



**UNIVERSIDAD TÉCNICA DE BABAHOYO**

**FACULTAD DE ADMINISTRACIÓN, FINANZAS E  
INFORMÁTICA**

**PROCESO DE TITULACIÓN**

**OCTUBRE 2017 – MARZO 2018**

**PROPUESTA TECNOLÓGICA DE GRADO O DE FIN DE CARRERA**

**PRUEBA PRÁCTICA**

**INGENIERÍA EN SISTEMAS**

**PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN SISTEMAS**

**TEMA:**

**Sistema de Notificaciones de Incidencias Escolares a través de Mensajería Móvil  
para los Estudiantes de la Escuela de Educación Básica Juan Antonio Montalván  
Cornejo de la Ciudad de Vinces**

**EGRESADO:**

**José Manuel Rosado Anzules**

**TUTOR:**

**Ing. Miguel Ángel Zúñiga Sánchez**

**AÑO 2018**

## ÍNDICE

ÍNDICE.....	II
ÍNDICE DE GRÁFICOS .....	III
ÍNDICE DE TABLAS.....	III
ÍNDICE DE ilustraciones .....	IV
INTRODUCCIÓN.....	1
CAPÍTULO I.....	2
1. Diagnóstico de necesidades y requerimientos .....	2
1.1.  Ámbito de Aplicación: descripción del contexto y hechos de interés .....	2
1.2.  Establecimiento de requerimientos .....	5
1.3.  Justificación de los requerimientos a complacer.....	12
CAPITULO II.....	14
2. Desarrollo del prototipo tecnológico .....	14
2.1.  Definición del prototipo. ....	14
2.2.  Fundamentación de teorías del prototipo. ....	15
2.3.  Objetivos del prototipo tecnológico. ....	33
2.4.  Diseño del prototipo tecnológico .....	34
2.5.  Ejecución del prototipo tecnológico.....	75
Al ingresar al sistema el docente y administrador visualizarán el formulario de inicio de sesión del sistema. ....	75
Formulario de incidencias podrán acceder tanto administradores como docentes .....	76
Módulo administrador en el cual se registran los docentes .....	76
Módulo de administrador, para la creación de aulas del periodo lectivo .....	77
Módulo de docente, para el registro de padres de familia .....	78
Módulo de docente, búsqueda de padres de familia .....	79
Módulo de docente, para visualizar el reporte de asistencia diario. ....	80
Formulario de búsqueda de docentes .....	80
CAPÍTULO III .....	81
3. Evaluación del prototipo.....	81
3.1.  Plan de evaluación.....	81
3.2.  Resultados de la evaluación. ....	82
CONCLUSIONES.....	83
RECOMENDACIONES .....	84
BIBLIOGRAFÍA .....	85

## ÍNDICE DE GRÁFICOS

Grafico 1.- Funcionalidad de la Aplicación: Administrador.....	35
Grafico 2.- Funcionalidad de la Aplicación: Docente.....	35
Grafico 3.- Diagrama de Clases de la Aplicación.....	36
Grafico 4.- Diagrama de Casos de uso de la aplicación.....	37
Grafico 5.- Diagrama Casos de Uso Inicio de Sesión al Sistema.....	38
Grafico 6.- Diagrama de Casos de Uso: Administración de Usuarios.....	38
Grafico 7.- Diagrama de Casos de Uso: Administración de Periodo Lectivo.....	38
Grafico 8.- Diagrama de Casos de Uso: Administración de Aulas.....	39
Grafico 9.- Diagrama de Casos de Uso: Administración de Notificaciones.....	39
Grafico 10.- Diagrama de Casos de Uso: Administración de Horarios.....	39
Grafico 11.- Diagrama de Casos de Uso: Administración de Estudiantes.....	40
Grafico 12.- Diagrama de Casos de Uso: Administración de Docentes.....	40
Grafico 13.- Diagrama de Casos de Uso: Administración de Asistencias.....	40
Grafico 14.- Diagrama de Casos de Uso: Administración de Tareas.....	41
Grafico 15.- Diagrama de Secuencias: Inicio de Sesión al Sistema.....	41
Grafico 16.- Diagrama de Secuencias: Buscar Usuarios.....	42
Grafico 17.- Diagrama de Secuencias:.....	42
Grafico 18.- Diagrama de Secuencias: Registro de Usuarios.....	43
Grafico 19.- Diagrama de Secuencias: Administración de Periodos.....	44
Grafico 20.- Diagrama de Secuencias: Envío de Notificaciones.....	44
Grafico 21.- Diagrama de Secuencias: Administración de Aulas.....	45
Grafico 22.- Diagrama de Secuencias: Administración de Asistencias.....	45
Grafico 23.- Diagrama de Secuencias: Administración de Representantes.....	46
Grafico 24.- Diagrama de Secuencias: Administración de Incidencias.....	47
Grafico 25.- Diagrama de Actividades: Inicio de Sesión al Sistema.....	47
Grafico 26: Diagrama de Actividades: Administración de Usuarios.....	48
Grafico 27.- Diagrama de Actividades: Administración de Incidencias.....	48
Grafico 28.- Diagrama de Actividades: Toma de Asistencias.....	49
Grafico 29.- Diagrama Entidad Relación.....	50
Grafico 30.- Diagrama Relacional.....	51

## ÍNDICE DE TABLAS

Tabla 1.- Perfiles de Usuarios.....	52
Tabla 2.- Estados.....	52
Tabla 3.- Usuarios.....	52
Tabla 4.- Periodo Lectivo.....	53
Tabla 5.- Perfiles de Usuario Activos.....	53
Tabla 6.- Universidad.....	53
Tabla 7.- Cargos del Plantel.....	54
Tabla 8.- Género.....	54

Tabla 9.- Áreas Pedagógicas .....	54
Tabla 10.- Grados .....	54
Tabla 11.- Docentes.....	55
Tabla 12.- Estudiantes .....	56
Tabla 13.- Representantes.....	56
Tabla 14.- Parentescos.....	57
Tabla 15.- Representantes y Estudiantes .....	57
Tabla 16.- Asignaturas.....	57
Tabla 17.- Jornada .....	58
Tabla 18.- Aulas .....	58
Tabla 19.- Horarios.....	58
Tabla 20.- Días .....	59
Tabla 21.- Detalle de Horario .....	59
Tabla 22.- Matriculas.....	59
Tabla 23.- Indicadores de Conducta .....	60
Tabla 24.- Parcial.....	60
Tabla 25.- Quimestre .....	60
Tabla 26.- Conducta .....	61
Tabla 27.- Indicadores de Calificación.....	61
Tabla 28.- Notas .....	61
Tabla 29.- Tipo de Incidencias .....	62
Tabla 30.- Notificaciones .....	62
Tabla 31.- Asistencia .....	62
Tabla 32.- Detalle de Asistencia.....	63

## **ÍNDICE DE ILUSTRACIONES**

Ilustración 1 Formulario de Inicio de Sesión .....	75
Ilustración 2 Pantalla de Bienvenida .....	75
Ilustración 3 Registro de Incidencias.....	76
Ilustración 4 Formulario de Registro de Docentes .....	76
Ilustración 5 Formulario de Registro de Aulas.....	77
Ilustración 6 Formulario de Registro de Estudiantes.....	77
Ilustración 7 Formulario de Registro de Representantes.....	78
Ilustración 8 Formulario de Registro de Matriculas.....	78
Ilustración 9 Formulario de Búsqueda de Representantes.....	79
Ilustración 10 Formulario de Registro de Asistencia .....	79
Ilustración 11 Reporte de Asistencia.....	80
Ilustración 12 Formulario de búsqueda de docentes. ....	80

## INTRODUCCIÓN

La educación es el eje fundamental para el avance de un país, principalmente la educación primaria, que además de enseñar a leer a escribir es la encargada de fomentar valores éticos, morales y cívicos; un asunto muy importante para el desarrollo de los niños/as. Sin embargo, conviene recordar que no sólo corresponde a la escuela el papel de formar a los niños/as, sino que es una responsabilidad compartida entre profesores y padres de familia.

