



EDUCACIÓN
SECRETARÍA DE EDUCACIÓN PÚBLICA



TECNOLÓGICO
NACIONAL DE MÉXICO®

Instituto Tecnológico de Orizaba
División de Estudios de Posgrado e Investigación
Maestría en Sistemas Computacionales

PROYECTO DE TESIS

TÍTULO DEL PROYECTO: “Análisis de perfiles de usuarios empleando tecnologías de web semántica”

PRESENTADO POR:

L.A.V. MARIO RICARDO PÉREZ HERNÁNDEZ M18010279

PARA OBTENER EL GRADO DE:

MAESTRO EN SISTEMAS COMPUTACIONALES

DIRECTOR DEL ANTEPROYECTO DE TESIS:

M.S.C LUIS ÁNGEL REYES HERNÁNDEZ

Agradecimientos

Al Tecnológico Nacional de México por la oportunidad de ingresar a la maestría en sistemas computacionales.

Al Consejo Nacional de Ciencia y Tecnología, CONACyT, por el apoyo económico otorgado para la realización de mis estudios de maestría.

A todos los maestros y doctores de la maestría en sistemas computacionales por sus enseñanzas y su paciencia.

A mis compañeros de clases por sus aportes.

A mi familia por el apoyo brindado.

Índice General

Índice de tablas	8
Índice de figuras	9
Resumen	11
Abstract.....	12
Introducción.....	13
Capítulo 1. Antecedentes.....	15
1.1. Marco teórico	15
1.1.1. E-Learning.....	15
1.1.2. Servicios Web	15
1.1.3 Ontologías	16
1.1.4 Aprendizaje Automático	17
1.1.5 Lenguaje de Consultas	17
1.2. Planteamiento del problema.....	18
1.3. Objetivos General y específicos.....	19
1.3.1. Objetivo General	19
1.3.2. Objetivos específicos.....	19
1.4. Justificación	19
Capítulo 2. Estado de la Práctica	20
2.1. Trabajos relacionados	20
2.2 Análisis comparativo	27
2.3 Propuesta de solución	33
2.3.1 Java.....	33
2.3.2 <i>Framework</i> JavaServer Faces.....	33
2.3.3 <i>Framework</i> PrimeFaces.....	34
2.3.4 <i>Framework</i> Apache Jena.....	34
2.3.5 NetBeans	34
Capítulo 3. Aplicación de la metodología	36
3.1 Análisis de requerimientos.....	36

3.2 Diseño del sistema	37
3.2.1 Diseño de la interfaz de usuario	37
3.2.2 Diagrama de paquetes	40
3.2.2.1 Diagrama de paquetes de la aplicación Web.....	40
3.2.2.2 Diagrama de paquetes de la aplicación móvil.....	41
3.2.3 Diagramas de clases.....	42
3.2.3.1 Diagramas de clases correspondientes a la aplicación Web	42
.....	46
3.2.3.2 Diagramas de paquetes correspondientes a la aplicación móvil	47
3.2.4 Arquitectura del sistema.....	50
3.2.5 Patrón MVC	50
3.2.6 Diagramas de secuencia	51
3.3 Codificación del software	56
3.3.1 Conexión a la base de datos	56
3.3.2 Esquema RDF	58
3.3.3 Consultas con SPARQL.....	59
3.3.3 Métodos de escritura sobre los archivos RDF.....	60
3.3.4 Métodos de los servicios REST	62
3.4 Pruebas.....	63
3.4.1 Pruebas de consulta de datos por medio de la ontología en la aplicación Web	63
3.4.2 Pruebas de consulta y modificación de datos de la aplicación móvil.....	65
3.5 Fase de implementación.....	67
3.5.1 Aplicación Web.....	67
3.5.2 Aplicación Móvil.....	67
Capítulo 4 Resultados.....	69
4.1 Caso de estudio	69
4.1.1 Las evaluaciones de los alumnos	69
4.2 Evaluación de los resultados.....	69
4.2.1 Establecimiento de las relaciones entre los alumnos utilizando el vocabulario FOAF	70

4.2.2 Funciones agregadas a la aplicación móvil.....	72
4.2.3 Datos disponibles para consulta del docente.....	73
Capítulo 5 Conclusiones y recomendaciones	76
5.1 Conclusiones.....	76
5.2 Recomendaciones	77
Productos Académicos	78
Referencias	79
Apéndice.....	82
Glosario.....	82
Anexos	82

Índice de tablas

Tabla 1. Comparativa de los trabajos relacionados.....	28
Tabla 2. Relación de la calificación de los alumnos con el lugar que ocupan dentro del aula.....	71
Tabla 3. Relación de alumnos reportados correspondiente a un mes.....	72

Índice de figuras

Figura 3.1 Diagrama de casos de uso para las aplicaciones Web y móvil	36
Figura 3.2.1 Diseño de la página principal de la aplicación Web	38
Figura 3.2.2 Menú de administrador desplegando la sección de Datos de los Alumnos	38
Figura 3.2.3 Ejemplo de formulario de captura para un nuevo alumno	39
Figura 3.2.4 Ejemplo de Consulta de información de un grupo	39
Figura 3.2.5 Ejemplo de la disposición de las pantallas de la aplicación móvil	40
Figura 3.3 Diagrama de paquetes de la aplicación Web.....	41
Figura 3.3.1 Diagrama de paquetes de la aplicación móvil.....	42
Figura 3.3.2 Diagrama de clases correspondiente al paquete <i>model</i>	43
Figura 3.3.3 Diagrama de clases correspondiente al paquete <i>filter</i>	43
Figura 3.3.4 Diagrama de clases correspondiente al paquete <i>control</i>	44
Figura 3.3.5 Diagrama de clases correspondiente al paquete <i>beans</i>	44
Figura 3.3.6 Diagrama de clases correspondiente al paquete <i>dao</i>	45
Figura 3.3.7 Diagrama de clases correspondiente al paquete <i>weka</i>	45
Figura 3.3.8 Diagrama de clases correspondiente al paquete <i>rest</i>	46
Figura 3.3.9 Diagrama de clases correspondiente al paquete <i>activities</i>	47
Figura 3.4 Diagrama de clases correspondiente al paquete <i>adapters</i>	48
Figura 3.4.1 Diagrama de clases correspondiente al paquete <i>gadgets</i>	48
Figura 3.4.2 Diagrama de clases correspondiente al paquete <i>model</i>	49
Figura 3.4.3 Diagrama de clases correspondiente al paquete <i>rest</i>	49
Figura 3.5 Arquitectura del sistema.....	50
Figura 3.5.1 Arquitectura de la aplicación	50
Figura 3.6.1 Diagrama de secuencia del inicio de sesión.....	51
Figura 3.6.2 Diagrama de secuencia de la eliminación de un registro	52
Figura 3.6.3 Diagrama de secuencia de la creación de un nuevo registro.....	52
Figura 3.6.4 Diagrama de secuencia de la modificación de un registro	53
Figura 3.6.5 Diagrama de secuencia de la consulta de datos	53
Figura 3.6.6 Diagrama de secuencia del inicio de sesión.....	54
Figura 3.6.7 Diagrama de secuencia de la creación de un plan de clase	54
Figura 3.6.8 Diagrama de secuencia de consulta de datos de un estudiante seleccionado.....	55
Figura 3.6.9 Diagrama de secuencia del inicio de clase en tiempo real	55
Figura 3.7 Código para la inicialización del servidor Fuseki al inicio de la aplicación Web ...	56
Figura 3.7.1 Código para la inicialización del servidor Fuseki incrustado	57
Figura 3.7.2 Código para la inicialización de la base de datos de transacciones sin el servidor Fuseki.....	57
Figura 3.8 Estructura del archivo RDF.....	58
Figura 3.9 Método en Java para la consulta de información de alumnos desde la clase modelo, se incluye un ejemplo de consulta mediante la sintaxis de SPARQL.....	59
Figura 3.9.1 Método de inserción de datos.....	60

Figura 3.9.1 Método de la clase de conexión a datos para obtener la información de un archivo RDF	60
Figura 3.9.2 Método de modificación de los datos de un archivo RDF	61
Figura 3.9.3 Método para eliminar datos de un archivo RDF	62
Figura 3.9.4 Métodos para la solicitud de lectura de datos	62
Figura 3.9.5 Métodos para la solicitud de lectura de datos	62
Figura 3.9.6 Pantalla de inicio de sesión	63
Figura 3.9.7 Pantalla con menú de administrador	64
Figura 3.9.8 Vista de la tabla con información de los alumnos	64
Figura 3.9.9 Ejemplo de los amigos relacionados con el alumno seleccionado.....	65
Figura 3.9.9.2 Vista de la prueba realizada para la obtención de datos con respecto al género de los alumnos	65
Figura 3.9.9.3 Vista de la pantalla de inicio de sesión y datos usuario con sesión iniciada.....	66
Figura 3.9.9.4 Vista de las pantallas que muestran información obtenida por los servicios REST	66
Figura 3.9.9.5 Vista de las pantallas de los menús que afectan los datos en el servidor por medio de los servicios REST	67
Figura 4.1 Relación alumno-posición. Marcados con borde rojo los alumnos con promedio menor a 70 y con amarillo menor a 80 y mayor a 70 para enfatizar la relación.....	70
Figura 4.2 Cuadrícula de selección de alumno donde se ha seleccionado a Alumno 13	71
Figura 4.3 Vista de	72
Figura 4.4 Captura de pantalla del menú de consulta de alumnos reportados.....	73
Figura 4.4.1 Detalle de la captura de pantalla que muestra el desglose de reportes pertenecientes a un alumno seleccionado.	73
Figura 4.4.2 Captura de pantalla que muestra el menú de las calificaciones de los alumnos ...	74
Figura 4.4.3 Vista del PDF generado para los reportes totales de los alumnos.....	74
Figura 4.4.4 Vista del PDF generado para los reportes pertenecientes a un alumno seleccionado	75
Figura 4.4.5 Vista del PDF generado para las calificaciones de un grupo seleccionado	75

Resumen

Los modelos educativos en México tienen dificultad para la adopción del *e-Learning*, la causa de esto es la saturación de las aulas en las escuelas públicas y privadas además de la rigidez de los programas educativos que se tienen hasta la fecha. Los avances tecnológicos han abaratado el costo de hardware como los teléfonos inteligentes, que al estar al alcance de un sector muy amplio de la población mexicana, permiten pensar en su inserción como herramienta útil para la educación. Este documento propone un primer paso al ofrecer al docente la posibilidad de automatizar varias de las tareas que comúnmente debe llevar a cabo en la parte administrativa de su trabajo, dándole la oportunidad de enfocarse en tareas más importantes como el seguimiento del desempeño de sus alumnos, para lo cual se propone el uso de web semántica generar información relevante que además pueda utilizarse diariamente de forma eficiente y eficaz.

Para emplear la web semántica se realizaron una aplicación web y una aplicación Android. Se utilizó el lenguaje FOAF (*Friend Of A Friend*, amigo de un amigo) con un manejo de datos mediante RDF (*Resource Description Framework*, marco de trabajo de descripción de recursos) con el fin de que los datos ingresados (en específico los perfiles de los estudiantes) sean analizados y por medio de las relaciones establecidas por el RDF se obtenga información que el docente pueda utilizar en sus labores cotidianas para la mejora continua de sus actividades en tiempo real durante las clases (por medio de la aplicación Android) y revisiones fuera del horario de trabajo (Por medio de la aplicación web). Los servicios Web se implementaron utilizando la arquitectura REST (*Representational State Transfer*, Transferencia Figurativa de Estado).

Las tecnologías seleccionadas para este proyecto son: como lenguajes Java y html5 con el marco de trabajo de JSF (*Java Server Faces*), para el aspecto visual se utilizó CSS (*Cascading Style Sheet*) y el *framework* de *PrimeFaces* en el caso de la aplicación web; los servicios Web utilizan la API (*Application Programming Interface*, Interfaz de Programación de Aplicaciones) de Jersey y las consultas de los datos son a través de SPARQL y SPARUL utilizando el paquete de Apache Jena 3.12.0 y como servidor Apache Jena Fuseki incrustado. Para la aplicación Android se utilizó Android Studio para todo su desarrollo donde para consumir los servicios Web se utilizó la API de Retrofit.

Como funcionalidad extra al propósito del proyecto se incluyó una SVM (*Support Vector Machine*, Máquina de Soporte Vectorial) para realizar predicciones de conducta de los estudiantes con base a su información académica.

Abstract

The educational models in Mexico have some difficulties for the adoption of e-Learning, the cause of this is the saturation of the classrooms in public and private schools as well as the rigidity of the educational programs that are available to date. Technological advances have lowered the cost of hardware such as smartphones, which, being within reach of a very large sector of the Mexican population, allow their inclusion as a useful tool for education. This document proposes a first step by offering the teacher the possibility of automating several of the tasks that they must commonly carry out in the administrative part of their work, giving them the opportunity to focus on more important tasks such as monitoring the performance of their students, for which the use of semantic web is proposed to generate relevant information that can also be used daily efficiently and effectively.

To use the semantic web techniques, a web application and an Android application were made. The FOAF language (Friend of a Friend) was used with data management through RDF (Resource Description Framework) so the data entered (specifically the profiles of students) can be analyzed and through the relationships established by the RDF, providing information that the teachers can use in their daily work for the continuous improvement of their activities; in real-time classes (through the Android application) and revisions outside working hours (Through the web application). Web services were implemented using the REST (Representational State Transfer) architecture.

The technologies selected for this project are; As Java and HTML5 languages with the JSF framework (Java Server Faces), CSS (Cascading Style Sheet) and the PrimeFaces framework were used for the visual aspect in the case of the web application; Web services use the API (Application Programming Interface, Jersey Application Programming Interface) and data queries are through SPARQL and SPARUL using the Apache Jena 3.12.0 package and as an embedded Apache Jena Fuseki server. For the Android application, Android Studio was used for all its development where the Retrofit API was used to consume Web services.

As an extra functionality to the purpose of the project, an SVM (Support Vector Machine) was included to make predictions of student behavior based on their academic information.

Introducción

En la actualidad los avances tecnológicos se manifiestan a gran velocidad, cada año llegan al mercado software y hardware más precisos, versátiles y eficaces, lo que hace que los usuarios de todos los niveles tengan la necesidad de avanzar al mismo paso para mantener su desempeño en condiciones óptimas. Este fenómeno abarca todas las áreas del conocimiento y forma una parte muy activa en la vida de los usuarios, quienes tienen a su alcance dispositivos inteligentes que cada año reducen su costo en relación con las nuevas versiones. Esto genera las nuevas tendencias como el *e-Learning*, que en países desarrollados comienza a ganar más territorios, pues muchas escuelas en todos los niveles están fomentando el aprendizaje electrónico con resultados muy positivos. No obstante, adoptar un nuevo esquema educativo de este tipo no es fácil en países donde los programas establecidos son muy rígidos y están apegados a las prácticas de la década anterior, tal es el caso de la gran mayoría de las escuelas públicas y particulares en México, donde la matrícula de estudiantes por aula es en promedio de 32 alumnos de acuerdo con los registros de la OCDE (Organización para la Cooperación y el Desarrollo Económicos) del año 2015 [1].

Este es un problema que pone en peligro el nivel educativo del país, pues la labor docente en los niveles básico y medio superior se vuelve bastante compleja al tratar de dar seguimiento a los más de 30 alumnos por cada grupo que se tenga a cargo, aunado a esto, el manejo de los registros del avance de cada alumno todavía se solicita por escrito en la gran mayoría de las instituciones y la plataforma digital proporcionada por la SEP (Secretaría de Educación Pública) sólo permite el acceso por usuario a esta información, por lo que no es una herramienta útil para el profesor, actividad que consume más tiempo a los docentes, pues deben manejar diariamente el papeleo requerido por la institución en la que trabajan. Esto se aborda de forma eficaz por la tecnología, en específico, por medio de un *Smartphone* (teléfono inteligente) gracias a la accesibilidad de estos productos que tanto alumnos como profesores utilizan diariamente y que de acuerdo con el resultado de la Encuesta Nacional sobre Disponibilidad y Uso de Tecnologías de la Información en los Hogares (ENDUTIH) realizada por el INEGI (Instituto Nacional de Estadística y Geografía) en el 2017 [2], en un año se pasó de 60.6 millones a 64.7 millones de usuarios. Esto abre muchas posibilidades para la educación en México, de aquí deriva el propósito de este proyecto, en donde como una primera fase se quiere abordar el problema de seguimiento de los alumnos dotando al catedrático de una aplicación que le permita automatizar varias de las tareas que debe cumplir diariamente, permitiéndole estar mejor informado del desempeño de cada uno de los estudiantes. No es realista pensar en reducir el número de la matrícula de alumnos por aula, si bien esto también resuelve la situación actual, implicaría problemas más graves como un mayor porcentaje de mexicanos sin derecho a estudiar, por ello este proyecto busca introducir a los teléfonos inteligentes en las aulas de las escuelas de México, comenzando en un ambiente controlado, es decir, en las manos del docente, y a futuro en las manos de cada uno de los alumnos.

En este documento se presenta el marco teórico y los trabajos relacionados con ontologías y web semántica, también se propone la realización de una aplicación web y una aplicación Android como solución al problema descrito. Se muestran también las tablas comparativas de las tecnologías seleccionadas para la realización del proyecto y finalmente una conclusión de lo aprendido.

Capítulo 1. Antecedentes

1.1. Marco teórico

A continuación, se definen algunos términos relacionados con el trabajo de investigación.

1.1.1. E-Learning

E-Learning es aprender utilizando tecnologías electrónicas para acceder al plan de estudios educativo fuera del entorno y las estrategias tradicionales del aula. Anteriormente el *e-Learning* se refería, en el mayor de los casos, a cursos o títulos entregados en línea. Debido al avance de las tecnologías móviles, el *e-Learning* abarca todas las áreas educativas donde se hace un uso efectivo de la tecnología para alcanzar algún aprendizaje [3].

1.1.2. Servicios Web

El término servicio web es genérico o específico, donde el primero se refiere a un servicio ofrecido por un dispositivo electrónico a otro dispositivo electrónico, comunicándose entre sí a través de la *World Wide Web*, o específico; un servicio web implementado en la tecnología o marca particular, W3C (*World Wide Web Consortium*, Consorcio de la Red Mundial) *Web Services*.

En un servicio web, la tecnología web, como HTTP (*Hypertext Transfer Protocol*, Protocolo de transferencia de hipertexto), originalmente diseñada para comunicación de persona a máquina, se utiliza para la comunicación de máquina a máquina, más específicamente para transferir formatos de archivos legibles por máquina, como XML (*Extensible Markup Language*, Lenguaje de Marcado Extensible) y JSON (*JavaScript Object Notation*, Notación de Objetos de Java Script).

Muchas organizaciones que proporcionan datos en páginas HTML formateadas también proporcionan esos datos en su servidor como XML o JSON, a menudo a través de un servicio web para permitir la sindicación. Otra aplicación ofrecida al usuario final es un *mashup*, donde un servidor web consume varios servicios web de diferentes proveedores y compila el contenido en una interfaz de usuario [4].

1.1.2.1 Servicios web asincrónicos JAX-WS

JAX-WS (API Java™ para servicios web XML) proporciona soporte para invocar servicios web mediante una invocación de cliente asíncrona. Si se invocan los servicios web de forma asíncrona, JAX-WS ofrece soporte tanto para el modelo de retorno de llamada como para el modelo de sondeo. Ambos modelos, el de retorno de llamada y el de sondeo, están disponibles en el cliente Dispatch y en el cliente Dynamic Proxy. [5].

1.1.2.2 Servicios REST

Representational State Transfer (REST) es una arquitectura que se ejecuta sobre HTTP (*Hypertext Transfer Protocol*, protocolo de transferencia de hipertexto).

El W3C establece el REST como una característica distintiva clave de los servicios web que cumplen con REST, en los cuales el propósito principal del servicio fue, en su origen, manipular representaciones XML de recursos web usando un conjunto uniforme de operaciones "sin estado"; y servicios web arbitrarios, en los que el servicio expone un conjunto arbitrario de operaciones. En la actualidad se prefiere JSON [6].

1.1.3 Ontologías

Es una especificación formal y explícita de una conceptualización compartida según [7], y toma una variedad de formas, pero necesariamente incluirá un vocabulario de términos y algunas especificaciones de su significado. Esto incluye definiciones y una indicación de cómo los conceptos están relacionados entre sí, que imponen en conjunto una estructura sobre el dominio y limitan las posibles interpretaciones de términos.

1.1.3.1 Lenguaje de Ontologías Web

El Lenguaje de Ontologías Web (OWL) es una familia de lenguajes de representación del conocimiento para la creación de ontologías. Las ontologías son una manera formal de describir taxonomías y redes de clasificación, definiendo esencialmente la estructura del conocimiento para varios dominios: los sustantivos que representan clases de objetos y los verbos que representan las relaciones entre los objetos. Las ontologías se asemejan a las jerarquías de clases en la programación orientada a objetos, pero hay varias diferencias críticas. Las jerarquías de clase están destinadas a representar las estructuras utilizadas en el código fuente que evolucionan con bastante lentitud (por lo general, revisiones mensuales) mientras que las ontologías están destinadas a representar información en Internet y se espera que evolucionen casi constantemente. Del mismo modo, las ontologías suelen ser mucho más flexibles, ya que están destinadas a representar información en Internet procedente de todo tipo de fuentes de datos heterogéneos. Por otro lado, las jerarquías de clases están destinadas a ser bastante estáticas y dependen de fuentes de datos mucho menos diversas y más estructuradas, como las bases de datos corporativas [8].

1.1.3.2 Esquema RDF

El esquema RDF (*Resource Description Framework*, Marco de Descripción de Recursos) se refiere a un lenguaje de representación declarativo influenciado por las ideas de la representación del conocimiento. A fines de la década de 1990, la actividad de metadatos del *World Wide Web Consortium* (W3C) comenzó a trabajar en esquema RDF (RDFS), un lenguaje para el intercambio de vocabulario RDF. El RDF se convirtió en una Recomendación W3C en febrero de 1999, y RDFS en una Recomendación candidata en marzo de 2000. En febrero de 2001, la Actividad de la Web Semántica reemplazó la Actividad de Metadatos. En 2004 (como parte de una revisión más amplia de RDF) RDFS se convirtió en una Recomendación W3C. Aunque RDFS proporciona cierto soporte para la especificación ontológica, la necesidad de un lenguaje de ontología más expresivo se ha vuelto clara [9].

1.1.3.3 Almacén de tripletas RDFS

Un almacén de tripletas [10] es una base de datos construida para el guardado y obtención de datos a través de consultas semánticas. Una entidad de datos está compuesta de sujeto, predicado y objeto. Tal como en una base de datos relacional, uno guarda la información en un almacén de tripletas y consultarla vía lenguaje de consultas. A diferencia de las bases de datos relacionales, están optimizadas para el guardado y consulta de tripletas, donde una tripleta es importada o exportada usando un marco de descripción de recursos y otros formatos.

1.1.3.4 FOAF

FOAF (*Friend Of A Friend*, Amigo de un Amigo) es un vocabulario definido por medio del cual se describe el mundo de acuerdo a propiedades, utiliza la sintaxis RDF/XML o RDFa construyendo la información mediante una representación en tripletas que se organiza en sujeto, predicado y objeto [11].

1.1.4 Aprendizaje Automático

El aprendizaje automático es el estudio científico de algoritmos y modelos estadísticos que utilizan los sistemas informáticos para realizar una tarea específica sin usar instrucciones explícitas, sino que se basan en patrones e inferencia. Es visto como un subconjunto de inteligencia artificial [12].

1.1.4.1 Máquina de Soporte Vectorial

Una SVM es un clasificador discriminativo definido formalmente por un hiperplano de separación. En otras palabras, dados los datos de entrenamiento etiquetados (aprendizaje supervisado), el algoritmo genera un hiperplano óptimo que categoriza nuevos ejemplos [13].

1.1.5 Lenguaje de Consultas

Un lenguaje para la especificación de procedimientos para la recuperación (y algunas veces también modificación) de información de una base de datos [14].

1.1.5.1 SPARQL

SPARQL (*SPARQL Protocol y RDF Query Language*, Protocolo y Lenguaje de Consulta RDF SPARQL) es un lenguaje de consulta RDF, es decir, un lenguaje de consulta semántico para bases de datos, capaz de recuperar y manipular datos almacenados en formato de Marco de Descripción de Recursos [15].

1.1.5.2 SPARUL

SPARUL, o SPARQL/*Update*, es un lenguaje para actualizar contenido en RDF, similar al lenguaje SPARQL que es usado para hacer consultas a datos en RDF. La especificación ha sido desarrollada por Hewlett-Packard y actualmente no es un estándar reconocido [16].

1.1.5.3 TDB

Una base de datos transaccional es un DBMS (*Database Management System*, Sistema de Administración de Bases de Datos) que tiene la capacidad de revertir o deshacer una transacción u operación de la base de datos si no se completa adecuadamente [17].

1.2. Planteamiento del problema

En la actualidad, los dispositivos móviles ofrecen un gran número de aplicaciones con las cuales los usuarios solucionan diversos problemas de una forma rápida y eficaz, esto conlleva a la constante transformación de las estrategias de enseñanza dentro y fuera del ambiente escolar, donde es necesario que los docentes se mantengan actualizados para incrementar su eficiencia de acuerdo con el avance de cada uno de sus alumnos.

De lo anterior deriva el uso cada vez mayor de software y tecnologías Web en la labor docente, apoyándose la mayoría en herramientas de transferencia y almacenamiento de datos como *Dropbox*, y servicios de correo electrónico como *G-Mail* y *Outlook*, por mencionar algunos, por otra parte, con base en el *Plan de Desarrollo 2011-2015* de la UNAM (Universidad Nacional Autónoma de México) [18], se puso en marcha un proyecto que consiste en dotar a cada estudiante de nivel medio superior con tabletas electrónicas incluyendo contenidos educativos, como libros de texto, cursos interactivos y aplicaciones didácticas. Los equipos contarían además con acceso a exámenes de autoevaluación vía Internet y en esas pruebas los alumnos podrían calificar su desempeño, reconocer sus deficiencias, y a partir de éstas, recibir información con recomendaciones para regularizarse.

Todas estas propuestas no tuvieron los resultados esperados y por ello no hubo un mayor seguimiento. La ventaja con la que se cuenta actualmente corresponde a la rápida evolución de los dispositivos móviles, los que además producen una baja en los precios de modelos anteriores haciéndolos mucho más accesibles a las economías media y media-baja. Esto en pocas palabras pone un dispositivo móvil (en específico un *Smart Phone*) en las manos del 57% de la población mexicana, lo que permite ver un nuevo panorama que ha sido ignorado frecuentemente por las instituciones educativas, donde un *Smart Phone* es una herramienta de gran utilidad para diversas actividades escolares, además de que por ser una herramienta que los usuarios utilizan diariamente, la adaptación y comprensión de las aplicaciones móviles es cada vez más rápida.

Si bien esto propone un futuro exitoso en el desarrollo de la educación, la realidad es que en México el uso de los dispositivos móviles dentro del aula está prohibido debido a que se les emplea con un fin ajeno al de los objetivos de la clase, convirtiéndolos en un distractor en lugar de una herramienta de aprendizaje. Otra de las problemáticas dentro del aula es la dificultad de detectar deficiencias académicas de los alumnos a tiempo (antes de una evaluación con valor en la calificación), situación que se manifiesta constantemente en aulas con un número mayor a 15 alumnos (habiendo en México escuelas con más de 40 estudiantes por aula), pues los docentes no tienen tiempo suficiente para asegurarse de que todos han comprendido el tema explicado en clase. Por esto se propone realizar una aplicación Web y otra móvil que permita a los docentes realizar actualizaciones en tiempo real de su grupo, para así no perder de vista ningún detalle en relación con el cumplimiento académico de cada estudiante, además de consultar información relevante para analizar la situación del alumno e implementar actividades que le permitan mejorar su rendimiento.

1.3. Objetivos General y específicos

A continuación, se describen los objetivos general y específicos.

1.3.1. Objetivo General

Aplicar técnicas de web semántica en un entorno Web para el análisis de los perfiles de los alumnos en alguna institución que permita el enlace a una aplicación móvil en el sistema operativo Android para la consulta e ingreso de información utilizando SPARQL para realizar las operaciones necesarias.

