



Reporte entrega final

Proyecto P00 – Gestión de asistencia de un colegio

Por Sebastián García, Guillermo González y Benjamín Navarrete

Índice

1. Entrega parcial 1	3
1.1. Realizar un análisis de los datos a utilizar y principales funcionalidades a implementar que dan sentido a la realización del proyecto	3
1.2. Diseño conceptual de clases del Dominio y su código en Java	4
1.3. Todos los atributos de todas las clases deben ser privados y poseer sus respectivos métodos de lectura y escritura (getter y setter)	5
1.4. Se deben incluir datos iniciales dentro del código	5
Opcional: Leer datos desde archivo/base de datos	5
2. Entrega parcial 2	7
2.1. Diseño conceptual y codificación de 2 (dos) niveles de anidación de colecciones de objetos	7
2.2. Diseño conceptual y codificación de 2 (dos) clases que utilicen sobrecarga de métodos	7
2.3. Diseño conceptual y codificación de al menos 1 clase mapa del JCF	8
3. Entrega parcial 3	9
3.1. Diseño de diagrama de clases UML de lo considerado hasta la EP3	9
3.2. Menú del sistema	12
3.2.1. Inserción manual/agregar elemento	13
3.2.2. Mostrar por pantalla listado de elementos. Esto para cada una de las 2 colecciones anidadas	14
A. Entrega acumulada A	16
A.2. Menú con funcionalidades de edición y eliminación de elementos	16
A.3. Se debe generar un reporte en archivo txt que considere mostrar datos de las 2 colecciones anidadas (ej: csv)	16
A.4. El código fuente debe estar bien modularizado de acuerdo a lo descrito en el informe además de seguir las buenas prácticas de documentación interna y legibilidad	16
A.5. Todas las funcionalidades pueden ser implementadas mediante consola	16
Opcional: La implementación de una o más interfaces gráficas de usuario, con la inclusión de ventanas con componentes AWT, SWING, JAVAFXu otro	16
A.6. Utilización de GitHub (Realización de al menos 3 Commit)	16
4. Entrega parcial 4	17
4.1. Se deben incluir al menos 2 funcionalidades propias que sean de utilidad para el negocio (distintas de la inserción, edición, eliminación y reportes)	17
4.1.1. Seleccionar un objeto por criterio, considera la selección de un objeto basado en un criterio específico, involucrando dos o más colecciones anidadas. Por ejemplo, selección del alumno con la nota final más baja de todos los cursos, o seleccionar el pasajero más joven de todos los buses de una compañía	17

4.1.2.	Subconjunto filtrado por criterio: considera la selección de un subconjunto de objetos basado en un criterio específico, involucrando dos o más colecciones anidadas. Por ejemplo, selección de los alumnos con nota final entre 4,0 y 7,0 de entre todos los cursos; o seleccionar a todos los pasajeros que tengan asiento impar de entre todos los buses de la compañía	17
4.2.	Diseño y codificación de 2 (dos) clases que utilicen sobreescritura de métodos	17
4.3.	Diseño y codificación de 1 (una) clase abstracta que sea padre de al menos 2 (dos) clases. La clase abstracta debe ser utilizada por alguna otra clase (contexto)	17
4.4.	Diseño y codificación de 1 (una) interfaz que sea implementada por al menos 2 (dos) clases. La interfaz debe ser utilizada por alguna otra clase (contexto)	17
	Opcional: Generar documentación con Javadoc.	18
B.	Entrega acumulada B	19
B.2.	Se deben implementar al menos 3 ventanas gráficas (GUIs en AWTo SWING): 1 ventana de menú, 1 ventana de agregar elemento y 1 ventana de listar elementos	19
B.3.	Se debe aplicar encapsulamiento y principios OO	19
B.4.	Continuidad en la utilización de GitHub (Realización de al menos 3 Commit adicionales a los ya hechos en la parte A)	19
F.	Entrega final	20
F.2.	Implementar el manejo de excepciones capturando errores potenciales específicos mediante Try-catch	20
F.3.	Crear 2 clases que extiendan de una Excepción y que se utilicen en el programa	20
F.4.	Aplicación del patrón de diseño Strategy (Estrategia)	20
	Opcional: Reemplazar los datos iniciales en el código por una conexión con base de datos local (MySQL), archivo de texto CSVo Excel, para la persistencia de datos	20
	Opcional: Considerar la implementación del patrón Modelo-Vista-Controlador (MVC) en la arquitectura del sistema	20
	Consideraciones generales	21
.1.	Bla bla bla	21

1. Entrega parcial 1

1.1. Realizar un análisis de los datos a utilizar y principales funcionalidades a implementar que dan sentido a la realización del proyecto

La problemática a abordar en este proyecto es la necesidad que tienen los colegios de llevar un registro de asistencia de sus estudiantes para cada año escolar, y para ello, la principal información que deberemos manejar son los estudiantes y los cursos. Los cursos se dividirán en diferentes niveles y cada nivel subdividirá en paralelos, cada uno de estos contará con la información del profesor a cargo de este curso y la lista de alumnos correspondiente.

Los alumnos contarán con información básica de ellos (nombre, rut, apellido, etc.) y cada uno contará con su registro de asistencia correspondiente al año escolar.

Con esto buscamos que se generen archivos que muestren la lista de estudiantes de cada curso y la asistencia de estos.

Funcionalidades principales:

- Registrar asistencia de un curso para cada día de clases.
- Registrar inasistencias.
- Registro de salidas antes de horario.
- Consultar registro de asistencia, inasistencias y salidas antes de horario de cualquier día del año.

Funcionalidades secundarias:

- Calcular porcentaje de asistencia
- Generar ranking de asistencia en distintos niveles del colegio
- Generar reportes en base a la asistencia

Entidades:

- Alumnos del colegio
- Cursos y niveles
- Profesores
- Apoderados
- Registro de asistencia

Manejo de datos:

- En principio, los datos relevantes para los procesos serán almacenados en un archivo de texto plano (CSV), el que contendrá las funcionalidades necesarias para agregar, modificar, listar o eliminar datos de él.
- Para efecto de trabajar con datos iniciales, se creó una clase Fakedata que incluye una librería externa (Faker). Esta clase permite crear datos iniciales aleatorios, y los guarda en los ficheros CSV respectivos, utilizando la clase Datafile de manejo de archivos.

1.2. Diseño conceptual de clases del Dominio y su código en Java

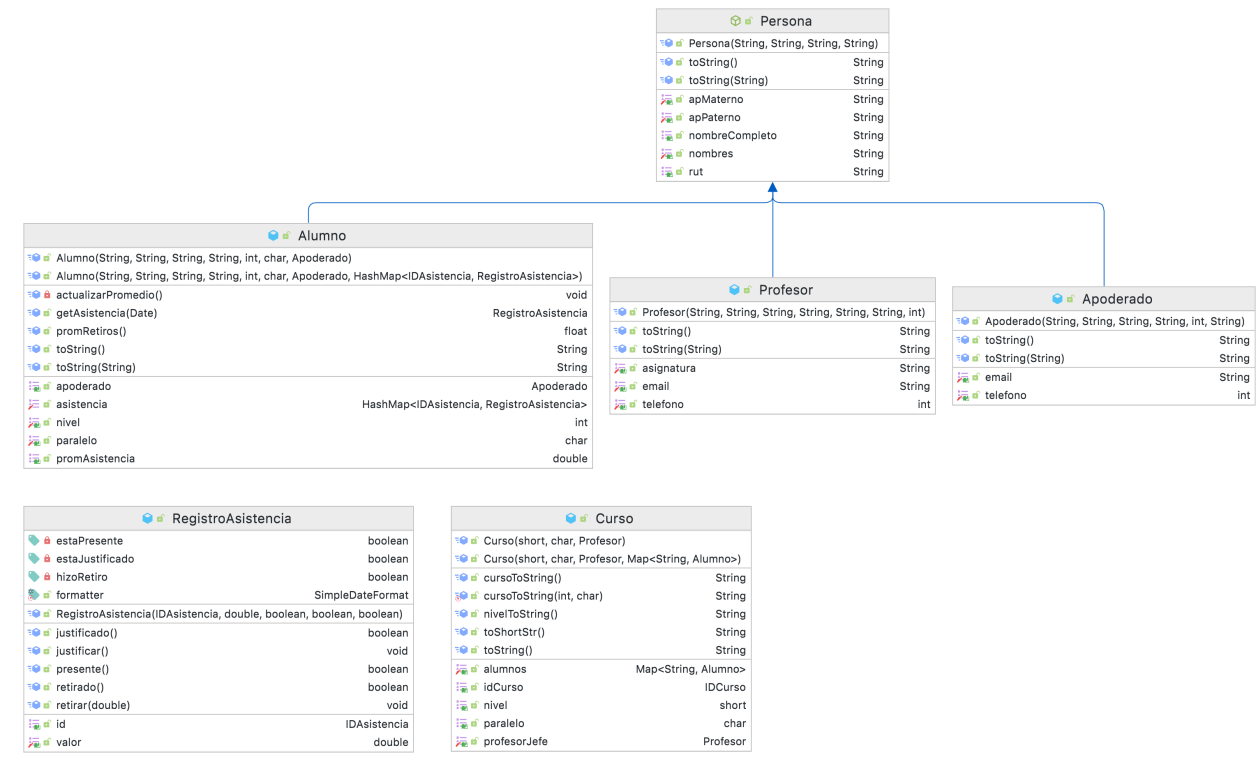


Figura 1: Clases del dominio

La clase **Persona** contendrá los datos genéricos de una persona, y será la clase padre de **Apoderado**, **Alumno** y **Profesor**. Además está la clase **Curso** que posee el nivel, valor numérico de 1 a 12, una letra identificadora de paralelo, un profesor jefe, y un Hashmap de alumnos, cuya llave será el RUT del alumno correspondiente.

