Proyecto I
Caso: Programa para Escuela de Basica

Carlo Bernucci Gómez
Profesor: Victor Hugo Aravena
ICC225-1: Programación

Problema

Se presenta el objetivo de diseñar un sistema y software para una escuela de básica, en el cual se pueda:

1. Ingresar la planificación de actividades de una asignatura.
2. Vincular cursos, alumnos, profesor.
3. Vincular apoderado a uno o más alumnos.
4. Ingresar las notas de evaluaciones.
5. Ingresar la asistencia a clases.
6. Ingresar anotación positivas o negativas a alumnos.

Junto a esto, la información almacenada se deberá exportar en distintos reportes a distintos formatos, los cuales serán:

1. XML
2. Json
3. Excel
4. Word
5. Html

Los reportes consisten en:

1. En un único archivo por apoderado, entregar la información de los hijos de un apoderado, la información consiste en notas de evaluaciones, actividades, anotaciones, promedios, y asistencia del alumno.
2. Reporte para cada profesor, en el cual se muestre el promedio de los alumnos que pertenecen a la asignatura que imparte.
3. Reporte por alumno que permita conocer el porcentaje de asistencia y notas por alumno.
4. Reporte que permita conocer los alumnos bajo cierto porcentaje de asistencia, dicho porcentaje entregado por el usuario.
5. Reporte que permita conocer los alumnos reprobados.
6. Reporte que permita conocer los apoderados con más de un alumno en la escuela.
7. Reporte para cada apoderado por cada uno de sus hijos, el cual de a conocer la planificación de actividades.

Aparte de estas condiciones, se debe crear un archivo XML o Json para hacer un poblamiento de datos para el programa. Este archivo deberá contar con las siguientes especificaciones.

1. Crear 8 niveles, cada uno con 2 Curso, A y B, y cada curso con 30 estudiantes.
2. Cada curso tiene 5 asignaturas, y cada asignatura tiene un profesor.
3. Cada asignatura tiene 5 notas (5 actividades evaluadas).
4. Cada asignatura tiene una planificación de 10 actividades.
5. Cada estudiante tiene una asistencia con 30 registros mínimo.
6. Cada estudiante tiene 1 apoderado, pero un apoderado puede tener más de 1 estudiante.

La entrega final consiste en entregar lo antes señalado, además de documentación de código, pruebas unitarias, diagrama de clases, Html Javadocs, este informe más un video explicativo.

Análisis

Se pide crear un programa para una escuela de básica, son 8 niveles con 2 cursos por nivel, cada uno con 30 estudiantes, cada curso tiene 5 asignaturas y cada una tiene un profesor, además cada alumno tiene un apoderado, los requisitos de asistencia y anotaciones solo se relacionan con el alumno así que no deberían presentar mayor problema.

Me propuse crear relaciones sin bucles para que la exportación de reportes fuera mas fácil.

Identifique las clases mas grandes como Escuela/Colegio y definí de que se componen, así con cada uno de las clases que dividen a una escuela, como actividad, asignatura, curso, nivel, escuela.

Después define a otro grupo de clases, que son los que se relacionan directamente con la información, como lo son los alumnos, cuando se les asignan actividades notas, anotaciones, asistencias, o profesores, cuando es necesario identificar y exportar en reporte la información de la asignatura que imparte un profesor, o los apoderados, que se relacionan con los alumnos y su respectiva información.

Así me di cuenta que ciertos atributos de clases podían dejar de ser referencias a otras clases y pasar a ser String para que la creación del archivo XML o Json inicial sea mas sencillo y no presente problemas. También se da a entender que existen ciertos atributos necesarios que no se nombran, que pueden servir para entregar la información de mejor manera, tal como promedio, o promedio por asignatura.

Junto con esto también planifique la creación de archivos xsl para la exportación de los reportes a archivos excel, word y html desde los xml que iba a generar por cada reporte.

Además de analizar lo que se pide dentro del sistema, tambien se entiende que se debe crear una interfaz grafica en la cual se pueda agregar información y datos, como el ingresar anotaciones, actividades, ya sean evaluadas o no, y si son evaluadas ingresar las notas correspondientes.