1.3.2. Objetivos específicos

- Analizar el funcionamiento de los servicios Web para su correcta implementación dentro del proyecto.
- Analizar las características de los servicios Web REST.
- Análisis del vocabulario FOAF aplicado en un esquema RDF.
- Diseñar una arquitectura adecuada para la aplicación propuesta.
- Realizar la aplicación Web.
- Realizar la aplicación Android.
- Aplicar el software con un caso de estudio específico.

1.4. Justificación

Del problema presentado anteriormente se deriva la necesidad de contar con una plataforma que permita a una institución educativa analizar los datos de sus alumnos para implementar las estrategias de enseñanza más adecuadas de acuerdo con las necesidades de cada uno de ellos y que además contemplan el tiempo fuera del aula, por ejemplo, el coordinador académico puede saber cuáles son los alumnos con mayor tendencia a reprobado los exámenes o qué grupo tiene el mejor desempeño y, con base en la información obtenida el coordinador y los docentes, por su parte, implementar un plan de apoyo, asesorías o la planeación de algún evento escolar. El profesor podrá localizar en todo momento aquellos alumnos que están en riesgo de reprobado su materia gracias a las notificaciones que recibe en su dispositivo móvil provenientes del proveedor de servicio Web, así como enviar mensajes automáticamente por razón de incumplimiento o mala conducta por medio de correo electrónico a los padres de familia en caso de contar con su información de contacto, permitiendo una comunicación constante entre el personal escolar y los padres de familia que no requerirá al docente invertir tiempo extra para concretar estas tareas.

Capítulo 2. Estado de la Práctica

Los proyectos tecnológicos que buscan impulsar la educación son muchos y muy variados, pues se plantean con respecto a las necesidades de cada lugar donde se implementan. Esto genera la necesidad de ajustarlos a cada institución educativa pues, incluso en una misma ciudad, las escuelas implementan diferentes estrategias de aprendizaje, no obstante, pertenecen a un mismo sistema educativo. En este capítulo se describen diferentes trabajos que abordan varios aspectos de la implementación del *e-learning*, también se analizó el contenido de artículos que presentan la aplicación de ontologías mediante FOAF y servicios web pues los resultados son de gran utilidad para este proyecto. Posteriormente se realiza una síntesis por medio de una tabla y finalmente se concluye con los hallazgos obtenidos mediante el análisis de cada uno de los trabajos.

2.1. Trabajos relacionados

El *e-learning* ha transformado la educación tradicional en un aprendizaje más flexible y eficiente en los países desarrollados. Sin embargo, el *e-learning* sigue siendo poco utilizado y además está en etapas tempranas de implementación en los países en desarrollo. Por lo tanto, entender los factores críticos detrás de la adopción y aceptación de la tecnología es una preocupación principal en países en desarrollo como Pakistán. En [19] se proporcionó y se examinó la línea base de adopción y aceptación para los sistemas de *e-learning* mediante la incorporación de factores externos críticos en el modelo de aceptación de la tecnología. Se propuso un modelo conceptual, el Modelo de Adopción de *E-Learning* de Pakistán, en el contexto de la educación superior. Se recogieron datos de 354 estudiantes de la Universidad Virtual de Pakistán y se emplearon modelos de ecuaciones estructurales para probar las hipótesis de investigación. La investigación empírica indicó que la autoeficacia de la computadora, la experiencia de Internet, el disfrute y las características del sistema son predictores significativos de la facilidad de uso percibida, mientras que las características del sistema son un fuerte predictor de la utilidad percibida. Además, el elemento subjetivo no se identificó como una utilidad percibida. Los hallazgos proporcionaron implicaciones prácticas para los responsables de las políticas, los profesionales y los desarrolladores en la implementación exitosa de sistemas de aprendizaje electrónico.

La educación basada en la competencia se desarrolló para cerrar la brecha entre lo que la universidad está produciendo y los requisitos de la industria. Por lo tanto, es necesario desarrollar herramientas que ayuden en el proceso de análisis. En [20] se propuso un sistema que modela las competencias requeridas por las ocupaciones la industria y el plan de estudios

de la educación superior y ayuda a combinar perfiles de los dos dominios. Los diferentes conceptos en el dominio fueron modelados como una ontología web, y un motor de inferencia de la información permite el emparejamiento de los mismos. Además del emparejamiento de perfiles, el sistema calcula un puntaje para el grado coincidente utilizando el método de proceso de jerarquía analítica (AHP).

En [21], combinaron y personalizaron FOAF y la ontología musical para el sistema de recomendaciones de conciertos de música en la aplicación de Facebook para permitir a los usuarios recuperar información filtrada de conciertos de diferentes fuentes. La información se filtra según el gusto musical del usuario definido en su perfil de Facebook. En la investigación, ARC2 se utilizó como marco PHP basado en web semántica para RDF. Los resultados del experimento mostraron que el uso de la función incorporada de Virtuoso durante el proceso de búsqueda aceleró el tiempo de procesamiento para búsquedas de artistas similares en un 43 por ciento aproximadamente. Para acelerar aún más el tiempo de procesamiento, la base de datos se utiliza como un caché para que el rendimiento se incremente en un 23 por ciento aproximadamente.

El volumen de contenido para adultos en la red mundial está aumentando rápidamente. Esto hace que la detección automática de contenido para adultos sea una tarea más desafiante, cuando se elimina el acceso a sitios web inadecuados. La mayoría de los sistemas de filtrado de páginas web pornográficas están basados en n-gramas, *Naive Bayes*, *K-nearest neighbor* y mecanismos de concordancia de palabras clave, que no proporcionan una extracción perfecta de datos útiles del contenido web no estructurado. Estos sistemas no tienen la capacidad razonable de filtrar de manera inteligente contenido web para clasificar páginas web médicas de páginas de contenido para adultos. Además, es fácil que los niños accedan a páginas Web pornográficas debido al contenido para adultos disponible en Internet. Todo esto crea un problema para los padres que desean proteger a sus hijos de un contenido inapropiado. Para resolver esto, Farman et al. [22] presentaron una SVM (*Support Vector Machine*, Máquina de Vectores de Soporte) y un sistema de conocimiento semántico basado en ontologías difusas para filtrar sistemáticamente el contenido web e identificar y bloquear el acceso a la pornografía. El sistema propuesto clasifica las URL en URLs para adultos y URL médicas mediante el uso de una lista negra de páginas web censuradas para proporcionar precisión y velocidad. La ontología difusa propuesta luego extrae contenido web para encontrar el tipo de sitio web (contenido para adultos, normal y médico) y bloquear el contenido pornográfico. Con el fin de examinar la eficiencia del sistema propuesto, la ontología difusa y las herramientas inteligentes se desarrollaron utilizando Protégé 5.1 y Java, respectivamente. El análisis experimental mostró que el rendimiento del sistema propuesto es eficiente para detectar y bloquear automáticamente el contenido para adultos.

En [23] se presentó un enfoque para usar servicios web semánticos en la gestión de procesos de producción. En particular, los dispositivos en los sistemas de producción tienen en cuenta la interfaz de servicio a través de la cual se controlan, mientras que las descripciones semánticas del servicio web formuladas en OWL-S (*Ontology Web Language for Service*, Lenguaje de Ontologías Web para Servicios) hacen posible determinar las condiciones y los efectos de invocar los servicios web. El enfoque involucró tres servicios web que cooperan para alcanzar los objetivos de producción usando el dominio de estos. En particular, uno de los tres servicios mantiene el modelo semántico del estado actual del sistema, mientras que otro utiliza el modelo para componer los servicios web de dominio para que logren conjuntamente los objetivos deseados. El modelo semántico del sistema se actualiza automáticamente en función de las notificaciones de eventos enviadas por los servicios del dominio.

Durante la última década, los sistemas de *e-learning* personalizados y los sistemas adaptativos de hiper-medios educativos atrajeron mucha atención de los investigadores en los campos de la informática y la educación. Yang et al. [24] afirmaron que la integración de estilos de aprendizaje en un sistema inteligente es una posible solución a los problemas de "desviación de aprendizaje" y "sobrecarga cognitiva". En su estudio, propusieron un método de predicción de estilo de aprendizaje basado en una técnica de reconocimiento de patrones. Las principales contribuciones de este método son: (1) es una forma de *middleware* que se aplica a otros sistemas de tutoría inteligente, y (2) procesa datos dependientes del tema para hacer predicciones y actualizar los perfiles del estilo de aprendizaje de forma recursiva. Las evaluaciones experimentales demostraron la efectividad de este método de predicción.

El uso de videoconferencias en aprendizaje a distancia implica los dos principales problemas de búsqueda y participación activa del usuario. En [25], se describió la implementación y el uso de una funcionalidad colaborativa de anotación de video educativo para superar estos dos desafíos. Se explican diferentes casos de uso y requisitos, así como detalles de la implementación. Grünwald y Meinel sugieren otras mejoras para fomentar una cultura de participación y un algoritmo para la extracción de datos semánticos. Finalmente, se presentan evaluaciones en forma de pruebas de usuario y cuestionarios en un entorno MOOC (*Massive Open Online Course*, Curso Masivo Abierto en Línea). Los resultados de la evaluación son prometedores, ya que indicaron no solo que los estudiantes lo perciben como útil, sino también que la efectividad del aprendizaje aumenta. La combinación de anotaciones de video de conferencias personales con un mapa temático semántico también se evaluó positivamente y se investigará más a fondo, al igual que la implementación en un contexto de MOOC.

En [26] se desarrolló un sistema de producción de material de aprendizaje por Internet basado en QR (QR-ULMPS) que proporciona a los docentes una herramienta educativa para motivar a

los estudiantes de nivel universitario matriculados en un curso de artes liberales. QR-ULMPS fue específicamente diseñado para apoyar el desarrollo de materiales de *u-learning* (Aprendizaje Ubicuo) y crear un entorno de aprendizaje atractivo para los estudiantes. Se utilizó un diseño de investigación experimental para evaluar la efectividad general de QR-ULMPS; la UTAUT (*Unified Theory of Acceptance and Use of Technology*, Teoría Unificada de la Aceptación y el Uso de la Tecnología) evaluó la viabilidad de usar QR-ULMPS para implementar actividades de enseñanza; mientras que la IMMS (*Instructional Materials Motivation Survey*, Encuesta de Motivación de Materiales de Instrucción) se usó para medir la motivación de aprendizaje de los estudiantes después de usar el sistema de *u-learning* propuesto. A partir de los resultados del cuestionario UTAUT, encontraron que los maestros calificaron positivamente el sistema y estaban dispuestos a aceptar y adoptar QR-ULMPS en el contenido de su curso. Los docentes también estuvieron de acuerdo en que QR-ULMPS era una herramienta útil para motivar el aprendizaje de los estudiantes durante las actividades de enseñanza al aire libre. Además, los resultados del cuestionario IMMS indicaron que los estudiantes asignados al sistema de aprendizaje propuesto lograron mejores resultados que los participantes que aprendieron a través de métodos convencionales. El sistema de *u-learning* propuesto es ventajoso porque mejora la motivación del estudiante y permite niveles más altos de participación, particularmente durante las actividades de aprendizaje al aire libre. Por lo tanto, el sistema de aprendizaje propuesto creó una experiencia de aprendizaje que suele interesar y atraer a los estudiantes. A pesar de que QR-ULMPS no es lo suficientemente maduro como para ser utilizado en varios dominios educativos, ofrece una oportunidad innovadora para que los docentes integren una metodología de enseñanza novedosa que desafía las normas educativas tradicionales.

Respecto al uso de dispositivos móviles para la obtención de datos en tiempo real, Zhu e Iglesias [27] establecieron que utilizando tecnología avanzada se monitorizan automáticamente las actividades para prevenir y controlar las enfermedades crónicas, en los últimos años, un número cada vez mayor de personas comenzó a preocuparse por su salud. Las enfermedades crónicas están relacionadas con el estilo de vida, y los registros diarios de actividad de diferentes usuarios se relacionan y utilizan como un indicador importante de la salud. Los datos utilizados en su documento se obtuvieron a partir de sensores de aceleración y giroscopios integrados en teléfonos inteligentes. Diseñaron un Adaboost-Stump eficiente que se ejecuta en un teléfono inteligente para clasificar cinco actividades de comunicación: ciclismo, correr, sentarse, pararse y caminar y obtener una precisión de clasificación satisfactoria del 98%. También un método de aprendizaje en línea, y el modelo de clasificación es requerido para la actualización continua con datos reales. Los parámetros en el modelo se adaptan cada vez más al usuario específico, lo que hace que la precisión de clasificación llegue al 95% en diferentes entornos de uso. Además, este documento también utilizó el marco OpenCL para diseñar el programa en paralelo. Este proceso es capaz de mejorar la eficiencia de la computación aproximadamente nueve veces.

En [28], se presentó un método para medir la similitud semántica entre conceptos en KGs (*Knowledge Graphs*, Gráficos de Conocimiento) como WordNet y DBpedia. El trabajo previo sobre métodos de similitud semántica se ha centrado en la estructura de la red semántica entre conceptos (por ejemplo, longitud y profundidad de la ruta), o solo en el contenido de información de conceptos (IC). Zhu e Iglesias propusieron un método de similitud semántica, llamado *wpath*, para combinar estos dos enfoques, usando IC para ponderar la longitud de ruta más corta entre los conceptos. El IC convencional basado en corpus se calcula a partir de la distribución de conceptos sobre el corpus textual, que se requiere para preparar un corpus de dominio que contenga conceptos anotados y que tenga un alto costo computacional. Como las instancias ya se extrajeron del corpus textual y se anotaron mediante conceptos en KG, se propuso el IC basado en gráficos para calcular el IC en función de las distribuciones de conceptos sobre instancias. A través de experimentos realizados en conjuntos de datos de similitud de palabras bien conocidos, mostraron que el método de similitud semántica de sendero produjo una mejora estadísticamente significativa sobre otros métodos de similitud semántica. Además, en una evaluación de clasificación de categoría real, el método *wpath* mostró el mejor rendimiento en términos de precisión y puntaje F.

El objetivo principal de los sistemas de *e-learning* es mejorar el rendimiento y la satisfacción del aprendizaje del alumno mediante el uso de las TICs. Esto se logra proporcionando una experiencia de aprendizaje personalizada que identifique y satisfaga los requisitos y habilidades del alumno individual. El rendimiento de los sistemas de *e-learning* mejoran significativamente mediante la explotación de capacidades dinámicas de autoaprendizaje que se adaptan rápidamente a las interacciones previas del usuario dentro del sistema y los continuos cambios en el entorno. Al-Tarabily et al. [23] propusieron un sistema dinámico de agentes múltiples que usa optimización de enjambre de partículas para los sistemas de aprendizaje electrónico. El sistema incorpora cinco agentes que toman en consideración las variaciones en las capacidades entre los diferentes usuarios. En primer lugar, el agente de agrupamiento de proyectos se utiliza para agrupar un conjunto de recursos / proyectos de aprendizaje en grupos similares. Segundo, el agente de agrupamiento de estudiantes (SCA) agrupa a los estudiantes de acuerdo con sus preferencias y habilidades. En tercer lugar, el agente de correspondencia estudiante-proyecto se utiliza para asignar el grupo de cada alumno a un proyecto adecuado o recursos de aprendizaje particulares de acuerdo con criterios de diseño específicos. Cuarto, el emparejamiento estudiante-alumno está diseñado para cumplir el mapeo eficiente entre los estudiantes independientes. Finalmente, el SCA dinámico (DSCA) se emplea para rastrear y analizar continuamente el comportamiento del estudiante dentro del sistema, como los cambios en los niveles de conocimiento y habilidad. En consecuencia, el DSCA adapta los entornos de *e-learning* para acomodar estas variaciones. Los resultados experimentales demostraron la efectividad del sistema propuesto para proporcionar soluciones casi óptimas en un tiempo computacional considerablemente menor a las estrategias existentes de autoaprendizaje.

El crecimiento sin precedentes de Internet, su accesibilidad generalizada y su facilidad de uso han aumentado las dependencias de los estudiantes en la Web para una búsqueda y recuperación rápida de recursos de aprendizaje. Sin embargo, los motores de búsqueda actuales tienden a confiar en las palabras clave correctas. Esto excluye otras características, como la capacidad de aprendizaje del individuo y su preparación para materiales de aprendizaje específicos. Como resultado, el mismo conjunto de palabras clave de búsqueda ofrece los mismos resultados de búsqueda. Esta situación dificulta la optimización de los motores de búsqueda en la Web para apoyar la heterogeneidad de sus usuarios en sus esfuerzos de aprendizaje. Rahman y Abdullah [29] intentaron aumentar la capacidad de los motores de búsqueda de Internet con recomendaciones personalizadas de resultados de búsqueda que combinan competencias y conductas de aprendizaje de los estudiantes. Los resultados obtenidos de los experimentos mostraron que proporcionan una mejora notable en términos de rendimiento y satisfacción para los estudiantes.

Con la proliferación de la tecnología, los programas de *e-learning* han llamado la atención de varias instituciones. Esto permite a los usuarios de alrededor del mundo aprender a acceder a nueva información y al mismo tiempo incrementar la información existente sumándose a la creciente cantidad de datos recopilados que ya se están generando a través de diferentes dispositivos y sensores empleados en todo el mundo. Esto llevó a la necesidad de analizar datos recopilados y extraer información útil de ellos. El ML (*Machine Learning*, Aprendizaje Automático) y el análisis de datos (DA) son técnicas propuestas que ayudan a extraer información y encontrar patrones valiosos dentro de los datos recopilados. En [30] el campo del *e-learning* se investigó en términos de definiciones y características. Además, se discutieron los diversos desafíos que enfrentan los diferentes participantes dentro de este proceso. Luego, se ofreció una breve descripción de las técnicas más populares de ML y DA. Finalmente, se propusieron algunas de las investigaciones disponibles que emplean tales técnicas para dar una idea de las áreas que merecen una mayor exploración e investigación.

A diferencia del análisis de información de datos de bajo nivel del usuario, como la información FOAF (*Friend of A Friend*, Amigo de un Amigo) básica: nombre, correo electrónico, dirección, entre otras, el análisis semántico convierte esa información de bajo nivel en una representación de conocimiento de alto nivel. Combinado con el usuario, el archivo FOAF y el mecanismo de representación del conocimiento semántico forman el marco de información unificada del usuario. A través del análisis semántico FOAF, el empresario no solo recomienda productos de manera eficiente de acuerdo con la información de registro estático del usuario, sino también reutiliza esta información rápida y fácilmente, dándose cuenta de la minería de intereses del usuario y del nuevo grupo de usuarios encontrado. En [31] se planteó que el análisis semántico de FOAF es la clave para resolver estos dos problemas "los usuarios interesados en la minería para la recomendación de productos" y "los amigos del usuario - nuevo grupo de usuarios

encontrado". En la recomendación del usuario y la administración de usuarios, se clasificó al usuario FOAF basado en SVM.

2.2 Análisis comparativo

Artículo	Problema	Tecnologías	Resultado	Estado	Enfoque
[11]	Implementar servicios de web semántica en procesos de producción automáticos.	Procesos OWL-S, SPARQL, monitores de servicios.	Los lenguajes OWL-S, SPARQL y SPARQL-Update son altamente expresivos, se requiere de una conversión a un formato más adecuado como PDDL.	Se concluyó que el enfoque permitiría a los ingenieros de dominio modelar los sistemas y servicios web en los idiomas expresivos y ampliamente admitidos antes mencionados, al tiempo que permite la composición eficiente y automática de los servicios web.	Industrial
[12]	La implementación de los actuales modelos educativos genera desviación del aprendizaje y sobrecarga cognitiva.	Sistema de predicción de estilo de aprendizaje basado en Java.	Creación de un método de predicción de estilos de aprendizaje en dos etapas.	A futuro se utilizarán redes neuronales profundas para incrementar la precisión de los datos obtenidos de más de 1,000 muestras.	Educativo
[13]	Las video-clases como propuesta de tecnologías de la educación tienen problemas de capacidad de búsqueda y no permiten la participación activa.	Anotaciones públicas, individuales y de grupo, en vivo y a petición, interfaces de usuario.	Por medio de encuestas se encontró que este tipo de dinámica es del agrado de los estudiantes quienes sugirieron que se agreguen marcadores a los videos para buscar el punto de interés específico.	Se planea agregar un manuscrito que permita buscar notas mediante <i>time-stamps</i> y funciones de marcadores.	Educativo

Artículo	Problema	Tecnologías	Resultado	Estado	Enfoque
[19]	Incrementar los niveles de motivación de los estudiantes por medio de la producción de material dinámico QR.	Encuestas IMMS, cuestionarios UTAUT,	El sistema de producción de materiales <i>U-Learning</i> por medio de QR.	Se espera que posteriormente los profesores generen un ambiente de <i>U-Learning</i> de acuerdo al contexto que permita a los estudiantes obtener conocimientos adecuados durante actividades fuera del aula.	Educativo
[20]	Utilizar el sensor de aceleración y el giroscopio de los <i>smartphones</i> para monitorizar la rutina del usuario.	OpenCl Framework.	Se monitorizaron las actividades físicas: caminar, correr, sentarse, levantarse, andar en bicicleta. Se clasificaron con una precisión del 98%.	A futuro se buscará agregar la monitorización de más actividades físicas del usuario.	Salud
[21]	Medir la similitud semántica de conceptos en grafos de conocimiento. Otros trabajos sólo se enfocan en la red semántica de conceptos o el contenido informativo de los conceptos.	Un método de similitud llamado wpath que se enfoca en ambos conceptos.	El método wpath demostró ser mejor que los métodos de similitud semántica.	Concluido.	Pruebas.

Artículo	Problema	Tecnologías	Resultado	Estado	Enfoque
[22]	Semántica heterogénea en el uso de ontologías.	Un <i>framework</i> de cuatro pasos para la asociación de ontologías.	El <i>framework</i> tiene una precisión promedio de 99.93% generando equivalencias con las técnicas de asociación.	Concluido. Los proyectos a futuro están dirigidos a la escalabilidad del <i>framework</i> .	Pruebas.
[23]	Los países en desarrollo no están adoptando las tecnologías para la mejora de la educación.	Recopilación de datos.	Los datos empíricos mostraron que las características del sistema es un fuerte predictor de la utilidad que tiene el <i>e-learning</i> .	Investigación concluida.	Educativo.
[24]	El volumen de contenido para adultos en la <i>world wide web</i> se incrementa a gran velocidad y es difícil su detección certera para eliminar el acceso a sitios web de contenido inapropiado.	Máquina de vectores de soporte, un sistema de conocimiento semántico difuso, Protégé 5.1 y Java.	El sistema clasifica las URL como de adultos y médicas, luego la ontología difusa extrae el contenido; de adultos, normal y médico. Después bloquea el material pornográfico. Los análisis mostraron la eficacia de este sistema.	Concluido.	Seguridad.

Artículo	Problema	Tecnologías	Resultado	Estado	Enfoque
[25]	La educación superior no se adapta en la mayoría de los casos al desarrollo tecnológico.	Ontologías de web semántica, un motor de inferencias en una aplicación de escritorio y un método AHP.	El sistema captura los conceptos del dominio y realiza un análisis.	A futuro se hará que la aplicación sea más automática en términos de definición de perfiles y mapeo de competencias, que son las actividades realizadas por el experto del dominio.	Educativo.
[26]	Un entorno <i>e-learning</i> requiere un nivel considerable de personalización donde la información proporcionada esté relacionada con la capacidad e intereses del usuario.	Sistema dinámico multi-agente DMAPSO.	El sistema realiza agrupamientos rápidos y eficientes en comparación con PSO convencional.	Concluido.	Educativo.
[27]	Los motores de búsqueda en internet requieren palabras claves correctas para el acceso a la información.	<i>Web crawler, google search, ontologías difusas.</i>	Los usuarios se beneficiaron de la búsqueda personalizada con base en el resultado de encuestas realizadas posterior a la aplicación. Las limitaciones del sistema hacen que sea necesario pensar en una forma de motivar a los estudiantes a utilizar motores de búsqueda obtenidos por las instituciones educativas.	Concluido.	Educativo.

Artículo	Problema	Tecnologías	Resultado	Estado	Enfoque
[28]	Las fuentes de información están dispersas en todo el mundo, se requiere extraer y concentrar información útil.	Algoritmos ML.	La investigación abarcó varios métodos para demostrar las características del <i>e-learning</i> que pueden ser exploradas.	Concluido.	Educativo.
[29]	La información que se obtiene de un usuario por medio de las ontologías comunes sólo ofrece conocimiento superficial del mismo haciendo difícil la tarea de ofrecer mejores servicios.	Vocabulario FOAF y SVM.	Se clasificó y se obtuvo información más detallada del usuario FOAF y nuevos grupos amigos del usuario.	Concluido.	Comercial
[30]	Las sugerencias de páginas web no ofrecen sugerencias específicas relacionadas con los gustos de los usuarios.	PHP basado en web semántica para RDF.	Se obtuvieron datos relacionados con los gustos del usuario en un 43% más rápido utilizando la base de datos como un caché.	Concluido.	Comercial.

De lo anterior se concluye que, con respecto a la implementación del *e-learning* como siguiente paso en la evolución de los sistemas educativos, se hicieron y se continúan haciendo grandes esfuerzos para poner las tecnologías de la educación al alcance del mayor número de usuarios posible. En los trabajos analizados se observa que sí hay un gran avance con respecto a los nuevos métodos y sistemas propuestos, lo que deja en claro la gran capacidad del campo de trabajo dedicado a las tecnologías de la educación. También se consideran los factores para la realización de estos proyectos, como el desarrollo económico de los países que ya implementan ampliamente el *e-learning*, lo que sugiere algunos obstáculos para el desarrollo de este proyecto. Por parte de los algoritmos y métodos para el desarrollo comercial y de las tecnologías, se retomaron algunos de los pasos que aportaron los autores, pero cambiando el enfoque a uno

educativo, donde las pruebas y los resultados de llevarlas a cabo, incrementen la capacidad de los entornos educativos de ofrecer mejores planes a sus alumnos dentro y fuera del aula.

En esta sección se hace la comparación de las tecnologías y las metodologías propuestas para el desarrollo del proyecto, así como el plan de trabajo para el logro de esta propuesta.

2.3 Propuesta de solución

Se propone crear un sistema a partir de dos aplicaciones, una Web y una móvil en el sistema operativo de Android, donde la primera es la principal y se utiliza de forma independiente, además de fungir como proveedor de datos para la aplicación móvil, la cual es un complemento que permite consultar y modificar los datos existentes en los registros de la aplicación Web.

A continuación, se muestran las diferentes tecnologías utilizadas para el desarrollo del sistema propuesto en esta tesis.

2.3.1 Java

Java [31] es un lenguaje de programación de alto nivel orientado a objetos desarrollado por *Sun Microsystems*, empresa adquirida recientemente por *Oracle Corporation*. Algunas características de este lenguaje se describen a continuación.

- **Orientado a objetos.** Los objetos agrupan en estructuras encapsuladas tanto sus datos como sus métodos.
- **Distribuido.** Java proporciona clases para su uso en aplicaciones de red, lo que facilita la creación de aplicaciones distribuidas.
- **Robusto.** Proporciona numerosas comprobaciones en compilación y tiempo de ejecución. Se ha prescindido por completo de los punteros y elimina la necesidad de liberación explícita de memoria.
- **Portable.** Java es un lenguaje compilado y ejecutado y esto es lo que permite la portabilidad.
- **Seguro.** Se implementaron barreras de seguridad en el lenguaje y en el sistema de ejecución en tiempo real.
- **Independiente de la arquitectura.** El compilador de Java genera *bytecodes*, un formato diseñado para transportar el código eficientemente a múltiples plataformas hardware y software.
- **Multihilo.** Java soporta la sincronización de múltiples hilos de ejecución, especialmente útiles en la creación de aplicaciones de red distribuidas.
- **Dinámico.** Las clases solo se enlazan a medida que se necesitan.

2.3.2 Framework JavaServer Faces

JSF [32] es un *framework* basado en el API de *Servlets* que proporciona un conjunto de componentes en forma de etiquetas definidas en páginas XHTML mediante *Facelets*.

