Reporte ejecutivo: “**Proyecto Sistema de Información (SIA) Avance**”

ICI2241-1- **PROGRAMACION AVANZADA** (1S2023)

**Gestión de Asistencia de alumnos de un colegio**

Integrantes: Martín Cevallos

Ignacio Cuevas

Vicente Jorquera

Profesor: Claudio Cubillos

Ayudante: Leonardo González

**Introducción**:

En el siguiente reporte ejecutivo procederemos a hacer el informe de nuestro proyecto de gestión de asistencia de alumnos de un colegio, en el cual haremos un registro de la asistencia de alumnos y alumnas, los distintos días del año escolar; junto con un registro de inasistencia extraordinarias y sus salidas anticipadas de horario. Para aquello, realizamos un proyecto en JAVA para poder realizar nuestro trabajo.

También hay que tener en cuenta que responderemos todo el “Proyecto Sistema de Información (SIA) Avance”, desde el SIA1.1 hasta SIA1.10, sumado con” Proyecto Sistema de Información (SIA) Final” desde el SIA2.1 hasta el SIA2.10.

ÍNDICE DE CONTENIDO

[Proyecto Sistema de Información (SIA) 1.1 4](#_Toc133864411)

[Proyecto Sistema de Información (SIA) 1.2 4](#_Toc133864412)

[Clase Alumno 5](#_Toc133864413)

[Clase Asistencia 5](#_Toc133864414)

[Clase Colegio 5](#_Toc133864415)

[Proyecto Sistema de Información (SIA) 1.3 5](#_Toc133864416)

[Proyecto Sistema de Información (SIA) 1.4 6](#_Toc133864417)

[Proyecto Sistema de Información (SIA) 1.5 6](#_Toc133864418)

[Proyecto Sistema de Información (SIA) 1.6 7](#_Toc133864419)

[Proyecto Sistema de Información (SIA) 1.7 7](#_Toc133864420)

[Proyecto Sistema de Información (SIA) 1.8 y 1.9 8](#_Toc133864421)

[Pasar lista: 8](#_Toc133864422)

[Buscar Alumno 9](#_Toc133864423)

[Riesgo de repitencia: 9](#_Toc133864424)

[Modificar Asistencia: 9](#_Toc133864425)

[Agregar alumno: 10](#_Toc133864426)

[Eliminar Alumno: 10](#_Toc133864427)

[Proyecto Sistema de Información (SIA) 2.1 11](#_Toc133864428)

[Proyecto Sistema de Información (SIA) 2.2 11](#_Toc133864429)

[Proyecto Sistema de Información (SIA) 2.3 11](#_Toc133864430)

[Proyecto Sistema de Información (SIA) 2.4 12](#_Toc133864431)

[Proyecto Sistema de Información (SIA) 2.5 12](#_Toc133864432)

[Proyecto Sistema de Información (SIA) 2.6 13](#_Toc133864433)

[Proyecto Sistema de Información (SIA) 2.7 14](#_Toc133864434)

[Proyecto Sistema de Información (SIA) 2.8 14](#_Toc133864435)

[Proyecto Sistema de Información (SIA) 2.9 14](#_Toc133864436)

[Proyecto Sistema de Información (SIA) 2.10 14](#_Toc133864437)

# Proyecto Sistema de Información (SIA) 1.1

Un profesor, Daniel Ayala, de un colegio, The Kingstown College, olvida constantemente pasar la lista en su curso. Lo que hace normalmente es pasarle una hoja al primer alumno, el cual pone su RUT, y luego le pasa la hoja al siguiente alumno, y así hasta llegar al ultimo alumno o alumna de la sala. Luego revisa el grupo de WhatsApp para anotar a cada alumno que faltó. Y, finalmente, en el transcurso del día, va anotando a los/las alumnas que se retiran anticipadamente.