No obstante, en la Escuela de Educación Básica Juan Antonio Montalván Cornejo de la Ciudad de Vinces, existen inconvenientes en la comunicación entre profesores y padres de familia, que provoca desconocimiento del estado académico de sus hijos, mostrando un cierto grado irresponsabilidad por parte de los padres, lo cual se ve reflejado en tareas incumplidas por parte de los niños/as.

La tecnología, actualmente crece a pasos agigantados, esta es muy buena razón para usarla con el objetivo de crear soluciones informáticas que automaticen procesos, que resuelvan los problemas que están presente en esta institución educativa.

El presente trabajo describe el desarrollo de un sistema de notificaciones a través de mensajería móvil en la Escuela de Educación Básica, para mantener informados a los padres de familia sobre los acontecimientos que suceden en dicha institución, además de notificar el informe académico de los estudiantes, así como de sus tareas.

# CAPÍTULO I

## 1. Diagnóstico de necesidades y requerimientos

### 1.1. **Ámbito de Aplicación: descripción del contexto y hechos de interés**

La educación primaria es el principal eje para el desarrollo del niño/a en el ámbito intelectual, ético y académico; y además de que este/a se pueda desenvolver tanto en la secundaria como en la universidad.

El problema principal de que se presenta en la Escuela de Educación Básica Juan Antonio Montalván Cornejo, es el incumplimiento de las tareas escolares por parte de los estudiantes, debido a que no hay un correcto control de los padres de familia porque no tienen conocimiento de las mismas, ya sea porque el niño/a no la copia en su cuaderno o simplemente se olvida de comunicarle y de realizarlas.

Así mismo se presenta el problema de que ciertos padres de familia no se presentan a las reuniones que se realizan en la escuela, para dar a conocer las actividades y el estado académico de los estudiantes, de igual manera este problema es ocasionado por la falta de comunicación entre padres e hijos/as, provocando una dificultad de comunicación entre la escuela y los padres de familia.

Actualmente existen herramientas informáticas, las cuales ayudan a resolver este tipo de inconvenientes. Estas herramientas cada día son más potentes cuando se trata automatizar procesos y tareas que hoy en día se siguen realizando de manera manual.

La presente propuesta plantea desarrollar un Sistema de Notificaciones de Incidencias Escolares a través de Mensajería Móvil para los estudiantes de la Escuela de Educación Básica Juan Antonio Montalván Cornejo de la ciudad de Vinces, el cual

permitirá a los padres de familia estar diariamente informados sobre los acontecimientos que suceden o sucederán en la escuela, de la misma manera podrán conocer el avance académico de sus hijos/as.

### **1.1.1. Planteamiento del problema**

La baja comunicación entre los padres de la familia y los estudiantes sobre las tareas que envía la escuela, induce a enterarse de las mismas a última hora dando como resultado que el estudiante realice sus tareas mal o sencillamente no las realice.

Ya que existe un grado de dificultad de notificar las incidencias escolares por parte de la escuela a los padres de familia, mucho de ellos faltan a las reuniones, lo que incita que no sepan del avance académico y se enteren de este cuando ya es demasiado tarde.

El incumplimiento de las tareas escolares de los estudiantes de la Escuela de Educación Básica Juan Antonio Montalván de la Ciudad de Vinces, debido al descuido, provoca el bajo rendimiento académico y además la pérdida de año.

Por lo tanto, la Escuela de Educación Básica Juan Antonio Montalván de la Ciudad de Vinces, busca nuevos métodos que estén acorde con las necesidades que debe cumplir a esta sociedad educativa, que permita a los docentes notificar cada acontecimiento escolar ya sean actividades curriculares, faltas, indisciplinas, estado académico de los estudiantes a sus respectivos padres, por medio de la mensajería móvil.

A medida que la tecnología avanza, las instrucciones educativas requieren automatizar sus procesos para mejorar sus servicios, es por eso que el desarrollo

de esta propuesta es muy importante para que los padres de familia y/o representantes estén informados diariamente sobre el avance académico de los estudiantes de la de Educación Básica Juan Antonio Montalván de la Ciudad de Vinces.

El Ministerio de Educación a través del portal Educar Ecuador, brinda un servicio automatizado en el cual el representante puede conocer el avance académico de los niño/as y jóvenes que se encuentran matriculados en alguna institución educativa; como las notas y faltas, desde la etapa inicial hasta que terminen la secundaria. (MINISTERIO DE EDUCACIÓN, 2017)

El padre de familia y/o representante, puede consultar el estado académico del estudiante, pero para acceder a este servicio se debe contar con acceso a internet, y además que sólo puede consultar las notas al final de cada parcial. Aunque en dicho sistema se pueden notificar tareas o actividades que se realizan en la escuela, muchos padres de familia y/o representantes no lo revisan diariamente, y sólo esperan a imprimir la matriz de las notas del estudiante al final.

A diferencia del servicio que ofrece el Ministerio de Educación, la presente propuesta tendrá como objetivo notificar de manera diaria cualquier incidencia escolar que se presente en la de Educación Básica Juan Antonio Montalván de la Ciudad de Vinces, directamente al teléfono celular del padre de familia y/o representante, por medio de mensajes de texto, sin importar si el teléfono es alta gama o este carece de acceso a internet, siendo esto una gran ventaja porque muchos padres de familia y/o representantes no cuentan con un Smartphone o con plan internet en sus hogares.



Esta propuesta contribuye a mejorar la comunicación entre la Escuela de Educación Básica Juan Antonio Montalván de la Ciudad de Vinces y los padres de familia y/o representantes sobre las incidencias escolares; y a la vez promover el mejoramiento académico de los estudiantes de dicha institución, para que en el futuro sean ciudadanos responsables capaces de desenvolverse solos en sus estudios.

### **1.1.2. Localización del problema objeto de estudio**

La investigación se localiza geográficamente en la provincia de Los Ríos; Cantón Vinces; en las calles Primero de Mayo entre Nicaragua y Teodoro Wold, cerca del Subcentro San Lorenzo; en la Escuela de Educación Básica Juan Antonio Montalván.

### **1.1.3. Problema general**

¿Cómo ayuda el desarrollo de un Sistema de Notificaciones de Incidencias Escolares a través de Mensajería Móvil para los Estudiantes de la Escuela de Educación Básica Juan Antonio Montalván Cornejo de la Ciudad de Vinces, para disminuir el incumplimiento de las tareas escolares por parte de los alumnos?

## **1.2. Establecimiento de requerimientos**

Una vez analizado los inconvenientes que existen en la Escuela de Educación Básica Juan Antonio Montalván Cornejo de la Ciudad de Vinces, se desarrollará un Sistema de Notificaciones de Incidencias Escolares a través de Mensajería Móvil, el cual contará con las siguientes funciones según los roles asignados en el prototipo:

### **Administrador del sistema (Directora del Plantel)**

- ✓ Administración y gestión de usuarios.

- Crear nuevos usuarios.
  - Modificar información de los usuarios.
  - Cambiar el estado de los usuarios (activo, inactivo).
  - Cambiar perfil de los usuarios.
  - Buscar usuarios.
- ✓ Gestión periodos lectivos.
- Crear el nuevo periodo lectivo.
- ✓ Gestión aulas
- Crear una nueva aula.
  - Asignar al aula un respectivo docente.
  - Actualizar aulas.
  - Eliminar aula.
  - Buscar aulas
- ✓ Administración y gestión de notificaciones enviadas.
- Búsqueda de notificaciones por fecha.
  - Búsqueda de notificaciones por aula.
  - Generar reportes de notificaciones.
  - Enviar notificaciones a docentes y padres de familia y/o representantes.

## **Docentes**

Una vez creado el periodo lectivo y las aulas los docentes tendrán las siguientes funciones:

- ✓ Administración de horarios

- Asignar al horario a las asignaturas según día y hora.
  - Actualizar horarios.
- ✓ Administración y gestión de estudiantes
    - Ingresar datos de estudiantes.
    - Actualizar datos de estudiantes.
    - Cambiar estado de estudiantes.
    - Asignar estudiantes al aula que el docente administra.
- ✓ Administración y gestión de padres de familia y/o representantes
    - Ingresar datos de padres de familia y/o representantes.
    - Actualizar datos de padres de familia y/o representantes.
    - Asignar al estudiante su respectivo representante.
- ✓ Administración y gestión de asistencias
    - Tomar asistencia a los estudiantes diariamente.
    - Asignar retrasos.
    - Notificar incidencias de asistencia.
    - Generar reporte de asistencia.
- ✓ Administración y gestión de tareas, evaluaciones y exámenes.
    - Agregar una nueva tarea
    - Notificar nueva tarea o evaluación a representantes.
    - Notificar cumplimiento de tareas.
- ✓ Administración y gestión de notas
    - Asignar notas de tareas, evaluaciones o exámenes de los estudiantes.