JSF proporciona las siguientes características destacables:

- Definición de las interfaces de usuario mediante vistas que agrupan componentes gráficos.
- Conexión de los componentes gráficos con los datos de la aplicación mediante los denominados *beans* gestionados.

- Conversión de datos y validación automática de la entrada del usuario.
- Navegación entre vistas.
- Internacionalización.

2.3.3 Framework PrimeFaces

PrimeFaces [33] es una biblioteca de componentes JSF de código abierto que cuenta con un conjunto de componentes enriquecidos que facilitan la creación de las aplicaciones Web.

Características:

- Conjunto de componentes ricos (Editor de HTML, autocompletar, cartas, gráficas o paneles, entre otros).
- Soporte de AJAX (“*Asynchronous JavaScript And XML*”, JavaScript asíncrono y XML) con despliegue parcial, lo que permite controlar qué componentes de la página actual se actualizarán y cuáles no.
- Componente para aplicaciones Web para teléfonos móviles, especiales para iPhones, Palm, Android y teléfonos móviles Nokia.

2.3.4 Framework Apache Jena

Apache Jena [34] es un marco de trabajo de Java para construir aplicaciones de Web semántica y datos relacionados.

Características:

- Crea y lee grafos RDF. Puede serializar las tripletas de datos utilizando los formatos populares como RDF/XML o Trurtle.
- Realiza consultas utilizando ARQ, un motor compilador de SPARQL 1.1. ARQ soporta búsquedas remotas federalizadas y búsquedas de texto libres.
- Cuenta con manejo especializado de almacén de tripletas TDB para un alto rendimiento y Fuseki para el manejo de los datos como un *end-point* accesible mediante http. Fuseki provee interacciones al estilo REST con los datos RDF.
- Interactúa con los modelos de Web semántica para la relación de los datos de un archivo RDF.

2.3.5 NetBeans

NetBeans [35] es un entorno de desarrollo integrado libre, desarrollado principalmente para el lenguaje de programación Java. Es un proyecto de código abierto con una gran base de usuarios, una comunidad en constante crecimiento.

Su plataforma permite desarrollar aplicaciones a partir de un conjunto de módulos, que se implementan independientemente. Las aplicaciones construidas a partir de estos se extienden agregándoles nuevos módulos.

Las aplicaciones son capaces de instalar módulos dinámicamente; e incluso algunas agregan un módulo de actualización para permitir a los usuarios descargar, a su vez, actualizaciones de firma digital y nuevas características directamente dentro de la aplicación en ejecución; reinstalando una actualización o una nueva versión, no obligando a los usuarios a descargar toda la aplicación de nuevo.

Esta plataforma ofrece servicios reutilizables comunes para las aplicaciones de escritorio, permitiendo a los desarrolladores centrarse en la lógica de sus aplicaciones. Entre sus características destacan: gestión de la interfaz de usuario (menús y barras de herramientas); gestión de configuración de usuario; gestión de almacenamiento; gestión de ventana; marco asistente; biblioteca visual de NetBeans y herramientas de desarrollo integrado.

Capítulo 3. Aplicación de la metodología

En este capítulo se muestra el desarrollo de la aplicación de acuerdo a las fases de la metodología UWE.

3.1 Análisis de requerimientos

Fija los requisitos funcionales de la aplicación web para reflejarlos en un modelo de casos de uso. Se adquieren, reúnen y especifican las características funcionales y no funcionales que deberán cumplir las aplicaciones web y móvil.

Se utiliza el diagrama de casos de uso para representar la interacción de los actores con la aplicación, identificando los requerimientos funcionales (Figura 3.1).

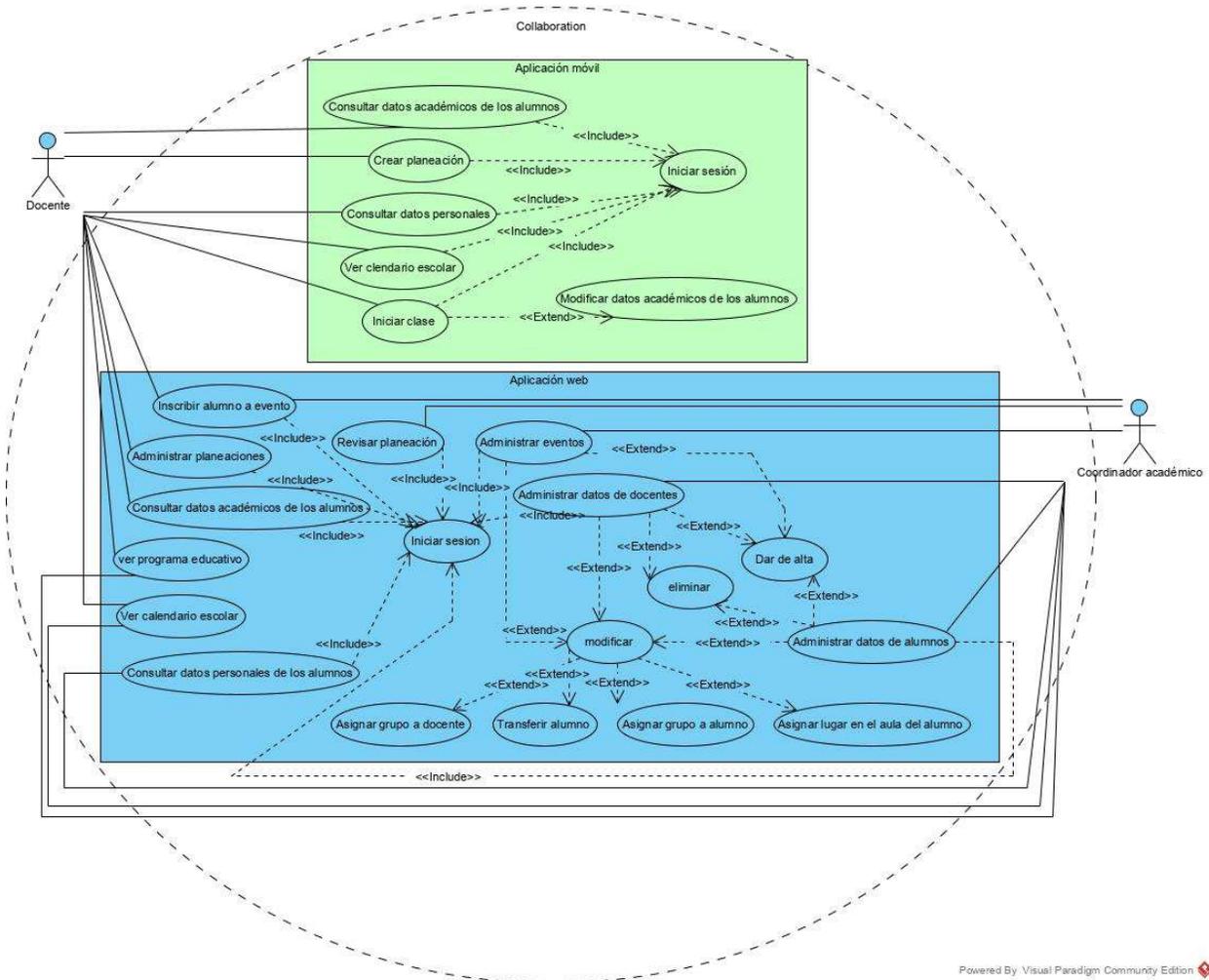


Figura 3.1 Diagrama de casos de uso para las aplicaciones Web y móvil

Los actores *Coordinador* y *Docente* deben autenticarse en el sistema para realizar sus operaciones, así mismo tendrán la opción de cerrar sesión su sesión en cualquier momento. El *Coordinador* es el encargado de administrar toda la información tanto de los alumnos como del personal docente por lo que tiene derecho de insertar nuevos datos en el sistema, modificar o eliminar cualquiera de los registros. También administra la información de los eventos escolares como festivales o concursos. Tiene acceso a las planeaciones creadas por los docentes; sólo puede consultar. El *Docente* tiene permiso de consultar toda la información académica de los alumnos y administrar sus propios planes de clase, crear nuevo, modificar o borrar existentes.

El actor *Docente* por medio de su teléfono inteligente tiene la opción de consultar el plan educativo utilizado por la institución donde labora, administrar los planes de clase; crear, modificar, eliminar. También puede configurar su cuenta para acceder a los servicios de la aplicación Web. En tiempo de clase llevará el control de su grupo mediante las actividades correspondientes a las establecidas en un plan de clase existente.

3.2 Diseño del sistema

Se basa en la especificación de requisitos producido por el análisis de requerimientos (fase de análisis), el diseño define cómo estos requisitos se cumplirán, al igual que la estructura que debe darse a la aplicación Web.

3.2.1 Diseño de la interfaz de usuario

Aquí se muestran los mockups correspondientes a la aplicación web y la aplicación móvil de este proyecto.

En el caso de la aplicación Web, se utiliza una pantalla principal a modo de matriz, donde el usuario accede a los diferentes sitios de la aplicación desde una sola ventana. En la figura 3.2.1 se observa un menú al centro de la ventana justo debajo del mensaje de bienvenida “Menú de Administrador”, el elemento que se utiliza corresponde a uno de los componentes de PrimeFaces llamado *MegaMenu*, el cual permite desplegar una lista en cada opción en la que se pasa el mouse por encima. Este elemento permite la integración de encabezados de sección, así como de imágenes para enriquecer el aspecto visual del menú de selección.

La página principal esta conformada por una plantilla, que mantiene a la vista del usuario la cabecera, el *MegaMenu* y el pie de página. El pie y cabecera de la página Web se utiliza para las opciones de regionalización, los avisos legales, la información de los términos y condiciones de uso, y si el usuario lo provee, vínculos a la página oficial de la institución educativa que utiliza la aplicación. Debajo del *MegaMenu* se encuentra la sección de *Contenidos* en la cual se mostrará la información en general para cualquiera de las opciones que elija el usuario.

Si el usuario pasa el puntero por encima de la sección de *Datos de los Alumnos* (Figura 3.2.2), el menú despliega una lista con las opciones; altas, bajas, modificaciones, revisar desempeño, revisar calificaciones. En el caso de que el usuario es un *Docente*, sólo la consulta de datos académicos de los alumnos está disponible. Para el manejo de la información de los docentes la funcionalidad es la misma. Las opciones de *Actividades Escolares* e *Información* están

disponibles para los dos perfiles, pero sólo el *Coordinador* tiene derecho a introducir actividades al calendario.

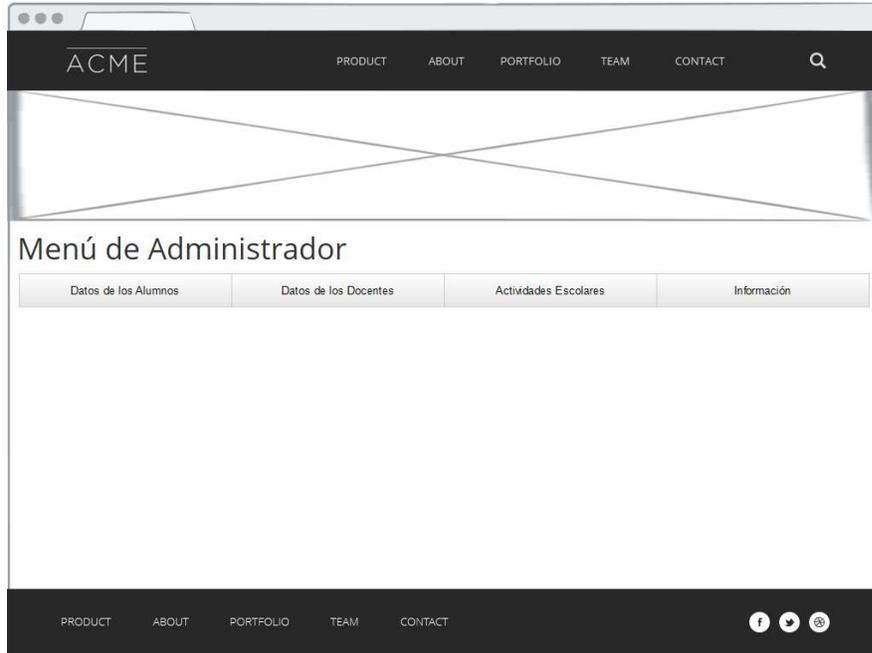


Figura 3.2.1 Diseño de la página principal de la aplicación Web

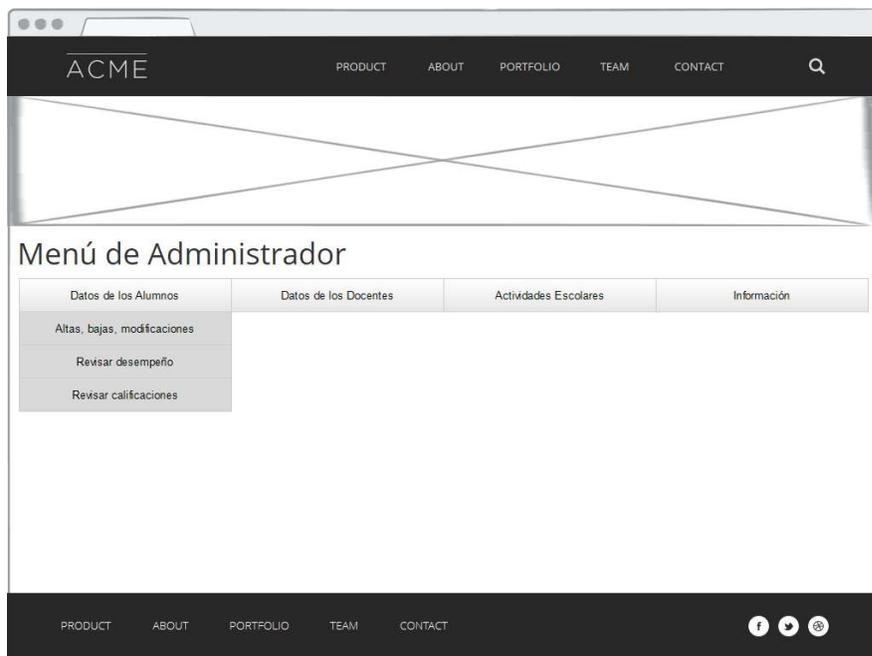


Figura 3.2.2 Menú de administrador desplegando la sección de Datos de los Alumnos

En la figura 3.2.3 se muestra ejemplo de formulario de captura, la información en general aparece en la división de *Contenidos* de la plantilla principal.

Figura 3.2.3 Ejemplo de formulario de captura para un nuevo alumno

Para la consulta de información de los alumnos los datos se muestran individuales o por grupo, en la figura 3.2.4 se muestra la distribución de alumnos dentro de un grupo seleccionado.

Figura 3.2.4 Ejemplo de Consulta de información de un grupo

En el caso de la aplicación móvil la disposición de los elementos debe ser simple para que el usuario pueda ubicarse rápidamente, puesto que la aplicación móvil es para utilizarse en tiempo real una clase. El esquema propuesto corresponde a no más de 3 botones en pantalla (sin contar

los botones por defecto del sistema operativo de Android) y un area de notificaciones en la parte inferior. En la figura 3.2.5 se muestra la disposición de varias pantallas de la aplicación móvil.

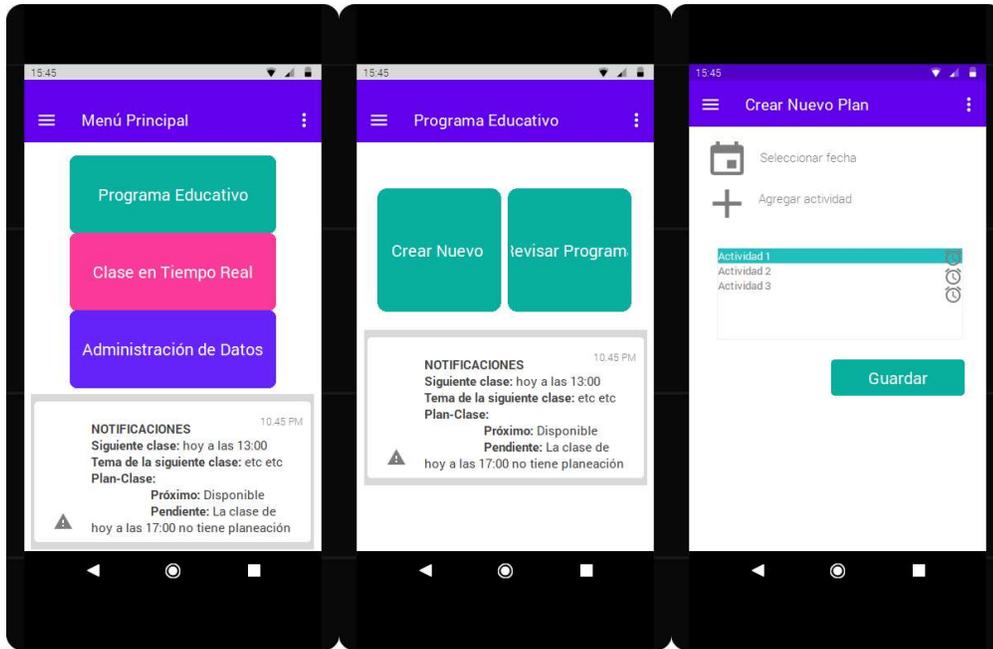


Figura 3.2.5 Ejemplo de la disposición de las pantallas de la aplicación móvil

3.2.2 Diagrama de paquetes

Aquí se muestra la organización de los paquetes de la aplicación Web y la aplicación móvil por medio de diagramas de paquetes.

3.2.2.1 Diagrama de paquetes de la aplicación Web

A continuación, se describe el contenido de los paquetes y la figura 3.3 muestra el diagrama.

- beans: Contiene todos los *beans* administrados para el uso del marco de trabajo de JavaServerFaces.
- control: Contiene formateadores y validadores.
- dao: Contiene la clase de conexión a los archivos RDF.
- filter: Contiene la clase para filtrar el acceso de los usuarios.
- model: Contiene las clases para el modelo de datos.
- rest: Contiene las clases relacionadas con los servicios REST.
- weka: Contiene la clase que utiliza la SVM.

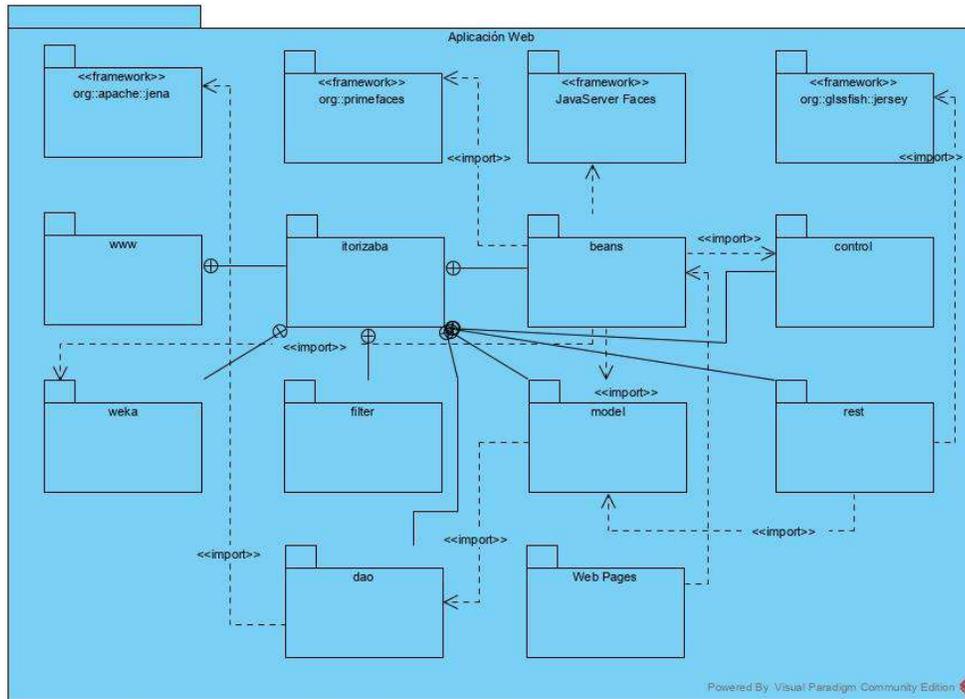


Figura 3.3 Diagrama de paquetes de la aplicación Web

3.2.2.2 Diagrama de paquetes de la aplicación móvil

En el caso de la aplicación móvil se utiliza la organización por defecto del IDE Android Studio, sólo se dividen los paquetes que contienen las clases de Java. Se muestra la distribución de los paquetes en la figura 3.3.1.

- activities: Contiene las clases que definen las actividades de la aplicación.
- adapters: Contiene las clases que se encargan de generar las listas para la visualización de los datos.
- gadgets: Contiene las clases necesarias para la generación de ventanas modales.
- model: Contiene las clases que representan el modelo de datos.
- rest: Contiene las clases necesarias para el uso de los servicios REST.
- res: Es el paquete por defecto en Android Studio para los archivos relacionados con las vistas.

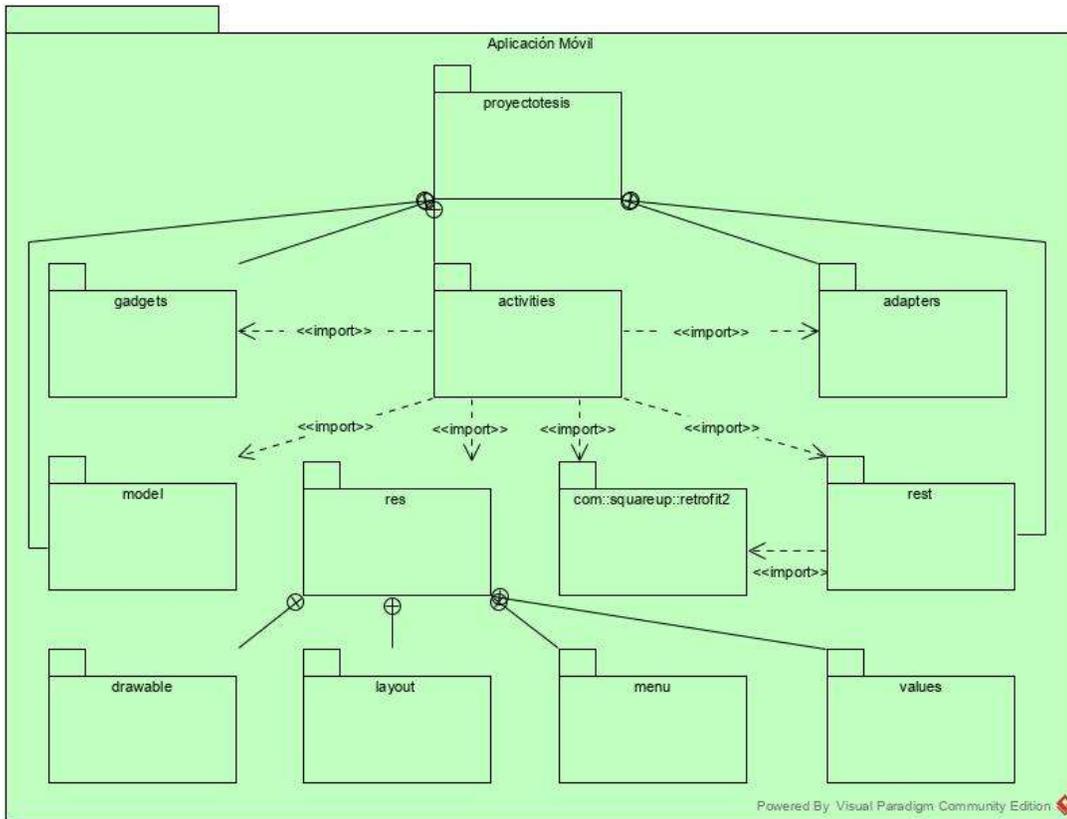


Figura 3.3.1 Diagrama de paquetes de la aplicación móvil

3.2.3 Diagramas de clases

Los diagramas de clases del sistema se muestran a continuación. Las clases aparecen de acuerdo al paquete en el que se encuentran.

3.2.3.1 Diagramas de clases correspondientes a la aplicación Web

Los siguientes diagramas muestran las clases organizadas en relación al paquete que las contiene. La figura 3.3.2 muestra el diagrama de las clases correspondientes al modelo, la figura 3.3.3 muestra las clases que se utilizan para filtrar la navegación del usuario, la figura 3.3.4 las clases correspondientes al manejo de sesión, validadores y convertidores, la figura 3.3.5 los beans administrados, la figura 3.3.6 la clase de acceso a datos y la figura 3.3.7 la máquina de soporte vectorial.

En la figura 3.3.8 se muestran las clases encargadas de los servicios REST, los métodos con los cuales se establece el enlace entre la aplicación Web y la móvil.