En el caso de **Apoderado**, este hereda de **Persona** y además tiene los datos de contacto del apoderado de cada alumno. **Alumno** contiene a su apoderado, y lleva su registro de asistencia correspondiente al año escolar.

Para el manejo de los datos, se utilizarán **interfaces** que nos permitan estandarizar la forma de obtener y manejar los datos, para poder incorporar en el futuro, de una forma más simple, manejo de bases de datos.

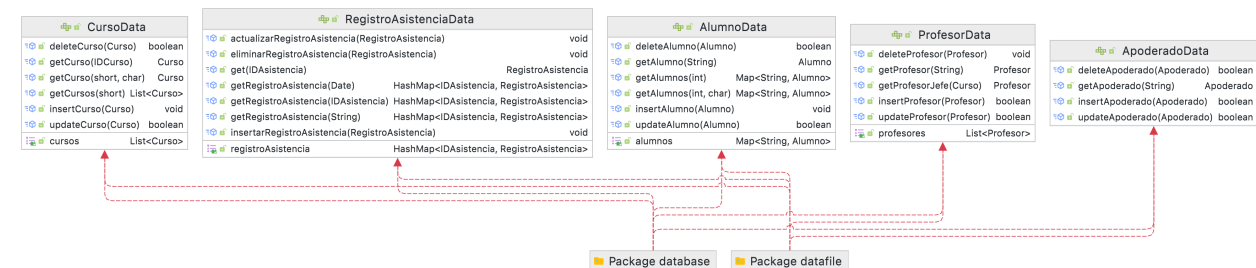


Figura 2: Clases para manejo de datos

Para el manejo de los archivos de texto en donde se almacenan los datos, se utilizará una clase dedicada a trabajar con ellos. Esta clase contendrá los métodos que posibilitan la inserción, listado, eliminación y actualización de los datos, entregados como texto CSV, separado por punto y coma (;). Para llevar los datos al formato CSV, la clase contiene un método estático que permite transformar una lista de String al texto CSV.

RegistroAsistenciaDF		
RegistroAsistenciaDF()		
actualizarRegistroAsistencia(RegistroAsistencia)		void
eliminarRegistroAsistencia(RegistroAsistencia)		void
get(IDAsistencia)		RegistroAsistencia
getRegistroAsistencia(Date)	HashMap<IDAsistencia, RegistroAsistencia>	
getRegistroAsistencia(IDAsistencia)	HashMap<IDAsistencia, RegistroAsistencia>	
getRegistroAsistencia(String)	HashMap<IDAsistencia, RegistroAsistencia>	
insertarRegistroAsistencia(RegistroAsistencia)		void
registroAsistenciaToCSV(RegistroAsistencia)		String
registroAsistenciaFromCSV(String)		RegistroAsistencia
registroAsistencia	HashMap<IDAsistencia, RegistroAsistencia>	

AlumnoDF		
AlumnoDF()		
alumnoFromCSV(String)	Alumno	
alumnoToCSV(Alumno)	String	
deleteAlumno(Alumno)	boolean	
getAlumno(String)	Alumno	
getAlumnos(int)	Map<String, Alumno>	
getAlumnos(int, char)	Map<String, Alumno>	
insertAlumno(Alumno)	void	
updateAlumno(Alumno)	boolean	
alumnos	Map<String, Alumno>	

CursoDF		
CursoDF()		
cursoFromCSV(String)	Curso	
cursoToCSV(Curso)	String	
deleteCurso(Curso)	boolean	
getCurso(IDCurso)	Curso	
getCurso(short, char)	Curso	
getCursos(short)	List<Curso>	
insertCurso(Curso)	void	
updateCurso(Curso)	boolean	
cursos	List<Curso>	

Datafile		
Datafile(String)		
clearFile()	void	
deleteLine(String)	boolean	
insertLine(String)	boolean	
listToCSV(List<String>)	String	
updateLine(String, String)	boolean	
data	List<String>	
fileExists	boolean	
filePath	String	

ProfesorDF		
ProfesorDF()		
deleteProfesor(Profesor)	void	
getProfesor(String)	Profesor	
getProfesorJefe(Curso)	Profesor	
insertProfesor(Profesor)	boolean	
profesorFromCSV(String)	Profesor	
profesorToCSV(Profesor)	String	
updateProfesor(Profesor)	boolean	
profesores	List<Profesor>	

ApoderadoDF		
ApoderadoDF()		
apoderadoFromCSV(String)	Apoderado	
apoderadoToCSV(Apoderado)	String	
deleteApoderado(Apoderado)	boolean	
getApoderado(String)	Apoderado	
insertApoderado(Apoderado)	boolean	
updateApoderado(Apoderado)	boolean	

Figura 3: Paquete Datafile

1.3. Todos los atributos de todas las clases deben ser privados y poseer sus respectivos métodos de lectura y escritura (getter y setter)

Bla bla Bla...

1.4. Se deben incluir datos iniciales dentro del código

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec et sem luctus, finibus mauris eget, euismod arcu. Morbi at mollis risus. Praesent consequat justo tellus, ac pretium urna lacinia quis. Ut sagittis cursus finibus. Morbi pellentesque vulputate tincidunt. Aliquam at pretium tellus, et vehicula velit. Nunc turpis metus, porttitor sit amet aliquam ut, venenatis quis elit. Nam tincidunt venenatis tortor, ac sodales libero varius nec. Donec vestibulum leo a metus rutrum, vitae elementum diam suscipit. Quisque non mauris rutrum, lacinia lectus id, viverra dui. In leo quam, ultrices vitae suscipit quis, porttitor quis turpis. Vestibulum semper ligula sit amet diam scelerisque gravida.

Opcional: Leer datos desde archivo/base de datos

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec et sem luctus, finibus mauris eget, euismod arcu. Morbi at mollis risus. Praesent consequat justo tellus, ac pretium urna lacinia quis. Ut sagittis cursus finibus. Morbi pellentesque

vulputate tincidunt. Aliquam at pretium tellus, et vehicula velit. Nunc turpis metus, porttitor sit amet aliquam ut, venenatis quis elit. Nam tincidunt venenatis tortor, ac sodales libero varius nec. Donec vestibulum leo a metus rutrum, vitae elementum diam suscipit. Quisque non mauris rutrum, lacinia lectus id, viverra dui. In leo quam, ultrices vitae suscipit quis, porttitor quis turpis. Vestibulum semper ligula sit amet diam scelerisque gravida.

2. Entrega parcial 2

2.1. Diseño conceptual y codificación de 2 (dos) niveles de anidación de colecciones de objetos

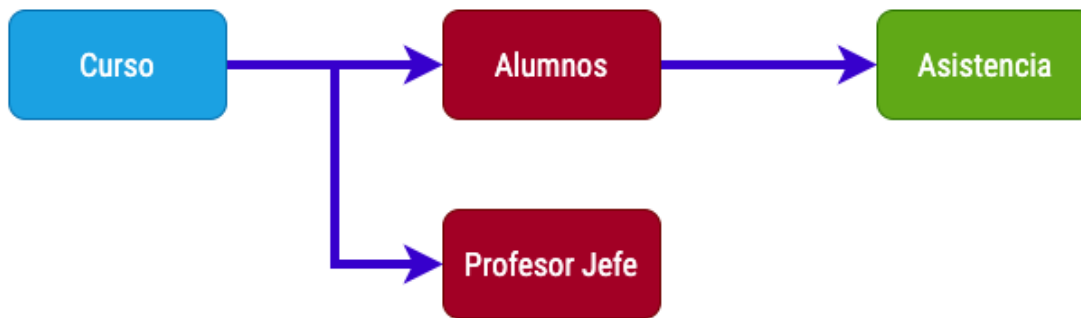


Figura 4: Anidación de colecciones de objetos

Como podemos ver en la imagen, el proyecto está planificado para contener la colección de cursos, la que a su vez contiene un **HashMap** de *Alumnos* y el *Profesor jefe* a cargo. En cuanto a la asistencia, cada *Alumno* contendrá un **HashMap** con su asistencia personal.