- Notificar notas a padres de familia y/o representantes.
  - Generar reporte de notas.
- ✓ Administración y gestión de otras incidencias.
- Notificar actos de indisciplina de los estudiantes.
  - Notificar convocatorias a reunión de padres de familia y/o representantes.
  - Notificar hora de salida.
  - Enviar notificaciones personalizadas a padres de familia y/o representantes.

El sistema tendrá interfaz web, que estará alojado en un servidor dentro de la dirección de la escuela, y funcionará por medio de una intranet.

### **1.2.1. Metodología**

Para el desarrollo del prototipo se ha escogido la metodología orientada a objetos, ya que es una metodología de la ingeniería del software, puesto que establece una representación del problema en relación al mundo real y lo hace corresponder con el ámbito de la solución, usando fases de una manera organizada para producir software de calidad.

Entre las fases de esta metodología tenemos: el Análisis Orientado a Objetos (AOO), el Diseño Orientado a Objetos (DOO) y la Programación Orientada a Objetos (POO) o implementación.

El análisis orientado a objetos hace énfasis en una investigación del problema y los requisitos del sistema antes de ponerlos en la solución. Durante esta fase se centra a descubrir y detallar los objetos en el dominio del problema.

El diseño orientado a objetos hace énfasis proyectar una solución conceptual que satisface los requisitos antes de ponerlos en la implementación. Durante esta fase se centra en la definición de los objetos de software y cómo estos colaboran para satisfacer los requerimientos para la solución del problema. Esta fase induce a pensar en cada una de las entidades que interactúan en términos de objetos, en vez de procedimientos, cuando planifican el código. Un objeto agrupa datos encapsulados y procedimientos para representar una entidad.

Durante las fases de análisis y diseño se crea un conjunto de modelos utilizando una notación acordada como, por ejemplo, el lenguaje unificado de modelado (UML). Una notación es un conjunto de diagramas normalizados que posibilita al analista o desarrollador a describir el comportamiento del sistema y los detalles de la arquitectura de forma no ambigua.

Finalmente en la programación orientada a objetos, los objetos creados en el diseño se traducen en una implementación concreta, utilizando un lenguaje de programación orientado a objetos, teniendo en cuenta los principios de la ingeniería de software, de modo que la implementación del software se flexible y extensible. (Larman, 2004)

### **1.2.2. Arquitectura**

La arquitectura del prototipo será cliente/servidor, la cual hace referencia al modelo de aplicación centralizada en el cual las tareas principales las realiza el servidor el cual comparte servicios o recursos a los clientes.

Los clientes de manera independiente realizan peticiones a una o varias aplicaciones hacia el servidor por medio de una conexión de red ya sean intranet

o internet. El servidor debe encontrarse siempre en ejecución para atender dichas peticiones.

### **1.2.3. Patrón de desarrollo**

La patrón desarrollo será modelo, vista, controlador (MVC), el cual es un patrón de diseño de software basado en la interconexión de tres tipos de componentes principales, en un lenguaje de programación como JAVA, a menudo con un fuerte enfoque en los paradigmas de software de programación orientada a objetos (POO). Los tres tipos de componentes se conocen como modelos, vistas y controladores.

El modelo es donde se guarda toda la lógica comercial de una aplicación. La lógica de negocios puede ser específica de cómo una aplicación almacena datos o utiliza servicios de terceros para cumplir con los requisitos de su negocio. Si la aplicación debe acceder a la información en una base de datos, el código para hacer eso se mantendrá en el modelo. Si fuera necesario, por ejemplo, buscar datos de stock o twittear sobre un nuevo producto, ese código también se mantendría en el modelo.

La vista es donde se guardan todos los elementos de la interfaz de usuario de nuestra aplicación. Esto puede incluir nuestro marcado HTML, hojas de estilo CSS y archivos JavaScript. Cualquier cosa que un usuario ve o interactúa se puede mantener en una vista.

El controlador es el componente que conecta modelos y vistas juntos. Los controladores aíslan la lógica comercial de un modelo de los elementos de la interfaz de usuario de una vista y manejan cómo la aplicación responderá a la interacción del usuario en la vista. Los controladores son el primer punto de

entrada a este trío de componentes, ya que la solicitud se pasa primero a un controlador, que luego creará una instancia de los modelos y vistas necesarios para cumplir con una solicitud a la aplicación. (Pitt, 2012)

#### **1.2.4. Requerimientos mínimos de funcionalidad**

##### **1.2.4.1. Requisitos de hardware**

✓ Para el servidor

- Memoria RAM: 4 GB.
- Espacio en Disco Duro: 500 GB.
- Procesador X86 o X64 de 3Ghz (mínimo)
- Arduino UNO
- Módulo GSM/GPRS SIM900 compatible con Arduino.
- Interfaz de red

✓ Para el cliente

- Memoria RAM 2GB (mínimo)
- Disco duro 5GB libres (mínimo)
- Procesador Intel/AMD de 2.16Ghz (mínimo)
- Interfaz de red.

##### **1.2.4.2. Requisitos de software**

✓ Para el servidor

- Sistema operativo CentOS 7.16.11
- Java JRE versión 8
- Tomcat Apache versión 8

- PostgreSQL versión 9.5
- ✓ Para el cliente
  - Sistema operativo de escritorio Windows (XP, 7, 8.1, 10) o Linux (Ubuntu, ZorinOS, Fedora).
  - Navegador Web actualizado (Mozilla Firefox, Google Chrome, Safari).

### **1.3. Justificación de los requerimientos a complacer**

En la Escuela de Educación Básica Juan Antonio Montalván de la Ciudad de Vinces, la comunicación entre padres y docentes se realiza de manera tradicional; el estudiante es el encargado de notificar a sus representantes, sobre las incidencias escolares que suceden en la institución. Con frecuencia aquella notificación no llega en el momento oportuno, lo que provoca que el representante no esté al tanto de estado de los estudiantes en la escuela.

Mediante una pequeña entrevista con la directora de la escuela, se notó que el problema principal que causa la falencia de comunicación entre la escuela y los padres de familia y/o representantes es el incumplimiento de las tareas escolares por parte de los estudiantes, teniendo como consecuencia el bajo rendimiento académico y hasta la pérdida del año lectivo.

Con el desarrollo del sistema se pretende mejorar las falencias de comunicación que existen entre los padres de familia y/o representantes con la escuela, con el objetivo resolver el problema de incumplimiento de tareas de los estudiantes y mejorar su rendimiento académico, permitiendo que sean niños/as y jóvenes más estudiosos y responsables con sus actividades curriculares.



La presente propuesta tiene como objetivo principal mantener a los padres de familia y/o representantes informados a diario sobre el avance de los estudiantes en la escuela, para que no estén apurados en últimas instancias al enterarse que sus hijos/as y/o representados que tienen tarea atrasada o incumplidas, que se involucren más con la escuela y que desempeñen su papel que corresponde en formar valores éticos, y morales a los niños para que ellos estén más comprometidos con sus estudios, además de cumplir con lo que establece el artículo 12 literal b, c, d, k de la (Ley Orgánica de Educación Intercultural, 2015)

**Art. 12.- Derechos.-** Las madres, los padres de y/o los representantes legales de las y los estudiantes tienen derecho a que se garantice a éstos, el pleno goce y ejercicio de sus derechos constitucionales en materia educativa; y, tienen derecho además a:

**b.-** Recibir informes periódicos sobre el progreso académico de sus representados así como de todas las situaciones que se presenten en la institución educativa y que requieran de su conocimiento;

**c.-** Participar, de conformidad con la reglamentación respectiva, en la evaluación de las y los docentes y de la gestión de las autoridades educativas;

**d.-** Elegir y ser elegidos como parte de los comités de padres y madres de familia y los demás órganos de participación de la comunidad educativa;

**k.-** Solicitar y acceder a la información que consideren pertinentes y que este en posesión de la institución educativa.