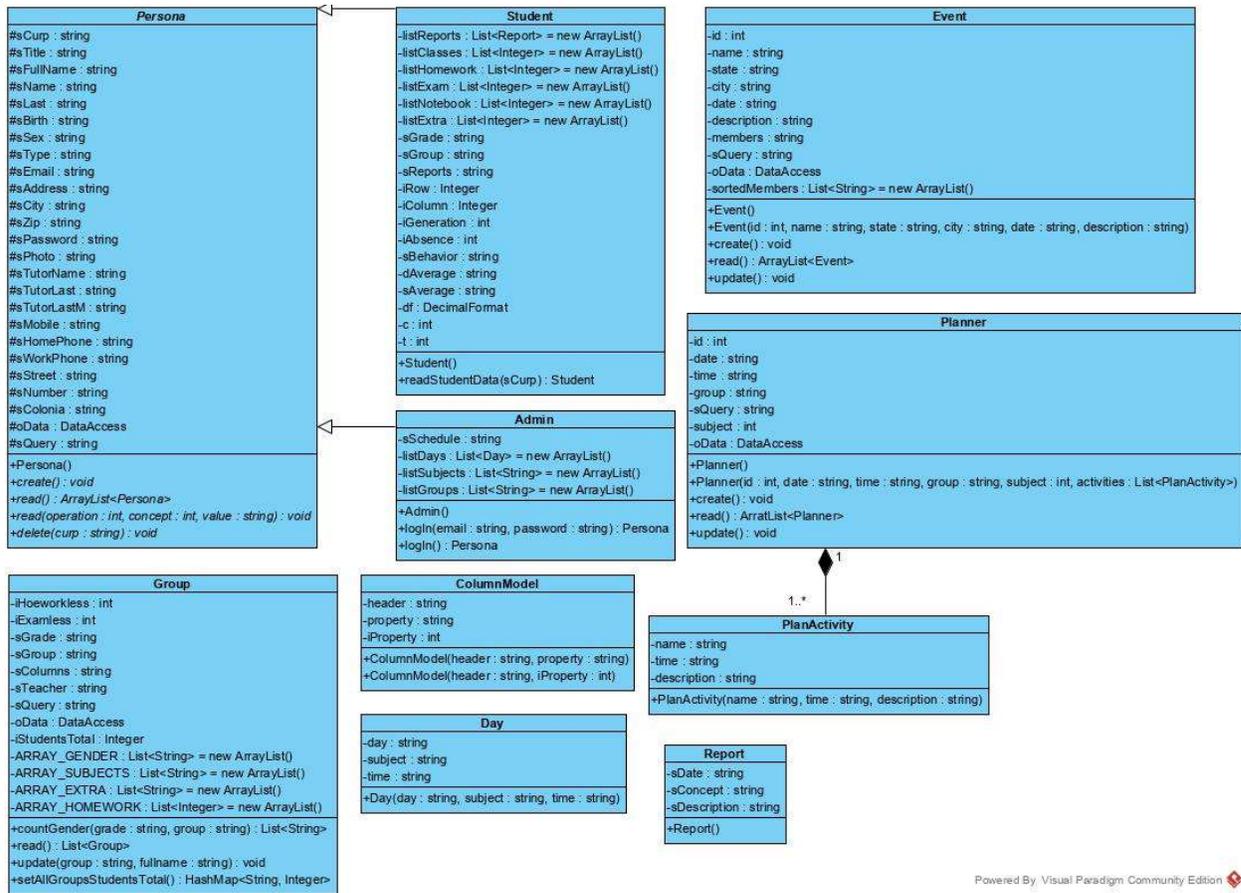


Figura 3.3.2 Diagrama de clases correspondiente al paquete *model*



Figura 3.3.3 Diagrama de clases correspondiente al paquete *filter*

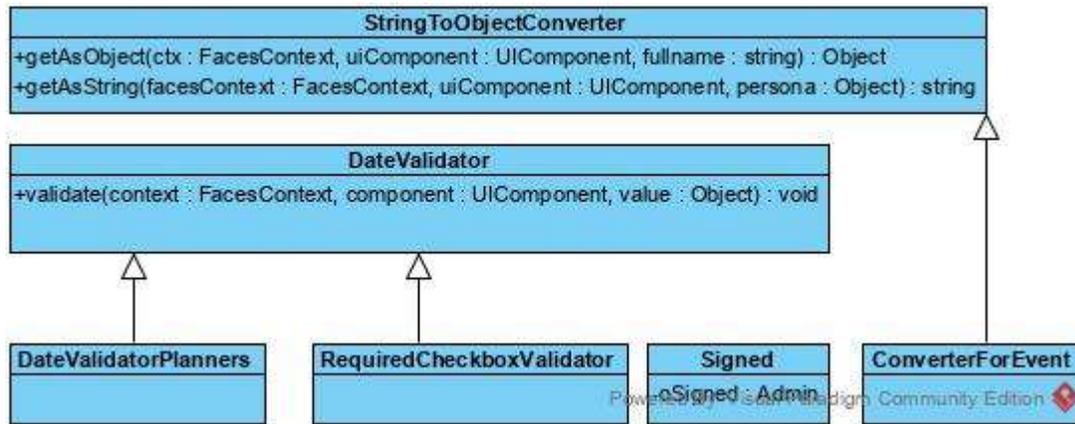


Figura 3.3.4 Diagrama de clases correspondiente al paquete control

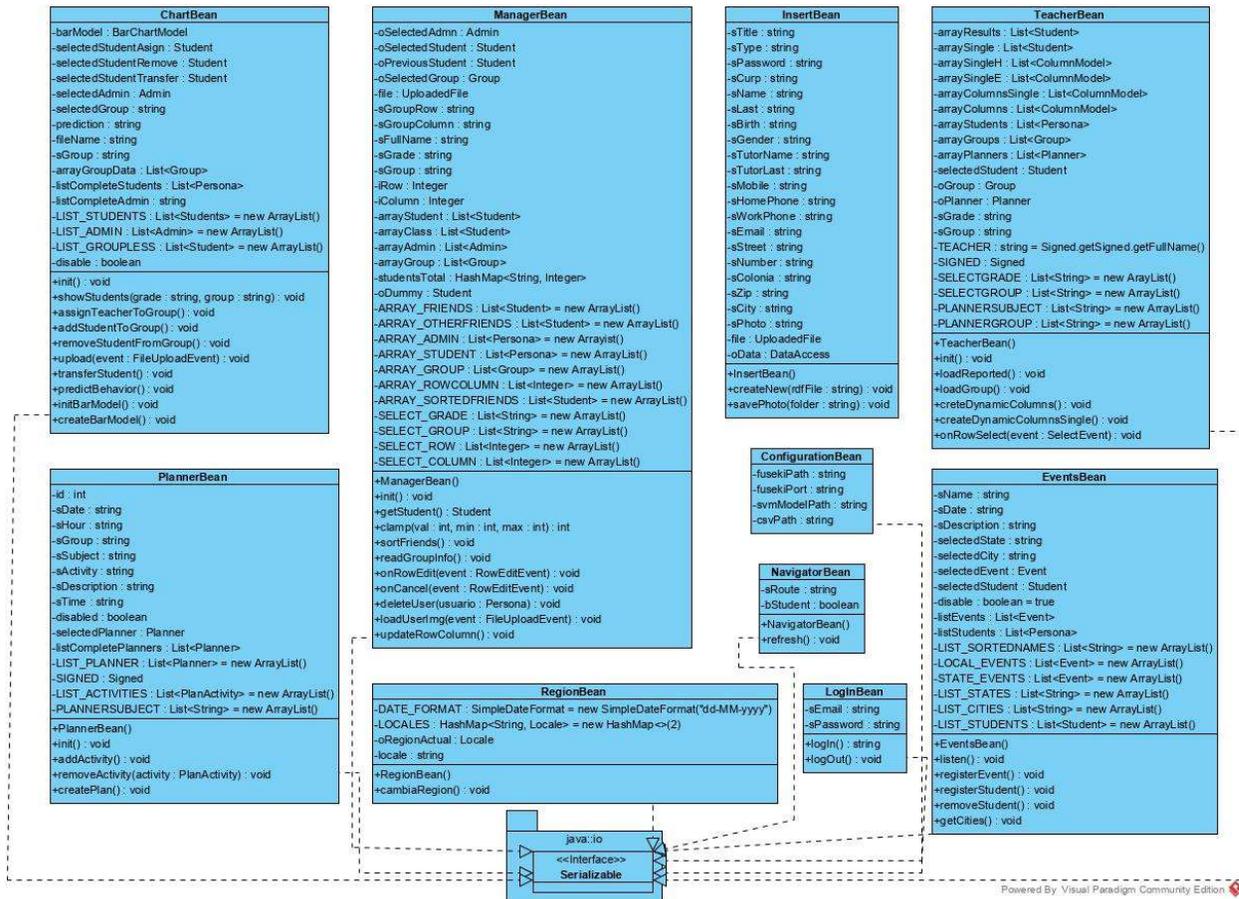


Figura 3.3.5 Diagrama de clases correspondiente al paquete beans

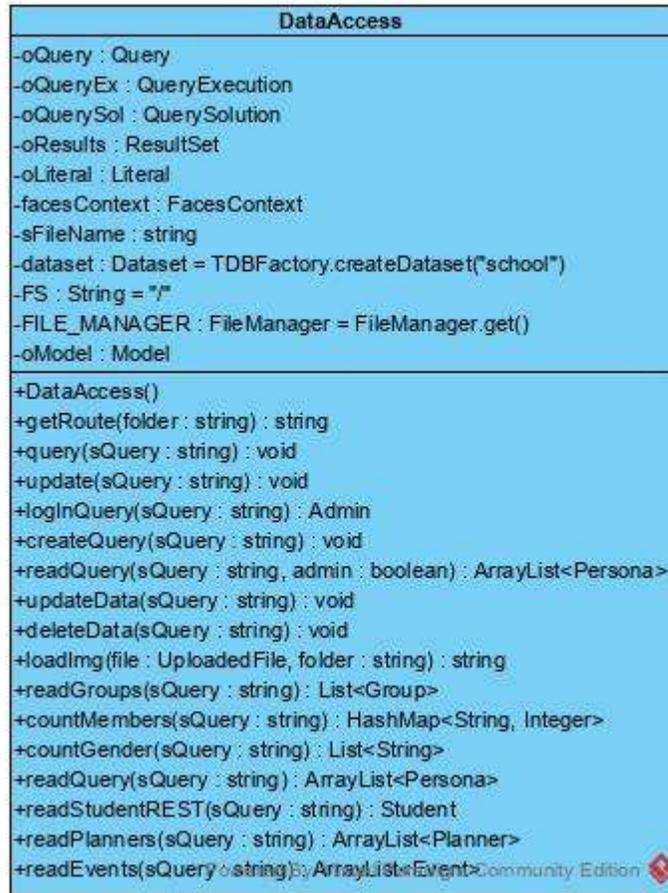


Figura 3.3.6 Diagrama de clases correspondiente al paquete *dao*

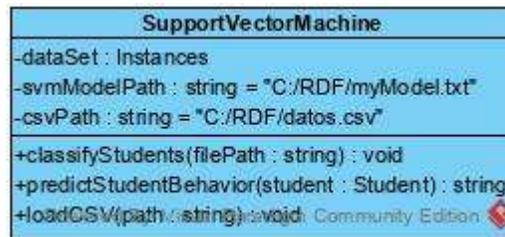


Figura 3.3.7 Diagrama de clases correspondiente al paquete *weka*

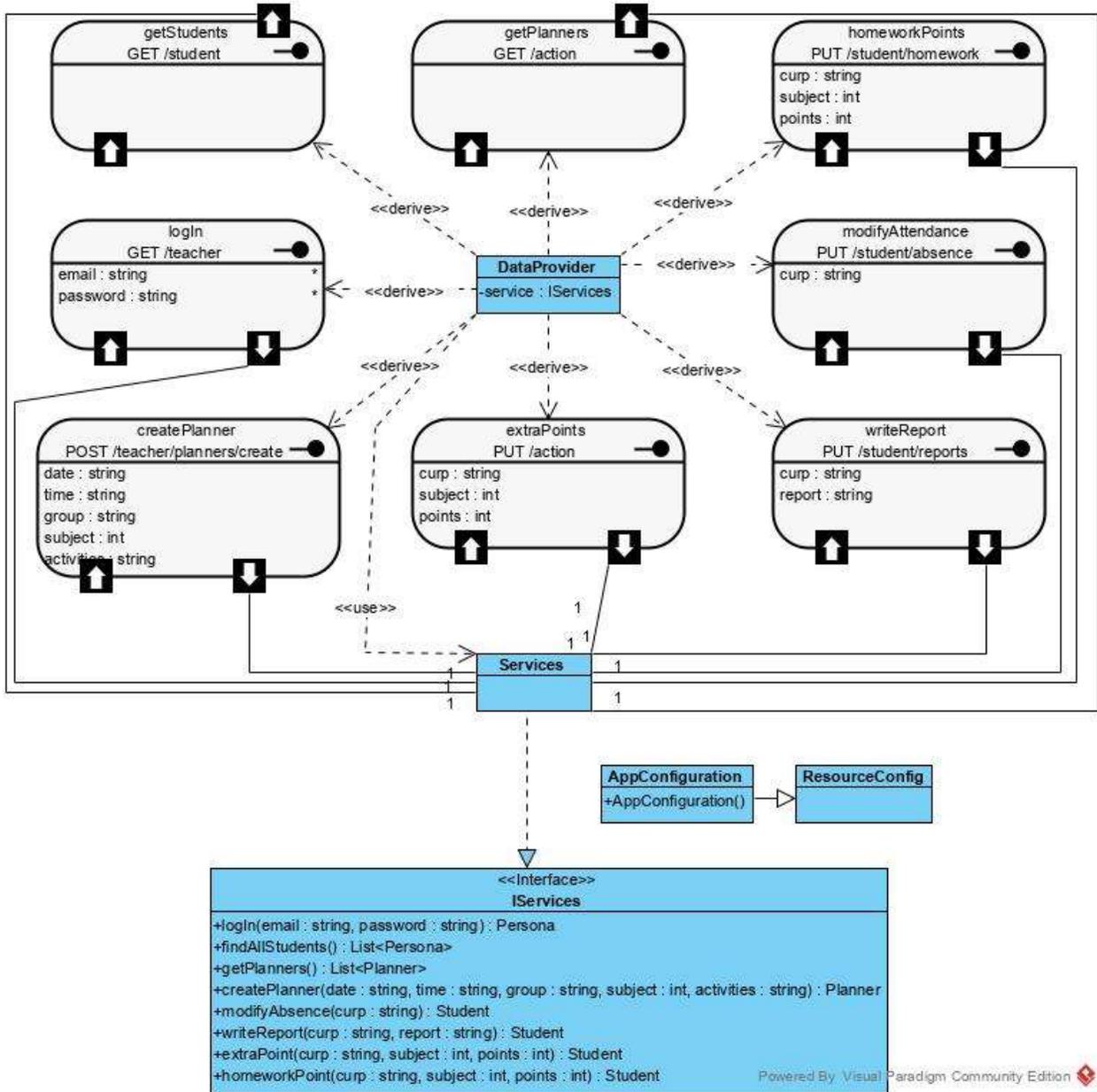


Figura 3.3.8 Diagrama de clases correspondiente al paquete *rest*

3.2.3.2 Diagramas de paquetes correspondientes a la aplicación móvil

En este apartado se muestran los diagramas de clases de la aplicación móvil, cada uno organizado en relación al paquete que contiene las clases respectivas. La figura 3.3.9 muestra el diagrama que contiene las clases que representan las actividades dentro del ciclo de vida de la aplicación Android. En la figura 3.4 se encuentran los adaptadores de los componentes visuales, los cuales son indispensables para crear listas con vistas personalizadas. Se utilizan también clases para operaciones específicas como la inserción de la fecha y la hora por medio de las clases que aparecen en el diagrama de la figura 3.4.1. Las clases que corresponden al modelo son mostradas por medio de la figura 3.4.2 y finalmente se utiliza un diagrama de clases con servicios REST para la visualización de estos en la parte móvil (figura 3.4.3).

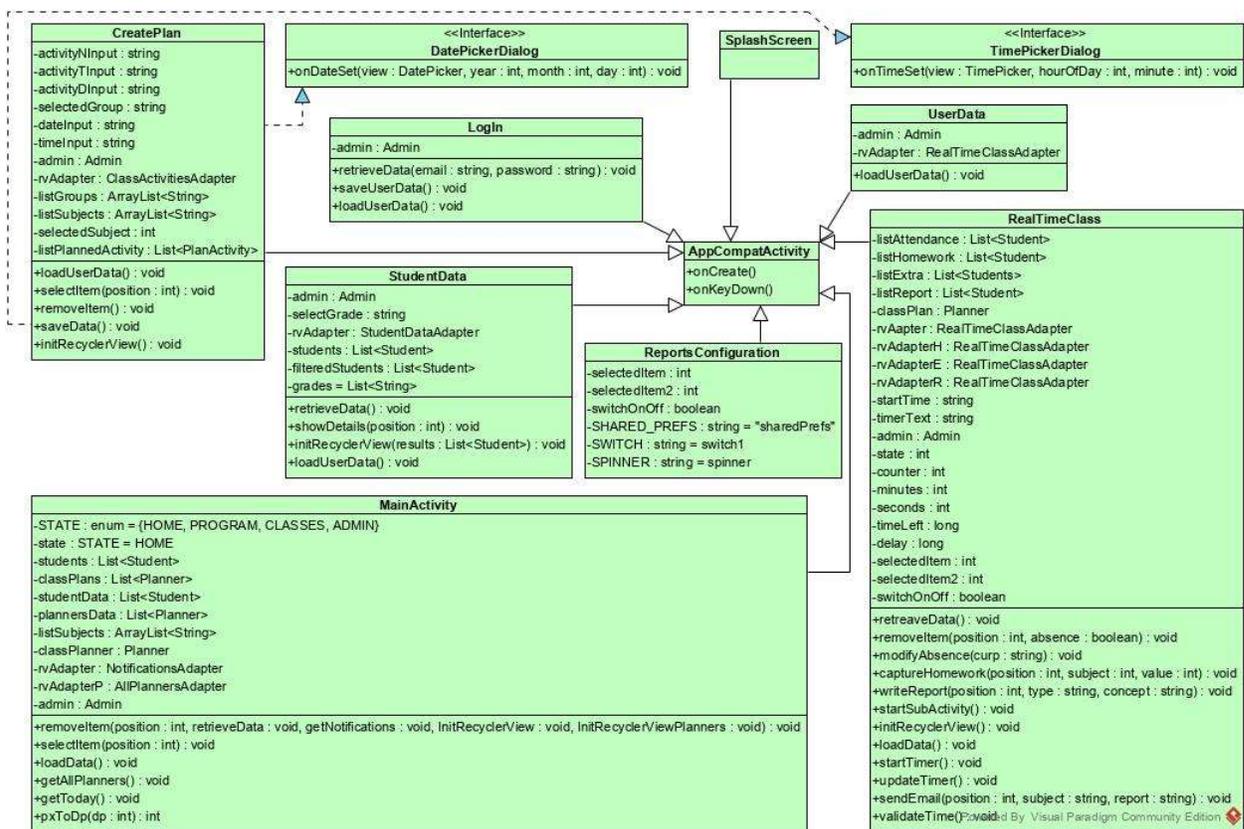


Figura 3.3.9 Diagrama de clases correspondiente al paquete *activities*

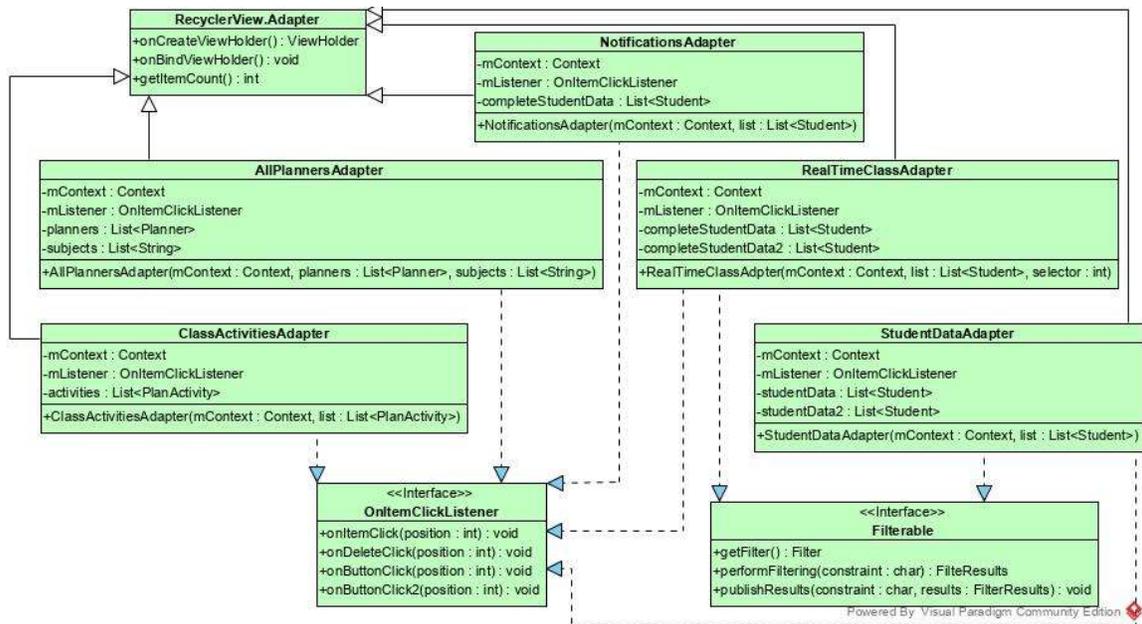


Figura 3.4 Diagrama de clases correspondiente al paquete *adapters*

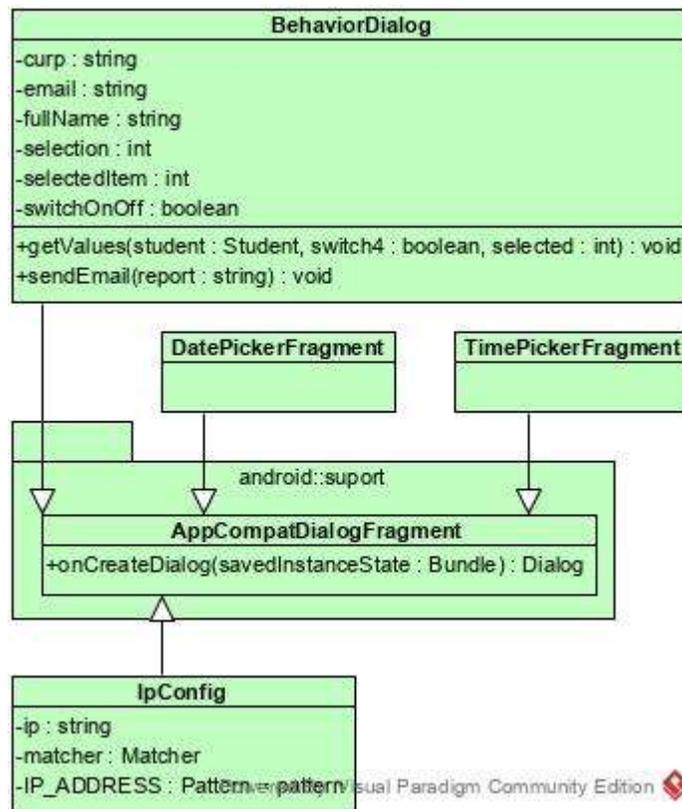


Figura 3.4.1 Diagrama de clases correspondiente al paquete *gadgets*

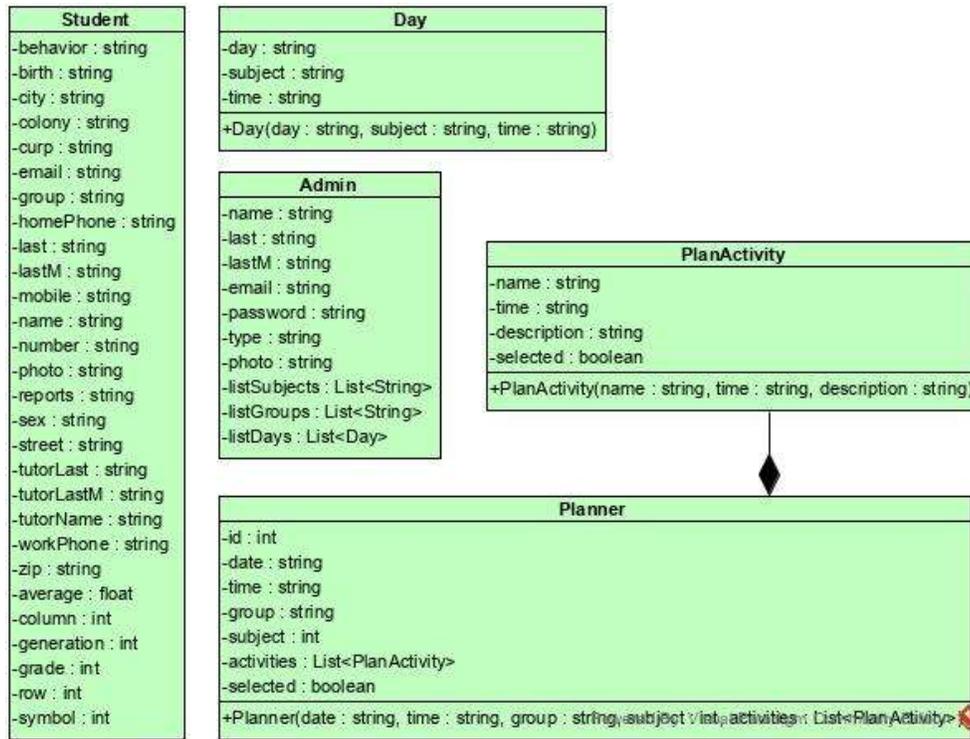


Figura 3.4.2 Diagrama de clases correspondiente al paquete *model*

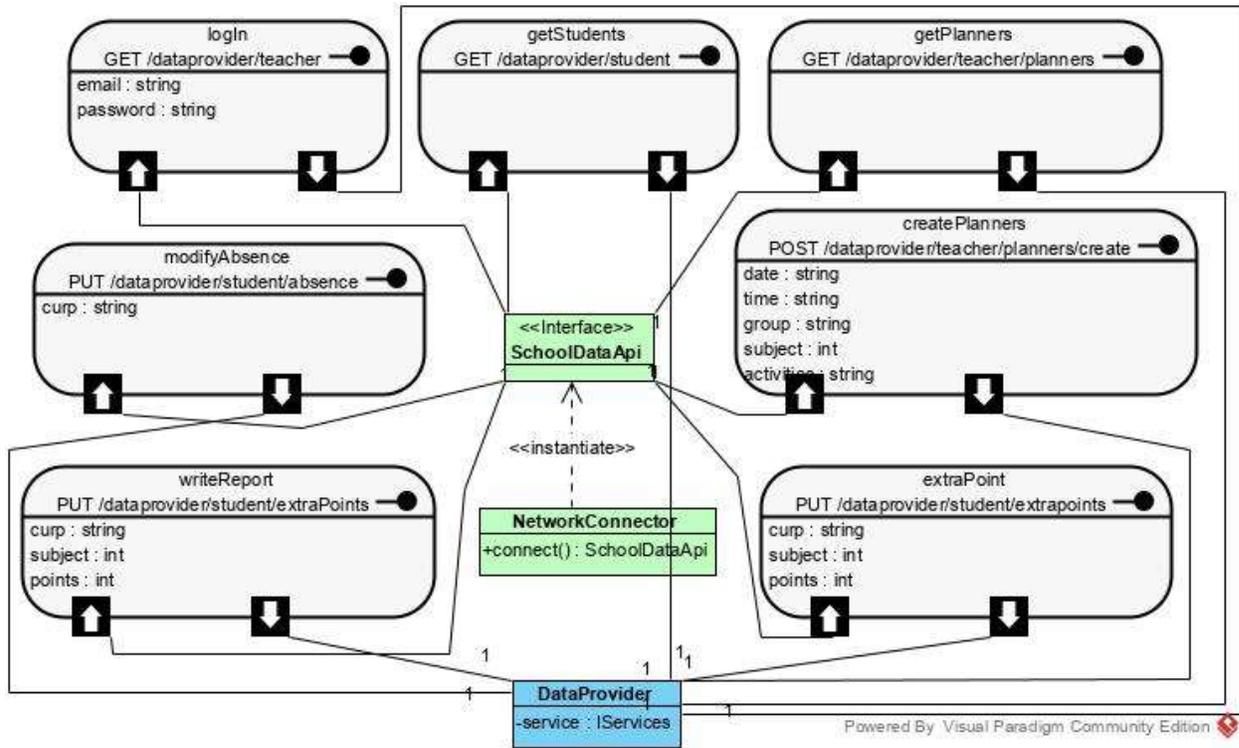


Figura 3.4.3 Diagrama de clases correspondiente al paquete *rest*.

3.2.4 Arquitectura del sistema

La arquitectura fue diseñada con base al patrón arquitectónico MVC (*Model, View, Controller*, Modelo, Vista, Controlador), de este modo la distribución de los componentes del sistema permite un fácil mantenimiento. La figura 3.5 presenta de forma abstracta esta arquitectura.

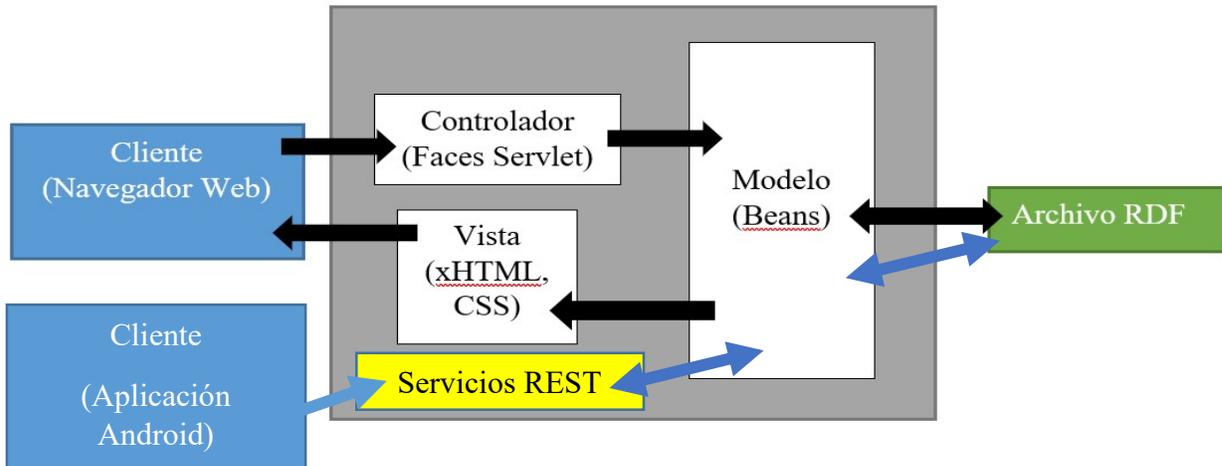


Figura 3.5 Arquitectura del sistema

3.2.5 Patrón MVC

El patrón arquitectónico MVC define tres capas dentro de la aplicación separando la lógica de negocio de la representación y la persistencia, como se observa en la Figura 3.5.1.