2.2. Diseño conceptual y codificación de 2 (dos) clases que utilicen sobrecarga de métodos

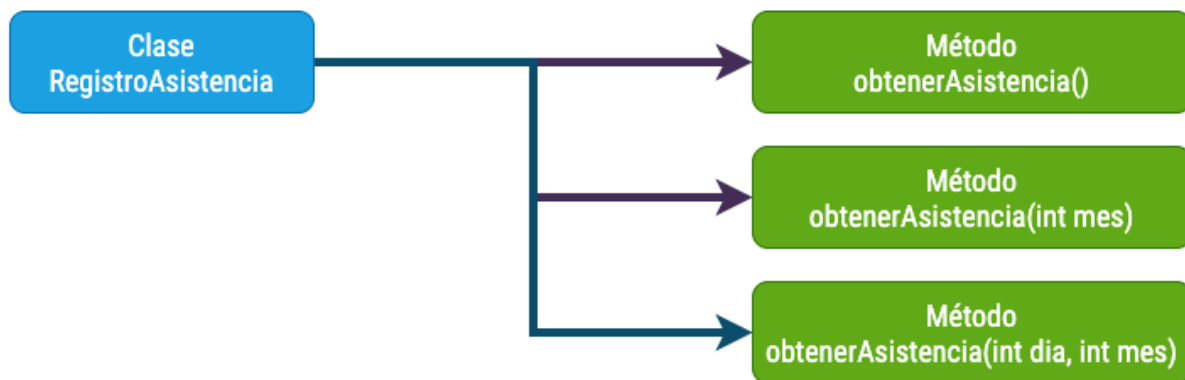


Figura 5: Sobrecarga de métodos: Clase RegistroAsistencia

Los métodos `obtenerAsistencia` de la clase **RegistroAsistencia** obtienen los datos de asistencia del alumno con los siguientes retornos:

- `public float obtenerAsistencia()`: Retorna el promedio de asistencia del alumno, calculado al día de realizada la consulta.
- `public float obtenerAsistencia(int mes)`: Retorna el promedio de asistencia del alumno, del mes especificado como parámetro.
- `public float obtenerAsistencia(int dia, int mes)`: Retorna el valor de asistencia del alumno en el día y mes especificados como parámetros.

Los métodos `getAlumnos` de la interfaz **AlumnoData** obtienen los datos de los alumnos con los siguientes retornos:

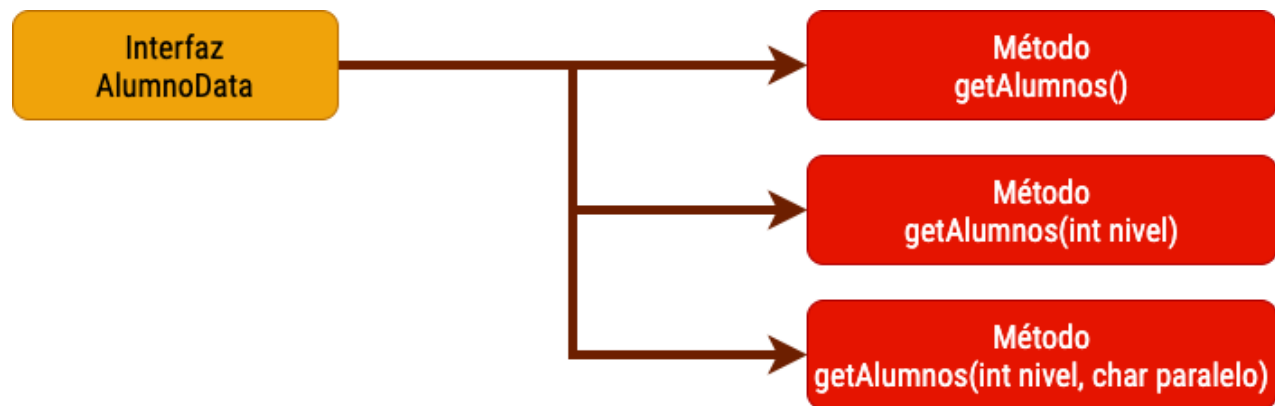


Figura 6: Sobrecarga de métodos: Interfaz AlumnoData

- `public Map<String, Alumno> getAlumnos():` Retorna un HashMap con todos los alumnos almacenados.
- `public Map<String, Alumno> getAlumnos(int nivel):` Retorna un HashMap con todos los alumnos del nivel especificado como parámetro.
- `public Map<String, Alumno> getAlumnos(int nivel, char paralelo):` Retorna un HashMap con todos los alumnos del curso especificado (nivel y paralelo).

2.3. Diseño conceptual y codificación de al menos 1 clase mapa del JCF

Cada **Curso** debe manejar un número indeterminado de **Alumnos**, y su recorrido debe ser lo más eficiente posible a la hora de buscar un alumno específico, dado que el programa debe manejar los registros de asistencia, que están anidados en cada alumno, y a su vez los alumnos están anidados en cada curso. Es por esto que se decidió por almacenar los alumnos en cada curso como **HashMap**, en donde la clave será el RUT del alumno, y el valor contendrá al alumno en sí.

3. Entrega parcial 3

3.1. Diseño de diagrama de clases UML de lo considerado hasta la EP3

El diagrama UML completo está disponible en el repositorio de GitHub, y será adjuntado a la entrega final del aula virtual de la asignatura. No obstante, se detallan los paquetes generados para el funcionamiento de la aplicación:

Package controllers

CursoControllerCLI	
CursoControllerCLI(MenuCLI)	
agregarCurso()	void
asignarNuevoProfesorJefeACurso()	void
eliminarCurso()	void
generarReporte(String, String)	Datafile
generarReporteListaCurso(short, char)	String
generarReporteTablaCursos()	String
mostrarCursos()	void
obtenerDatosCurso()	Curso
obtenerDatosCurso(Curso)	Curso
obtenerIDCurso()	IDCurso
verDatosCurso()	void

MenuControllerCLI	
MenuControllerCLI(MenuCLI)	
menu(String)	void
menuAdministracion(short)	void
menuAlumnos(short)	void
menuAsistencia(short)	void
menuCursos(short)	void
menuOpc(String, short)	void
menuPrincipal(short)	void
menuReportes(short)	void

UtilsGUI	
UtilsGUI()	
errorMsg(String)	void
getCursoFromCB(JComboBox, List<Curso>)	Curso
getCursosCB(List<Curso>)	DefaultComboBoxModel
getIcon(IconCode, float, Color)	Icon
infoMsg(String)	void
msjSiNo(String)	int
actualDate	String

AlumnoControllerCLI	
AlumnoControllerCLI(MenuCLI)	
agregarAlumno(CursoControllerCLI)	void
editarAlumno()	void
eliminarAlumno()	void
obtenerDatosAlumno(Alumno, int, char)	Alumno
obtenerDatosAlumno(int, char)	Alumno
verAlumno()	void
verAlumnos(CursoControllerCLI)	void

AsistenciaControllerCLI	
AsistenciaControllerCLI(MenuCLI)	
alumnoConMasRetiros()	void
alumnoEntrePorcentajesRetiros()	void
alumnoMejorAsistencia()	void
alumnosEntrePorcentajesAsistencia()	void
getAlumnoRequerido(int)	Alumno
getEntrePorcentajes(int, int, int)	Map<String, Alumno>








CustomColors	
CustomColors()	
getColor(boolean)	Color
valueOf(String)	CustomColors
values()	CustomColors[]
color	Color











Panels	
Panels()	
valueOf(String)	Panels
values()	Panels[]
panel	JPanel











ReportesControllerCLI	
ReportesControllerCLI(MenuCLI)	
generarListaCurso(CursoControllerCLI)	void
generarTablaCursos(CursoControllerCLI)	void










ReportesControllerGUI	
ReportesControllerGUI()	
generarReporte(String, String)	Datafile
generarReporteListaCurso(Curso)	String










Package data.datafile








RegistroAsistenciaDF		
 RegistroAsistenciaDF()		
 actualizarRegistroAsistencia(RegistroAsistencia)		void
 eliminarRegistroAsistencia(RegistroAsistencia)		void
 get(IDAsistencia)		RegistroAsistencia
 getRegistroAsistencia(Date)	HashMap<IDAsistencia, RegistroAsistencia>	
 getRegistroAsistencia(IDAsistencia)	HashMap<IDAsistencia, RegistroAsistencia>	
 getRegistroAsistencia(String)	HashMap<IDAsistencia, RegistroAsistencia>	
 insertarRegistroAsistencia(RegistroAsistencia)		void
 registroAsistenciaToCSV(RegistroAsistencia)		String
 registroAsistenciaFromCSV(String)		RegistroAsistencia
 registroAsistencia	HashMap<IDAsistencia, RegistroAsistencia>	

AlumnoDF		
 AlumnoDF()		
 alumnoFromCSV(String)	Alumno	
 alumnoToCSV(Alumno)	String	
 deleteAlumno(Alumno)	boolean	
 getAlumno(String)	Alumno	
 getAlumnos(int)	Map<String, Alumno>	
 getAlumnos(int, char)	Map<String, Alumno>	
 insertAlumno(Alumno)	void	
 updateAlumno(Alumno)	boolean	
 alumnos	Map<String, Alumno>	

CursoDF		
 CursoDF()		
 cursoFromCSV(String)	Curso	
 cursoToCSV(Curso)	String	
 deleteCurso(Curso)	boolean	
 getCurso(IDCurso)	Curso	
 getCurso(short, char)	Curso	
 getCursos(short)	List<Curso>	
 insertCurso(Curso)	void	
 updateCurso(Curso)	boolean	
 cursos	List<Curso>	