## CAPITULO II

### 2. Desarrollo del prototipo tecnológico

#### 2.1. Definición del prototipo.

Sistema de Notificaciones de Incidencias Escolares a través de Mensajería Móvil para los Estudiantes de la Escuela de Educación Básica Juan Antonio Montalván Cornejo de la Ciudad de Vinces, será una Aplicación Web, es un prototipo que cumple las necesidades de la institución, las cuales fueron encontradas mediante entrevista directa con la directora de la escuela.

El presente prototipo será una aplicación Web, la cual funcionará en una Intranet de la Escuela. El servidor donde se alojará la aplicación será CentOS, este se reconstruye a partir de otra plataforma Linux empresarial, *Red Hat Enterprise Linux*. CentOS contiene un sinnúmero de características útiles para la administración y gestión de servidores. (Membrey, Verhoeven, & Angenendt , 2009)

El lenguaje de programación en el que se desarrollará la aplicación será Java. La tecnología Java se compone de una gama de productos basados en el poder de la red y en la idea de que el software debe ser capaz de ejecutarse en diferentes máquinas, sistemas y dispositivos. (Arias, 2014)

Para el almacenamiento de los datos se ha usado PostgreSQL, un poderoso gestor de bases de datos relacional orientado a objetos. No solo tiene funciones avanzadas, sino que las realiza rápidamente. Puede superar a muchas otras bases de datos, incluidas las propietarias para muchos tipos de cargas de trabajo de bases de datos. (Obe & Hsu, 2012)

Cabe recalcar que todas estas herramientas de software que se utilizaron son de libre distribución. El sistema además contará con un circuito electrónico, que utiliza la tecnología de hardware libre Arduino, que se encargará de enviar las notificaciones de texto (SMS), al teléfono celular de cada representante legal.

## **2.2. Fundamentación de teorías del prototipo.**

El presente prototipo ofrecerá soluciones informáticas a los problemas y necesidades que están presentes en la Escuela de Educación Básica Juan Antonio Montalván Cornejo de la Ciudad de Vinces, y mejorar la participación del representante legal en las actividades que realiza la escuela.

La aplicación tendrá interfaz amigable, en la cual el usuario podrá interactuar de manera fácil con las tareas orientadas las necesidades específicas de cada uno de los mismos, de acuerdo a los perfiles designados del sistema.

Cualquier usuario que esté conectado a la intranet, podrá visualizar la aplicación, y podrá acceder a la información de la escuela siempre y cuando tenga sus credenciales acceso, para asegurar la integridad y confiabilidad de la información la institución.

La aplicación web facilitará que el acceso a la información se rápido, además de visualizarla de una forma que el usuario de sienta cómodo con el sistema. Además la implementación de una aplicación Web, disminuye el tiempo de instalación en cada dispositivo, además de poder acceder a ella desde cualquier sistema operativo que se pueda conectar a la red.

### **2.2.1. Metodología**

La metodología orientada a objetos presenta características que lo hacen idóneo para el análisis, diseño y programación de sistemas; sin embargo, el análisis de requisitos, que es la relación entre la asignación de software al nivel del sistema y el diseño del software, se quedó atrás por lo que empezaron a surgir diferentes métodos de análisis y diseño orientado a objetos, entre los que destacan los métodos Booch, OOSE (Object Oriented Software Engineering) y OMT (Object Modeling Technique). Para poner fin a la "guerra de métodos" que se presentó en ese momento, se creó el Lenguaje Unificado de Modelado (UML). (Román Zamitiz, Páginas Personales de Profesores , 2007)

#### **2.2.1.1. Ventajas de la metodología orientada a objetos**

- *Reusabilidad.*- El objetivo principal de desarrollar software reutilizable es reducir el tiempo y el esfuerzo del desarrollo y mantenimiento de otros sistemas similares. La metodología orientada a objetos nos ayuda a crear software reutilizable debido a sus mecanismos para encapsular datos y funciones relacionadas, para ocultar información privada.
- *Estabilidad.* Los objetos se modelan para que puedan ser utilizados una y otra vez, y así corregir errores y hacerlos estables, así como sucede con las unidades de procesamiento y otros integrados que se vuelven estables.
- *El analista oculta comportamiento de objetos y no piensa en detalles de bajo nivel.* La encapsulación de código oculta la lógica de los grandes objetos para que sean de fácil uso.

- *Herencia de clases.* Se crean objetos a partir de otros objetos, las cuales a su vez están compuestos por otros objetos. Esto permite armar módulos de software más sofisticados, que a su vez se convierten en bloques de instrucciones de software más grandes. Es de decir, ayuda a crear módulos de software que se pueden conectar entre sí, lo que permite la creación de nuevos programas
- *Alta Calidad.* Otro de los objetivos de la metodología orientada a objetos es hacer que los elementos del sistema sean más modulares, mejorando así la calidad del sistema y la eficiencia del análisis y diseño de sistemas.
- *Un rápido diseño.* El análisis y diseño orientado a objetos es un enfoque que está destinado a facilitar el desarrollo de sistemas que deben cambiar rápidamente en respuesta a entornos empresariales dinámicos, utilizando o mejorando módulos ya existentes.
- *Integridad de procesos.* En cada clase se modelan procesos explícitos los cuales se instancian en un objeto específico para que pueda hacer uso de ese proceso, siendo esto de vital importancia en caso de que una persona no autorizada intente entrar al sistema.
- *Mantenimiento sencillo.* Provee la garantía de que el sistema disfrutará de una vida más larga a la vez que tendrá costos de mantenimiento mucho más pequeños. Debido a que la mayoría de los procesos dentro del sistema están encapsulados, los comportamientos pueden reutilizarse e incorporarse en nuevos comportamientos.

- *Modelado del mundo real:* el sistema orientado a objetos tiende a modelar el mundo real de una manera más completa que los métodos tradicionales. Los objetos se organizan en clases de objetos y los objetos se asocian con comportamientos. El modelo se basa en objetos, más que en datos y procesamiento.
- *Independencia del diseño.* El sistema orientado a objetos promete ser mucho más confiable que los sistemas tradicionales, principalmente porque se pueden "construir" nuevos comportamientos a partir de objetos existentes. Debido a que los objetos se pueden llamar y acceder dinámicamente, se pueden crear nuevos objetos en cualquier momento.
- *Interacción.* Los nuevos objetos pueden heredar atributos de datos de uno u otros objetos. Los comportamientos pueden heredarse de las superclases y pueden agregarse nuevos comportamientos sin afectar las funciones existentes del sistema.
- *Aplicaciones Cliente-Servidor.* Por su propia naturaleza, las aplicaciones cliente / servidor envuelven la transmisión de mensajes hacia adelante y hacia atrás a través de una red, y la metodología orientada a objetos se combina bien con la arquitectura física y conceptual de las aplicaciones cliente / servidor.
- *Alta compatibilidad con bases de datos.* Nos permite crear clases, organizar objetos, estructurar una jerarquía de herencia y llamar a métodos de otras clases. Además de estos, también proporciona las instalaciones asociadas con los sistemas de bases de datos estándar.

- *Migración.* Los beneficios de la orientación a objetos justifican el diseño y el desarrollo de métodos de reingeniería apuntan a migrar sistemas procesales heredados a plataformas modernas orientadas a objetos.
- *Uso de herramientas CASE.* Se puede lograr un nivel adicional de productividad llevando el resultado de la herramienta CASE al diseño de un programa orientado a objetos, que tiene como objetivo lograr que se desarrollen mejores sistemas de alta calidad utilizando un enfoque orientado a objetos y una herramienta de modelado para obtener una comprensión sólida del diseño del sistema que se desarrollará. (Román Zamitiz, Páginas Personales de Profesores, 2017).

#### **2.2.1.2. UML**

Las notaciones nos permiten formular ideas complejas en forma resumida y precisa. En los proyectos que involucran a muchos participantes, a menudo con diferentes conocimientos técnicos y culturales, la precisión y claridad son críticas conforme se incrementa rápidamente el costo de la falta de comunicación.