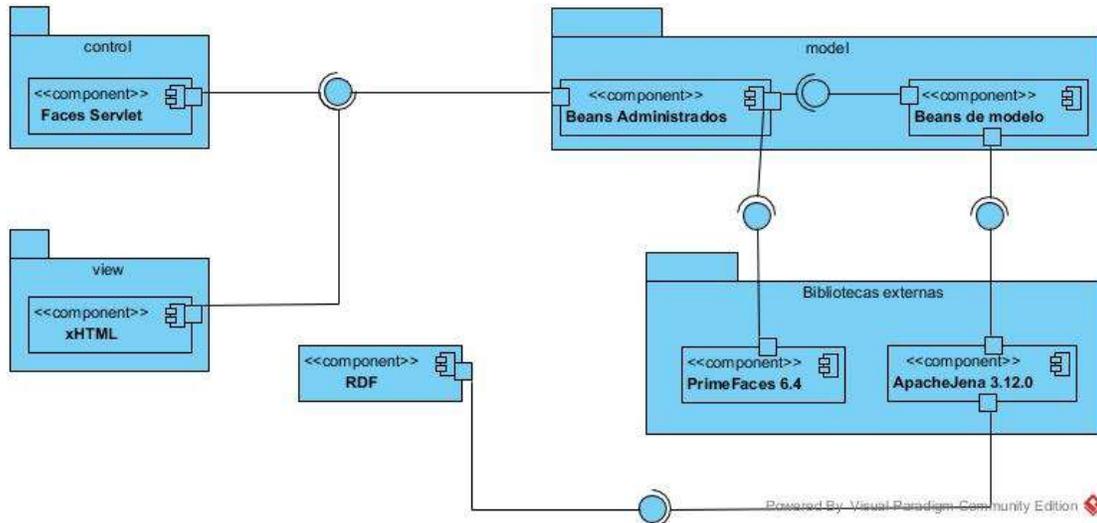


Figura 3.5.1 Arquitectura de la aplicación

3.2.6 Diagramas de secuencia

Se incluyen en esta sección los diagramas de secuencia para las aplicaciones web y móvil.

3.2.6.1 Diagramas de secuencia de la aplicación Web.

Se incluyen los diagramas de secuencia para el inicio de sesión, altas, bajas, modificaciones y consultas. En la figura 3.6.1 se encuentra el diagrama de secuencia de inicio de sesión, dado que la interacción del usuario *Coordinador* y el usuario *Docente* funciona de la misma manera, sólo se muestra al usuario *Coordinador* como actor. En las figuras 3.6.2, 3.6.3 y 3.6.4 se observan las secuencias correspondientes a la funcionalidad CRUD del sistema en la parte de los registros de estudiantes y docentes a los cuales sólo el coordinador académico puede acceder.

El diagrama en la figura 3.6.5 ejemplifica la interacción del coordinador o el docente con respecto a las consultas de los datos.

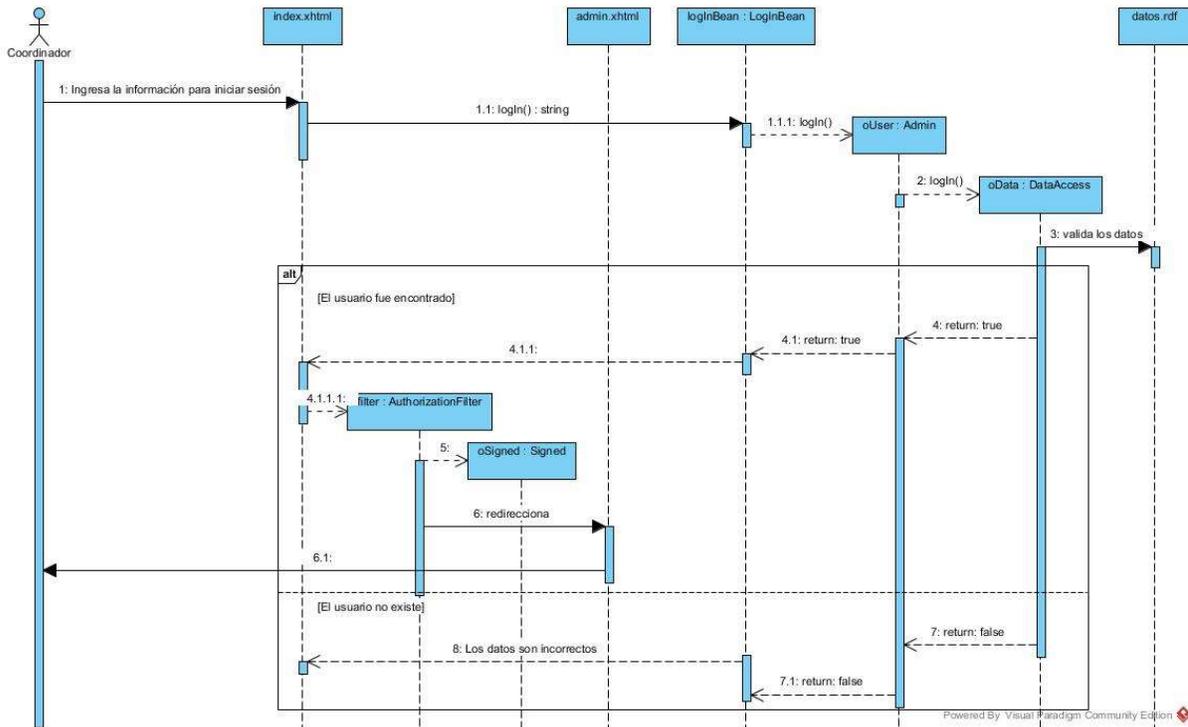


Figura 3.6.1 Diagrama de secuencia del inicio de sesión

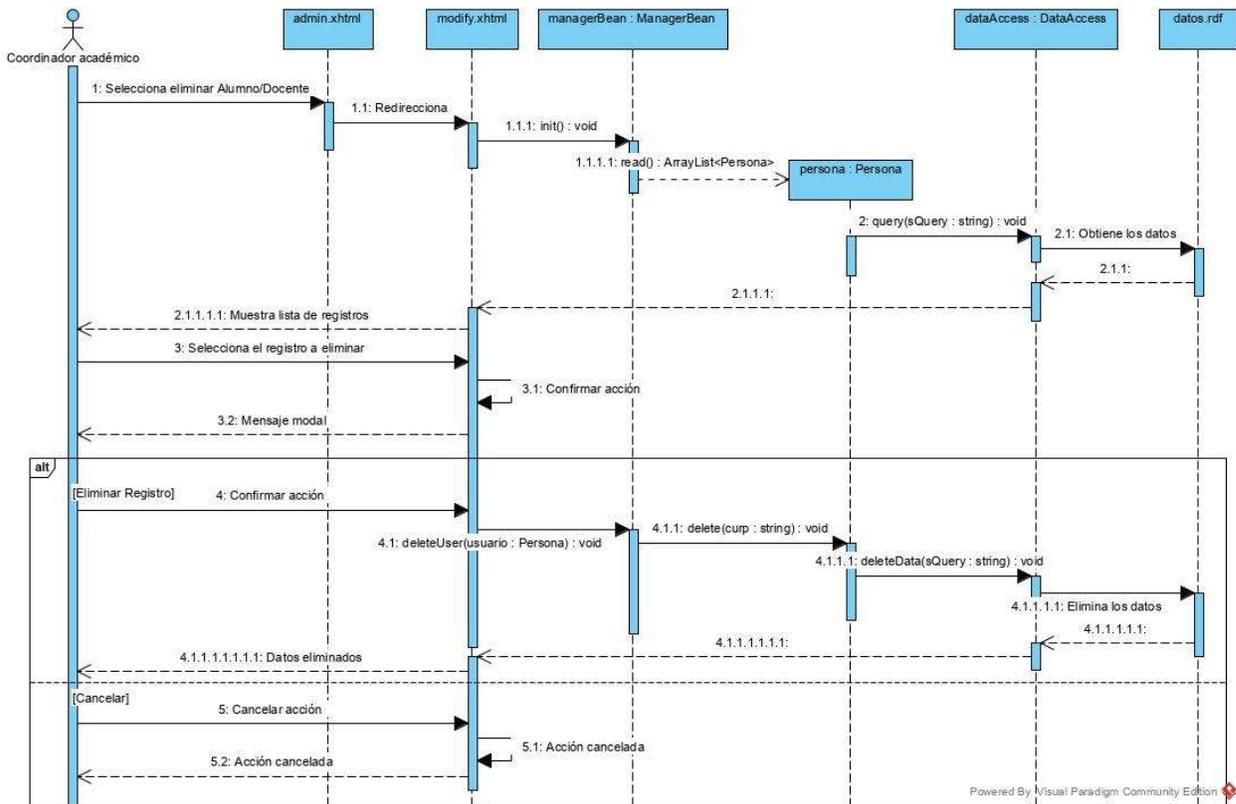


Figura 3.6.2 Diagrama de secuencia de la eliminación de un registro

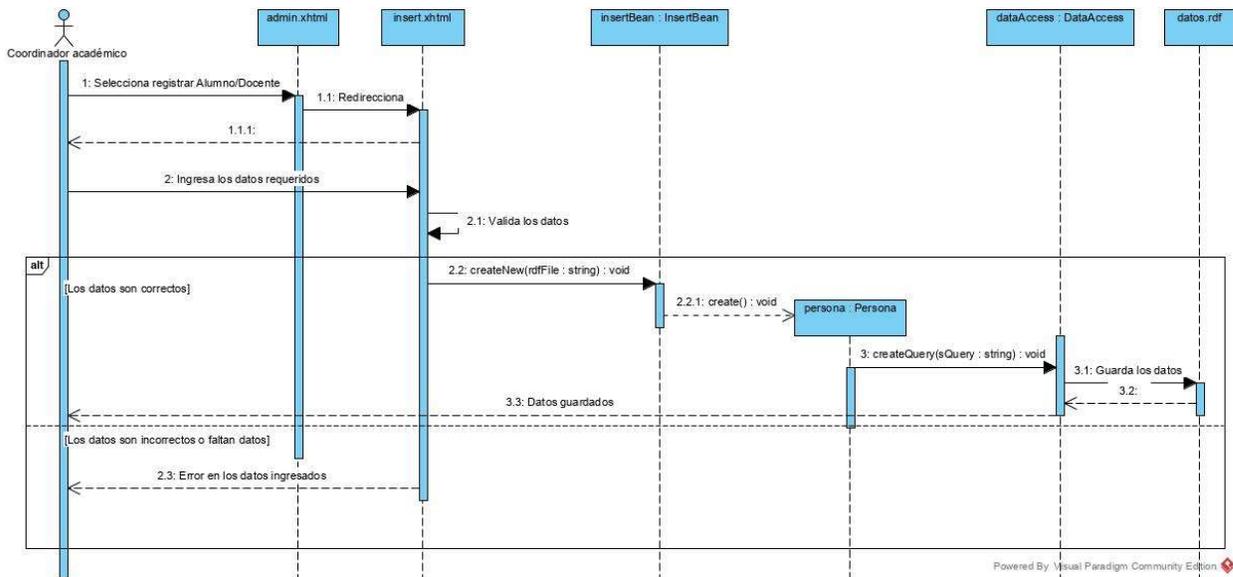


Figura 3.6.3 Diagrama de secuencia de la creación de un nuevo registro

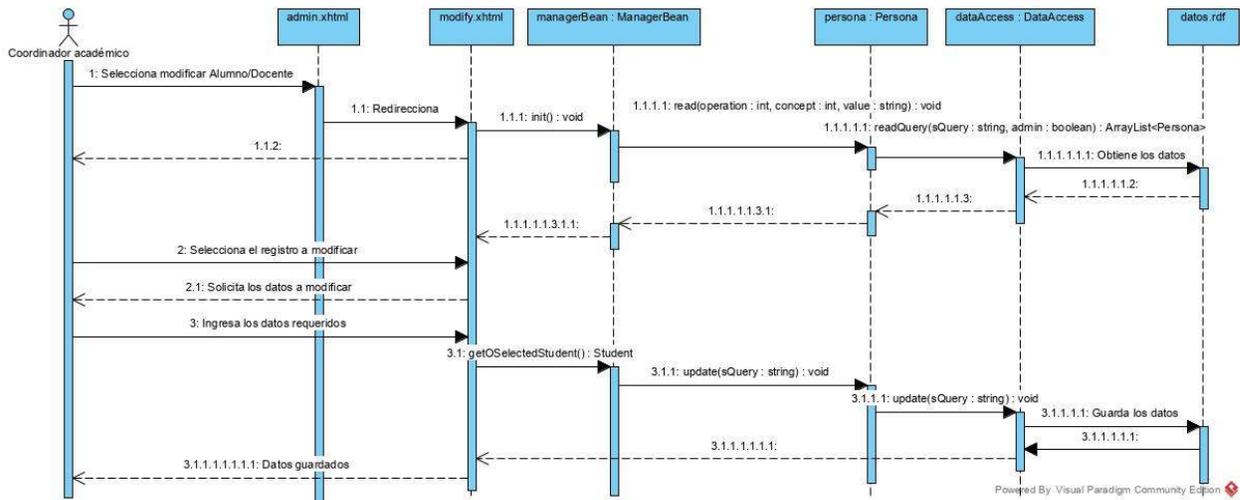


Figura 3.6.4 Diagrama de secuencia de la modificación de un registro

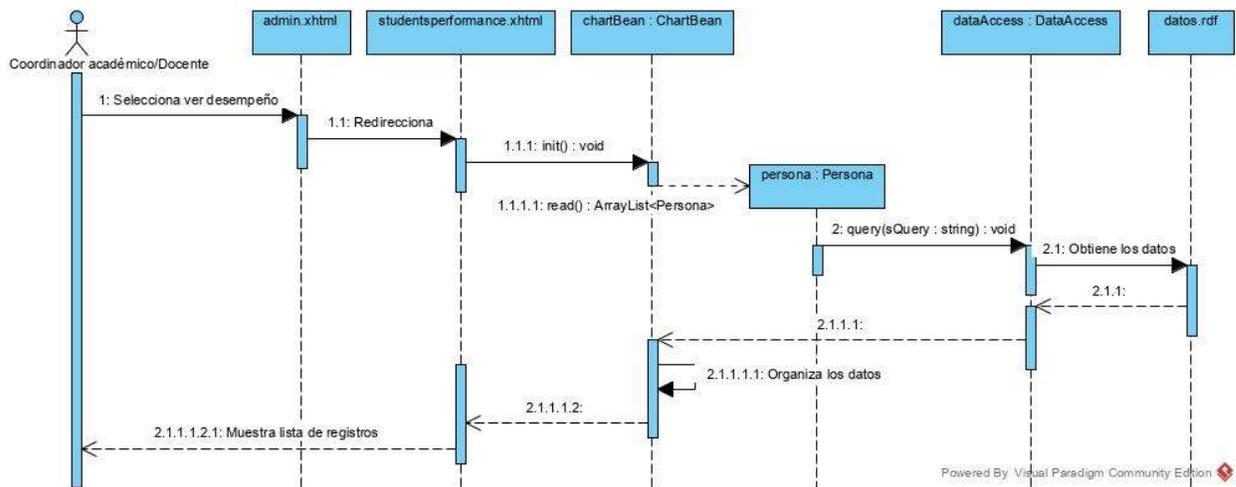


Figura 3.6.5 Diagrama de secuencia de la consulta de datos

3.2.6.2 Diagramas de secuencia de la aplicación móvil

En este apartado se muestran los diagramas correspondientes a la aplicación móvil. La aplicación móvil depende de la aplicación Web para la obtención de los datos, por lo que, para facilitar la lectura de los diagramas, se utiliza el color verde en los objetos pertenecientes a la aplicación Android y el color azul para los objetos que forman parte de los servicios REST dentro de la aplicación Web.

Se incluyen los diagramas de secuencia para las actividades; inicio de sesión (figura 3.6.6), crear plan clase (figura 3.6.7), consultar datos de los alumnos (figura 3.6.8), inicio de la clase en tiempo real (figura 3.6.9).

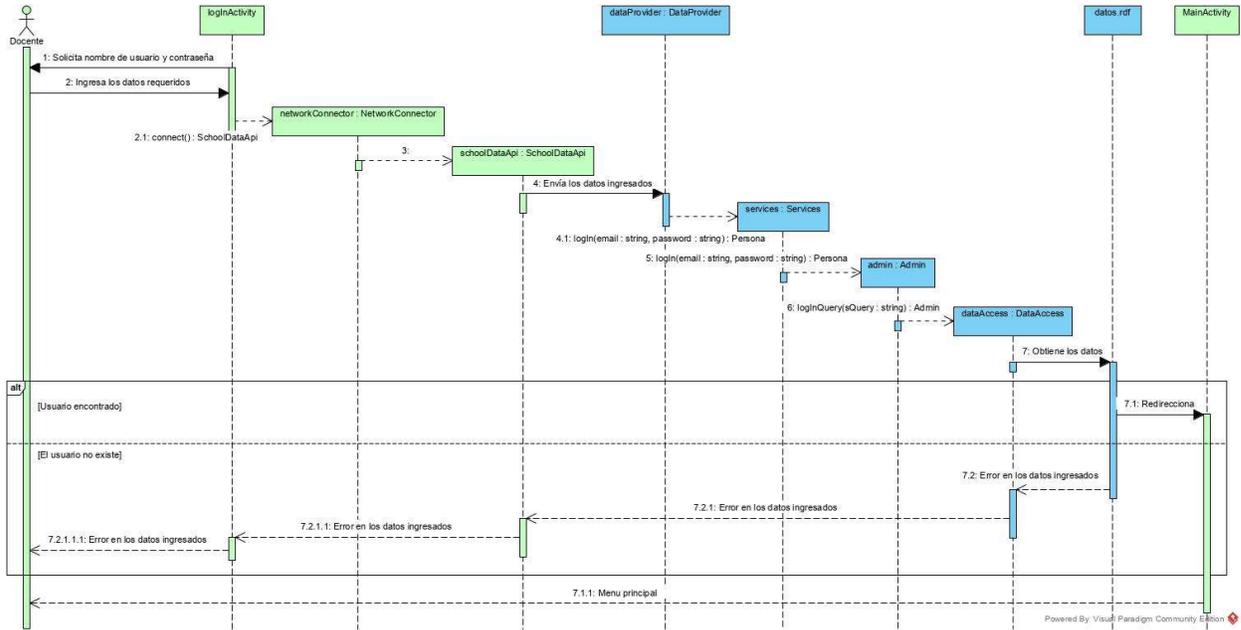


Figura 3.6.6 Diagrama de secuencia del inicio de sesión

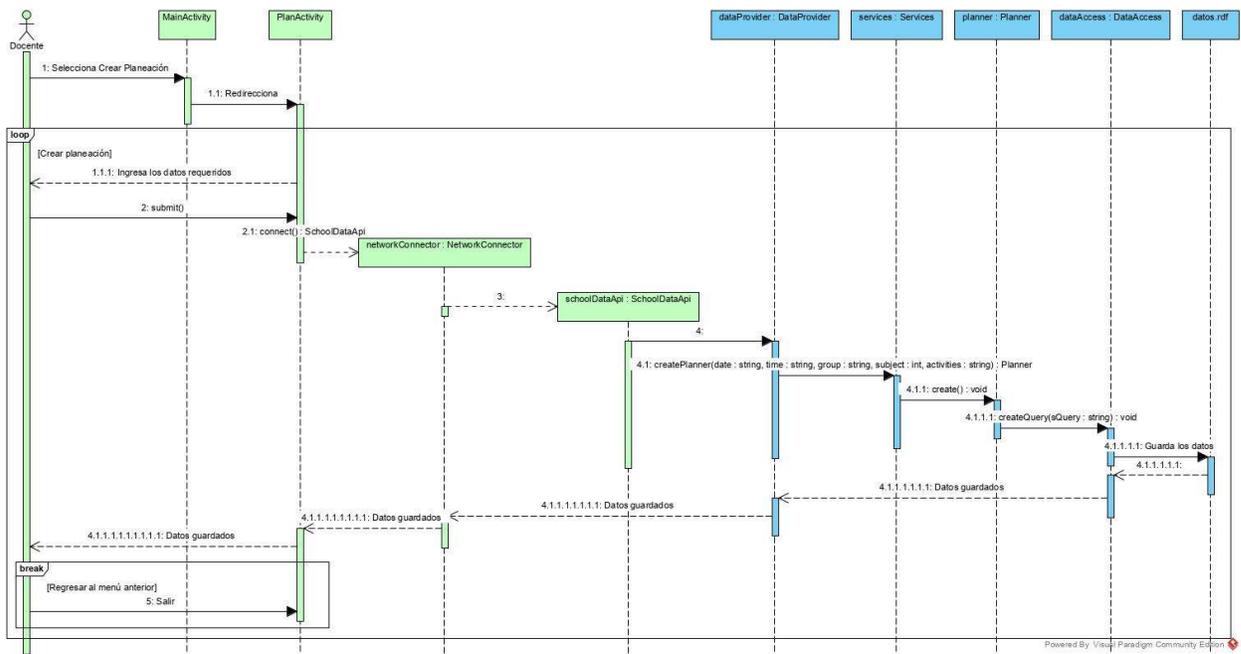


Figura 3.6.7 Diagrama de secuencia de la creación de un plan de clase

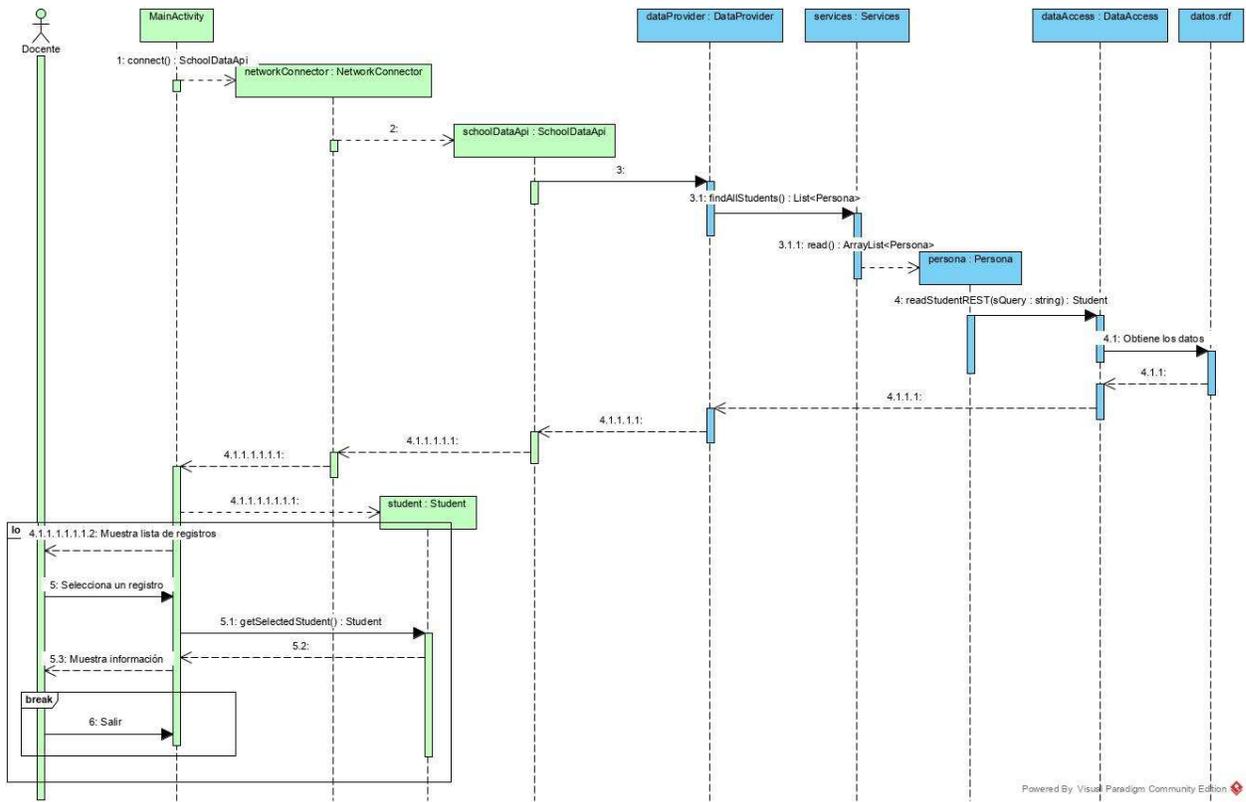


Figura 3.6.8 Diagrama de secuencia de consulta de datos de un estudiante seleccionado

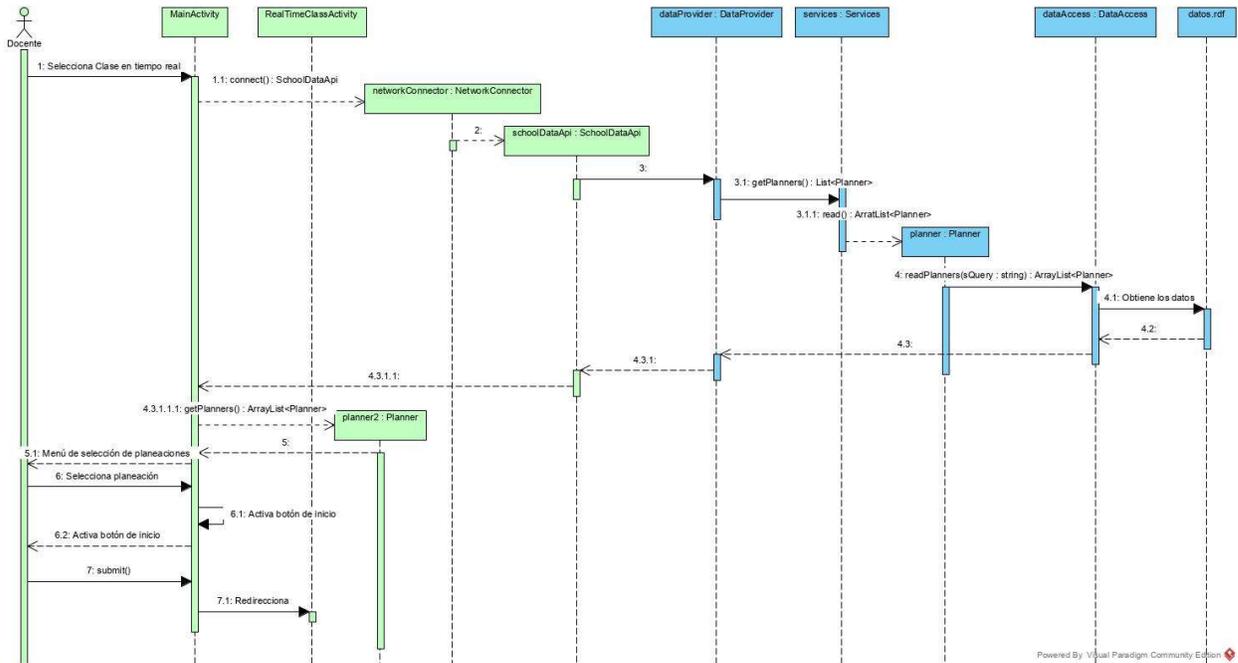


Figura 3.6.9 Diagrama de secuencia del inicio de clase en tiempo real

3.3 Codificación del software

Durante esta etapa se realizan las tareas que comúnmente se conocen como programación; que consiste, esencialmente, en llevar a código fuente, en el lenguaje de programación elegido, todo lo diseñado en la fase anterior.