Diseño

Partí creando un diagrama de clases en Visual Paradigm, definí las clases mas grandes y como se componían, y siempre entendiendo como fluye la información por este sistema, en el cual cada set o get de algún dato debe terminar en un callejón sin salida para no generar bucles.

Las primeras clases que relacione fueron Colegio←-Niveles←-Cursos←-Asignaturas, según las especificaciones del problema. Después define la forma de identificar cada uno, por lo que un nivel tiene un Int, un curso un Char y una asignatura un String.

La clase alumno, la relacione con un curso, así definí que Curso←-(Alumnos y Asignaturas), siempre pensando en los reportes y como mostrar la información por la interfaz gráfica es que también a Alumno le di como atributo Asignaturas, así con este diseño el llamado de información la puedo dividir en 2.

Si un profesor necesita ver o cambiar los datos de alguna nota de alguna asignatura, el recorrido es de Curso a Asignatura, ya que en esta se guardan las notas de todos los alumnos que tienen dicha asignatura, pero si necesito saber la información de 1 alumno por ejemplo para entregarla a apoderas, el recorrido seria Curso a Alumno a Asignatura, donde solo se encuentran las actividades de este alumno, y además de esta forma es más automático calcular promedios por asignatura y promedio general.

La clase Apoderado está al mismo nivel de Niveles del Colegio y se relaciona solo con alumnos, ya que apoderado solo necesita que le muestren información de sus alumnos.

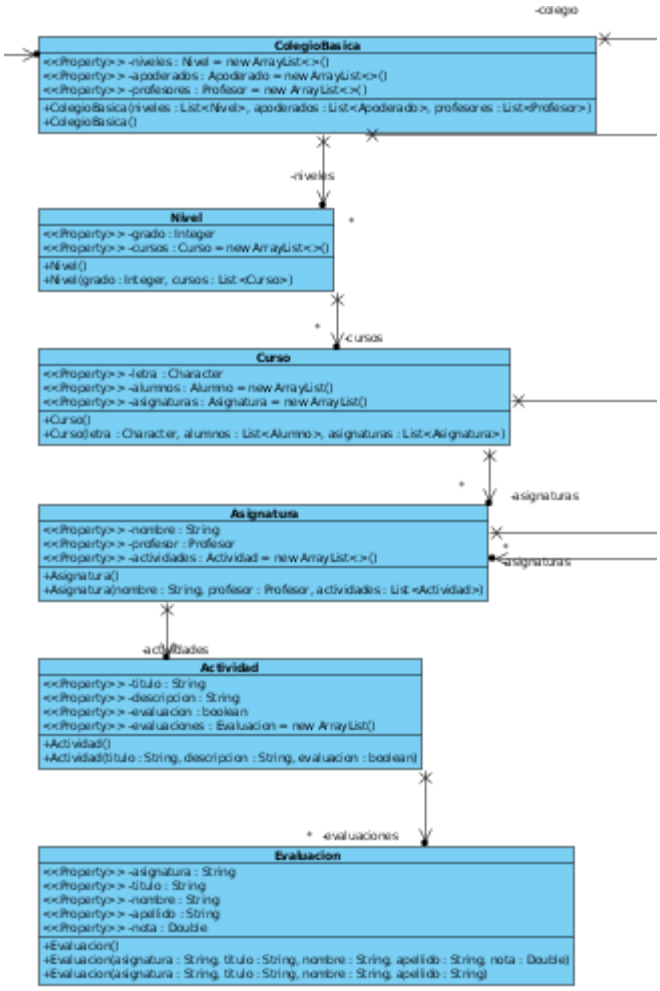
La clase Profesor se relaciona con Alumnos y la Asignatura se relaciona con profesor, así el profesor tiene la información de sus alumnos.

Además tengo una clase extra en la cual tengo todos los métodos que crean Xml y Json y los transforman en excel, word y html.