Para que una notación permita la comunicación precisa debe tener una semántica bien definida, debe ser muy adecuada para la representación de un aspecto dado de un sistema y debe ser bien comprendida por los participantes del proyecto.

UML es una notación que se produjo como resultado de la unificación de la técnica de modelado de objetos. Por lo tanto, proporciona construcciones para un amplio rango de sistemas y actividades (por ejemplo, sistemas de tiempo real,

sistemas distribuidos, análisis, diseño del sistema, entregas). El desarrollo de sistemas se enfoca en tres modelos diferentes del sistema:

- El modelo funcional, representado en UML con diagramas de caso de uso, describe la funcionalidad del sistema desde el punto de vista del usuario.
- El modelo de objetos, representado en UML con diagramas de clase, describe la estructura de un sistema desde el punto de vista de objetos, atributos, asociaciones y operaciones.
- El modelo dinámico, representado en UML con diagramas de secuencia, diagramas de gráfica de estado y diagramas de actividad, describe el comportamiento interno del sistema. Los diagramas de secuencia describen el comportamiento como una secuencia de mensajes intercambiados entre un conjunto de objetos, mientras que los diagramas de gráfica de estado describen el comportamiento desde el punto de vista de estados de un objeto individual y las transiciones posibles entre estados.  
(Bruegge & Dutoit, 2002)

### **2.2.2. Arquitectura**

En la arquitectura cliente servidor hay una poderosa computadora (conocida como servidor) que actúa como proveedor de información a otras estaciones de trabajo subordinadas. La arquitectura cliente-servidor define una relación entre el usuario de una estación de trabajo (el cliente frontal) y un servidor posterior de archivos, impresión, comunicaciones, u otro tipo de sistema proveedor de servicios. (López Gutiérrez, 2017).



### 2.2.2.1. Ventajas

Las ventajas más relevantes que ofrece son:

- *Servidor centralizado.* Tener un servidor único que aloje la base de datos de nombres de usuario y contraseñas y, al mismo tiempo, administrar qué niveles de acceso pueden tener los usuarios individuales y las computadoras para recursos de red específicos es una ventaja distintiva de la arquitectura del servidor cliente. Se necesita menos personal de apoyo para gestionar las cuentas de seguridad centralizadas de lo que sería necesario si se tuviera que configurar la seguridad y el acceso a los recursos en cada computadora individual de la red. Esta ventaja es aún mayor en las redes donde los usuarios suelen acceder a los recursos de varias computadoras, ya que el acceso a los recursos y los permisos de inicio de sesión tendrían que configurarse para cada usuario en cada computadora que usan.
- Centralización de hardware y servicios. Desde varias plataformas en la red, se puede acceder al servidor de forma remota, y utilizar el hardware, software o servicios; que este provee.
- Seguridad. Las reglas que definen los derechos de seguridad y acceso se pueden definir en el momento de la configuración del servidor..
- Ventaja de escalabilidad. Puede agregar recursos en forma de segmentos de red, computadoras y servidores a una red de servidor cliente sin grandes interrupciones en la red. El acceso a cualquier recurso nuevo se puede administrar desde la base de datos de seguridad centralizada, almacenada

en un único servidor de red. Sin servidores centralizados que alberguen nuevos recursos, las nuevas computadoras agregadas a la red necesitarían configurarse individualmente con los nombres de usuario y las contraseñas de todos los usuarios que pudieran necesitar utilizarlas, y el acceso a los recursos almacenados en múltiples computadoras necesitaría ser configurado en cada recurso de alojamiento de la computadora, para cada usuario que necesite acceder a ellos.. (Ruiz, 2013)

### **2.2.3. Tecnología**

#### **2.2.3.1. Página Web**

Una página web es un documento electrónico capaz de contener texto, sonido, vídeo, archivos, enlaces, imágenes y que puede ser accedida mediante un navegador web. Esta información se encuentra generalmente en formato HTML o XHTML, y puede proporcionar acceso a otras páginas web mediante enlaces de hipervínculo.

La información de la World Wide Web está contenida en las páginas Web. La primera página es la que se suele llamar home page, o página de presentación, y es desde la que se da referencia a todas las demás que forman parte de un sitio. Una página Web personal puede contener lo que su autor quiera. (Hobbs, 1999)

Una vez publicadas en el World Wide Web, cualquier usuario puede acceder a ella, quien puede visualizar la información que contiene la página web de una forma entendible.

### **2.2.3.2. Aplicación Web**

Una aplicación web es aquella herramienta a la que accedemos a través de un navegador, también conocido como cliente HTTP, utilizando internet o una intranet, puesto que se encuentra alojada dentro de un servidor, independientemente al equipo que accede a ella. Es decir es una aplicación que se desarrolla en un lenguaje interpretable por los navegadores web, sin depender de un sistema operativo específico. (Ramos Martín & Ramos Martín, 2014 )

Las aplicaciones Webs pueden ser accedidas por una amplia gama de dispositivos, que tengan una conexión a la red, sea internet o intranet, y además cuenten con una navegador actualizado, que cumpla con los requerimientos que pide la aplicación para su correcto funcionamiento.

### **2.2.3.3. CentOS 7**

CentOS es el sistema operativo Linux de nivel empresarial, que es 100% binario compatible con Red Hat Enterprise Linux (RHEL). Actúa como una alternativa gratuita a la oferta comercial de Linux de RedHat, con solo un cambio en la marca. (Cánepa, 2016)

CentOS 7 es uno de los sistemas operativos Linux más confiables que se utiliza para múltiples funcionalidades en una infraestructura informática. Es como la Caja de Pandora para cualquier administrador de sistemas, ya que puede configurarlo para realizar cualquier tarea para su entorno.

Tener un servidor CentOS 7 en cualquier infraestructura puede ayudar a implementar una cantidad de servicios útiles para mantener, proteger y

administrar la infraestructura de una manera inteligente y automatizada. (Alibi & Roy, 2016)

#### **2.2.3.4. Apache TomCat**

Apache TomCat es un contenedor de servlet y de web de código abierto basado en Java, que se utiliza para alojar aplicaciones escritas en Java. Fue desarrollado por primera vez para Jakarta TomCat. Debido a un aumento en la demanda, más tarde se alojó como un proyecto separado llamado Apache TomCat, que es respaldado por The Apache Software Foundation.

Inicialmente fue desarrollado por James Duncan Davidson, un arquitecto de software en Sun Microsystems. Más tarde ayudó a que este proyecto fuera de código abierto y jugó un papel clave en la donación de este proyecto de Sun Microsystems a The Apache Software Foundation. TomCat implementa las especificaciones Java Servlet y JavaServer Pages (JSP) de Sun Microsystems, y proporciona un entorno de servidor web HTTP "puro Java" para ejecutar código Java.

Diferentes versiones de Apache TomCat están disponibles para diferentes versiones de las especificaciones de Servlet y JSP, que además incluyen muchas características adicionales que la convierten en una plataforma útil para desarrollar y desplegar aplicaciones web y servicios web. (Khare, 2012)

#### **2.2.3.5. PostgreSQL**

PostgreSQL es el más popular sistema gestor de base de datos relacional de código abierto. Su reciente aumento en popularidad proviene de usuarios de otras bases de datos que buscan un sistema con mejores garantías de

confiabilidad, mejores capacidades de consulta, más funciones, operaciones más predecibles o simplemente que quieran algo más fácil de aprender, entender, y usar. Entre las principales características de este gestor de bases de datos tenemos:

***Fácil de usar:*** los comandos SQL de PostgreSQL son consistentes entre sí y estándar. Todas las herramientas de línea de comandos toman los mismos argumentos.

***Seguro:*** PostgreSQL es completamente transaccional, incluidos los cambios estructurales destructivos. Esto significa que puede probar cualquier cosa con seguridad en una transacción, incluso eliminar datos o cambiar las estructuras de la tabla, con la seguridad de que si revierte la transacción, se revertirán todos los cambios que haya realizado. La copia de seguridad y la restauración sencillas hacen que sea trivial clonar una base de datos en la red y experimentar en una copia de base de datos privada.

