje C++ y el paradigma de Programación Orientada a Objetos.

Objetivos Específicos

- Aplicar el concepto de Herencia para compartir atributos comunes y Polimorfismo para la visualización de datos.
- Implementar métodos de búsqueda y ordenación en el manejo de los registros de estudiantes, profesores, notas.
- Generar reportes a documento de texto de la información de notas que maneja el sistema.

5. Marco teórico

- Clase y Objeto: Una clase es la plantilla que define atributos y métodos; un objeto es una instancia concreta de dicha clase.
- Herencia (Inheritance): Mecanismo por el cual una clase derivada adquiere propiedades de una clase base (Persona). Compartiendo atributos y métodos pero diferenciándose con los propios.
- Polimorfismo: Capacidad de una función (mostrarDatos()) de comportarse diferente según el objeto que la invoque.
- Encapsulamiento: Protección de los datos (private y protected) accesibles sólo a través de métodos públicos (getters y setters).

6. Desarrollo de la solución

a. Tecnología de Desarrollo

Lenguaje de programación C++: Lenguaje de propósito general que ofrece control a bajo nivel y características de alto nivel orientadas a objetos.

Compilador utilizado: Dev c++ Version 6.3

Librerías utilizadas: <iostream> (entrada/salida), <fstream> (archivos)

b. Metodología de implementación (VISIÓN, SRS, SAD)

- **Visión**

El presente proyecto de sistema busca crear una herramienta de consola ligera y rápida que permita a los usuarios de un colegio consultar y manipular información en una interfaz ligera, priorizando la lógica, funcionalidades y la estructura de datos.

- **Especificación de requerimientos de software**

- **Requerimientos funcionales**

#	Descripción
Requisito Funcional 01:	El sistema debe desplegar diferentes opciones según el tipo de usuario.
Requisito Funcional 02:	El estudiante debe poder ver sus datos personales, notas y generar reportes.
Requisito Funcional 03:	El profesor debe poder ver el curso a cargo, modificar notas, y generar reportes.
Requisito Funcional 04:	Se debe poder ordenar y buscar notas de usuarios.

- **Requerimientos no funcionales**

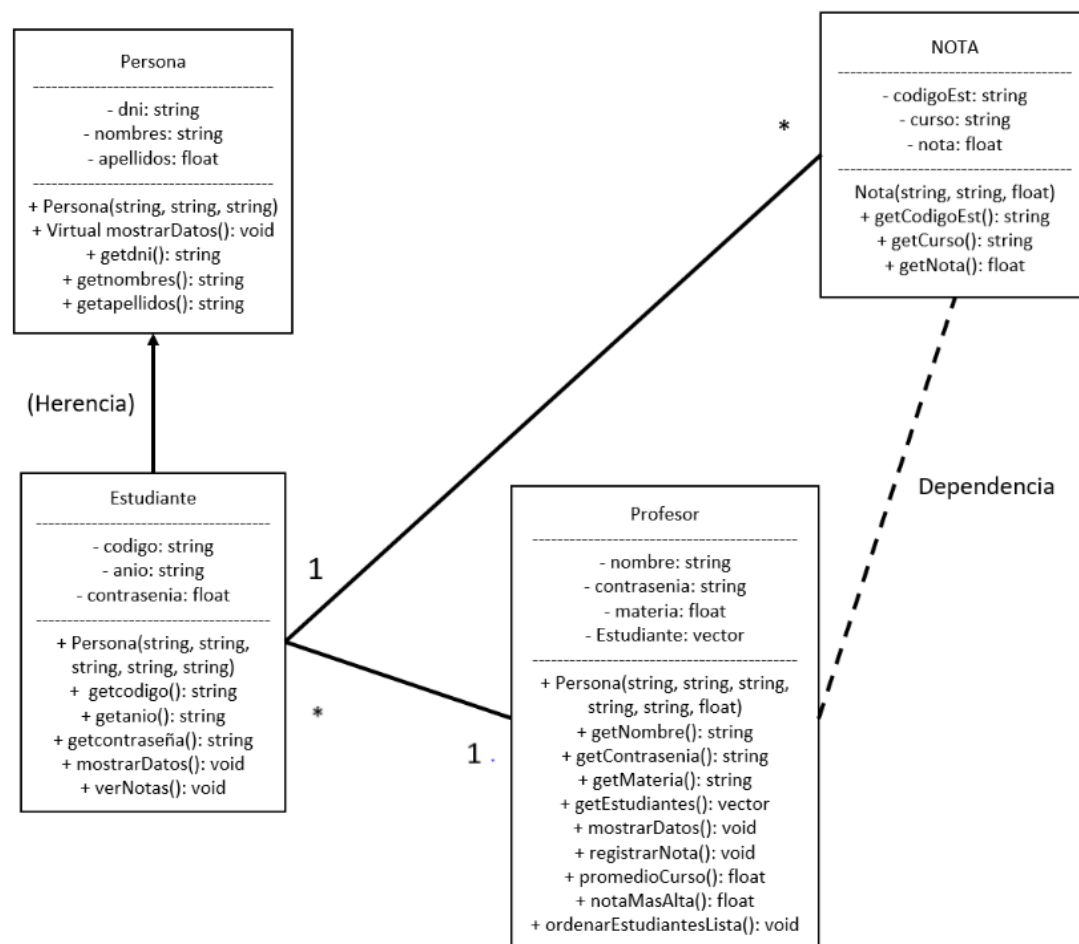
Necesidad	ID Req.	Descripción
1 Facilidad	RNF-01	Que sea sencillo en su manejo y muestre mensajes de error, no tener demasiadas opciones.
2 Escalabilidad	RNF-02	Que pueda agregarse, modificarse o eliminarse funciones.
3 Seguridad	RNF-03	Contempla que por diferentes usuarios su acceso es diferente, así como la manipulación de datos. Y que para acceder a un usuario que pueda modificar pida credenciales para acceder.