Para poder ayudar al profesor, necesitamos crear un sistema similar al que usa él, pero hacer de manera digital, para poder tener un registro de la asistencia más ordenado, rápido y efectivo. Para aquello, implementaremos un programa en JAVA básico, pero eficiente, en el cual el profesor podrá pasar la asistencia de la clase, mostrar las asistencias de la clase y agregar una asistencia manualmente, ya sea un alumno nuevo o de un alumno existente. Lo anteriormente comentado sería lo aparecería en el menú de la toma de asistencia del profesor.

# Proyecto Sistema de Información (SIA) 1.2

En el siguiente apartado, haremos un diseño conceptual de clases del dominio y su respectivo código en Java.

## Clase Alumno

La clase alumno será la encargada de tener la información primordial de los estudiantes, con nombres intuitivos. Esto se maneja con un “*String*” llamado nombre, el cual entrega el nombre del estudiante; y un “*Integer*” de nombre rut, que entrega el RUT del alumno.

Y por último, tenemos arreglo de asistencias, es decir, un arreglo de tipo asistencia, el cual guarda todas las asistencias del alumno.

Texto

Descripción generada automáticamente

## Clase Asistencia

En segundo lugar, tenemos la clase mencionada anteriormente, “Asistencia”, la cual posee 5 variables de instancia. Las 4 primeras: presente, ausente, retiro\_anticipado y ausencia\_justificada, los cuales son “*booleanos*” que representan la asistencia de ese día. El día se guarda en una variable llamada fecha, que es de tipo “*LocalDate*” (se nos hizo más fácil manejar los días con este tipo para que en un futuro podamos hacer un calendario de asistencias).

Texto

Descripción generada automáticamente

## Clase Colegio

La clase Colegio es la clase principal del programa, que es donde se van guardando los datos del alumnado. Esto se maneja con un *HashMap* de tipo Alumno.

Texto

Descripción generada automáticamente

# Proyecto Sistema de Información (SIA) 1.3

Como se mostró en SIA1.2, cada variable de instancia está con su visibilidad privada (private). A su vez, cada variable instanciada posee su getter y setter respectivamente.

# Proyecto Sistema de Información (SIA) 1.4

Como se mencionó en SIA1.2, en el apartado de Clase Base de datos, se crea esta clase con el fin de poder crear datos iniciales del sistema, para la demostración funcional de esta. Esta clase solo posee un método, que le llamamos EnvasarDatos, que, como muestra a continuación, inicializa un arreglo de string con 3 nombres (los nombres usados en la demostración fueron creados al azar, cualquier parecido con la realidad son mera coincidencia).

Luego, a cada alumno se le asigna un RUT al azar (con un numero random entre 15 millones y 20 millones). También se crea un día al azar de marzo, entre el 1 y el 20 (con el fin de que sean 3 días consecutivos y no tope con el 30). Finalmente, se agregan los 3 alumnos al HashMap creado, el cual será explicado a mayor profundidad más adelante.

Esta clase, EnvasarDatos, ya no existe dentro de nuestras clases debido a que no la necesitaremos, porque ahora tenemos un LeerCsv que obtendrá datos desde un archivo externo.

Texto

Descripción generada automáticamente

# Proyecto Sistema de Información (SIA) 1.5

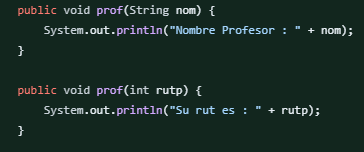
Como mencionamos anteriormente, nuestro sistema tiene dos Java Collection Framework (JCF) anidados entre sí; uno es un HashMap (primera colección), el cual lleva como Key un entero (el Rut del alumnado), y Value lleva los alumnos del colegio. A su vez, el alumnado lleva un arreglo de tipo Asistencia, el cual, como dijimos anteriormente, lleva la asistencia de cada alumno y alumna del colegio.

Diagrama

Descripción generada automáticamente

# Proyecto Sistema de Información (SIA) 1.6

Se utiliza la sobrecarga de métodos en la clase profesor, al usar 2 métodos del mismo nombre con distinto tipo de entrada para mostrar el Rut y el nombre del profesor



# Proyecto Sistema de Información (SIA) 1.7

Como mencionamos en el SIA1.5, el mapa del JCF es de tipo HashMap, el cual almacena los datos del estudiante por día, y los diferencia por el RUT de cada estudiante.