***Potente:*** PostgreSQL admite muchos tipos de datos sofisticados listos para usar, incluidos JSON, XML, objetos geométricos, jerarquías, etiquetas y matrices. Los nuevos tipos de datos y funciones se pueden escribir en SQL, C o en muchos lenguajes de procedimientos integrados, incluidos Python, Perl, TCL y otros. Las extensiones agregan diversas capacidades de forma rápida y sencilla, incluida la búsqueda de texto completo, la supervisión de consultas lentas, el cifrado de contraseñas y más.

***Fiable:*** PostgreSQL es muy amigable con la administración tanto del sistema como de la base de datos. Todas las conexiones son procesos simples y pueden ser administradas por las utilidades del sistema operativo. El diseño de

carpeta estándar facilita el control de dónde se almacenan los datos para que pueda aprovechar al máximo su partición. Utiliza las instalaciones de inicio del sistema estándar en cada plataforma.

**Rápido:** PostgreSQL hace un uso estratégico de la indexación y la optimización de consultas para hacer el menor trabajo posible. Tiene uno de los planificadores de consultas más avanzados de cualquier base de datos relacional, pero expone su enunciado de razonamiento interno, por lo que puede encontrar y solucionar los problemas de rendimiento si surgen. (Lyons, 2013)

#### **2.2.3.6. Java**

Java es un lenguaje de programación de propósitos generales. Podemos usar Java para desarrollar el mismo tipo de aplicaciones que programamos con otros lenguajes como C o Pascal. Es uno de los lenguajes de programación que tiene mayor importancia sobre los demás es Java, el cual es dominante en la creación de aplicaciones empresariales, elaboración de videojuegos, sitios Web, etcétera.

Otra característica importante es que Java es multiplataforma, lo que significa que las aplicaciones programadas con este lenguaje pueden “correr” en cualquier sistema operativo y hardware. Esto lo posiciona como el lenguaje de programación más versátil de los últimos años. (Sznajdleder, 2013)

La tecnología Java que permite construir aplicaciones Webs está basada en el desarrollo de Servlets, pero esto es parte de lo que se conoce como JEE (Java Enterprise Edition). En general, Java ha disfrutado de una carrera exitosa

en el espacio empresarial durante casi dos décadas, pero todos entendemos que tener éxito hoy no garantiza el éxito mañana.

La necesidad de soluciones flexibles, robustas y escalables entregadas a través de Internet usando la web y los móviles solo está creciendo. Si bien Java cubre la mayoría de estas necesidades, el cambio es inevitable para que se adapte a los nuevos desafíos. Afortunadamente para Java, con una gran comunidad de desarrolladores a su alrededor, hay una gran cantidad de herramientas, bibliotecas y patrones arquitectónicos establecidos para ofrecer soluciones para estas complejidades comerciales.

Java EE estandariza estas soluciones y permite al desarrollador de Java aprovechar sus habilidades. El espacio empresarial para aplicaciones empresariales nunca ha sido más vibrante que ahora. Java EE 8 trae consigo APIs más nuevo y mejoras para los existentes en la creación de aplicaciones empresariales. (Padmanabhan, 2018)

#### **2.2.3.7. HTML**

Para crear páginas web, los desarrolladores utilizan un lenguaje informático llamado lenguaje de marcado de hipertexto (Hypertext Markup Language, HTML). El lenguaje de marcado de hipertexto es la especificación de programación de cómo se pueden escribir las páginas web para que puedan ser entendidas y visualizadas correctamente por computadoras u otros dispositivos con acceso a Internet. (DeBolt, 2007)

Cada día, los usuarios de todo el mundo navegan por la Web, visitando sitios que presentan una amplia gama de contenido usando texto, imágenes,

videos y otras formas de multimedia. Para ver el contenido basado en web, los usuarios ejecutan un navegador web, que es un programa que los usuarios usan para ver contenido en la World Wide Web. En el sentido simple, un navegador solo sabe cómo recuperar una página de un sitio remoto y luego utilizar las instrucciones de formato que contiene la página mostrar su contenido. (Jamsa, 2013 )

El lenguaje de marcado ha ido evolucionado; y actualmente está HTML5 para el diseño de páginas webs, y se ha vuelto extremadamente popular porque admite una increíble variedad de plataformas. La misma aplicación puede funcionar bien en un escritorio, tableta y teléfono inteligente del usuario sin ninguna codificación especial por parte del desarrollador. A menudo, las bibliotecas, APIS y microservicios proporcionan contenido en una forma que coincide con el sistema host automáticamente, sin intervención del desarrollador. (Mueller, 2015 )

#### **2.2.3.8. JSP**

Java Server Pages, o JSP para abreviar, es una tecnología del lado del servidor que toma el lenguaje Java, con su simplicidad y elegancia inherentes, y lo usa para crear aplicaciones web altamente interactivas y flexibles. En el actual clima económico inseguro, tener el lenguaje Java detrás hace que JSP sea particularmente atractivo para las empresas: Java es un lenguaje abierto, lo que significa que no requiere licencias costosas y, por lo tanto, las soluciones JSP pueden ser muy rentables.

La premisa fundamental de JSP es que HTML se puede utilizar para crear la estructura básica de una página web, y el código de Java se puede combinar



con él para proporcionar los componentes dinámicos de la página que esperan los usuarios de la web. Si se tiene en claro los conceptos de HTML y páginas web, JSP ofrece una forma inmejorable de aprender a crear contenido innovador e interactivo, así como de comprender el popular lenguaje de Java.

Una página JSP es simplemente un archivo de texto común que contiene marcado (generalmente HTML) adecuado para mostrar dentro de un navegador. Los elementos que se integran en una página JSP se utilizan para proporcionar lógica de procesamiento que permite que el contenido dinámico se produzca a mediante una solicitud o petición por parte del cliente, para luego ser resuelta por el servidor. La lógica de procesamiento se desarrolla mediante código puro de Java, donde se pueden instanciar clases, para utilizar objetos con sus respectivos atributos y métodos. (Raible, y otros, 2008)

#### **2.2.3.9. Struts 2**

Struts 2 es un marco de aplicación web moderna, extensible y ágil, que es adecuado para aplicaciones web de pequeña y gran escala. Struts 2 comenzó como Trabajo web. Fue una respuesta a algunas de las deficiencias percibidas en Struts 1, posiblemente el framework de aplicaciones web Java más popular y de mayor duración.

Struts 2 reduce (y en la mayoría de los casos elimina) los vínculos con la especificación del servlet, lo que hace que el proceso de prueba sea sustancialmente más fácil. Struts 2 también permite la Inyección de Dependencia en muchos niveles, lo que significa que se mejoran la capacidad de prueba como la capacidad de reutilización. (Newton, 2009)

Apache Struts es uno de los proyectos de código abierto más exitosos jamás creados. Con la excepción de proyectos de "infraestructura" como Linux, MySQL o varios lenguajes de programación, algunos otros frameworks de código abierto han logrado tener el éxito, la popularidad, el dominio del mercado y la capacidad de cambiar la forma en que los desarrolladores piensan como Struts.

Struts 2 provee características para realizar la validación de formularios y la conversión de tipos, lo que le permite tratar los valores de sus formularios como objetos de dominio sin saturar su código. Struts 2, permite implementar JavaScript, el Modelo de Objetos de Documento y CSS, que ayuda cómo crear un comportamiento limpio y conciso del lado del cliente. (Roughley, 2008 )

#### **2.2.3.10. CSS**

CSS es un acrónimo de hojas de estilo en cascada (Cascading Style Sheets), otra especificación de programación. CSS usa reglas llamadas estilos para determinar la presentación visual. Las reglas de estilo están integradas con el contenido de la página web de varias maneras. CSS es fácil de aprender y entender, pero proporciona un control poderoso sobre la presentación de un documento HTML. Más comúnmente, CSS se combina con los lenguajes de marcado HTML o XHTML. (Jamsa, 2013 )

En HTML, los estilos se pueden escribir en el flujo del HTML, o en línea, también. CSS también se puede integrar en páginas web de otras maneras. A veces no tienes control sobre estas reglas. Los navegadores permiten a los usuarios configurar ciertas reglas de estilo CSS, de acuerdo con sus propias

preferencias. Las preferencias del usuario pueden anular la escritura de reglas de estilo, además todos los navegadores vienen con reglas de estilo incorporadas.