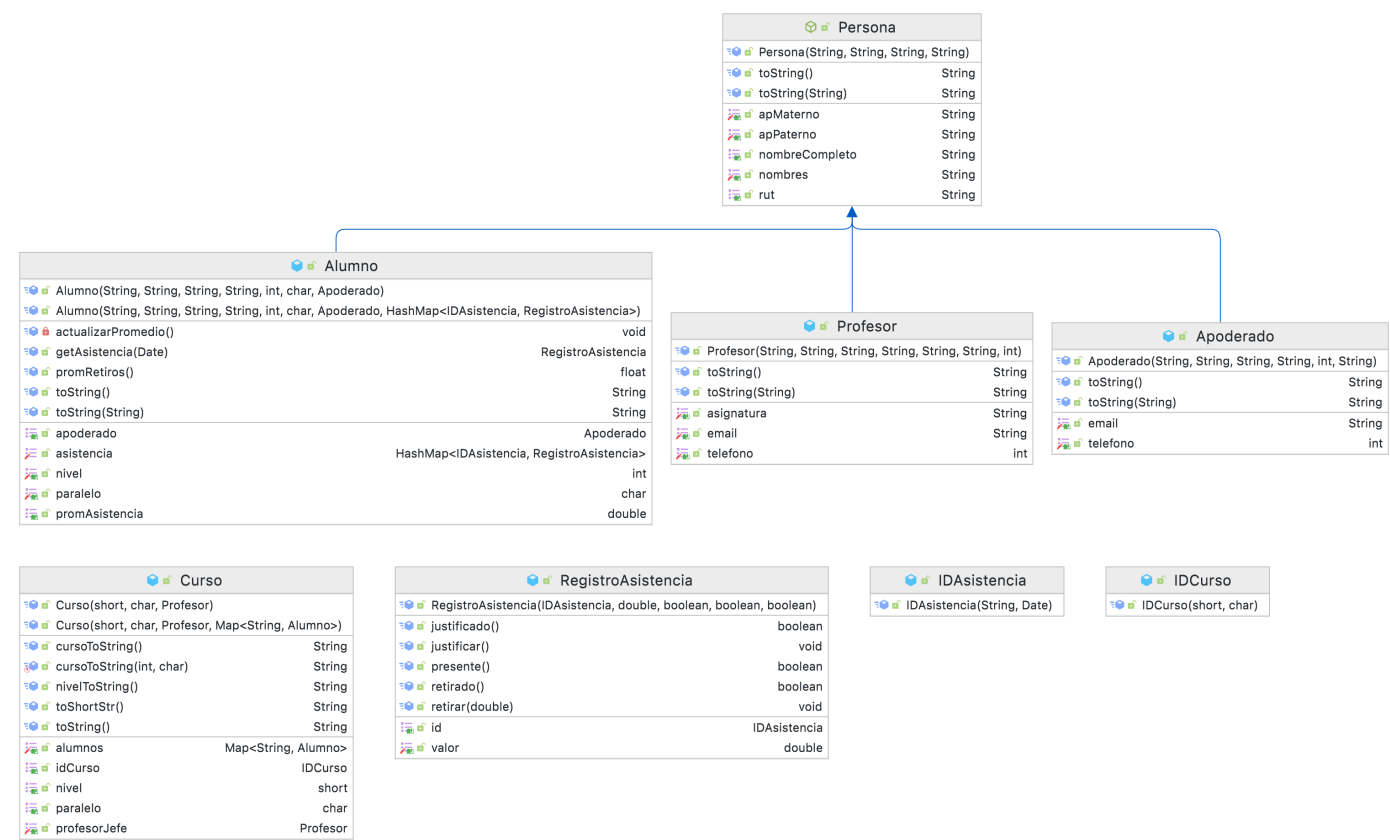
Datafile		
 Datafile(String)		
 clearFile()		void
 deleteLine(String)		boolean
 insertLine(String)		boolean
 listToCSV(List<String>)		String
 updateLine(String, String)		boolean
 data	List<String>	
 fileExists	boolean	
 filePath	String	

ProfesorDF		
 ProfesorDF()		
 deleteProfesor(Profesor)		void
 getProfesor(String)	Profesor	
 getProfesorJefe(Curso)	Profesor	
 insertProfesor(Profesor)		boolean
 profesorFromCSV(String)	Profesor	
 profesorToCSV(Profesor)	String	
 updateProfesor(Profesor)		boolean
 profesores	List<Profesor>	

ApoderadoDF		
 ApoderadoDF()		
 apoderadoFromCSV(String)	Apoderado	
 apoderadoToCSV(Apoderado)	String	
 deleteApoderado(Apoderado)		boolean
 getApoderado(String)	Apoderado	
 insertApoderado(Apoderado)		boolean
 updateApoderado(Apoderado)		boolean

Generated by JFDG

Package models



Package views.cli

UtilsCLI	
headers	Map<String, String[]>
UtilsCLI()	
anchoColumnas(Object[][], String[])	int[]
anchoTotalTabla(int[])	int
imprimirCaracter(char, int)	void
imprimirCurso(Curso)	void
imprimirEsperaEnter()	void
imprimirFilaTabla(Object[], int[])	void
imprimirFuncionalidadNoImplementada()	void
imprimirIngresarOpcion(String)	void
imprimirIngresarOpcion(String, String)	void
imprimirMenu(Object[][], String)	void
imprimirPersona(Persona)	void
imprimirSolicitar(String, String)	void
imprimirTabla(Object[][], String[], String)	void
imprimirTitulo(String)	void
imprimirTitulo(String, int)	void
mensajeBienvenida()	void
mensajeDespedida()	void
mensajeErrIngresado()	void
mensajeErrorNoEncontrado(String, int)	void
mensajeExitoConexionMySQL()	void
mensajeIntentandoConexionMySQL()	void
mensajeUtilizandoDatafile()	void

AlumnoViewCLI	
AlumnoViewCLI()	
mostrarTablaAlumnos(Map<String, Alumno>, String)	void

MenuCLI	
menuControllerCLI	MenuControllerCLI
opciones	Map<String, Object[]>
MenuCLI(BufferedReader, AlumnoData, ApoderadoData, CursoData, ProfesorData)	
mostrarMenu(String)	void
mostrarMenu(String, String)	void
alumnoData	AlumnoData
apoderadoData	ApoderadoData
cursoData	CursoData
lector	BufferedReader
profesorData	ProfesorData

AsistenciaViewCLI	
AsistenciaViewCli()	
mostrarTablaAlumnosAsistencia(Map<String, Alumno>, String)	void

CLI	
menuCLI	MenuCLI
CLI()	
finalizarCLI()	void
iniciarCLI()	void
main(String[])	void

CursoViewCLI	
CursoViewCLI()	
mostrarTablaCursos(List<Curso>)	void

3.2. Menú del sistema

La interfaz de línea de comandos de la aplicación (CLI) se ejecuta a partir del método main contenido en la clase CLI, dentro del package views.cli. Cuenta con un menú principal, que es el primero que se muestra por pantalla, y permite la interacción con el usuario a través del ingreso de opciones numéricas.

>_ Consola: Menú principal del sistema

```
Menú principal
-> ( 1 ) - Administrar colegio
    Permite gestionar los cursos, alumnos, profesores y asistencia
-> ( 2 ) - Generar reportes
    Permite generar reportes con los datos del sistema
-> ( 3 ) - Gráficos (próximamente)
    Permite obtener gráficos sobre los datos del sistema
-> ( 0 ) - Salir
    Salir de la aplicación
Ingrese su opción (numérica):
```

La opción 1 nos muestra el menú de administración del colegio, en donde se permite gestionar los datos con los que trabaja la aplicación.

>_ Consola: Menú administración

```
Menú administracion
-> ( 1 ) - Gestionar cursos
    Permite gestionar los datos de los cursos
-> ( 2 ) - Gestionar alumnos
    Permite gestionar los datos de los alumnos
-> ( 3 ) - Gestionar asistencia
    Permite gestionar la asistencia de los alumnos
-> ( 9 ) - Salir
    Salir de la aplicación
-> ( 0 ) - Volver al menú principal
    Salir de la administración del colegio
Ingrese su opción (numérica):
```

Cada menú permite gestionar los datos de cada uno de los entes involucrados. En este caso, se mostrará el menú de gestión de cursos:

>_ Consola: Menú gestión de cursos

```
Menú cursos
-> ( 1 ) - Agregar curso
    Permite agregar un curso al sistema
-> ( 2 ) - Ver datos de curso
    Permite ver los datos de un curso en específico
-> ( 3 ) - Ver cursos
    Permite ver la lista de cursos disponibles
-> ( 4 ) - Asignar nuevo profesor jefe a curso
    Permite asignar un nuevo profesor jefe al curso
-> ( 5 ) - Eliminar curso
    Permite eliminar un curso en específico
-> ( 9 ) - Salir
    Salir de la aplicación
-> ( 0 ) - Volver al menú de administración
    Salir de la gestión de cursos
Ingrese su opción (numérica):
```

3.2.1. Inserción manual/agregar elemento

Para agregar elementos al programa, se accede a través del menú en específico del tipo de dato que queremos almacenar. En este caso de ejemplo, para agregar un curso (según el menú mostrado en la parte superior) se realiza con la opción numérica 1. Al presionar el 1, el sistema solicita los datos necesarios para ser agregados:

>_ Consola: Funcionalidad: Agregar curso

```
Ingrese el nivel del curso(número de 1 a 12): 5
Ingrese el paralelo del curso(caracter): C
Ingrese el RUT del profesor jefe(RUT): 12345678-9
Ingrese los nombres del profesor jefe(texto): Profesor Jefe
Ingrese el apellido paterno del profesor jefe(texto): Ejemplo
Ingrese el apellido materno del profesor jefe(texto): Ejemplo
Ingrese la asignatura del profesor jefe(texto): Asignatura de ejemplo
Ingrese el email del profesor jefe(texto): ejemplo@ejemplo.com
Ingrese el teléfono del profesor jefe(texto): 912345678
```

Luego de agregados los datos, se genera un nuevo objeto de tipo Curso y se agrega al archivo de texto plano (a través de los métodos de la clase CursoDatafile, del package data.datafile). Una vez insertados los datos nuevos, el programa muestra la tabla con todos los cursos (y el nuevo curso también presente).