# Proyecto Sistema de Información (SIA) 1.8 y 1.9

El menú para el sistema está pensado en el profesor de aula para que pueda pasar la lista de manera rápida y eficaz. Para ello, el profesor debe ingresar la contraseña secreta, con tal de que ningún alumno pueda acceder a la información tan fácil.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamenteLuego se despliega la información necesaria para un correcto funcionamiento de la asistencia:

## Pasar lista:

Este botón desplegará toda la lista de los alumnos registrados por el alumno.

Interfaz de usuario gráfica

Descripción generada automáticamente con confianza media

## Buscar Alumno

En este caso, el botón permite buscar un alumno por el RUT:

Cuadrado

Descripción generada automáticamente con confianza media

## Riesgo de repitencia:

Este botón filtrará el alumnado que tiene riesgo de repitencia por asistencia. Más adelante se explicará a detalle esta función.

## Modificar Asistencia:

Esta función te permite cambiar/modificar la asistencia de un alumno o alumna de un día en específico.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

## Agregar alumno:

Esta función es por si hay algún alumno o alumna nueva, al cual le puede poner un tipo de asistencia especial. Se agregará como Presente el día en el que se accede.

Icono

Descripción generada automáticamente

## Eliminar Alumno:

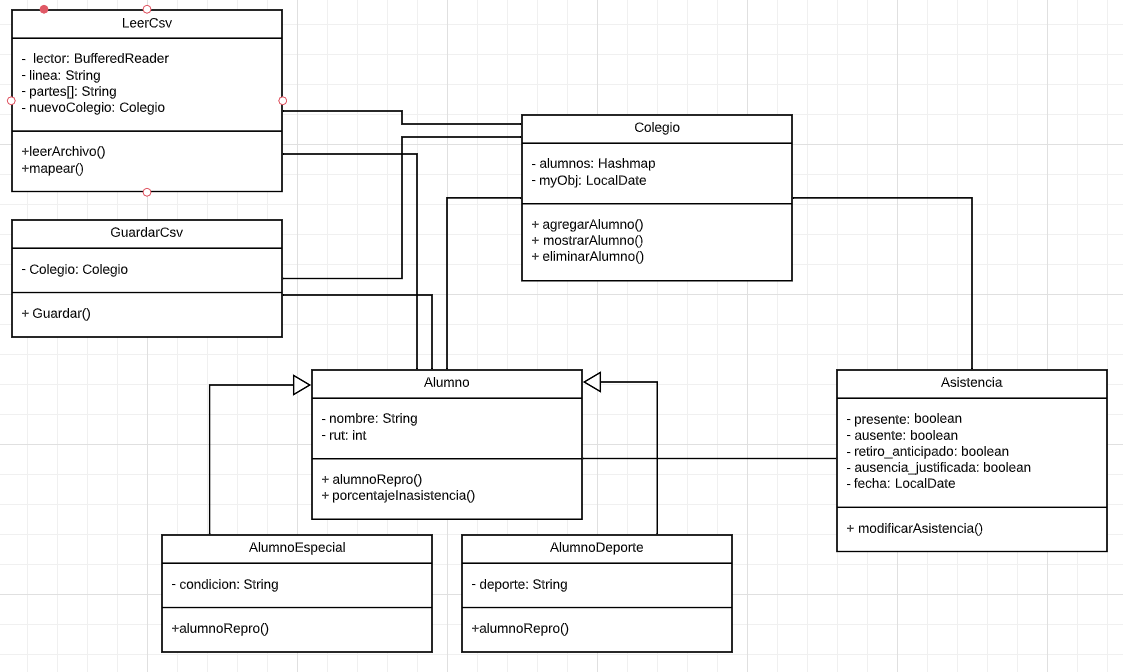
Esta función permite eliminar un alumno de la lista.

Imagen que contiene Icono

Descripción generada automáticamente

# Proyecto Sistema de Información (SIA) 2.1

A continuación se presenta el diagrama de clases UML, señalando a través de flechas, la herencia producida, y a través de líneas, las clases que tienen asociación con la otra.