CSS no es nada sin HTML. Puesto que HTML proporciona páginas web con contenido y estructura significativa, y aunque puede no ser bonita por sí misma, la web no existiría sin ella. Por lo tanto, para aprovechar al máximo su capacitación CSS, necesita saber cómo escribir HTML para crear una base sólida y bien construida. La buena noticia es que cuando usa CSS en todo su sitio, HTML realmente se vuelve más fácil de escribir. (McFarland, 2015)

#### **2.2.3.11. JavaScript**

JavaScript es un lenguaje de programación utilizado principalmente como un lenguaje de scripting del lado del cliente para controlar cómo se comportan los sitios web. En este entorno, el lenguaje complementa HTML que describe la estructura de los documentos web; y CSS, el lenguaje de estilo que define su apariencia. Cuando se integra en una página web, JavaScript permite que la página ejecute código y responda dinámicamente a las acciones del usuario, proporcionando capacidad de respuesta e interactividad de una manera que no se puede lograr utilizando solo HTML.

El nombre JavaScript se deriva del lenguaje de programación Java, como una táctica de marketing de Netscape para capitalizar la popularidad de Java. Netscape quería un lenguaje simple e interpretado que fuera más fácil que Java para los programadores no profesionales para usar en la Web. JavaScript toma mucha de su sintaxis de Java, y los lenguajes están orientados a objetos, pero todavía son dos idiomas muy diferentes. A diferencia de Java, el código

JavaScript no necesita compilarse y no puede producir aplicaciones independientes.

Como lenguaje JavaScript está diseñado para ejecutarse en un entorno host, donde el host proporciona mecanismos para comunicarse con el mundo exterior. Este modo más común es el navegador web, pero los motores de JavaScript también se pueden encontrar en muchos otros entornos, como escritorios, bases de datos, sistemas integrados y entornos del lado del servidor. (Paxton, Resig, & Ferguson, 2015)

#### **2.2.3.12. SMS**

SMS (Short Message Service) es un servicio básico para intercambiar mensajes de texto cortos (con un máximo de 160 caracteres simples). A pesar de sus limitaciones, el SMS es ampliamente utilizado en la actualidad y representa uno de los servicios que aun ofrecen los operadores de telefonía móvil. (Bodic, 2005)

SMS hoy en día tiene muchas aplicaciones en todo el mundo, y aunque hace mucho tiempo fue muy popular entre la mensajería de texto personal, fue aplicación de SMS más común. Hoy en día muchas empresas aun lo utilizan para notificar a sus clientes sobre sus productos o servicios. (Hillebrand, 2010 )

#### **2.2.3.13. Arduino**

Arduino es una plataforma de creación de prototipos de electrónica de código abierto basada en hardware y software flexible y fácil de usar. Esta placa utiliza la serie de microcontroladores Atmel. Hay muchos modelos de hardware Arduino que puedes usar. (Castro, 2015)

. También es un entorno de desarrollo que permite escribir software para la placa en el lenguaje de programación Arduino. Se usa para una variedad de propósitos y efectos diferentes, desde proyectos simples como construir un termostato, hasta otros más avanzados como robótica, servidores web, sismógrafos, sistemas de seguridad para el hogar y sintetizadores. (Kulkarni, 2017)

#### **2.2.3.14. Módulo GSM/GPRS SIM900**

El modulo SIM 900 GSM es una solución completa de servicio de radio por paquetes (GPRS) GSM de cuatro bandas en un módulo de tecnología de montaje en superficie (SMT) que puede integrarse en la aplicación del cliente, permitiendo usar la red móvil para enviar SMS.

Se controla mediante comandos AT (GSM 07.07. 07.05 y Comandos AT mejorados por SIMCOM) y es totalmente compatible con Arduino. (Krishna Hari K, N, & Salim BOUHLEL, 2013)

### **2.3. Objetivos del prototipo tecnológico.**

#### **2.3.1. Objetivo General**

Mejorar la comunicación entre representantes y los docentes de la Escuela de Educación Básica Juan Antonio Montalván Cornejo de la Ciudad de Vines con el desarrollo un Sistema de Notificaciones de Incidencias Escolares a través de Mensajería Móvil de las actividades académicas de los Estudiantes.

#### **2.3.2. Objetivos Específicos**

- Diseñar una interfaz gráfica de usuario comprensible y entendible para el manejo de las actividades escolares de la escuela.

- Automatizar los procesos académicos que realiza la escuela con los docentes, estudiantes y representantes legales.
- Enviar notificaciones a los representantes legales de los estudiantes mensajes de texto móviles (SMS).

#### **2.4. Diseño del prototipo tecnológico**

La escuela será encargada de notificar directamente a cada representante legal de los estudiantes, cada una de las actividades académicas que realiza la escuela. Para ello se procedió a automatizar la mayoría de los procesos que realiza el docente con los estudiantes para luego de ello proceder a enviar las notificaciones por medio de mensajes de texto a los padres de familia y/o representantes legales.

Se utilizará la tecnología adecuada para que el uso de la aplicación sea fácil y amigable para que supla las necesidades que tiene cada usuario del sistema. Toda la información que manipule el sistema será almacenada en la base de datos, manteniendo como datos históricos, a la cual se podría acceder desde cualquier momento que se desee.

##### **2.4.1. Fase de análisis**

La directora de la escuela tiene como objetivo principal dar una solución óptima en caso de que ocurra algún problema en la institución educativa, para todos los procesos académicos se cumplan de la mejor manera, su objetivo es también. Cada docente de la escuela es el encargado de crear el horario de clases en sus respectivas aulas, así como también hacer llegar los comunicados a los respectivos representantes de alguna incidencia que ocurra en la escuela.

Durante la entrevista con la directora se pudo deducir que muchas veces estos comunicados de la escuela no llegan a los representantes, lo que provoca muchos inconvenientes en cuanto en la responsabilidad tanto de ellos y de sus representados.

#### 2.4.2. Fase de diseño

La aplicación web que se ha desarrollado además de automatizar la creación de horarios, la toma de asistencia y el control de las tareas escolares, también se encargará de enviar notificaciones a través de mensajes SMS a los determinados padres de familia, lo que facilitará al representante legal a cumplir su rol en la escuela y está informado con lo que sucede en la misma en relación con su estudiante representado.

<b>Funcionalidad de la Aplicación</b>		
Ingreso a la aplicación web	Ingresos de datos (crear , actualizar, activar, inactivar usuarios, aulas)	Enviar notificaciones Generar reportes

*Grafico 1.- Funcionalidad de la Aplicación: Administrador.*

<b>Funcionalidad de la Aplicación</b>		
Ingreso a la aplicación web	Ingresos de datos (crear , actualizar, activar, horarios, estudiantes, representantes)	Enviar notificaciones Generar reportes