Para el caso de los Alumnos, también se agregan a través de su menú de gestión correspondiente, en donde se solicitarán los datos requeridos:

>_ Consola: Funcionalidad: Agregar alumno

```
Ingrese el nivel del curso(número de 1 a 12): 12
Ingrese el paralelo del curso(caracter): B
Ingrese el RUT del alumno(RUT): 12345678-9
Ingrese los nombres del alumno(texto): Alumno De
Ingrese el apellido paterno del alumno(texto): Ejemplo
Ingrese el apellido materno del alumno(texto): Ejemplo
Ingrese el RUT del apoderado(RUT): 91234567-8
Ingrese los nombres del apoderado(texto): Apoderado De
Ingrese el apellido paterno del apoderado(texto): Ejemplo
Ingrese el apellido materno del apoderado(texto): Ejemplo
Ingrese el email del apoderado(texto): apoderado@ejemplo.com
Ingrese el teléfono del apoderado(texto): 987654321
```

Al igual que con los Cursos, luego de ser agregado el nuevo alumno, se muestra una tabla con todos los alumnos del curso al que fue incorporado, incluyendo el dato nuevo.

3.2.2. Mostrar por pantalla listado de elementos. Esto para cada una de las 2 colecciones anidadas

Al igual que con la funcionalidad de agregar elementos, en este caso para listar los elementos, se debe acceder desde el menú que gestiona el tipo de elemento que deseamos recuperar.

Para el caso de los cursos, son listados de la siguiente manera:

>_ Consola: Funcionalidad: Mostrar cursos

Cursos						
Curso	Nivel	Paralelo	Nombre profesor jefe	Email profesor jefe	Telefono profesor jefe	
1BA	1	A	Shanta Hilll Senger	sue.casper@hotmail.com	987659326	
1BB	1	B	Elden Kautzer Abshire	maxima.tremblay@hotmail.com	978042546	
2BA	2	A	Grant Cartwright Ferry	jaymie.anderson@gmail.com	947869804	
2BB	2	B	Claudette Baumbach Roob	lasonya.swaniawski@yahoo.com	954534919	
3BA	3	A	Alex Parisian Hintz	darell.lueilwitz@hotmail.com	952017013	
3BB	3	B	Darrel Nicolas Cole	milagros.emmerich@hotmail.com	948755395	
4BA	4	A	Tamisha Volkman Fadel	joaquin.stiedemann@gmail.com	985662322	
4BB	4	B	Jacques Wolff Bednar	donella.lockman@gmail.com	984890419	
5BA	5	A	Marylee Donnelly Bernier	evelin.abshire@gmail.com	990485383	
5BB	5	B	Inge Fay McCullough	sharda.fahay@yahoo.com	991602692	
6BA	6	A	Bruce Kulas Goodwin	george.champlin@gmail.com	975853419	
6BB	6	B	Marcelle Emmerich Williamson	nickolas.pouros@gmail.com	961721076	
7BA	7	A	Kory O'Kon Brakus	thanh.waters@hotmail.com	974036963	
7BB	7	B	Elease White Jerde	elli.mayer@gmail.com	988854022	
8BA	8	A	Kandace Hahn Hammes	edmond.corwin@yahoo.com	950970290	
8BB	8	B	Loyce Krajcik Jakubowski	jenine.keebler@hotmail.com	943740407	
1MA	9	A	Teodoro Pfeffer Ankunding	lottie.weissnat@gmail.com	993288595	
1MB	9	B	Denita Kuvalis Heller	shila.morissette@yahoo.com	969915007	
2MA	10	A	Raphael Walter Stoltenberg	hyacinth.senger@gmail.com	984581430	
2MB	10	B	Elton Beatty McLaughlin	joya.barrows@gmail.com	941208950	
3MA	11	A	Lacy Cassin Bauch	harlan.mitchell@hotmail.com	975709937	
3MB	11	B	Dede Goodwin Schuppe	ryan.hand@gmail.com	966145453	
4MA	12	A	Noelia Hagenes Swift	renate.swaniawski@hotmail.com	941278587	
4MB	12	B	Leif Zulauf Boyle	celina.keeling@yahoo.com	973953363	

Y en el caso de los alumnos, se lista de la siguiente manera:

>_ Consola: Funcionalidad: Mostrar alumnos

Alumnos del 1 medio A					
RUT	Ap. Paterno	Ap. Materno	Nombres	Nombre completo apoderado	Teléfono apoderado
45720666-8	Greenholt	Becker	Jamison	Dante Greenholt Beier	985207105
46254622-3	Cassin	Beahan	Byron	Solomon Cassin Goldner	971498469
43100387-7	Wilkinson	Fay	Delmer	Elvin Wilkinson Bechtelar	973228091
37002298-3	Murray	Cummings	Antony	Felica Murray Lang	946952504
47438615-1	White	Mayer	Katharyn	Katherine White Moore	950651632
45933943-8	Emard	Sauer	Deon	Doretta Emard Mertz	982425038
34581069-3	Reilly	Medhurst	Kasey	Melodee Reilly Walter	947910091
34665351-7	Carroll	Jenkins	Jesica	Weldon Carroll Wisoky	949098199
38226629-8	Legros	Beier	Kristofer	Damion Legros Hegmann	985372752
47026929-7	Huels	Schuppe	Raisa	Elvera Huels Reilly	989935616
38273902-5	Quigley	Conn	Delmy	Saul Quigley Paucek	979615506
31365566-2	D'Amore	Haag	Exie	Darwin D'Amore Schumm	950907383
49860824-4	Torp	Morissette	Michael	Prince Torp Schneider	946302232
49642433-6	Kris	Johns	Ehtel	Christin Kris Zemlak	996654940
37953682-2	MacGyver	Leuschke	Candance	Dane MacGyver Hettinger	997309006

A. Entrega acumulada A

A.2. Menú con funcionalidades de edición y eliminación de elementos

El menú generado también nos permite editar y eliminar datos relacionados a los Cursos y los Alumnos. Específicamente, se puede:

- Editar o eliminar alumnos: Se pueden modificar los datos del alumno (exceptuando el RUT). No así los datos del apoderado, ya que es una funcionalidad que estará disponible en las entregas siguientes. También se pueden eliminar los alumnos, lo que provoca a su vez que se elimine del sistema al apoderado que tenga asociado.
- Asignar nuevo profesor jefe a curso: Se puede modificar un curso, asignándole un nuevo profesor jefe y eliminando al anterior.
- Eliminar curso: El programa permite la eliminación completa de un curso, la que provocará también la eliminación de los datos del profesor, alumnos, y por consiguiente, de los apoderados.

A.3. Se debe generar un reporte en archivo txt que considere mostrar datos de las 2 colecciones anidadas (ej: csv)

El programa permite la generación de reportes en formato CSV (separados por comas). Los reportes que se pueden generar son los siguientes:

- Lista del curso: Este archivo CSV contendrá los datos de los alumnos de un curso en específico, entregado por el usuario.
- Tabla de cursos: Este archivo CSV contendrá los datos generales de todos los cursos a los que el programa tiene acceso. Esta tabla, además de mostrar los datos del curso mismo, muestra datos generales del profesor jefe a cargo del mismo.

A.4. El código fuente debe estar bien modularizado de acuerdo a lo descrito en el informe además de seguir las buenas prácticas de documentación interna y legibilidad

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec et sem luctus, finibus mauris eget, euismod arcu. Morbi at mollis risus. Praesent consequat justo tellus, ac pretium urna lacinia quis. Ut sagittis cursus finibus. Morbi pellentesque vulputate tincidunt. Aliquam at pretium tellus, et vehicula velit. Nunc turpis metus, porttitor sit amet aliquam ut, venenatis quis elit. Nam tincidunt venenatis tortor, ac sodales libero varius nec. Donec vestibulum leo a metus rutrum, vitae elementum diam suscipit. Quisque non mauris rutrum, lacinia lectus id, viverra dui. In leo quam, ultrices vitae suscipit quis, porttitor quis turpis. Vestibulum semper ligula sit amet diam scelerisque gravida.