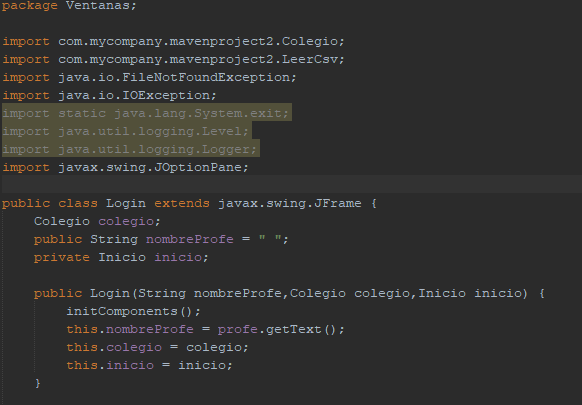


# Proyecto Sistema de Información (SIA) 2.2

El programa utiliza la clase "LeerCsv" para leer los datos de un archivo CSV y luego mapearlos a objetos "Alumno" y "Asistencia". Los datos son luego almacenados en un objeto "Colegio". La clase "GuardarCsv" se utiliza para escribir los datos de los alumnos y sus asistencias en un archivo CSV.

# Proyecto Sistema de Información (SIA) 2.3

El código implementa una interfaz gráfica de usuario, utilizando el framework SWING para Java. Esta consiste en una ventana principal con un panel que contiene una serie de botones que permiten al usuario realizar diversas acciones, como agregar, buscar y eliminar alumnos, modificar la asistencia, guardar los datos y ver si un alumno está en riesgo de repitencia. Véase (SIA)1.8 Y (SIA 1.9)

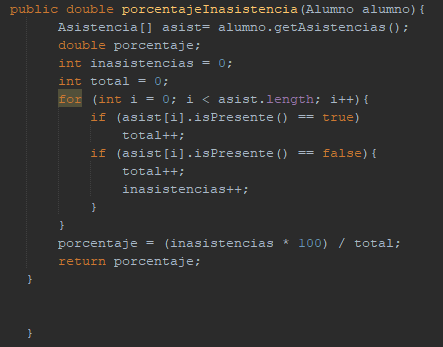


# Proyecto Sistema de Información (SIA) 2.4

Como se muestra en los puntos (SIA)1.8 Y (SIA 1.9) se han utilizado ventanas para tener una interfaz gráfica y de esta manera el usuario tenga una interacción más amena con el programa

# Proyecto Sistema de Información (SIA) 2.5

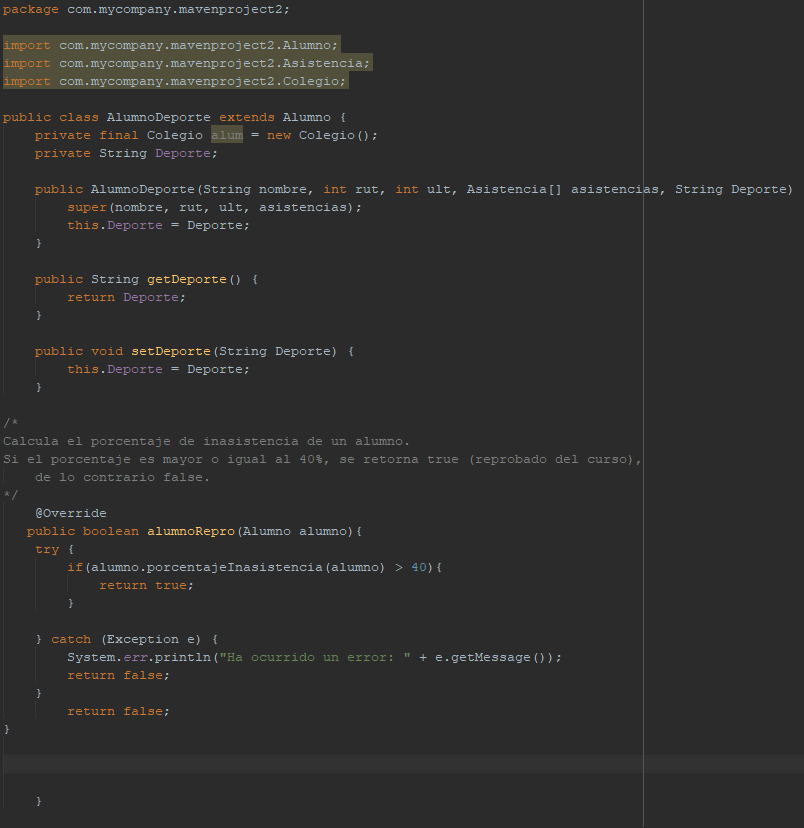
El código cumple con el requisito de incluir al menos una funcionalidad propia que sea útil para el negocio. Esta funcionalidad se logra mediante la implementación del método porcentajeInasistencia, que calcula el porcentaje de inasistencias de un alumno y lo retorna como un valor de tipo double. Además, en una de las ventanas esta la opción de ver alumnos que estén en riesgo de repitencia, filtrándolos.