3.3.1 Conexión a la base de datos

Para la implementación de la base de datos se utilizaron diferentes métodos; en el primero se utilizó el servidor de Apache Jena Fuseki como una aplicación independiente, este tipo de implementación fue viable durante el periodo de pruebas ya que todas estas se realizaron de manera local, pero una vez iniciadas las pruebas en un servidor en la red, no fue posible conectarse al servidor de Fuseki por lo que se cambió el código para implementar el servicio como una aplicación incrustada.

```

43 @WebListener
44 public class FusekiConnect implements ServletContextListener {
45
46     private static final int PORT = 3030;
47
48     public void contextInitialized(ServletContextEvent sce) {
49         try {
50             if(!isRunning()) {
51                 try {
52                     ExternalContext context = FacesContext.getCurrentInstance().getExternalContext();
53                     ProcessBuilder pb = new ProcessBuilder("java", "-jar", "fuseki-server.jar");
54                     pb.directory(new File("D:/apache-jena-fuseki-3.12.0/"));
55                     Process p = pb.start();
56                 } catch(IOException e) {}
57             }
58         } catch (IOException ex) {Logger.getLogger(FusekiConnect.class.getName()).log(Level.SEVERE, null, ex);}
59     }
60 }
61
62 @Override
63 public void contextDestroyed(ServletContextEvent sce) {}
64
65 private boolean isRunning() throws IOException {
66     Process process = Runtime.getRuntime().exec("sc query java");
67     Scanner reader = new Scanner(process.getInputStream(), "UTF-8");
68     while(reader.hasNextLine())
69         if(reader.nextLine().contains("java")) return true;
70     return false;
71 }
72
73
74 public static int getPORT() {return PORT;}
75
76 }

```

Figura 3.7 Código para la inicialización del servidor Fuseki al inicio de la aplicación Web

En la figura 3.7 se muestra el código que inicia el servidor Fuseki cuando este se utiliza como una aplicación independiente. Al iniciar la aplicación Web el método *contextInitialized* en la línea 48 (el cual es implementado por medio de la interfaz *ServletContextListener*) revisa si el servicio en el puerto 3030 se encuentra activo, en caso de que no inicia el proceso, en este caso, el archivo ejecutable del servidor (línea 54). Este tipo de implementación funciona correctamente pero no tiene utilidad para el usuario final, los pasos para su instalación no son sencillos para un usuario sin experiencia, y un problema mayor es la validación que hace el método *isRunning()* (línea 66), pues el proceso del servidor Fuseki tiene como nombre 'java' lo

que da un falso positivo si existe otra máquina virtual ejecutándose bajo el mismo nombre. Por estas razones se decidió eliminar el uso de Fuseki como aplicación independiente.

Las ventajas de tener el servidor de datos incrustado en la aplicación Web fueron; la eliminación de la configuración del servidor como paso indispensable para la inicialización del sistema y la capacidad de desplegar la aplicación en un *host* en internet sin la necesidad de más configuraciones o impedimentos como en las pruebas con el servidor independiente. En una iteración más se optó por trabajar directamente desde el *framework* de Apache Jena sin la necesidad de incluir las clases relacionadas con Fuseki ya que se implementa un modo de base de datos de transacción desde Jena que en efecto son las mismas clases que utiliza Fuseki para su funcionamiento. Realizada esta parte el tamaño del archivo generado para el despliegue de la aplicación se redujo significativamente, en la figura 3.7.1 se muestra el código de la primera iteración para el servidor de datos incrustado donde se configura un servidor Fuseki en la línea 55, luego en la línea 58 se obtiene el archivo RDF que se utilizará, finalmente en la línea 59 se introduce el modelo al servidor.

```

43 @WebListener
44 public class FusekiConnect implements ServletContextListener {
45     private static final int PORT = 3330;
46     private FusekiServer server;
47     private static final Location LOCATION = Location.create("C:/RDF/school");
48     private static final Dataset DS = TDBFactory.createDataset(LOCATION);
49     private static final Dataset DS = TDBFactory.createDataset(LOCATION);
50
51     private static final Dataset DS = TDBFactory.createDataset(LOCATION);
52
53     @Override
54     public void contextInitialized(ServletContextEvent sce) {
55         server = FusekiServer.create().add("/school", DS).build();
56         server.start();
57         FileManager FILE_MANAGER = FileManager.get();
58         Model oModel = FILE_MANAGER.loadModel("C:/RDF/alumnos.rdf");
59         DatasetAccessor dataAccessor = DatasetAccessorFactory.createHTTP("http://localhost:"+PORT+"/school/data");
60         dataAccessor.putModel(oModel);
61     }
62
63     @Override
64     public void contextDestroyed(ServletContextEvent sce) {
65         server.stop();
66     }

```

Figura 3.7.1 Código para la inicialización del servidor Fuseki incrustado

En la figura 3.7.2 se muestra el código final donde se establece una base de datos de transacciones (línea 99) en un código reducido y sin la necesidad de una clase servidor.

```

96 public DataAccess() {
97     FILE_MANAGER.addLocatorClassLoader(DataAccess.class.getClassLoader());
98     oModel = FileManager.get().loadModel("/school/data/alumnos.rdf");
99     dataset = TDBFactory.createDataset("/school/data/mainTDB");
100    dataset.begin(ReadWrite.WRITE);
101    DatasetAccessor accessor = DatasetAccessorFactory.create(dataset);
102    accessor.putModel(oModel);
103    dataset.commit();
104    dataset.end();
105 }

```

Figura 3.7.2 Código para la inicialización de la base de datos de transacciones sin el servidor Fuseki

3.3.2 Esquema RDF

Para almacenar la información del personal docente y de los estudiantes se utiliza un archivo RDF basado en el vocabulario FOAF. La figura 3.8 muestra la organización de los atributos que cada individuo posee, el esquema es similar a un XML básico, pero los datos son escritos uno por línea.

```
<?xml version="1.0"?>

<!DOCTYPE rdf:RDF [
  <!ENTITY foaf "http://xmlns.com/foaf/0.1/" >
  <!ENTITY tesis "http://localhost:8080/tesis/" >
  <!ENTITY owl "http://www.w3.org/2002/07/owl#" >
  <!ENTITY xsd "http://www.w3.org/2001/XMLSchema#" >
  <!ENTITY rdfs "http://www.w3.org/2000/01/rdf-schema#" >
  <!ENTITY rdf "http://www.w3.org/1999/02/22-rdf-syntax-ns#" >
]>

<rdf:RDF xmlns="http://localhost:8080/tesis/"
  xml:base="http://localhost:8080/tesis/"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:tesis="http://localhost:8080/tesis/"
  xmlns:foaf="http://xmlns.com/foaf/0.1/"
  xmlns:owl="http://www.w3.org/2002/07/owl#"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
  <owl:Ontology rdf:about="http://localhost:8080/tesis/">
    <owl:imports rdf:resource="http://xmlns.com/foaf/0.1/" />
  </owl:Ontology>

  <!--NAMEDINDIVIDUALS-->

  <!--PEHM881013HVZRRR05-->
  <owl:NamedIndividual rdf:about="&tesis;PEHM881013HVZRRR05">
  <rdf:type rdf:resource="&foaf;Person"/>
  <foaf:column rdf:datatype="&xsd;integer">3</foaf:column>
  <foaf:row rdf:datatype="&xsd;integer">5</foaf:row>
  <foaf:spanish rdf:datatype="&xsd;integer">6</foaf:spanish>
  <foaf:history rdf:datatype="&xsd;integer">5</foaf:history>
  <foaf:chemistry rdf:datatype="&xsd;integer">5</foaf:chemistry>
  <foaf:math rdf:datatype="&xsd;integer">9</foaf:math>
  <foaf:givenname>Mario Ricardo</foaf:givenname>
  <foaf:family_name>Perez</foaf:family_name>
  <foaf:family_mname>Hernandez</foaf:family_mname>
  <foaf:birth_date>1988-10-13</foaf:birth_date>|
  <foaf:curp>PEHM881013HVZRRR05</foaf:curp>
  <foaf:member rdf:resource="&tesis;1A"/>
  <foaf:member rdf:resource="&tesis;2019"/>
  <foaf:knows rdf:resource="&tesis;AAAA999999HVZRRR05"/>
  <foaf:knows rdf:resource="&tesis;FFFF881013HVZRRR06"/>
  <foaf:report>
  2019-05-05_Mala conducta_El alumno agredió verbalmente a un compañero_
  2019-05-06_Retardo_El alumno llegó 30 minutos tarde_
  </foaf:report>

  </owl:NamedIndividual>
```

Figura 3.8 Estructura del archivo RDF

3.3.3 Consultas con SPARQL

Las consultas se realizan mediante la sintaxis de SPARQL en el caso de la lectura de los archivos RDF. En las figuras 3.9 y 3.9.1 se muestran las consultas y la implementación en Java para obtener la información deseada.

```

234 public Student readStudentData(String curp) throws Exception {
235     Student student;
236     sQuery = "PREFIX foaf: <http://xmlns.com/foaf/0.1/>"
237             + "SELECT * WHERE {"
238             + " ?person foaf:givenname ?a ."
239             + " ?person foaf:family_name ?b ."
240             + " ?person foaf:family_mname ?c ."
241             + " ?person foaf:birth_date ?d ."
242             + " ?person foaf:gender ?e ."
243             + " ?person foaf:curp ?f ."
244             + " ?person foaf:parent_givenname ?g ."
245             + " ?person foaf:parent_family_name ?h ."
246             + " ?person foaf:parent_family_mname ?i ."
247             + " ?person foaf:email ?j ."
248             + " ?person foaf:mobile ?k ."
249             + " ?person foaf:home ?l ."
250             + " ?person foaf:work ?m ."
251             + " ?person foaf:street ?n ."
252             + " ?person foaf:number ?o ."
253             + " ?person foaf:colonia ?p ."
254             + " ?person foaf:zip ?q ."
255             + " ?person foaf:city ?r ."
256             + " ?person foaf:photo ?s ."
257             + " ?person foaf:member ?group ."
258             + " ?group foaf:grade ?t ."
259             + " ?group foaf:identifier ?u ."
260             + " ?person foaf:column ?v ."
261             + " ?person foaf:row ?w ."
262             + " ?person foaf:member ?generation ."
263             + " ?generation foaf:generation ?x ."
264             + " ?person foaf:absense ?y ."
265             + " ?person foaf:behavior ?z ."
266
317             + " FILTER (?f = '" + curp + "'" )
318             + "};";
319
320     if (oData == null) {
321         oData = new DataAccess();
322         student = oData.readStudentREST(sQuery);
323     } else student = oData.readStudentREST(sQuery);
324
325     if(student != null) return student;
326     else throw new Exception("No hay datos");
327 }

```

Figura 3.9 Método en Java para la consulta de información de alumnos desde la clase modelo, se incluye un ejemplo de consulta mediante la sintaxis de SPARQL (Campos removidos para brevedad)

```

752 public ArrayList<Planner> readPlanner(String sQuery) {
753     ArrayList<Planner> listPlanners = new ArrayList<>();
754     Planner oPlanner;
755     String id, date, hour, group, activities;
756     int subject;
757     dataset.begin(ReadWrite.READ);
758     try {
759         query(sQuery);
760         oResults = oQueryEx.execSelect();
761         while(oResults.hasNext()) {
762             oQuerySol = oResults.nextSolution();
763             oLiteral = oQuerySol.getLiteral("id");
764             id = oLiteral.getString();
765             oLiteral = oQuerySol.getLiteral("a");
766             date = oLiteral.getString();
767             oLiteral = oQuerySol.getLiteral("b");
768             hour = oLiteral.getString();
769             oLiteral = oQuerySol.getLiteral("c");
770             group = oLiteral.getString();
771             oLiteral = oQuerySol.getLiteral("d");
772             subject = oLiteral.getInt();
773             oLiteral = oQuerySol.getLiteral("e");
774             activities = oLiteral.getString();
775             String[] parts = activities.split("_");
776             List<PlanActivity> listActivities = new ArrayList<>();
777             for(int i = 0; i < parts.length; i += 3) {
778                 PlanActivity pa = new PlanActivity(parts[i], parts[i+1], parts[i+2]);
779                 listActivities.add(pa);
780             }
781             oPlanner = new Planner(Integer.valueOf(id), date, hour, group, subject, listActivities);
782             listPlanners.add(oPlanner);
783         }
784     } finally {
785         oQueryEx.close();
786         dataset.end();
787     }
788     return listPlanners;
789 }

```

Figura 3.9.1 Método de la clase de conexión a datos para obtener la información de un archivo RDF

3.3.3 Métodos de escritura sobre los archivos RDF

A continuación, se muestran los métodos para inserción, modificación y borrado de datos de un archivo RDF. Se utiliza SPARUL para todos los casos. Las figuras 3.9.2, 3.9.3 y 3.9.4 muestran la cadena estructurada para las operaciones mencionadas (líneas 105, 442, 466, 473, 480, 496, 503, 510, 517 y 535).

```

103 @Override
104 public void create() throws Exception {
105     sQuery = "PREFIX foaf: <http://xmlns.com/foaf/0.1/>"
106             + "PREFIX : <http://localhost:8080/tesis/>"
107             + "INSERT DATA { :'+sCurp+' foaf:curp '"+sCurp+''; foaf:givenname '"+sName+'";
108             + " foaf:family_name '"+sLast+''; foaf:family_mname '"+sLastM+''; foaf:birth_date '"+sBirth+'";
109             + " foaf:gender '"+sSex+''; foaf:curp '"+sCurp+''; foaf:parent_givenname '"+sName+'";
110             + " foaf:parent_family_name '"+sLast+''; foaf:parent_family_mname '"+sLastM+'";
111             + " foaf:email '"+sEmail+''; foaf:mobile '"+sTel+''; foaf:home '"+sHomePhone+'";
112             + " foaf:work '"+sWorkPhone+''; foaf:street '"+sStreet+''; foaf:number '"+sNumber+'";
113             + " foaf:colonia '"+sColonia+''; foaf:zip '"+sZip+''; foaf:city '"+sCity+''; foaf:photo '"+sPhoto+'";
114             + " foaf:column 0 ; foaf:row 0 ; foaf:absence 0 ; foaf:behavior '' ; foaf:member :NA ; foaf:member :2016
115             + " foaf:class0 0 ; foaf:class1 0 ; foaf:class2 0 ; foaf:class3 0 ; foaf:class4 0 ; foaf:class5 0 ; foaf:
116             + " foaf:homework0 0 ; foaf:homework1 0 ; foaf:homework2 0 ; foaf:homework3 0 ; foaf:homework4 0 ; foaf:
117             + " foaf:exam0 0 ; foaf:exam1 0 ; foaf:exam2 0 ; foaf:exam3 0 ; foaf:exam4 0 ; foaf:exam5 0 ; foaf:exam6
118             + " foaf:notebook0 0 ; foaf:notebook1 0 ; foaf:notebook2 0 ; foaf:notebook3 0 ; foaf:notebook4 0 ; foaf:
119             + " foaf:extra0 0 ; foaf:extra1 0 ; foaf:extra2 0 ; foaf:extra3 0 ; foaf:extra4 0 ; foaf:extra5 0 ; foaf:
120             + " }";
121
122     if (oData == null){
123         oData = new DataAccess();
124         oData.createQuery(sQuery);
125     } else oData.createQuery(sQuery);
126 }

```

Figura 3.9.2 Método de inserción de datos

```

438 @Override
439 public void update(int operation, int concept, String value) throws Exception {
440     switch(operation){
441         case 0:
442             sQuery = "PREFIX foaf: <http://xmlns.com/foaf/0.1/>"
443                 + "PREFIX : <http://localhost:8080/tesis/>"
444                 + "DELETE { :"+sCurp+" foaf:givenname ?a ; foaf:family_name ?b ; foaf:family_mname ?c ;"
445                 + " foaf:birth_date ?d ; foaf:gender ?e ; foaf:CURP ?f ; foaf:parent_givname ?g ;"
446                 + " foaf:parent_family_name ?h ; foaf:parent_family_mname ?i ; foaf:email ?j ;"
447                 + " foaf:mobile ?k ; foaf:home ?l ; foaf:work ?m ; foaf:street ?n ; foaf:number ?o ;"
448                 + " foaf:colonia ?p ; foaf:zip ?q ; foaf:city ?r ; foaf:photo ?s ."
449                 + "}"
450                 + "INSERT { :"+sCurp+" foaf:givenname '"+sName+"' ;"
451                 + " foaf:family_name '"+sLast+"' ; foaf:family_mname '"+sLastM+"' ;"
452                 + " foaf:birth_date '"+sBirth+"' ; foaf:gender '"+sSex+"' ; foaf:CURP '"+sCurp+"' ;"
453                 + " foaf:parent_givname '"+sTutorName+"' ; foaf:parent_family_name '"+sTutorLast+"' ;"
454                 + " foaf:parent_family_mname '"+sTutorLastM+"' ; foaf:email '"+sEmail+"' ; foaf:mobile '"+sMobile+"' ;"
455                 + " foaf:home '"+sHomePhone+"' ; foaf:work '"+sWorkPhone+"' ; foaf:street '"+sStreet+"' ;"
456                 + " foaf:number '"+sNumber+"' ; foaf:colonia '"+sColonia+"' ; foaf:zip '"+sZip+"' ;"
457                 + " foaf:city '"+sCity+"' ; foaf:photo '"+sPhoto+"' ."
458                 + "}"
459                 + "WHERE { :"+sCurp+" foaf:givenname ?a ; foaf:family_name ?b ; foaf:family_mname ?c ;"
460                 + " foaf:birth_date ?d ; foaf:gender ?e ; foaf:CURP ?f ; foaf:parent_givname ?g ; foaf:parent_family_name ?h ;"
461                 + " foaf:parent_family_mname ?i ; foaf:email ?j ; foaf:mobile ?k ; foaf:home ?l ; foaf:work ?m ;"
462                 + " foaf:street ?n ; foaf:number ?o ; foaf:colonia ?p ; foaf:zip ?q ; foaf:city ?r ; foaf:photo ?s ."
463                 + "}" ;
464             break;
465         case 1:
466             sQuery = "PREFIX foaf: <http://xmlns.com/foaf/0.1/>"
467                 + "PREFIX : <http://localhost:8080/tesis/>"
468                 + "DELETE { :"+sCurp+" foaf:class"+concept+" ?a .}"
469                 + "INSERT { :"+sCurp+" foaf:class"+concept+" "+value+" .}"
470                 + "WHERE { :"+sCurp+" foaf:class"+concept+" ?a .}" ;
471             break;
472         case 2:
473             sQuery = "PREFIX foaf: <http://xmlns.com/foaf/0.1/>"
474                 + "PREFIX : <http://localhost:8080/tesis/>"
475                 + "DELETE { :"+sCurp+" foaf:homework"+concept+" ?a .}"
476                 + "INSERT { :"+sCurp+" foaf:homework"+concept+" "+value+" .}"
477                 + "WHERE { :"+sCurp+" foaf:homework"+concept+" ?a .}" ;
478             break;
479         case 3:
480             String allReports = sReports + value;
481             String cleanString = allReports.replaceAll("\n", "");
482             sQuery = "PREFIX foaf: <http://xmlns.com/foaf/0.1/>"
483                 + "PREFIX : <http://localhost:8080/tesis/>"
484                 + "DELETE { :"+sCurp+" foaf:report ?a .}"
485                 + "INSERT { :"+sCurp+" foaf:report '"+cleanString+" .}"
486                 + "WHERE { :"+sCurp+" foaf:report ?a .}" ;
487             break;
488         case 5:
489             sQuery = "PREFIX foaf: <http://xmlns.com/foaf/0.1/>"
490                 + "PREFIX : <http://localhost:8080/tesis/>"
491                 + "DELETE { :"+sCurp+" foaf:photo ?a .}"
492                 + "INSERT { :"+sCurp+" foaf:photo '"+sPhoto+" .}"
493                 + "WHERE { :"+sCurp+" foaf:photo ?a .}" ;
494             break;
495         case 6:
496             sQuery = "PREFIX foaf: <http://xmlns.com/foaf/0.1/>"
497                 + "PREFIX : <http://localhost:8080/tesis/>"
498                 + "DELETE { :"+sCurp+" foaf:member :'+value.substring(0,2)+' ;}"
499                 + "INSERT { :"+sCurp+" foaf:member :'+value.substring(2)+' ;}"
500                 + "WHERE { :"+sCurp+" foaf:member :'+value.substring(0,2)+' ;}" ;
501             break;
502         case 7:
503             sQuery = "PREFIX foaf: <http://xmlns.com/foaf/0.1/>"
504                 + "PREFIX : <http://localhost:8080/tesis/>"
505                 + "DELETE { :"+sCurp+" foaf:row ?a ; foaf:column ?b ;}"
506                 + "INSERT { :"+sCurp+" foaf:row '"+iRow+"' ; foaf:column '"+iColumn+" ;}"
507                 + "WHERE { :"+sCurp+" foaf:row ?a ; foaf:column ?b ;}" ;
508             break;
509         case 8:
510             sQuery = "PREFIX foaf: <http://xmlns.com/foaf/0.1/>"
511                 + "PREFIX : <http://localhost:8080/tesis/>"
512                 + "DELETE { :"+sCurp+" foaf:row ?a ; foaf:column ?b ;}"
513                 + "INSERT { :"+sCurp+" foaf:row '"+iRow+"' ; foaf:column '"+iColumn+" ;}"
514                 + "WHERE { :"+sCurp+" foaf:row ?a ; foaf:column ?b ;}" ;
515             break;
516         case 8:
517             sQuery = "PREFIX foaf: <http://xmlns.com/foaf/0.1/>"
518                 + "PREFIX : <http://localhost:8080/tesis/>"
519                 + "INSERT DATA { :"+sCurp+" foaf:knows :'+value+' ;}";
520             default: break;
521     }
522     if (oData == null){
523         oData = new DataAccess();
524         oData.updateData(sQuery);
525     } else oData.updateData(sQuery);
526 }

```

Figura 3.9.3 Método de modificación de los datos de un archivo RDF

```

533     @Override
534     public void delete(String curp) throws Exception {
535         sQuery = "PREFIX foaf: <http://xmlns.com/foaf/0.1/>"
536                 + "PREFIX : <http://localhost:8080/tesis/>"
537                 + "DELETE {?s ?p ?o}"
538                 + "WHERE {?s ?p ?o . "
539                 + "FILTER (?s = :"+curp+"}"
540                 + "}" ;
541
542         if (oData == null) {
543             oData = new DataAccess();
544             oData.deleteData(sQuery);
545         } else oData.deleteData(sQuery);
546     }

```

Figura 3.9.4 Método para eliminar datos de un archivo RDF

3.3.4 Métodos de los servicios REST

Los métodos para la implementación de los servicios REST tienen como base las clases de Jersey. En las siguientes figuras (3.9.5 y 3.9.6) se encuentran los ejemplos de la consulta y modificación de los datos del archivo RDF.

```

35     /**
42     @GET
43     @Path("/teacher")
44     @Produces(MediaType.APPLICATION_JSON)
45     public Response login(@QueryParam("email") String email, @QueryParam("password") String password) throws Exception {
46         Persona teacher = service.login(email, password);
47         GenericEntity<Persona> teacherEntity = new GenericEntity<Persona>(teacher);
48         return Response.status(200).entity(teacherEntity).build();
49     }
50
51     /**
57     @GET
58     @Path("/student")
59     @Produces(MediaType.APPLICATION_JSON)
60     public Response getStudents() throws Exception {
61         List<Persona> studentsList = service.findAllStudents();
62         GenericEntity<List<Persona>> studentsEntity = new GenericEntity<List<Persona>>(studentsList);
63         return Response.status(200).entity(studentsEntity).build();
64     }

```

Figura 3.9.5 Métodos para la solicitud de lectura de datos

```

183     @PUT
184     @Path("/student/reports")
185     @Consumes(MediaType.APPLICATION_FORM_URLENCODED)
186     public Response writeReport(@FormParam("curp") String curp, @FormParam("report") String report) throws IOException {
187         @SuppressWarnings("UnusedAssignment")
188         Response response = null;
189         try{
190             if(curp.equals("") || report.equals("")) response = Response.notAcceptable(null).build();
191             else {
192                 service.writeReport(curp, report);
193                 response = Response.ok().build();
194             }
195         } catch(Exception e) { response = Response.serverError().build();}
196         return response;
197     }
198 }

```

Figura 3.9.6 Métodos para la solicitud de modificación de datos

3.4 Pruebas

Las pruebas se realizan para asegurar el correcto funcionamiento de secciones de código.

3.4.1 Pruebas de consulta de datos por medio de la ontología en la aplicación Web

Para verificar la integridad del código se realizaron las pruebas correspondientes a la consulta de información mediante el uso de la ontología basada en el vocabulario FOAF. Los resultados son ilustrados en este apartado.

3.4.1.1 Inicio de sesión

Antes de que el usuario pueda acceder al sistema, él debe autenticarse. En este proceso se realiza una lectura del archivo RDF recibiendo como parámetros de búsqueda, el correo electrónico del usuario y su contraseña tal como se observa en la figura 3.9.7.

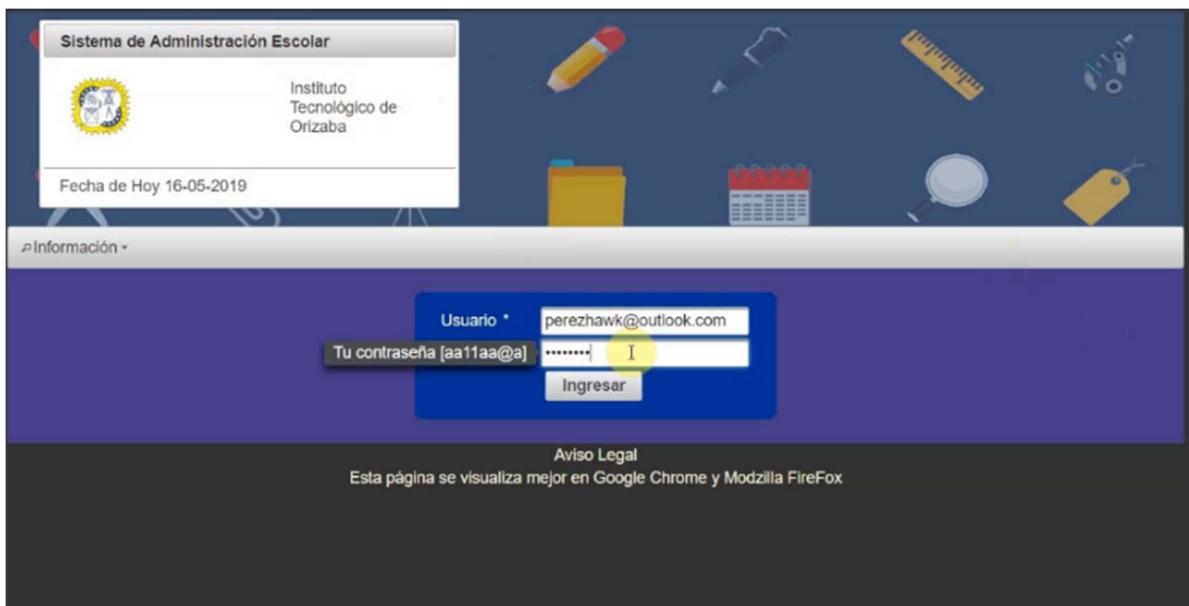


Figura 3.9.7 Pantalla de inicio de sesión

3.4.1.2 Vista del menú de Administrador

Una vez validados los datos el usuario es redireccionado a la página correspondiente a su tipo (administrador o docente). Ahí se muestra el MegaMenu que despliega una lista con las acciones que es posible realizar. En la figura 3.9.8 se muestra el despliegue de la lista al pasar el puntero por encima de una de las pestañas del MegaMenu. En la parte superior de la pantalla se muestra la foto, el nombre y un saludo para quien acaba de ingresar al sistema.