A.5. Todas las funcionalidades pueden ser implementadas mediante consola

Opcional: La implementación de una o más interfaces gráficas de usuario, con la inclusión de ventanas con componentes AWT, SWING, JAVAFXu otro

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec et sem luctus, finibus mauris eget, euismod arcu. Morbi at mollis risus. Praesent consequat justo tellus, ac pretium urna lacinia quis. Ut sagittis cursus finibus. Morbi pellentesque vulputate tincidunt. Aliquam at pretium tellus, et vehicula velit. Nunc turpis metus, porttitor sit amet aliquam ut, venenatis quis elit. Nam tincidunt venenatis tortor, ac sodales libero varius nec. Donec vestibulum leo a metus rutrum, vitae elementum diam suscipit. Quisque non mauris rutrum, lacinia lectus id, viverra dui. In leo quam, ultrices vitae suscipit quis, porttitor quis turpis. Vestibulum semper ligula sit amet diam scelerisque gravida.

A.6. Utilización de GitHub (Realización de al menos 3 Commit)

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec et sem luctus, finibus mauris eget, euismod arcu. Morbi at mollis risus. Praesent consequat justo tellus, ac pretium urna lacinia quis. Ut sagittis cursus finibus. Morbi pellentesque vulputate tincidunt. Aliquam at pretium tellus, et vehicula velit. Nunc turpis metus, porttitor sit amet aliquam ut, venenatis quis elit. Nam tincidunt venenatis tortor, ac sodales libero varius nec. Donec vestibulum leo a metus rutrum, vitae elementum diam suscipit. Quisque non mauris rutrum, lacinia lectus id, viverra dui. In leo quam, ultrices vitae suscipit quis, porttitor quis turpis. Vestibulum semper ligula sit amet diam scelerisque gravida.

4. Entrega parcial 4

4.1. Se deben incluir al menos 2 funcionalidades propias que sean de utilidad para el negocio (distintas de la inserción, edición, eliminación y reportes)

4.1.1. Seleccionar un objeto por criterio, considera la selección de un objeto basado en un criterio específico, involucrando dos o más colecciones anidadas. Por ejemplo, selección del alumno con la nota final más baja de todos los cursos, o seleccionar el pasajero más joven de todos los buses de una compañía

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec et sem luctus, finibus mauris eget, euismod arcu. Morbi at mollis risus. Praesent consequat justo tellus, ac pretium urna lacinia quis. Ut sagittis cursus finibus. Morbi pellentesque vulputate tincidunt. Aliquam at pretium tellus, et vehicula velit. Nunc turpis metus, porttitor sit amet aliquam ut, venenatis quis elit. Nam tincidunt venenatis tortor, ac sodales libero varius nec. Donec vestibulum leo a metus rutrum, vitae elementum diam suscipit. Quisque non mauris rutrum, lacinia lectus id, viverra dui. In leo quam, ultrices vitae suscipit quis, porttitor quis turpis. Vestibulum semper ligula sit amet diam scelerisque gravida.

4.1.2. Subconjunto filtrado por criterio: considera la selección de un subconjunto de objetos basado en un criterio específico, involucrando dos o más colecciones anidadas. Por ejemplo, selección de los alumnos con nota final entre 4,0 y 7,0 de entre todos los cursos; o seleccionar a todos los pasajeros que tengan asiento impar de entre todos los buses de la compañía

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec et sem luctus, finibus mauris eget, euismod arcu. Morbi at mollis risus. Praesent consequat justo tellus, ac pretium urna lacinia quis. Ut sagittis cursus finibus. Morbi pellentesque vulputate tincidunt. Aliquam at pretium tellus, et vehicula velit. Nunc turpis metus, porttitor sit amet aliquam ut, venenatis quis elit. Nam tincidunt venenatis tortor, ac sodales libero varius nec. Donec vestibulum leo a metus rutrum, vitae elementum diam suscipit. Quisque non mauris rutrum, lacinia lectus id, viverra dui. In leo quam, ultrices vitae suscipit quis, porttitor quis turpis. Vestibulum semper ligula sit amet diam scelerisque gravida.

4.2. Diseño y codificación de 2 (dos) clases que utilicen sobreescritura de métodos

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec et sem luctus, finibus mauris eget, euismod arcu. Morbi at mollis risus. Praesent consequat justo tellus, ac pretium urna lacinia quis. Ut sagittis cursus finibus. Morbi pellentesque vulputate tincidunt. Aliquam at pretium tellus, et vehicula velit. Nunc turpis metus, porttitor sit amet aliquam ut, venenatis quis elit. Nam tincidunt venenatis tortor, ac sodales libero varius nec. Donec vestibulum leo a metus rutrum, vitae elementum diam suscipit. Quisque non mauris rutrum, lacinia lectus id, viverra dui. In leo quam, ultrices vitae suscipit quis, porttitor quis turpis. Vestibulum semper ligula sit amet diam scelerisque gravida.

4.3. Diseño y codificación de 1 (una) clase abstracta que sea padre de al menos 2 (dos) clases. La clase abstracta debe ser utilizada por alguna otra clase (contexto)

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec et sem luctus, finibus mauris eget, euismod arcu. Morbi at mollis risus. Praesent consequat justo tellus, ac pretium urna lacinia quis. Ut sagittis cursus finibus. Morbi pellentesque vulputate tincidunt. Aliquam at pretium tellus, et vehicula velit. Nunc turpis metus, porttitor sit amet aliquam ut, venenatis quis elit. Nam tincidunt venenatis tortor, ac sodales libero varius nec. Donec vestibulum leo a metus rutrum, vitae elementum diam suscipit. Quisque non mauris rutrum, lacinia lectus id, viverra dui. In leo quam, ultrices vitae suscipit quis, porttitor quis turpis. Vestibulum semper ligula sit amet diam scelerisque gravida.

4.4. Diseño y codificación de 1 (una) interfaz que sea implementada por al menos 2 (dos) clases. La interfaz debe ser utilizada por alguna otra clase (contexto)

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec et sem luctus, finibus mauris eget, euismod arcu. Morbi at mollis risus. Praesent consequat justo tellus, ac pretium urna lacinia quis. Ut sagittis cursus finibus. Morbi pellentesque vulputate tincidunt. Aliquam at pretium tellus, et vehicula velit. Nunc turpis metus, porttitor sit amet aliquam ut, venenatis quis elit. Nam tincidunt venenatis tortor, ac sodales libero varius nec. Donec vestibulum leo a metus rutrum, vitae elementum diam suscipit. Quisque non mauris rutrum, lacinia lectus id, viverra dui. In leo quam, ultrices vitae suscipit quis, porttitor quis turpis. Vestibulum semper ligula sit amet diam scelerisque gravida.

Opcional: Generar documentación con Javadoc.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec et sem luctus, finibus mauris eget, euismod arcu. Morbi at mollis risus. Praesent consequat justo tellus, ac pretium urna lacinia quis. Ut sagittis cursus finibus. Morbi pellentesque vulputate tincidunt. Aliquam at pretium tellus, et vehicula velit. Nunc turpis metus, porttitor sit amet aliquam ut, venenatis quis elit. Nam tincidunt venenatis tortor, ac sodales libero varius nec. Donec vestibulum leo a metus rutrum, vitae elementum diam suscipit. Quisque non mauris rutrum, lacinia lectus id, viverra dui. In leo quam, ultrices vitae suscipit quis, porttitor quis turpis. Vestibulum semper ligula sit amet diam scelerisque gravida.

B. Entrega acumulada B

B.2. Se deben implementar al menos 3 ventanas gráficas (GUIs en AWT o SWING): 1 ventana de menú, 1 ventana de agregar elemento y 1 ventana de listar elementos

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec et sem luctus, finibus mauris eget, euismod arcu. Morbi at mollis risus. Praesent consequat justo tellus, ac pretium urna lacinia quis. Ut sagittis cursus finibus. Morbi pellentesque vulputate tincidunt. Aliquam at pretium tellus, et vehicula velit. Nunc turpis metus, porttitor sit amet aliquam ut, venenatis quis elit. Nam tincidunt venenatis tortor, ac sodales libero varius nec. Donec vestibulum leo a metus rutrum, vitae elementum diam suscipit. Quisque non mauris rutrum, lacinia lectus id, viverra dui. In leo quam, ultrices vitae suscipit quis, porttitor quis turpis. Vestibulum semper ligula sit amet diam scelerisque gravida.

B.3. Se debe aplicar encapsulamiento y principios OO

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec et sem luctus, finibus mauris eget, euismod arcu. Morbi at mollis risus. Praesent consequat justo tellus, ac pretium urna lacinia quis. Ut sagittis cursus finibus. Morbi pellentesque vulputate tincidunt. Aliquam at pretium tellus, et vehicula velit. Nunc turpis metus, porttitor sit amet aliquam ut, venenatis quis elit. Nam tincidunt venenatis tortor, ac sodales libero varius nec. Donec vestibulum leo a metus rutrum, vitae elementum diam suscipit. Quisque non mauris rutrum, lacinia lectus id, viverra dui. In leo quam, ultrices vitae suscipit quis, porttitor quis turpis. Vestibulum semper ligula sit amet diam scelerisque gravida.

B.4. Continuidad en la utilización de GitHub (Realización de al menos 3 Commit adicionales a los ya hechos en la parte A)

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec et sem luctus, finibus mauris eget, euismod arcu. Morbi at mollis risus. Praesent consequat justo tellus, ac pretium urna lacinia quis. Ut sagittis cursus finibus. Morbi pellentesque vulputate tincidunt. Aliquam at pretium tellus, et vehicula velit. Nunc turpis metus, porttitor sit amet aliquam ut, venenatis quis elit. Nam tincidunt venenatis tortor, ac sodales libero varius nec. Donec vestibulum leo a metus rutrum, vitae elementum diam suscipit. Quisque non mauris rutrum, lacinia lectus id, viverra dui. In leo quam, ultrices vitae suscipit quis, porttitor quis turpis. Vestibulum semper ligula sit amet diam scelerisque gravida.