- **Reglas del negocio**

- Los usuarios debe usar cuenta y contraseña para acceder al sistema
- Los profesores pueden modificar las notas.
- Los reportes pueden generarse en cualquier momento.

- **Arquitectura de software**

- Diagrama de clase



7. Conclusiones y recomendaciones

Conclusiones

- Se logró implementar exitosamente una arquitectura basada en clases, donde la creación de la clase base y sus derivadas demostró la eficiencia para evitar la redundancia de código. El uso de métodos virtuales validó el polimorfismo, haciendo que el sistema responda diferente según el usuario.
- La integración de algoritmos de búsqueda y ordenación es útil para el manejo de los registros. La aplicación de estos algoritmos permite a los usuarios localizar información específica y organizarlos como crean conveniente.
- El uso de la librería <fstream> permitió concretar exportación de datos. La funcionalidad de generar reportes en documentos de texto .txt es clave, porque permite extraer la información accesible al momento por consola hacia un archivo portable.
- El sistema logró una distinción de accesos donde los docentes pueden modificar la información académica y los estudiantes únicamente consultarla.

Recomendaciones

- Aprovechar la arquitectura base Persona para agregar futuros roles, como Administrativos, y expandir clases siendo eficiente y reutilización de código.
- Optimizar los algoritmos de búsqueda y ordenamiento para que mantengan su velocidad y eficiencia en caso de que aumente la cantidad de registros.
- Ampliar el uso de <fstream> para que, aparte de documentos de texto .txt, pueda exportarse en formatos como .csv, facilitando más la portabilidad de datos.
- Validar permisos de accesos para asegurar que ninguna entrada errónea permita a un usuario Estudiante pueda hacer modificaciones.

8. Bibliografía

Joyanes Aguilar, L. (2008). Programación en C++: Algoritmos, estructuras de datos y objetos (2ª ed.). McGraw-Hill Interamericana.

Stroustrup, B. (2013). El lenguaje de programación C++ (4ª ed.). Addison-Wesley.

cplusplus.com. (2024). C++ Reference: Input/Output with files (fstream).

Recuperado de <https://cplusplus.com/reference/fstream/>

9. Anexos

Información del compilador DEV C++ Versión 6.3