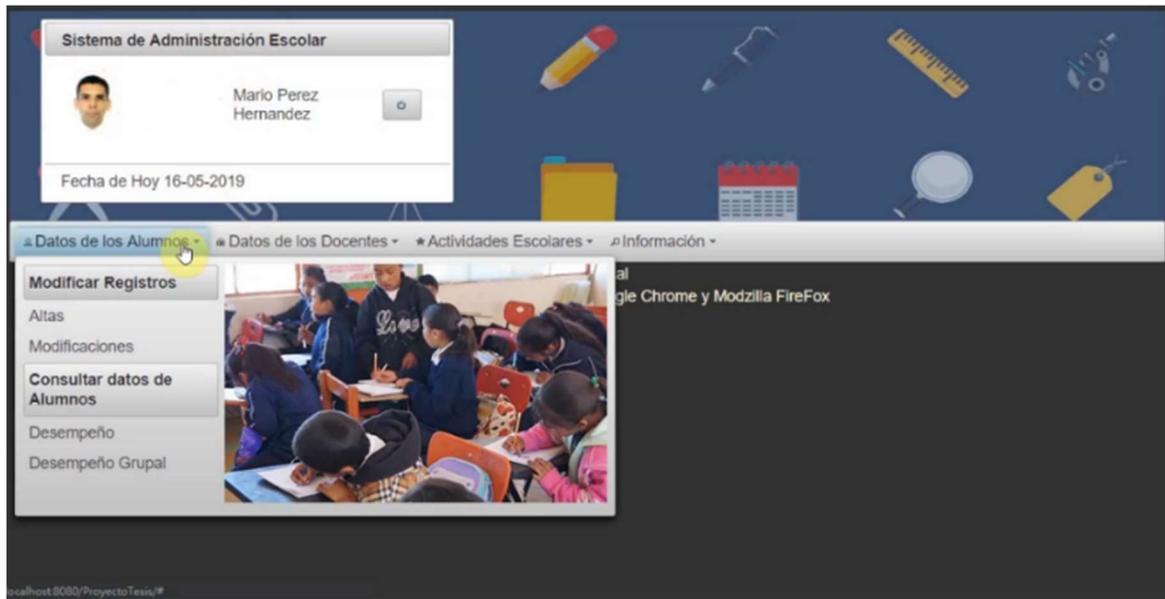


Figura 3.9.8 Pantalla con menú de administrador

3.4.1.3 Vista de la información general de los alumnos

Al seleccionar *Desempeño* se carga la información en el área de contenido. Una tabla con todos los registros de los alumnos aparece en pantalla, el administrador tiene la oportunidad de consultar los datos relacionados del alumno queha elegido (figura 3.9.9).

Registros de los Estudiantes											
CURP	Nombre(s)	Apellido Paterno	Apellido Materno	Fecha de Nacimiento	Género	Foto	Datos del Padre o Tutor	Apellido Paterno	Apellido Materno	Correo Electrónico	C
RAMF060530HVZMRR44	Alumno 16	Ramirez	Morales	2012/09/30	Masculino		X	X	X	a@outlook.com	2281
JIVA060530HVZMRR44	Alumno 16	Ramirez	Morales	2012/09/30	Femenino		X	X	X	a@outlook.com	2281
COMJ060607MVZZDRA8	Alumno 6	Contreras	Martinez	2012/09/30	Femenino		X	X	X	a@outlook.com	2281

Figura 3.9.9 Vista de la tabla con información de los alumnos

3.4.1.4 Vista de los compañeros relacionados con el alumno seleccionado

Al seleccionar a un estudiante, su información se muestra en una ventana sobrepuesta, por medio de la ontología se accede a la información respecto de las amistades del alumno seleccionado tal como se muestra en la figura 3.9.9.1.

Compañeros de Clase Cercanos			
	C 2	C 3	C 4
F 2			
	Promedio: 78.2	Promedio: 71.5	Promedio: 76.8
F 3			
	Promedio: 73.4	Promedio: 71.1	Promedio: 85.9
F 4			
	Promedio: 64.4	Promedio: 67.4	Promedio: 66.6

Figura 3.9.9.1 Ejemplo de los amigos relacionados con el alumno seleccionado

3.4.1.5 Prueba de consulta de la información de un grupo seleccionado

En el caso de la consulta de información por grupo, se realizaron las pruebas exitosamente, obteniendo la información específica de cada grupo y mostrándola en gráficas (figura 3.9.9.2).

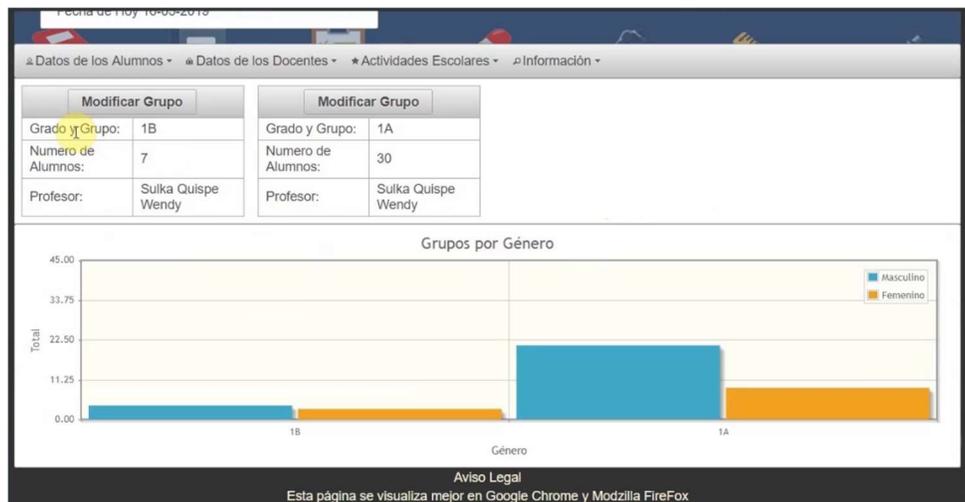


Figura 3.9.9.2 Vista de la prueba realizada para la obtención de datos con respecto al género de los alumnos

3.4.2 Pruebas de consulta y modificación de datos de la aplicación móvil

Para verificar la conectividad entre los dispositivos móviles y el servidor se realizaron varias pruebas desde un teléfono inteligente Xiao Mi modelo A2 como cliente. Se muestran algunas capturas de la aplicación funcionando.

3.4.2.1 Inicio de sesión

Los datos que se solicitan al usuario son los mismos que los de la aplicación Web, el método para acceder a los datos corresponde a aquel anotado con la palabra clave GET pues sólo se requiere leer la información. La figura 3.9.9.3 muestra la pantalla de inicio de sesión y la pantalla de la información del usuario que inició la sesión.

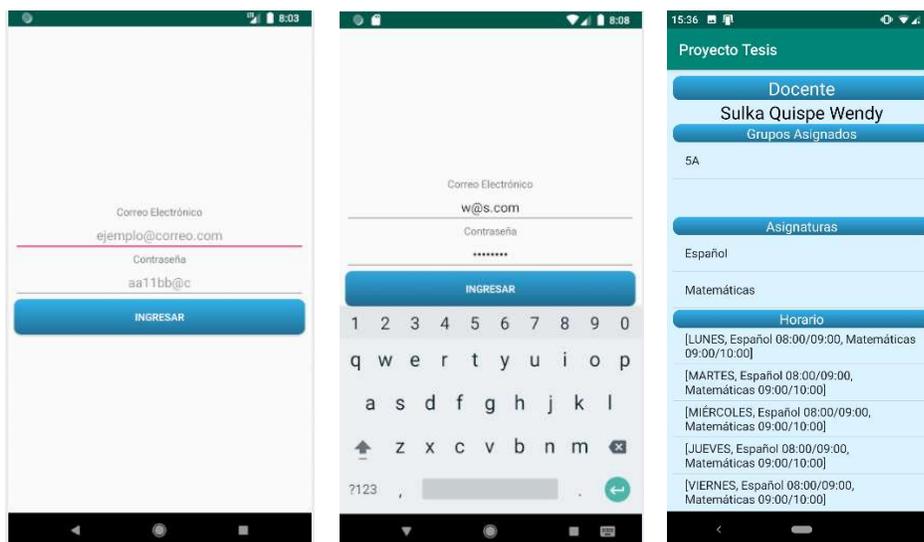


Figura 3.9.9.3 Vista de la pantalla de inicio de sesión y datos usuario con sesión iniciada

3.4.2.2 Operaciones de consulta de datos

La aplicación móvil en su interfaz de usuario constantemente muestra los datos de los alumnos registrados en el archivo RDF, esto con la intención de mantener al usuario informado de todos los datos académicos de los alumnos que requieren más atención ya sea por mostrar un bajo rendimiento o mala conducta (figura 3.9.9.4).

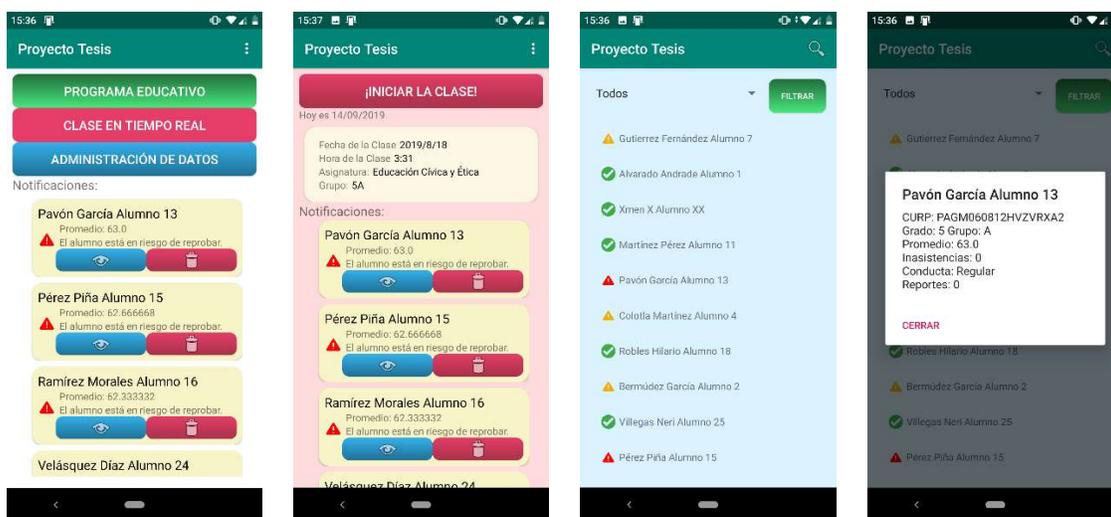


Figura 3.9.9.4 Vista de las pantallas que muestran información obtenida por los servicios REST

3.4.2.3 Operaciones de modificación de datos

De la misma manera se probaron las operaciones correspondientes a la modificación del archivo RDF desde el cliente por medio de la app móvil. Se muestran las capturas de pantalla de la aplicación siendo utilizada en los menús que utilizan métodos de modificación de datos tales como modificar asistencia y reportar alumno. En la figura 3.9.9.5 de izquierda a derecha se encuentran la pantalla de clase en tiempo real, la pantalla de creación de una nueva planeación y la pantalla de configuración del envío de reportes.

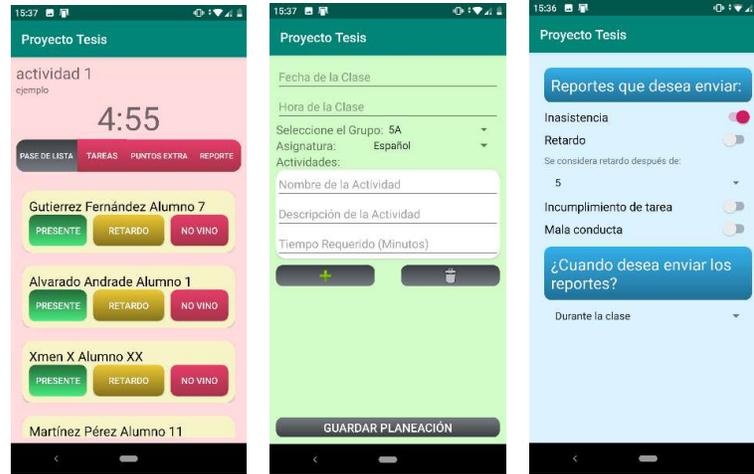


Figura 3.9.9.5 Vista de las pantallas de los menús que afectan los datos en el servidor por medio de los servicios REST

3.5 Fase de implementación

Corresponde al proceso por el cual los programas desarrollados son transferidos apropiadamente al computador destino, inicializados, y eventualmente, configurados; todo ello con el propósito de ser utilizados por el usuario final.

Esto incluye la implementación de la arquitectura, el modelo del usuario, la interfaz del usuario, los mecanismos adaptativos y las tareas referentes a la integración de todas estas implementaciones.

3.5.1 Aplicación Web

La implementación de la aplicación Web se realizó en un servidor local, no obstante, también se realizó el despliegue en el servidor de Jelastic el cual permite a los usuarios registrados probar sus servicios por un tiempo limitado a 15 días, aún siendo un periodo corto, fue tiempo suficiente para probar la aplicación por medio de su interfaz de usuario y como proveedor de servicios REST.

3.5.2 Aplicación Móvil

Para la instalación y fácil acceso a la aplicación móvil se realizó el despliegue en la Google Playstore, de este modo cualquier usuario puede adquirirla sin dificultades, está especificado

que la aplicación no funciona por sí sola, se requiere de estar registrado en la base de datos de la institución que está haciendo uso de la aplicación Web E-Clases.

Se añadió al menú principal la opción de configuración de conexión, de esta forma el usuario final puede elegir entre conectarse a una red local o a un servidor en la Web. Para la conexión local se utiliza la dirección IP acompañada del puerto del *host*, y para utilizar un servidor en la Web se utiliza el nombre del dominio donde fue desplegada la aplicación Web.

Capítulo 4 Resultados

Este capítulo describe los resultados obtenidos al realizar las pruebas y las decisiones que fueron tomadas para llegar a ellos.

4.1 Caso de estudio

Los datos que se utilizaron para las pruebas del sistema fueron proporcionados por la escuela primaria Ramón López Velarde, estos datos contienen los registros de dos grupos de educación básica correspondientes al quinto grado, ambos grupos con un total de 25 alumnos.

4.1.1 Las evaluaciones de los alumnos

La forma en la que es evaluado cada alumno se basa en rasgos seleccionados por el docente en algunos casos, es decir, la institución permite a los docentes asignar un porcentaje de la calificación a alguna actividad que ellos elijan; para resolver esta problemática se diseñó el archivo RDF con el campo “foaf:extra(n)” lo que permite al usuario docente asignar una calificación al rasgo seleccionado sin la necesidad de cambiar la estructura de la ontología establecida.

La aplicación Web principalmente, fue ajustada en base a los registros del docente por lo que el sistema en esta versión cumple únicamente con los requerimientos de una institución educativa de nivel básico, no obstante, el sistema tiene la capacidad de ser utilizado ampliamente en cualquier otro nivel educativo realizando pocos ajustes. Para los fines de este proyecto y el tiempo disponible para su realización fue importante mantener el caso de estudio limitado a escuelas primarias, dejando su implementación en otros niveles educativos para trabajos a futuro fuera de este proyecto.

La funcionalidad CRUD de la aplicación Web del sistema se implementó correctamente, los datos son modificados en nivel de administrador y docente, siendo el primero capaz de modificar los datos personales y académicos de los alumnos y los docentes, y el segundo sólo los datos académicos de los alumnos a su cargo. Esta parte del proyecto es la base para la inserción de las técnicas de Web Semántica, pues si bien la funcionalidad CRUD no representa a simple vista el uso de una ontología, el uso de RDF cobra sentido al momento de realizar las consultas y es justo en ese momento cuando el análisis de los alumnos se nutre de las relaciones establecidas en la ontología.

4.2 Evaluación de los resultados

A continuación, se presentan los análisis de los resultados obtenidos, en relación con lo presentado en el punto anterior, el tener un caso de estudio de cierto nivel educativo definió; los datos disponibles para consultas y modificaciones realizadas por el docente, las funciones agregadas a la aplicación móvil, la manera en que las relaciones se establecen entre los alumnos.

4.2.1 Establecimiento de las relaciones entre los alumnos utilizando el vocabulario FOAF

Una de las limitaciones del caso de estudio seleccionado fue la manera en la que se relacionan los alumnos en los niveles básicos, pues en estos casos un grupo de alumnos es estático y tiene siempre al mismo docente encargado. Para vincular a un alumno con otro las operaciones se realizaron de forma manual, es decir, el docente seleccionó cuáles alumnos se conocen entre sí, siendo de diferentes grupos. En el caso de la relación establecida por lugar en el aula fue de la misma manera, cualquier alumno perteneciente a un grupo es relacionado por medio del vocabulario “foaf:member” con cualquier otro que sea miembro del mismo grupo, esto de forma automática desde que al alumno se le es asignado su grupo, pero la relación de amigos cercanos también se genera manualmente cuando el docente organiza el lugar de los alumnos dentro del aula en el menú de la aplicación Web “Asignar lugar en el aula”.

Esta limitación no impide probar las funciones de consulta de datos enriquecidos, siempre que el docente conozca y esté al tanto de los alumnos, mantiene al sistema informado de estas relaciones, lo que finalmente le permite tener más información útil de sus estudiantes. En modelos educativos donde el alumno debe cambiar de grupo en cada clase, estas relaciones se implementarían automáticamente.

La figura 4.1 muestra las relaciones alumno-posición, en el grupo de 5A. Es importante mencionar que no se toman en cuenta las características clínicas de los alumnos, por ejemplo, cuando un estudiante tiene problemas de la vista y el profesor no se ha percatado de ello, estos datos tienen relevancia, pero está fuera de las capacidades del sistema analizar estas situaciones. El resultado obtenido muestra una fuerte tendencia a bajas calificaciones en la fila 4.

Grado 5		Grupo A		Ver Información	
Nombre	Alumno 1	Nombre	Alumno 9	Nombre	Alumno 11
Promedio	81.8	Promedio	91.9	Promedio	83.6
Nombre	Alumno 14	Nombre	Alumno 12	Nombre	Alumno 4
Promedio	88.2	Promedio	89.2	Promedio	76.6
Nombre	Alumno 19	Nombre	Alumno 2	Nombre	Alumno 10
Promedio	78.2	Promedio	71.5	Promedio	76.8
Nombre	Alumno 17	Nombre	Alumno 25	Nombre	Alumno 7
Promedio	77.6	Promedio	87.6	Promedio	73.4
Nombre	Alumno 5	Nombre	Alumno 18	Nombre	Alumno 3
Promedio	71.1	Promedio	85.9	Promedio	65.4
Nombre	Alumno 24	Nombre	Alumno 6	Nombre	Alumno 8
Promedio	67.8	Promedio	64.4	Promedio	67.4
Nombre	Alumno 13	Nombre	Alumno 21	Nombre	Alumno 22
Promedio	66.6	Promedio	72.9	Promedio	61.6
Nombre	Alumno 15	Nombre	Alumno 16	Nombre	Alumno 20
Promedio	64.6	Promedio	63.8	Promedio	87.6
Nombre	Alumno 23	Nombre	Alumno 21	Nombre	Alumno 22
Promedio	81.9	Promedio	72.9	Promedio	61.6

Figura 4.1 Relación alumno-posición. Marcados con borde rojo los alumnos con promedio menor a 70 y con amarillo menor a 80 y mayor a 70 para enfatizar la relación

En la figura 4.2 se seleccionó al Alumno 13 dado que es el alumno con más compañeros a su alrededor de promedios bajos, siendo sólo 2 de 8 compañeros aquellos con promedio mayor a 80.

	C 3	C 4	C 5
F 3			
	Promedio: 71.1	Promedio: 85.9	Promedio: 65.4
F 4			
	Promedio: 67.4	Promedio: 66.6	Promedio: 64.6
F 5			
	Promedio: 72.9	Promedio: 61.6	Promedio: 81.9

4.2 Cuadrícula de selección de alumno donde se ha seleccionado a Alumno 13

Para el grupo de 5A se expone en la tabla 2 lo mencionado anteriormente. Se observa a simple vista el sector del salón donde ocurren las calificaciones más bajas, este dato es analizado por el personal académico para tomar las decisiones pertinentes en base a la experiencia del profesor a cargo del grupo, donde su convivencia constante con cada estudiante determina si lo que ocurre es provocado por la mala conducta (que es el factor más fácil de detectar dentro del aula) o se debe a alguna situación que aún no ha sido identificada.

Número de fila	Alumnos con promedio menor a 7	Alumnos con promedio de 7 a 8	Alumnos con promedio mayor a 8
1	0	0	5
2	0	5	0
3	1	2	2
4	5	0	0
5	2	1	2

Tabla 2. Relación de la calificación de los alumnos con el lugar que ocupan dentro del aula

Otro de los datos que apoyan el análisis que cada catedrático lleva a cabo respecto de sus alumnos es el control de reportes de los alumnos, los cuales se consultan de diversas formas dentro de la aplicación Web y la aplicación móvil. Para el grupo de 5A se tiene que, la mayoría de los reportes levantados en un periodo de un mes corresponde a incumplimiento de tareas, segundo factor el incumplimiento de tareas y por último los reportes de mala conducta. Otro hecho a considerar es que los alumnos que aparecen como reportados frecuentemente son; Alumno 24, Alumno 8, Alumno 13, Alumno 15, Alumno 16 y Alumno 22, todos ellos pertenecientes al sector de riesgo. En la tabla 3 se muestra la relación de los alumnos reportados.

Alumno Reportado	Reportes por Inasistencia	Reportes por Incumplimiento	Reportes por mala conducta
Alumno 3	0	4	0
Alumno 4	2	0	0
Alumno 8	1	3	1
Alumno 13	0	3	0
Alumno 15	0	5	0
Alumno 16	1	4	1
Alumno 22	0	3	0
Alumno 24	0	4	0
Total:	4	26	2

Tabla 3. Relación de alumnos reportados correspondiente a un mes

4.2.2 Funciones agregadas a la aplicación móvil

Para el manejo de la clase en tiempo real se agregó la función de “Reportes en tiempo real” con la cual el profesor puede enviar reportes de manera instantánea; de mala conducta, de inasistencia o de incumplimiento, a los padres del alumno que cometió la falta. Es posible que esta función no tenga mucha utilidad en niveles educativos donde los estudiantes son adultos, pero en el escenario contrario es una herramienta útil para mantener a los padres de familia informados. La figura 4.3 muestra las pantallas donde se registran los reportes.

Las pruebas se realizaron con un correo propio debido a las políticas de privacidad por las cuales la escuela no nos proporcionó datos personales de sus alumnos ni de los padres de familia.

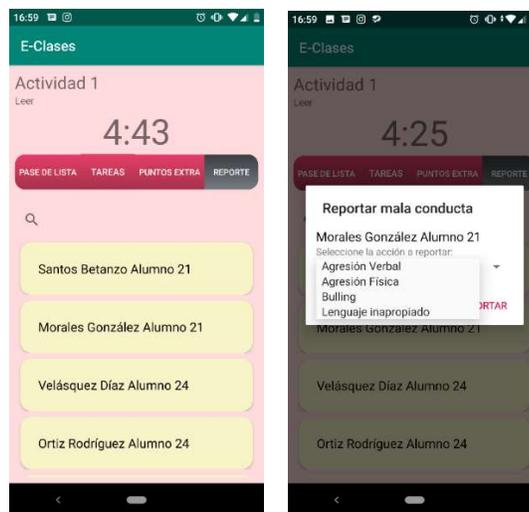


Figura 4.3 Vista de “Clase en tiempo real” con la opción de reportes seleccionada y la vista del menú de selección de reporte habiendo elegido a un alumno.

Los reportes son registrados en la base de datos y si el usuario ha seleccionado el envío inmediato, se activa un menú para seleccionar el servicio de correo electrónico con el cual se enviará el reporte. Los datos del emisor, destinatario, asunto y contenido del mensaje, son llenados automáticamente por la aplicación, el usuario sólo requiere confirmar el envío. Esta tarea toma entre 5 y 10 segundos para realizarla, por lo que el flujo de la clase no se ve afectado por llevar a cabo esta tarea de forma inmediata.

4.2.3 Datos disponibles para consulta del docente

Respecto a los datos que cada profesor consulta y modifica de los alumnos que tiene a su cargo, se añadió la opción de exportación de las tablas en formato PDF ya que el personal académico en ocasiones requiere estos datos impresos. En la figura 4.4 se muestra el menú al que tiene acceso el docente donde se encuentran los alumnos que han sido reportados por mala conducta, inasistencia o incumplimiento.

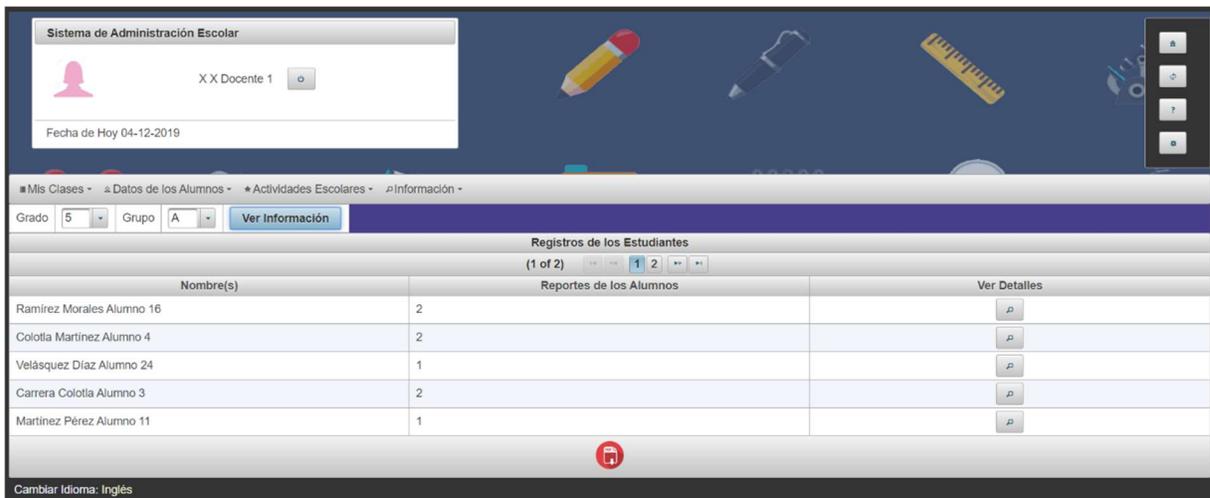


Figura 4.4 Captura de pantalla del menú de consulta de alumnos reportados

El docente entonces selecciona al alumno del cual desea obtener la información detallada de cada reporte como se muestra en la figura 4.4.1. En ambos casos se exporta la información al dar click al botón rojo.



Figura 4.4.1 Detalle de la captura de pantalla que muestra el desglose de reportes pertenecientes a un alumno seleccionado.

De la misma manera se obtiene el archivo PDF de las calificaciones de todos los alumnos pertenecientes a un grupo seleccionado, como se muestra en la figura 4.4.2, donde el menú está disponible tanto para docentes como para el coordinador académico.

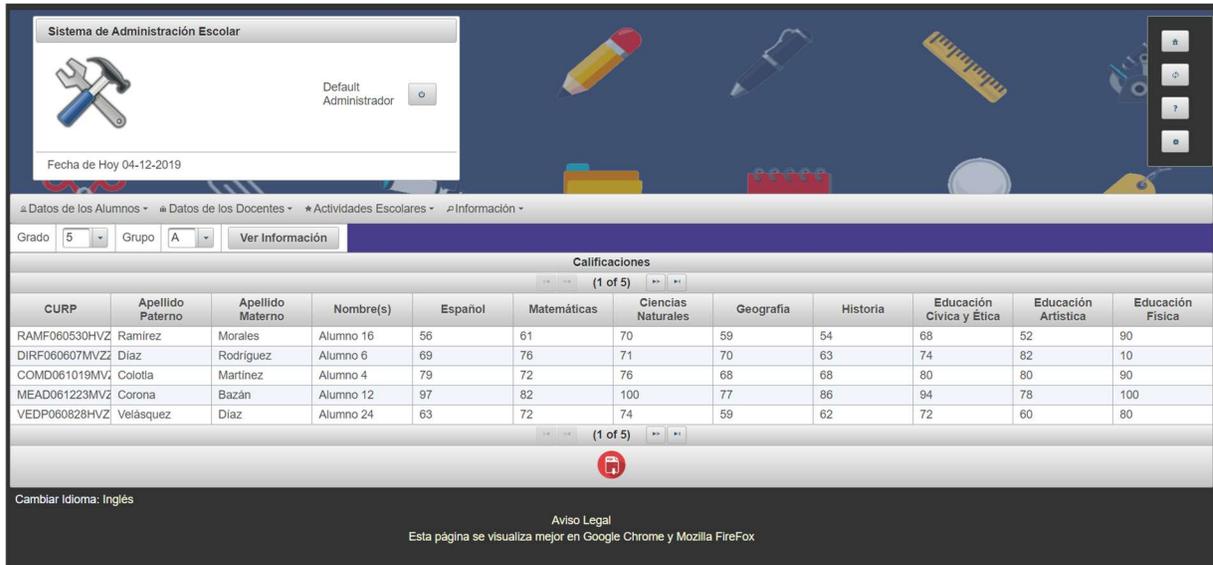


Figura 4.4.2 Captura de pantalla que muestra el menú de las calificaciones de los alumnos

Se observa en las siguientes figuras las tablas generadas por el exportador PDF descargadas desde la aplicación; registro de reportes (figura 4.4.3), reportes en detalle por alumno (figura 4.4.4) y calificaciones generales de un grupo seleccionado (figura 4.4.5).