F. Entrega final

F.2. Implementar el manejo de excepciones capturando errores potenciales específicos mediante Try-catch

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec et sem luctus, finibus mauris eget, euismod arcu. Morbi at mollis risus. Praesent consequat justo tellus, ac pretium urna lacinia quis. Ut sagittis cursus finibus. Morbi pellentesque vulputate tincidunt. Aliquam at pretium tellus, et vehicula velit. Nunc turpis metus, porttitor sit amet aliquam ut, venenatis quis elit. Nam tincidunt venenatis tortor, ac sodales libero varius nec. Donec vestibulum leo a metus rutrum, vitae elementum diam suscipit. Quisque non mauris rutrum, lacinia lectus id, viverra dui. In leo quam, ultrices vitae suscipit quis, porttitor quis turpis. Vestibulum semper ligula sit amet diam scelerisque gravida.

F.3. Crear 2 clases que extiendan de una Excepción y que se utilicen en el programa

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec et sem luctus, finibus mauris eget, euismod arcu. Morbi at mollis risus. Praesent consequat justo tellus, ac pretium urna lacinia quis. Ut sagittis cursus finibus. Morbi pellentesque vulputate tincidunt. Aliquam at pretium tellus, et vehicula velit. Nunc turpis metus, porttitor sit amet aliquam ut, venenatis quis elit. Nam tincidunt venenatis tortor, ac sodales libero varius nec. Donec vestibulum leo a metus rutrum, vitae elementum diam suscipit. Quisque non mauris rutrum, lacinia lectus id, viverra dui. In leo quam, ultrices vitae suscipit quis, porttitor quis turpis. Vestibulum semper ligula sit amet diam scelerisque gravida.

F.4. Aplicación del patrón de diseño Strategy (Estrategia)

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec et sem luctus, finibus mauris eget, euismod arcu. Morbi at mollis risus. Praesent consequat justo tellus, ac pretium urna lacinia quis. Ut sagittis cursus finibus. Morbi pellentesque vulputate tincidunt. Aliquam at pretium tellus, et vehicula velit. Nunc turpis metus, porttitor sit amet aliquam ut, venenatis quis elit. Nam tincidunt venenatis tortor, ac sodales libero varius nec. Donec vestibulum leo a metus rutrum, vitae elementum diam suscipit. Quisque non mauris rutrum, lacinia lectus id, viverra dui. In leo quam, ultrices vitae suscipit quis, porttitor quis turpis. Vestibulum semper ligula sit amet diam scelerisque gravida.

Opcional: Reemplazar los datos iniciales en el código por una conexión con base de datos local (MySQL), archivo de texto CSV o Excel, para la persistencia de datos

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec et sem luctus, finibus mauris eget, euismod arcu. Morbi at mollis risus. Praesent consequat justo tellus, ac pretium urna lacinia quis. Ut sagittis cursus finibus. Morbi pellentesque vulputate tincidunt. Aliquam at pretium tellus, et vehicula velit. Nunc turpis metus, porttitor sit amet aliquam ut, venenatis quis elit. Nam tincidunt venenatis tortor, ac sodales libero varius nec. Donec vestibulum leo a metus rutrum, vitae elementum diam suscipit. Quisque non mauris rutrum, lacinia lectus id, viverra dui. In leo quam, ultrices vitae suscipit quis, porttitor quis turpis. Vestibulum semper ligula sit amet diam scelerisque gravida.

Opcional: Considerar la implementación del patrón Modelo-Vista-Controlador (MVC) en la arquitectura del sistema

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec et sem luctus, finibus mauris eget, euismod arcu. Morbi at mollis risus. Praesent consequat justo tellus, ac pretium urna lacinia quis. Ut sagittis cursus finibus. Morbi pellentesque vulputate tincidunt. Aliquam at pretium tellus, et vehicula velit. Nunc turpis metus, porttitor sit amet aliquam ut, venenatis quis elit. Nam tincidunt venenatis tortor, ac sodales libero varius nec. Donec vestibulum leo a metus rutrum, vitae elementum diam suscipit. Quisque non mauris rutrum, lacinia lectus id, viverra dui. In leo quam, ultrices vitae suscipit quis, porttitor quis turpis. Vestibulum semper ligula sit amet diam scelerisque gravida.

Consideraciones generales

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec et sem luctus, finibus mauris eget, euismod arcu. Morbi at mollis risus. Praesent consequat justo tellus, ac pretium urna lacinia quis. Ut sagittis cursus finibus. Morbi pellentesque vulputate tincidunt. Aliquam at pretium tellus, et vehicula velit. Nunc turpis metus, porttitor sit amet aliquam ut, venenatis quis elit. Nam tincidunt venenatis tortor, ac sodales libero varius nec. Donec vestibulum leo a metus rutrum, vitae elementum diam suscipit. Quisque non mauris rutrum, lacinia lectus id, viverra dui. In leo quam, ultrices vitae suscipit quis, porttitor quis turpis. Vestibulum semper ligula sit amet diam scelerisque gravida.

Cras vestibulum, orci id rhoncus facilisis, ipsum erat volutpat elit, eu elementum ligula velit eget ex. Fusce in felis massa. Aliquam commodo augue ac tellus fringilla interdum. Curabitur ac magna orci. Cras dolor turpis, gravida sed semper non, dictum ut eros. Sed vehicula urna at fringilla volutpat. Curabitur a tellus nec enim venenatis cursus vel a purus. Pellentesque gravida hendrerit urna, quis sodales erat faucibus pharetra.

Nunc erat mauris, finibus vitae velit a, fermentum lobortis arcu. Nunc luctus viverra augue id tincidunt. Pellentesque vitae libero nec erat pharetra dapibus eget vel nunc. Mauris tempus laoreet lorem non pretium. Nullam a neque condimentum, lacinia sapien sit amet, elementum urna. Mauris ac metus nec neque vulputate volutpat nec id nunc. Nulla augue leo, finibus eget consectetur at, imperdiet at magna.

.1. Bla bla bla

Nullam tincidunt mollis odio et volutpat. In vel velit dui. Phasellus pulvinar venenatis lorem, quis volutpat risus euismod sed. Fusce at odio vel nunc faucibus vulputate vitae vitae tortor. Phasellus rhoncus aliquam lectus, et cursus ipsum interdum sed. Mauris et nisl non diam tincidunt sagittis. Maecenas ultrices nunc nec tellus commodo, non interdum augue laoreet. Nullam auctor lacus sed hendrerit tempor. Aliquam nec luctus orci, eu varius eros. Nam sit amet lorem id nisi ultricies rutrum vitae ac nibh. Curabitur ultrices tortor non pulvinar placerat.

Nulla fermentum efficitur nulla at blandit. Donec dignissim malesuada sapien ac tempus. Quisque hendrerit ipsum eget interdum faucibus. Suspendisse odio eros, dapibus sit amet purus id, vestibulum commodo nunc. Proin placerat augue nibh, vitae semper diam varius vel. Nulla pretium felis et lacus pellentesque mollis. Ut sodales orci id libero pretium commodo. Pellentesque vitae urna vel tellus ullamcorper volutpat eget et diam.

Praesent porta dolor quam, et congue dui aliquet vitae. Phasellus dignissim in nisl vel volutpat. Morbi id turpis quis mi ornare egestas vel sed sem. Duis molestie condimentum ipsum, non tincidunt felis. Praesent rutrum placerat odio a semper. Ut vulputate scelerisque convallis. Curabitur vitae consectetur est. Nunc iaculis tincidunt libero ut varius. Donec elementum vitae erat vitae bibendum. Suspendisse mattis ultricies pharetra. Integer ac urna congue justo commodo varius. Ut vel rutrum eros. In hac habitasse platea dictumst.

Sed quis sem id est porta ornare vel aliquet lorem. Fusce tincidunt blandit ultrices. Duis ullamcorper, sem nec faucibus viverra, nisi est mattis massa, congue tincidunt ante risus ac lacus. Cras lobortis condimentum turpis at auctor. Nulla facilisi. Proin quis ligula sed tortor placerat tincidunt. Nullam vitae euismod enim. Pellentesque nec risus vel risus congue dapibus. Aenean non nisi at augue volutpat porttitor. Proin ac massa eu dui pulvinar ornare. Curabitur ac tortor sodales, aliquet enim eu, suscipit massa. Nam purus libero, ullamcorper ac sodales et, ullamcorper ut tortor. Maecenas sodales vitae leo sit amet tempor. Suspendisse sodales fermentum pharetra. Praesent at tellus viverra lectus dictum dictum at sit amet massa.

Vivamus ante nibh, lacinia pharetra tristique in, pharetra sed lacus. Curabitur a pellentesque neque. Nulla venenatis metus ante, sit amet accumsan augue placerat a. Vivamus tristique nunc vitae massa suscipit, vitae eleifend sapien aliquet. Ut laoreet quam quis tellus sagittis mollis ut ut sem. Nam pretium purus lorem, eu hendrerit metus dapibus vitae. Vestibulum auctor sollicitudin risus sed mattis.