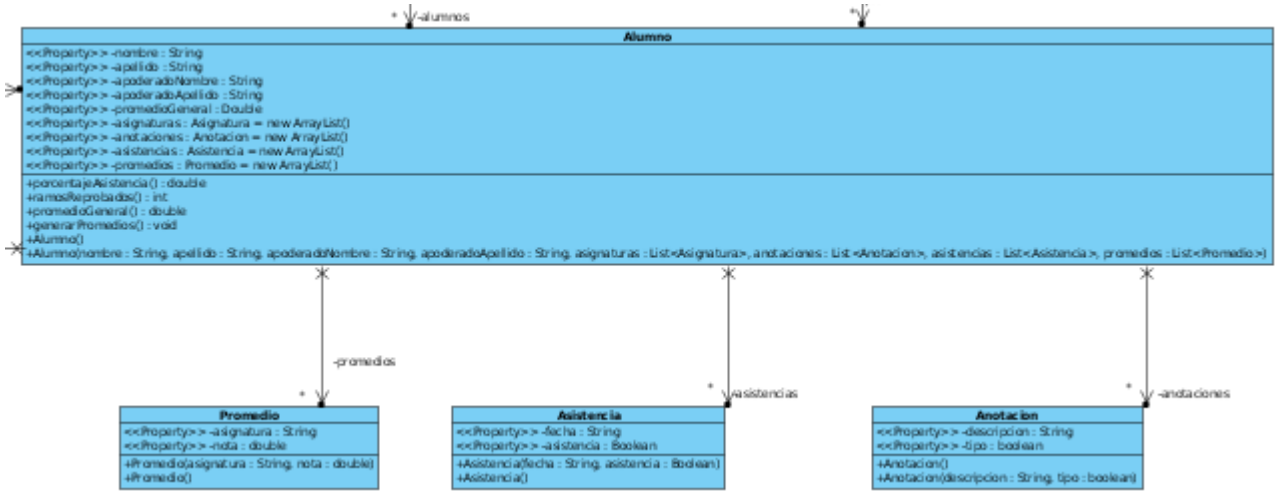
Las clases que define como más “pequeñas” son anotación, actividad, asistencia, evaluación (de una actividad), y promedio, todas estas me ayudan a que al crear Arrays de dichos objetos, estos puedan almacenar mas y diferente tipos de información, como por ejemplo asistencia, que se compone de una fecha tipo String y un “asiste” de tipo Boolean.

El uml es demasiado grande, ya que también muestra los métodos as importantes, así que se podra ver parte de la estructura del sistema en las siguientes imagenes.

Estructura del sistema colegio sin alumnos, profesor y apoderado.



Estructura Alumno



Conclusión

Por medio de un buen análisis previo, se puede tener en mente como se desea crear el sistema, y como se abordaran los problemas que van surgiendo, además de que se debe tener y como se va a mostrar las opciones en la interfaz gráfica.

El buen análisis termina en un sistema bien estructurado, tal ves no el más óptimo, pero estable, según esto fue fácil después generar datos aleatorios, para después estos exportarlos a un archivo JSON y tenerlo como el poblamiento base del programa.

Es necesario resaltar que se entiende después de realizar el proyecto que es mejor generar cambio a través del programa, que intentar resolverlo directamente en el archivo de poblamiento inicial, aunque sea un cambio mínimo.

Si el sistema tiene una estructura sin bucles permite pasar objetos con datos directamente a Xml o Json, con lo que exportarlos a otros formatos es sencillo, la desventaja de esto se centra en que el xml puede contener datos que no son relevantes para el reporte, pero esto varia si es que en algún momento es necesario agregar más datos que mostrar en el momento de la exportación.

Cabe recalcar que crear los métodos para llevar los datos ingresados a varios objetos y dividir esta información según sea necesario fue complicado, pero al momento de querer crear un reporte por ejemplo de un alumno, el hacer el xsl y recorrer nodos se hace mucho más fácil.

Extras como el Javadoc ayudo a mantener orden en los métodos y no tener que revisarlos nuevamente para entender que hacían, además de las pruebas unitarias, las cuales me sirvieron en probar que objetos se puedan crear de cierta forma, ya que algunos tienen mas de 1 constructor.

Junto a todo esto puedo concluir y resaltar que llevar un itinerario ayuda bastante en avanzar el proyecto y mantener un orden en las tareas y detalles pendientes.