Registros de los Estudiantes	
Nombre(s)	Reportes de los Alumnos
Ramírez Morales Alumno 16	2
Colotla Martínez Alumno 4	2
Velasquez Díaz Alumno 24	1
Carrera Colotla Alumno 3	2
Martínez Pérez Alumno 11	1
Huerta Vargas Alumno 8	3
Pérez Piñal Alumno 15	3
Tzanahua Cano Alumno 22	1
Pavón García Alumno 13	2

Figura 4.4.3 Vista del PDF generado para los reportes totales de los alumnos

Huerta Vargas Alumno 8		
Fecha	Concepto	Descripción
2019-05-06	Retardo	El alumno llegó 30 minutos tarde
2019-06-07	Tarea	El alumno no cumplió con la tarea de español
2019-06-15	Mala conducta	El alumno agredió a un compañero

Figura 4.4.4 Vista del PDF generado para los reportes pertenecientes a un alumno seleccionado

CURP	Apellido Paterno	Apellido Materno	Nombre(s)	Calificaciones								
				Español	Matemáticas	Ciencias Naturales	Geografía	Historia	Educación Cívica y Física	Educación Artística	Educación Física	
RAMF060530HV ZMRRA4	Ramírez	Morales	Alumno 16	56	61	70	59	54	68	52	90	
DIRF060607MV ZZDRA8	Díaz	Rodríguez	Alumno 6	69	76	71	70	63	74	82	10	
COMD061019MV VZLRNA9	Colotla	Martínez	Alumno 4	79	72	76	68	68	80	80	90	
MEAD061223MV VZLSNA6	Corona	Bazán	Alumno 12	97	82	100	77	86	94	78	100	
VEDP060828HV ZLZBA2	Velázquez	Díaz	Alumno 24	63	72	74	59	62	72	60	80	
COBE061016HV ZRZDA0	Corona	Bazán	Alumno 5	62	65	71	66	63	79	83	80	
CKCA050617HV ZRLNA1	Carrera	Colotla	Alumno 3	61	65	66	61	58	70	80	62	
ROPM060613MV VZSZNAS	Rosales	Peña	Alumno 19	73	74	86	75	72	88	68	90	
MAPA060711MV VZRLA3	Martínez	Pérez	Alumno 11	82	92	87	74	86	91	90	67	
SACM060809MV VZNLNAS	Santiago	Celis	Alumno 20	88	88	92	88	77	90	78	100	
AAAE060131HV ZLNLA6	Alvarado	Andrade	Alumno 1	77	74	96	81	66	84	96	80	
VINA060601MV ZLRDA5	Villegas	Neri	Alumno 25	97	81	90	77	80	90	86	100	
HUVM060314HV VZRRRA6	Huerta	Vargas	Alumno 8	69	57	70	67	54	72	60	90	
PEPH060125HV ZRNVZA9	Pérez	Piña	Alumno 15	62	57	69	60	55	73	61	80	
GUFJ061221MV ZTRNA2	Gutiérrez	Fernández	Alumno 7	76	72	72	72	67	65	73	90	
TACG060417HV ZZNVZA9	Tzannahua	Cano	Alumno 22	59	59	64	55	56	48	72	80	
JAHS060503HV ZMRMA7	Jaimés	Hernández	Alumno 9	94	91	92	85	79	100	94	100	
ROAA060519MV VZBGRAS	Robles	Aguilar	Alumno 17	88	66	90	69	68	90	70	80	
ROHB061115HV ZBLRA9	Robles	Hilario	Alumno 18	85	87	82	77	80	98	78	100	
PAGM060812HV VZVRCA2	Pavón	García	Alumno 13	58	61	70	58	58	76	72	80	
VAOA051021HV VZSRLA2	Vázquez	Orosco	Alumno 23	74	78	94	71	66	92	80	100	
SABR061228MV ZNTSA0	Santos	Betanzo	Alumno 21	65	68	75	70	65	79	71	90	
PEMT060519MV ZRRNA1	Pérez	Martínez	Alumno 14	90	86	86	80	84	96	84	100	
LOOU060125HV ZPRZA2	López	Ortiz	Alumno 10	77	75	77	72	67	93	73	80	
BEGP060513MV ZRRTA5	Bermúdez	García	Alumno 2	80	78	81	76	73	89	85	10	

Figura 4.4.5 Vista del PDF generado para las calificaciones de un grupo seleccionado

Capítulo 5 Conclusiones y recomendaciones

Los objetivos de este proyecto se cumplieron completamente dando como resultado las aplicaciones Web y móvil con el funcionamiento propuesto y la adición extraordinaria del predictor de conducta para la aplicación Web.

5.1 Conclusiones

Dada la importancia de la clara administración de los datos de los alumnos de cualquier institución educativa, se propuso un sistema de gestión escolar capaz de ofrecer al personal académico información más completa de sus estudiantes, esto por medio del análisis de los perfiles de los usuarios utilizando el esquema RDF y el vocabulario FOAF. Además del manejo de la información, era muy importante el diseño de una herramienta que no complicara el trabajo de los docentes, sino lo contrario, de esta idea surge el crear un sistema cuya funcionalidad se completa entre una aplicación Web y una aplicación móvil.

El propósito de este proyecto se cumplió para el nivel de educación básica, dejando la base para lo que puede ser el desarrollo de un trabajo más ambicioso en el cual ningún nivel escolar queda fuera. Las herramientas elaboradas permitieron encontrar información que normalmente pasa desapercibida, además de ser fáciles de usar.

La metodología UWE permitió tener mayor control sobre las funcionalidades de ambas aplicaciones del sistema ya que por medio de los diagramas UML se puede obtener un resultado más acorde con los objetivos que se planea alcanzar, incluso algunos ajustes que surgieron posteriormente debido a imprevistos provenientes del caso de estudio y del servidor de datos, se realizaron sin inconvenientes gracias al orden establecido en los diagramas UML.

Para la implementación de la ontología en el esquema RDF con vocabulario FOAF se requirió una cantidad considerable de tiempo debido a que no se encontraron fuentes más claras sobre el uso de FOAF en otros artículos científicos, por lo que se utilizó la documentación oficial. Muchos de los campos que aparecen en cada individuo ya sea “foaf:Person” o “foaf:Group” también tienen el prefijo “foaf” esto con el fin de tener un recurso homogéneo ya que esos campos no están especificados en el vocabulario como tal.

Las pruebas realizadas permitieron ver que las funciones propuestas para el sistema se pueden mejorar en gran medida introduciendo inteligencia artificial. Para la aplicación Web se introdujo el predictor de conducta y se creó un modelo con los datos de los alumnos; es sólo un ejemplo de las posibilidades que puede alcanzar este sistema. El predictor de conducta alcanzó una precisión del 92%, aunque la muestra constaba sólo de 25 estudiantes, los datos seleccionados para realizar el entrenamiento del modelo fueron sus calificaciones, dato que se encuentra en todas las entidades de alumno y que en relación a las pruebas realizadas arrojó resultados confiables.

El rápido avance de la tecnología tiene la gran ventaja de bajar los costos de aquellos dispositivos que ya no están a la “moda”, este es un factor que debe aprovecharse para el establecimiento de nuevas estrategias educativas. Este proyecto abre las puertas al aprendizaje electrónico al introducir al teléfono inteligente como herramienta de trabajo dentro del aula no como una aplicación que puede usarse para realizar algunas actividades, sino como una aplicación que debe usarse para concretar el trabajo día a día del catedrático. Queda abierta la posibilidad de una segunda fase donde cada alumno puede utilizar su propio teléfono inteligente para cumplir con su programa educativo.

5.2 Recomendaciones

Si bien este trabajo cumple su objetivo, su funcionamiento ya puede ser considerado “anticuado” pues la tendencia de la tecnología actual apuesta a la automatización de las tareas de todo tipo, por lo que el paso inmediato es la introducción de inteligencia artificial para vincular a un alumno con otro de forma automática, la elaboración de las planeaciones de clases basada en el uso cotidiano del docente, o sugerencias para el apoyo a estudiantes en riesgo de reprobación. Las opciones son vastas y estos sistemas pueden estar al alcance de muchos.

Significaría una gran aportación a la educación si se continúa buscando la inserción de los *smartphones* en los salones de clases, pero esta vez en manos de los alumnos. Este es un reto cuya dificultad no se encuentra en la elaboración del sistema sino en la conducta de los usuarios [31], pues se tiene que considerar la distracción que producen estos dispositivos que si no están bien regulados pueden ser perjudiciales para el aprendizaje.

Para el caso de estudio un modelo educativo donde los estudiantes deben cambiar de grupo en cada clase (como generalmente se hace en los modelos universitarios) ofrecería más opciones con respecto a la implementación de la ontología y la inteligencia artificial. El sistema aquí presentado es una base para futuros proyectos enfocados en la adopción del aprendizaje electrónico.

Productos Académicos



Aplicación Android disponible en la Google Playstore bajo el nombre de E-clases. Publicada el 29 de octubre de 2019. La aplicación es gratuita y la última actualización fue el día 8 de noviembre de 2019 en su versión 1.0.3. Se requiere estar registrado en el sistema de la aplicación Web para utilizarla.

Aplicación Web E-Clases para gestión escolar.



Mario Ricardo Pérez Hernández, Luis Ángel Reyes Hernández, Celia Romero Cruz, Gustavo Peláez Camarena y Beatriz Alejandra Olivares Zepahua.

Resource Description Framework and Friend Of A Friend-Based Web and Mobile Applications Prototype in an Education Environment. Ciudad de México, México. Estatus: Aceptado (Presentado 25-10-19). Indizado en la biblioteca virtual de IEEEExplore en “2019 7th International Conference in Software Engineering Research and Innovation (CONISOF)”

Referencias

- [1] “Panorama de la educación 2015”, OCDE. [Online]. Available: <https://www.oecd.org/mexico/Education-at-a-glance-2015-Mexico-in-Spanish.pdf> [Accessed: 30/noviembre/2018].
- [2] “Encuesta Nacional sobre Disponibilidad y Uso de Tecnologías de la Información en los Hogares”, INEGI. [Online]. Available: http://www.inegi.org.mx/saladeprensa/boletines/2016/especiales/especiales2016_03_01 [Accessed: 30/noviembre/2018].
- [3] F. Kanwal and M. Rehman, "Factors Affecting E-Learning Adoption in Developing Countries–Empirical Evidence From Pakistan’s Higher Education Sector," in *IEEE Access*, vol. 5, pp. 10968-10978, 2017.
- [4] Jian Wang, Panpan Gao, Yutao Ma, Keqing He, Patrick C. K. Hung, "A Web Service Discovery Approach Based on Common Topic Groups Extraction", *Access IEEE*, vol. 5, pp. 10193-10208, 2017.
- [5] “Invocación asíncrona a los servicios web JAX-WS”, IBM. [Online]. Available: https://www.ibm.com/support/knowledgecenter/es/SSAW57_9.0.0/com.ibm.websphere.nd.mutlplatform.doc/ae/twbs_jaxwsclientasync.html [Accessed: 19/marzo/2019].
- [6] Lagares Angel, Florian Daniel, Boulaem Benatallah, “Web Service Composition: A Survey of Techniques and Tools”, *ACM Computing Surveys (CSUR)*, vol. 48, Issue 3, February 2016.
- [7] Frederik Bülthoff and Maria Maleshkova. 2014. RESTful or RESTless-Current state of today's top Web APIs. In *European Semantic Web Conference*. Springer, 64--74.
- [8] H. Gasmi and A. Bouras, "Ontology-Based Education/Industry Collaboration System," in *IEEE Access*, vol. 6, pp. 1362-1371, 2018.
- [9] A. T. A. Wardhana and H. T. Nugroho I., "Combining FOAF and music ontology for music concerts recommendation on facebook application," 2013 Conference on New Media Studies (CoNMedia), Tangerang, 2013, pp. 1-5.
- [10] “What is an ontology?”, Steven, Roberts. [Online]. Available: <http://www.cs.man.ac.uk/~stevensr/onto/node3.html> [Accessed: 30/noviembre/2018].
- [11] Triplestore [Online]. Available: <https://en.wikipedia.org/wiki/Triplestore>, [Accessed: 1/Junio/2019]
- [12] “Definición de FOAF”. [Online] Available <http://www.foaf-project.org/> [Accessed: 16/Octubre/2018].

- [13] F. Ali *et al.*, "A Fuzzy Ontology and SVM–Based Web Content Classification System," in *IEEE Access*, vol. 5, pp. 25781-25797, 2017.
- [14] J. Puttonen, A. Lobov and J. L. Martinez Lastra, "Semantics-Based Composition of Factory Automation Processes Encapsulated by Web Services," in *IEEE Transactions on Industrial Informatics*, vol. 9, no. 4, pp. 2349-2359, Nov. 2013.
- [15] J. Yang, Z. X. Huang, Y. X. Gao and H. T. Liu, "Dynamic Learning Style Prediction Method Based on a Pattern Recognition Technique," in *IEEE Transactions on Learning Technologies*, vol. 7, no. 2, pp. 165-177, April-June 2014.
- [16] F. Grünewald and C. Meinel, "Implementation and Evaluation of Digital E-Lecture Annotation in Learning Groups to Foster Active Learning," in *IEEE Transactions on Learning Technologies*, vol. 8, no. 3, pp. 286-298, 1 July-Sept. 2015.
- [17] K. Chin, K. Lee and Y. Chen, "Impact on Student Motivation by Using a QR-Based U-Learning Material Production System to Create Authentic Learning Experiences," in *IEEE Transactions on Learning Technologies*, vol. 8, no. 4, pp. 367-382, 1 Oct.-Dec. 2015.
- [18] "Plan de desarrollo institucional 2011-2015". [online] Available at: https://www.planeacion.unam.mx/consulta/Plan_desarrollo.pdf, Accessed 3 Dic. 2019
- [19] P. Li, Y. Wang, Y. Tian, T. Zhou and J. Li, "An Automatic User-Adapted Physical Activity Classification Method Using Smartphones," in *IEEE Transactions on Biomedical Engineering*, vol. 64, no. 3, pp. 706-714, March 2017.
- [20] G. Zhu and C. A. Iglesias, "Computing Semantic Similarity of Concepts in Knowledge Graphs," in *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 1, pp. 72-85, 1 Jan. 2017.
- [21] X. Hu, Z. Feng, S. Chen, K. Huang, J. Li and M. Zhou, "Accurate Identification of Ontology Alignments at Different Granularity Levels," in *IEEE Access*, vol. 5, pp. 105-120, 2017.
- [22] M. M. Al-Tarabily, R. F. Abdel-Kader, G. Abdel Azeem and M. I. Marie, "Optimizing Dynamic Multi-Agent Performance in E-Learning Environment," in *IEEE Access*, vol. 6, pp. 35631-35645, 2018.
- [23] M. M. Rahman and N. A. Abdullah, "A Personalized Group-Based Recommendation Approach for Web Search in E-Learning," in *IEEE Access*, vol. 6, pp. 34166-34178, 2018.
- [24] A. Moubayed, M. Injadat, A. B. Nassif, H. Lutfiyya and A. Shami, "E-Learning: Challenges and Research Opportunities Using Machine Learning & Data Analytics," in *IEEE Access*, vol. 6, pp. 39117-39138, 2018.
- [25] Ying Li, Li Li and Yunlu Zhang, "FOAF user classification based on SVM," 2013 IEEE 4th International Conference on Software Engineering and Service Science, Beijing, 2013, pp. 271-274.

- [26] Pekka Abrahamson, et al., "Mobile-D: An Agile Approach for Mobile Application Development", Companion to the 19th Annual ACM SIGPLAN, 2012, pp. 174 – 175.
- [27] Y.D. Amaya Balaguera, "Metodologías ágiles en el desarrollo de aplicaciones para dispositivos móviles. Estado actual", *Revista de Tecnología*, 2013, vol. 12, No. 2, pp. 111-124.
- [28] Oracle, "Java Developer Center" [Online]. Available: <http://www.oracle.com/technetwork/es/Java/index.html>. Accessed: Oct. 30, 2018.
- [29] JavaServer Faces, "Introducción a JavaServer Faces". [online] Available at: <http://www.jtech.ua.es/>, Accessed 24 Nov. 2018
- [30] PrimeFaces, "Why PrimeFaces" [online] Available at: <https://www.primefaces.org/whyprimefaces/>, Accessed 24 Nov. 2018.
- [31] Apache, Apache Jena [online] Available at: <https://jena.apache.org>, Accessed 6 May. 2019.
- [32] NetBeans, "NetBeans Docs & support," 2016. [Online]. Available: <https://netbeans.org/kb/index.html>. Accessed: Oct. 30, 2018.
- [33] S. Kandry, R. Roufayel, "How to use effectively smartphone in the classroom". *IEEE Global Engineering Education Conference (EDUCON)*. IEEE 2017.

Apéndice

Glosario

**u-learning*.- Supone una deliberada adaptación de la propuesta de enseñanza al contexto en el cual se encuentran inmersos los estudiantes. Está basado en el aprendizaje por dispositivos móviles.

**Middleware*.- Software que sirve como puente entre un sistema operativo o base de datos y las aplicaciones.

Anexos

Tabla A.1. Relación de tareas de los alumnos de 5A

NÚMERO DE TAREAS SOLICITADAS POR MATERIA														
ESPAÑOL	7	C. NATURALES	1	HISTORIA	2	E. ARTISTICA	1							
MATEMATICAS	7	GEOGRAFIA	2	FCYE	1									
TAREAS ENTREGADAS POR ALUMNO														
ALUMNO	ESPAÑOL	10%	MATEMATICAS	10%	C. NATURALES	10%	GEOGRAFIA	10%	HISTORIA	10%	FCYE	10%	EA	10%
Alumno 1	4.5	0.64	2.5	0.36	1	1.00	0.5	0.25	0	0.00	1	1.00	1	1.00
Alumno 2	6.5	0.93	5	0.71	1	1.00	1	0.50	0.5	0.25	1	1.00	1	1.00
Alumno 3	3.5	0.50	6.5	0.93	1	1.00	1	0.50	0.5	0.25	1	1.00	1	1.00
Alumno 4	5	0.71	4.5	0.64	1	1.00	0.5	0.25	0.5	0.25	1	1.00	1	1.00
Alumno 5	1	0.14	3	0.43	1	1.00	1	0.50	0.5	0.25	1	1.00	1	1.00
Alumno 6	6	0.86	5.5	0.79	0.5	0.50	2	1.00	0.5	0.25	1	1.00	1	1.00
Alumno 7	2	0.29	3.5	0.50	0.5	0.50	1	0.50	0	0.00	1	1.00	1	1.00
Alumno 8	5	0.71	2	0.29	1	1.00	1	0.50	0	0.00	1	1.00	1	1.00
Alumno 9	4	0.57	6.5	0.93	1	1.00	1	0.50	0.5	0.25	1	1.00	1	1.00
Alumno 10	4	0.57	5.5	0.79	1	1.00	1	0.50	0	0.00	1	1.00	1	1.00
Alumno 11	1	0.14	4	0.57	0	0.00	1	0.50	1.5	0.75	1	1.00	1	1.00
Alumno 12	6	0.86	7	1.00	1	1.00	1	0.50	2	1.00	1	1.00	1	1.00
Alumno 13	0	0.00	2	0.29	0	0.00	0	0.00	0	0.00	1	1.00	1	1.00
Alumno 14	6	0.86	7	1.00	1	1.00	2	1.00	2	1.00	1	1.00	1	1.00
Alumno 15	0.5	0.07	1.5	0.21	1	1.00	1	0.50	0	0.00	1	1.00	1	1.00
Alumno 16	1.5	0.21	5	0.71	1	1.00	1	0.50	0	0.00	1	1.00	1	1.00
Alumno 17	1	0.14	1.5	0.21	1	1.00	1	0.50	0	0.00	1	1.00	1	1.00
Alumno 18	6.5	0.93	6.5	0.93	1	1.00	1	0.50	1.5	0.75	1	1.00	1	1.00
Alumno 19	2.5	0.36	3	0.43	1	1.00	1	0.50	0.5	0.25	1	1.00	1	1.00
Alumno 20	7	1.00	7	1.00	1	1.00	2	1.00	1	0.50	1	1.00	1	1.00
Alumno 21	0	0.00	2	0.29	1	1.00	1	0.50	0	0.00	1	1.00	1	1.00
Alumno 22	2	0.29	6	0.86	1	1.00	1	0.50	0	0.00	1	1.00	1	1.00
Alumno 23	3	0.43	5.5	0.79	1	1.00	1	0.50	0	0.00	1	1.00	1	1.00
Alumno 24	6	0.86	6.5	0.93	1	1.00	1	0.50	1.5	0.75	1	1.00	1	1.00
Alumno25	6	0.86	6.5	0.93	1	1.00	1	0.50	1.5	0.75	1	1.00	1	1.00

Tabla A.2. Relación de exámenes de los alumnos de 5A

CALIFICACIÓN DE EXAMENES POR MATERIA Y POR ALUMNO														
ALUMNO	ESPAÑOL	40%	MATEMÁTICAS	40%	C. NATURALES	40%	GEOGRAFÍA	40%	HISTORIA	40%	FCYE	40%	EA	40%
Alumno 1	4	1.60	4	1.60	8	3.20	6	2.40	3	1.20	5	2.00	8	3.20
Alumno 2	3	1.20	3	1.20	3	1.20	3	1.20	3	1.20	5	2.00	4	1.60
Alumno 3	3	1.20	3	1.20	3	1.20	3	1.20	3	1.20	4	1.60	2	0.80
Alumno 4	4.5	1.80	3	1.20	3	1.20	3	1.20	3	1.20	4	1.60	4	1.60
Alumno 5	3	1.20	3	1.20	3	1.20	3	1.20	3	1.20	5	2.00	6	2.40
Alumno 6	3	1.20	5	2.00	4.5	1.80	3	1.20	3	1.20	4	1.60	6	2.40
Alumno 7	4.5	1.80	3	1.20	3	1.20	3	1.20	3	1.20	0	0.00	2	0.80
Alumno 8	5	2.00	3	1.20	4.5	1.80	5	2.00	3	1.20	5	2.00	2	0.80
Alumno 9	7	2.80	5.5	2.20	5.5	2.20	5	2.00	4	1.6	8	3.20	6	2.40
Alumno 10	4	1.60	3	1.20	3	1.20	3	1.20	3	1.2	7	2.80	2	0.80
Alumno 11	7	2.80	8.5	3.40	8.5	3.40	4	1.60	6.5	2.6	7	2.80	1	0.40
Alumno 12	7	2.80	3	1.20	8	3.20	3	1.20	4	1.6	6	2.40	2	0.80
Alumno 13	3	1.20	3	1.20	6	2.40	3	1.20	3	1.2	5	2.00	4	1.60
Alumno 14	6	2.40	4.5	1.80	4.5	1.80	3	1.20	4	1.6	7	2.80	4	1.60
Alumno 15	4.5	1.80	3	1.20	4	1.60	3	1.20	3	1.2	5	2.00	2	0.80
Alumno 16	3	1.20	3	1.20	4.5	1.80	3	1.20	3	1.2	4	1.60	0	0.00
Alumno 17	8.5	3.40	3	1.20	7	2.80	3	1.20	4	1.6	7	2.80	2	0.80
Alumno 18	4	1.60	4.5	1.80	3	1.20	3	1.20	3	1.2	7	2.80	2	0.80
Alumno 19	3	1.20	3	1.20	4.5	1.80	3	1.20	3	1.2	5	2.00	0	0.00
Alumno 20	4.5	1.80	4.5	1.80	5.5	2.20	4.5	1.80	3	1.2	5	2.00	2	0.80
Alumno 21	3	1.20	3	1.20	3	1.20	3	1.20	3	1.2	4	1.60	2	0.80
Alumno 22	4.5	1.80	3	1.20	4	1.60	3	1.20	4.5	1.8	0	0.00	6	2.40
Alumno 23	4	1.60	4	1.60	7.5	3.00	3	1.20	3	1.2	7	2.80	4	1.60
Alumno 24	3	1.20	5	2.00	5.5	2.20	3	1.20	3	1.2	5	2.00	2	0.80
Alumno 25	7	2.80	3	1.20	5	2.00	3	1.20	3	1.2	5	2.00	4	1.60

Tabla A.3. Relación de asistencia de los alumnos de 5A

PUNTOS POR ASISTENCIA			
	CERO FALTAS	1	PUNTO (10%)
	CINCO FALTAS	0.5	PUNTOS (5%)
	MÁS DE CINCO FALTAS	0	PUNTOS (0%)
PUNTOS POR ASISTENCIA			
ALUMNO	PUNTOS		
Alumno 1	1		
Alumno 2	1		
Alumno 3	1		
Alumno 4	1		
Alumno 5	0.5		
Alumno 6	0.5		
Alumno 7	1		
Alumno 8	1		
Alumno 9	1		
Alumno 10	1		
Alumno 11	0.5		
Alumno 12	1		
Alumno 13	1		
Alumno 14	1		
Alumno 15	0.5		
Alumno 16	1		
Alumno 17	1		
Alumno 18	1		
Alumno 19	1		
Alumno 20	1		
Alumno 21	1		
Alumno 22	1		
Alumno 23	1		
Alumno 24	1		
Alumno 25	1		

Tabla A.4. Relación de libreta de los alumnos de 5A

PUNTOS DE LIBRETA POR MATERIA							
	LIB. COMPLETA	1.5	PUNTOS (15%)				
	LIB. CON OBSERVACIONES	1	PUNTOS (10%)				
CALIFICACIÓN DE LIBRETA POR MATERIA Y POR ALUMNO							
ALUMNO	ESPAÑOL	MATEMATICAS	C. NATURALES	GEOGRAFÍA	HISTORIA	FCYE	EA
Alumno 1	1.5	1.5	1.5	1.5	1.5	1.5	1.5
Alumno 2	1.5	1.5	1.5	1.5	1.5	1.5	1.5
Alumno 3	1.5	1.5	1.5	1.5	1.5	1.5	1.5
Alumno 4	1.5	1.5	1.5	1.5	1.5	1.5	1.5
Alumno 5	1.5	1.5	1.5	1.5	1.5	1.5	1.5
Alumno 6	1.5	1.5	1.5	1.5	1.5	1.5	1.5
Alumno 7	1.5	1.5	1.5	1.5	1.5	1.5	1.5
Alumno 8	1.5	1.5	1.5	1.5	1.5	1.5	1.5
Alumno 9	1.5	1.5	1.5	1.5	1.5	1.5	1.5
Alumno 10	1.5	1.5	1.5	1.5	1.5	1.5	1.5
Alumno 11	1.5	1.5	1.5	1.5	1.5	1.5	1.5
Alumno 12	1.5	1.5	1.5	1.5	1.5	1.5	1.5
Alumno 13	1.5	1.5	1.5	1.5	1.5	1.5	1.5
Alumno 14	1.5	1.5	1.5	1.5	1.5	1.5	1.5
Alumno 15	1.5	1.5	1.5	1.5	1.5	1.5	1.5
Alumno 16	1.5	1.5	1.5	1.5	1.5	1.5	1.5
Alumno 17	1.5	1.5	1.5	1.5	1.5	1.5	1.5
Alumno 18	1.5	1.5	1.5	1.5	1.5	1.5	1.5
Alumno 19	1.5	1.5	1.5	1.5	1.5	1.5	1.5
Alumno 20	1.5	1.5	1.5	1.5	1.5	1.5	1.5
Alumno 21	1.5	1.5	1.5	1.5	1.5	1.5	1.5
Alumno 22	1.5	1.5	1.5	1.5	1.5	1.5	1.5
Alumno 23	1.5	1.5	1.5	1.5	1.5	1.5	1.5
Alumno 24	1.5	1.5	1.5	1.5	1.5	1.5	1.5
Alumno25	1.5	1.5	1.5	1.5	1.5	1.5	1.5