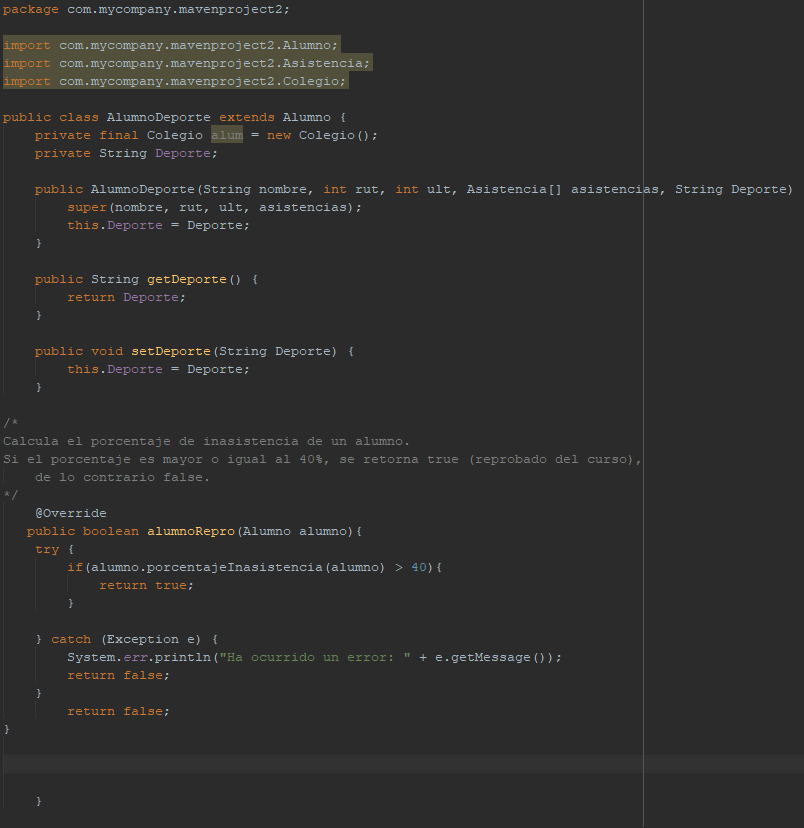
# Proyecto Sistema de Información (SIA) 2.6