Praesent interdum suscipit consectetur. Morbi a viverra nulla. Nulla sollicitudin est nunc, a tristique dui rutrum ac. In tincidunt massa orci. Vivamus nec ligula luctus, bibendum sapien non, tempus mi. Morbi laoreet vitae eros eu ullamcorper. Nunc vel placerat ante. Fusce nec leo eget est malesuada aliquam. Cras vel massa fringilla, sollicitudin nisi vitae, pharetra nunc. Donec quis sapien quis risus volutpat auctor. Curabitur eget massa elementum, faucibus ex at, viverra nisl.

Nunc nisl nunc, molestie eu libero at, tempor feugiat nisi. Praesent tristique fermentum nulla. In nec consequat magna, a auctor orci. Integer elementum justo a risus dapibus, id tempor dui malesuada. Interdum et malesuada fames ac ante ipsum primis in faucibus. Morbi ipsum lorem, posuere nec maximus ut, tempus vitae tortor. Vestibulum consectetur justo id dui tristique, nec faucibus orci ultrices. Pellentesque ultricies, turpis a laoreet consectetur, risus tellus dictum orci, ut porttitor nulla enim nec est.

Sed ac mi rutrum, mollis urna blandit, volutpat felis. In hac habitasse platea dictumst. Vestibulum sit amet nunc sed sem aliquam interdum. Morbi venenatis molestie odio, et euismod lorem malesuada at. Aenean sed felis et lectus rutrum placerat nec pharetra nibh. Sed auctor volutpat lacus eu molestie. Nulla dui massa, condimentum a odio ut, euismod eleifend metus. Curabitur porttitor quam eu velit pellentesque malesuada.

Phasellus non arcu tristique, maximus dolor eget, tempor enim. Sed vel tincidunt dui. Nam viverra leo a ultrices sagittis. Vestibulum pellentesque semper vulputate. Aenean laoreet interdum lectus, ut pharetra magna varius in. Vivamus

vestibulum molestie hendrerit. In eget dignissim ligula. Aliquam elementum, quam a fringilla efficitur, nisi elit malesuada tellus, in rhoncus enim arcu id ex. Sed fringilla diam vel luctus efficitur. Nulla facilisi. Maecenas ac luctus mi.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec et sem luctus, finibus mauris eget, euismod arcu. Morbi at mollis risus. Praesent consequat justo tellus, ac pretium urna lacinia quis. Ut sagittis cursus finibus. Morbi pellentesque vulputate tincidunt. Aliquam at pretium tellus, et vehicula velit. Nunc turpis metus, porttitor sit amet aliquam ut, venenatis quis elit. Nam tincidunt venenatis tortor, ac sodales libero varius nec. Donec vestibulum leo a metus rutrum, vitae elementum diam suscipit. Quisque non mauris rutrum, lacinia lectus id, viverra dui. In leo quam, ultrices vitae suscipit quis, porttitor quis turpis. Vestibulum semper ligula sit amet diam scelerisque gravida.

Cras vestibulum, orci id rhoncus facilisis, ipsum erat volutpat elit, eu elementum ligula velit eget ex. Fusce in felis massa. Aliquam commodo augue ac tellus fringilla interdum. Curabitur ac magna orci. Cras dolor turpis, gravida sed semper non, dictum ut eros. Sed vehicula urna at fringilla volutpat. Curabitur a tellus nec enim venenatis cursus vel a purus. Pellentesque gravida hendrerit urna, quis sodales erat faucibus pharetra.

Nunc erat mauris, finibus vitae velit a, fermentum lobortis arcu. Nunc luctus viverra augue id tincidunt. Pellentesque vitae libero nec erat pharetra dapibus eget vel nunc. Mauris tempus laoreet lorem non pretium. Nullam a neque condimentum, lacinia sapien sit amet, elementum urna. Mauris ac metus nec neque vulputate volutpat nec id nunc. Nulla augue leo, finibus eget consectetur at, imperdiet at magna.

Nullam tincidunt mollis odio et volutpat. In vel velit dui. Phasellus pulvinar venenatis lorem, quis volutpat risus euismod sed. Fusce at odio vel nunc faucibus vulputate vitae vitae tortor. Phasellus rhoncus aliquam lectus, et cursus ipsum interdum sed. Mauris et nisl non diam tincidunt sagittis. Maecenas ultrices nunc nec tellus commodo, non interdum augue laoreet. Nullam auctor lacus sed hendrerit tempor. Aliquam nec luctus orci, eu varius eros. Nam sit amet lorem id nisi ultricies rutrum vitae ac nibh. Curabitur ultrices tortor non pulvinar placerat.

Nulla fermentum efficitur nulla at blandit. Donec dignissim malesuada sapien ac tempus. Quisque hendrerit ipsum eget interdum faucibus. Suspendisse odio eros, dapibus sit amet purus id, vestibulum commodo nunc. Proin placerat augue nibh, vitae semper diam varius vel. Nulla pretium felis et lacus pellentesque mollis. Ut sodales orci id libero pretium commodo. Pellentesque vitae urna vel tellus ullamcorper volutpat eget et diam.

Praesent porta dolor quam, et congue dui aliquet vitae. Phasellus dignissim in nisl vel volutpat. Morbi id turpis quis mi ornare egestas vel sed sem. Duis molestie condimentum ipsum, non tincidunt felis. Praesent rutrum placerat odio a semper. Ut vulputate scelerisque convallis. Curabitur vitae consectetur est. Nunc iaculis tincidunt libero ut varius. Donec elementum vitae erat vitae bibendum. Suspendisse mattis ultricies pharetra. Integer ac urna congue justo commodo varius. Ut vel rutrum eros. In hac habitasse platea dictumst.

Sed quis sem id est porta ornare vel aliquet lorem. Fusce tincidunt blandit ultrices. Duis ullamcorper, sem nec faucibus viverra, nisi est mattis massa, congue tincidunt ante risus ac lacus. Cras lobortis condimentum turpis at auctor. Nulla facilisi. Proin quis ligula sed tortor placerat tincidunt. Nullam vitae euismod enim. Pellentesque nec risus vel risus congue dapibus. Aenean non nisi at augue volutpat porttitor. Proin ac massa eu dui pulvinar ornare. Curabitur ac tortor sodales, aliquet enim eu, suscipit massa. Nam purus libero, ullamcorper ac sodales et, ullamcorper ut tortor. Maecenas sodales vitae leo sit amet tempor. Suspendisse sodales fermentum pharetra. Praesent at tellus viverra lectus dictum dictum at sit amet massa.

Vivamus ante nibh, lacinia pharetra tristique in, pharetra sed lacus. Curabitur a pellentesque neque. Nulla venenatis metus ante, sit amet accumsan augue placerat a. Vivamus tristique nunc vitae massa suscipit, vitae eleifend sapien aliquet. Ut laoreet quam quis tellus sagittis mollis ut ut sem. Nam pretium purus lorem, eu hendrerit metus dapibus vitae. Vestibulum auctor sollicitudin risus sed mattis.

Praesent interdum suscipit consectetur. Morbi a viverra nulla. Nulla sollicitudin est nunc, a tristique dui rutrum ac. In tincidunt massa orci. Vivamus nec ligula luctus, bibendum sapien non, tempus mi. Morbi laoreet vitae eros eu ullamcorper. Nunc vel placerat ante. Fusce nec leo eget est malesuada aliquam. Cras vel massa fringilla, sollicitudin nisi vitae, pharetra nunc. Donec quis sapien quis risus volutpat auctor. Curabitur eget massa elementum, faucibus ex at, viverra nisl.

Nunc nisl nunc, molestie eu libero at, tempor feugiat nisi. Praesent tristique fermentum nulla. In nec consequat magna, a auctor orci. Integer elementum justo a risus dapibus, id tempor dui malesuada. Interdum et malesuada fames ac ante ipsum primis in faucibus. Morbi ipsum lorem, posuere nec maximus ut, tempus vitae tortor. Vestibulum consectetur justo id dui tristique, nec faucibus orci ultrices. Pellentesque ultricies, turpis a laoreet consectetur, risus tellus dictum orci, ut porttitor nulla enim nec est.

Sed ac mi rutrum, mollis urna blandit, volutpat felis. In hac habitasse platea dictumst. Vestibulum sit amet nunc sed sem aliquam interdum. Morbi venenatis molestie odio, et euismod lorem malesuada at. Aenean sed felis et lectus rutrum placerat nec pharetra nibh. Sed auctor volutpat lacus eu molestie. Nulla dui massa, condimentum a odio ut, euismod eleifend metus. Curabitur porttitor quam eu velit pellentesque malesuada.

Phasellus non arcu tristique, maximus dolor eget, tempor enim. Sed vel tincidunt dui. Nam viverra leo a ultrices sagittis. Vestibulum pellentesque semper vulputate. Aenean laoreet interdum lectus, ut pharetra magna varius in. Vivamus vestibulum molestie hendrerit. In eget dignissim ligula. Aliquam elementum, quam a fringilla efficitur, nisi elit malesuada tellus, in rhoncus enim arcu id ex. Sed fringilla diam vel luctus efficitur. Nulla facilisi. Maecenas ac luctus mi.