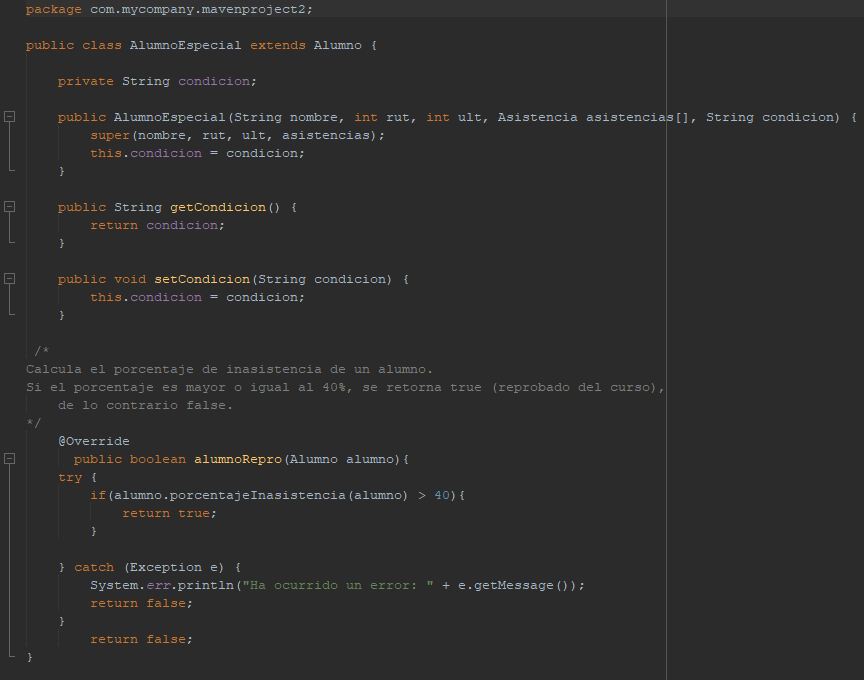
se puede concluir que cumple con el enunciado. El código está bien modularizado y sigue las buenas prácticas de documentación interna y legibilidad.

Un ejemplo de esto es en la subclase AlumnoDeporte



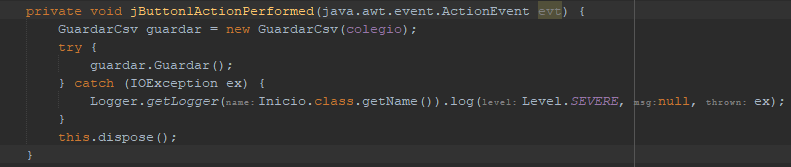
# Proyecto Sistema de Información (SIA) 2.7

En el proyecto se diseñaron y codificaron dos clases que utilizan sobreescritura de métodos, a continuación, se puede ver una imagen de ambas.

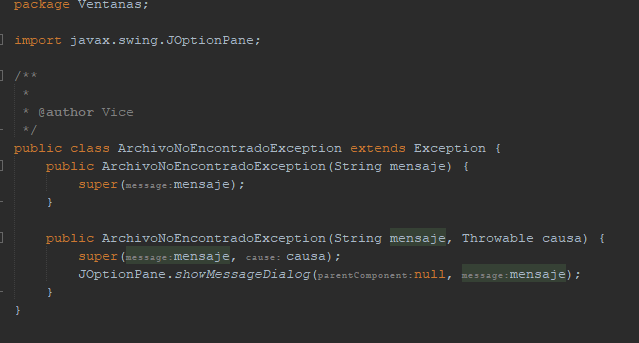


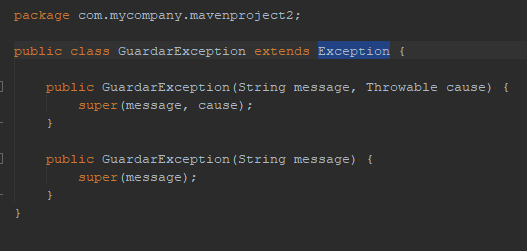
# Proyecto Sistema de Información (SIA) 2.8

Se hace uso del try catch en la ventana “inicio” para capturar algún error al ejecutar el método “guardar()”. También se hace uso en la ventana “login” al momento de cargar los archivos del CSV.



# Proyecto Sistema de Información (SIA) 2.9

Se crearon 2 clases llamadas “LeerArchivoException.java” y “GuardarException.java” que se extienden de la super clase “Exception” las cuales manejan los mensajes de error. Uno se ocupa en Login.java y el otro en GuardarCsv.java



# Proyecto Sistema de Información (SIA) 2.10

El proyecto ha sido subido a GitHub, el cual posee 25 commits realizadas por naxoruto (Ignacio Cuevas), pero editadas por la totalidad del grupo.

El link del GitHub es <https://github.com/naxoruto/Proyecto-JAVA>