*Grafico 2.- Funcionalidad de la Aplicación: Docente.*

## Diagrama de Clases

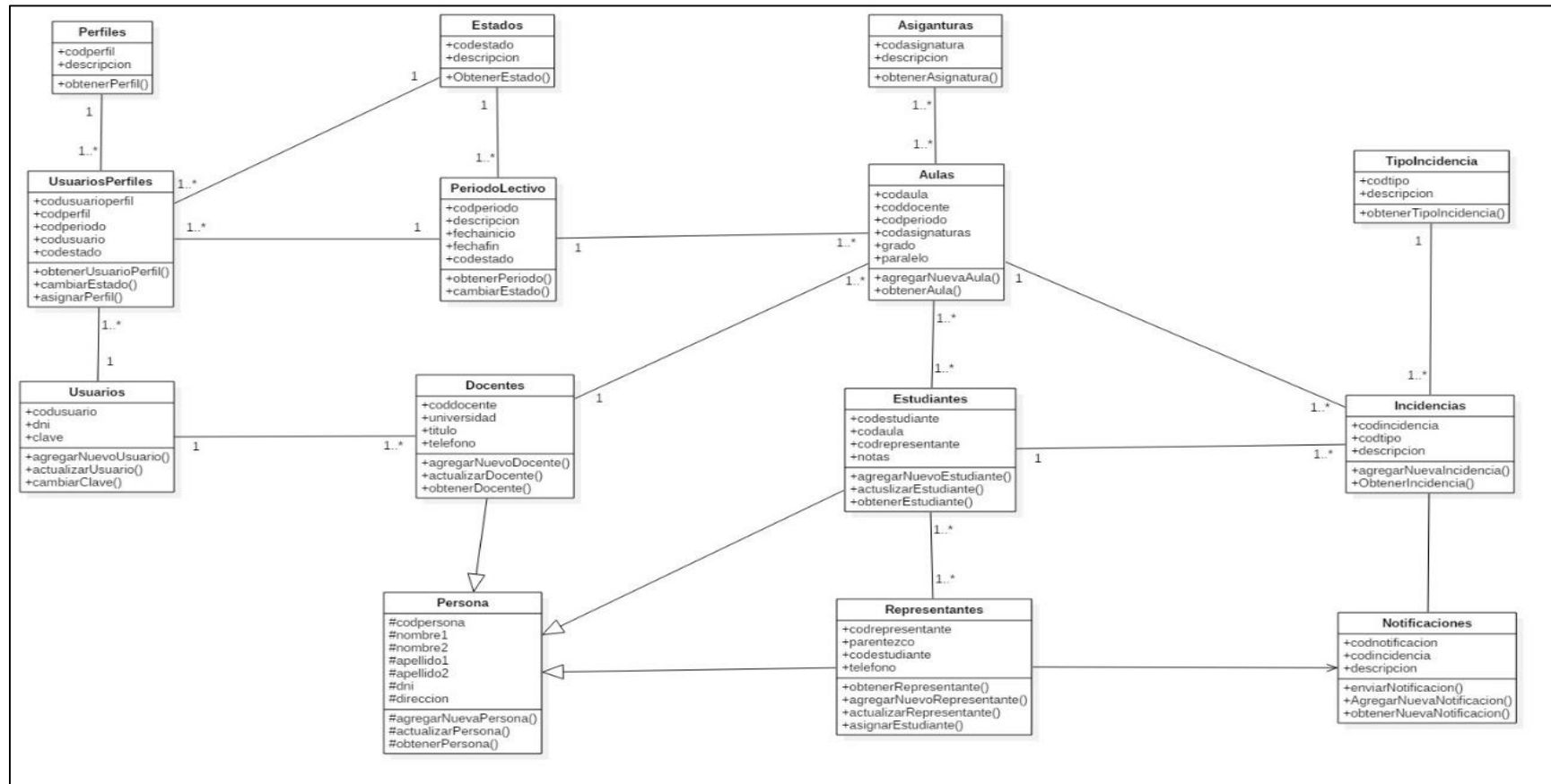


Grafico 3.- Diagrama de Clases de la Aplicación



## Diagramas de Casos de Uso

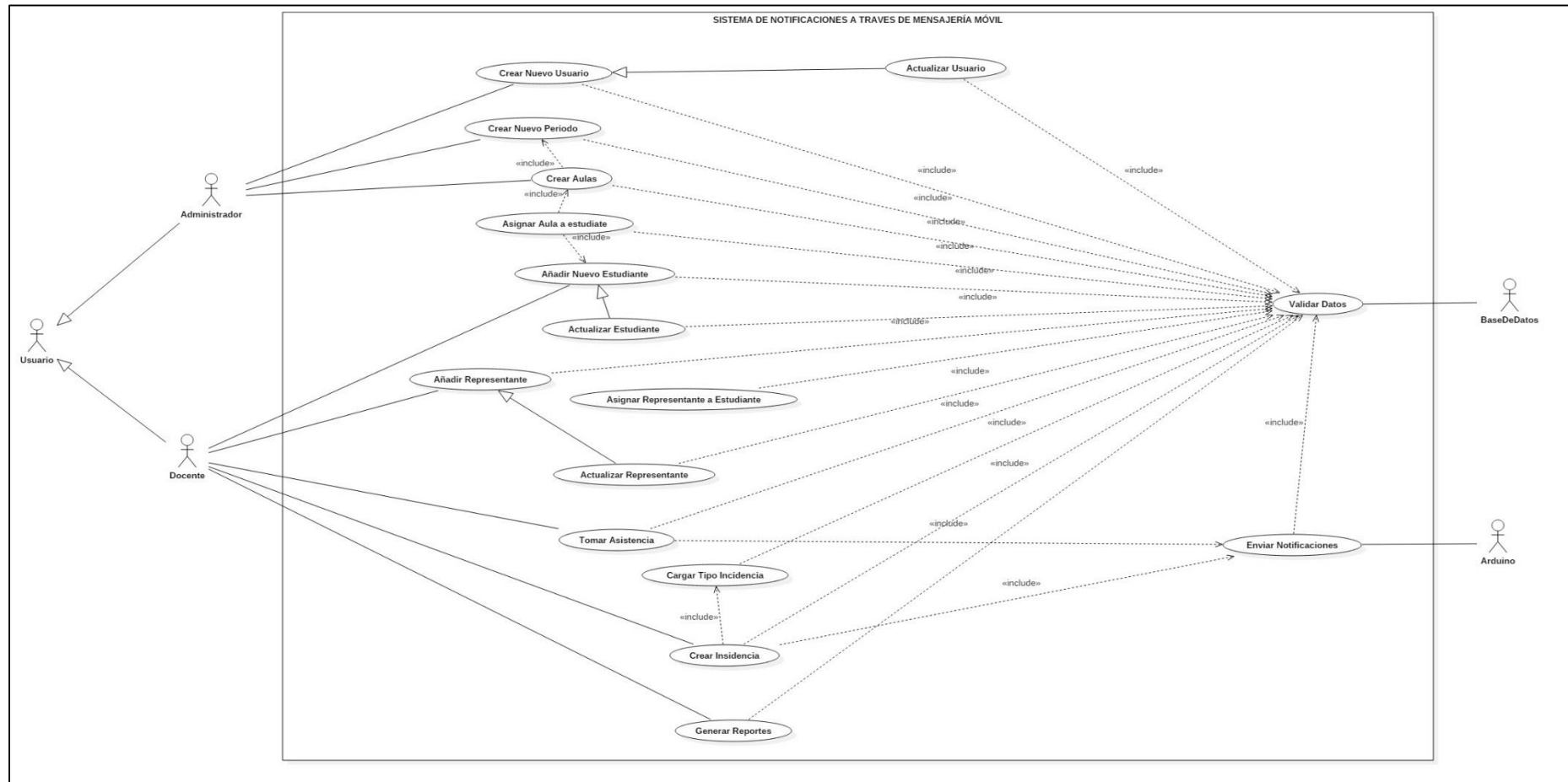


Grafico 4.- Diagrama de Casos de uso de la aplicación

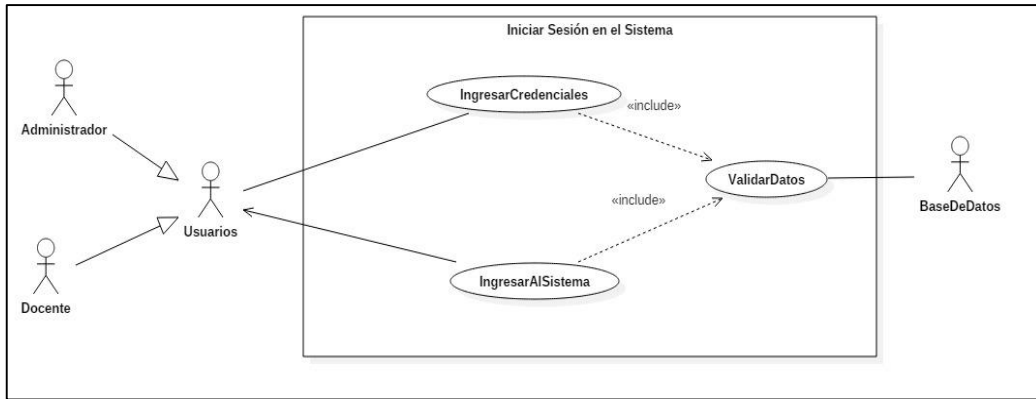


Grafico 5.- Diagrama Casos de Uso Inicio de Sesión al Sistema

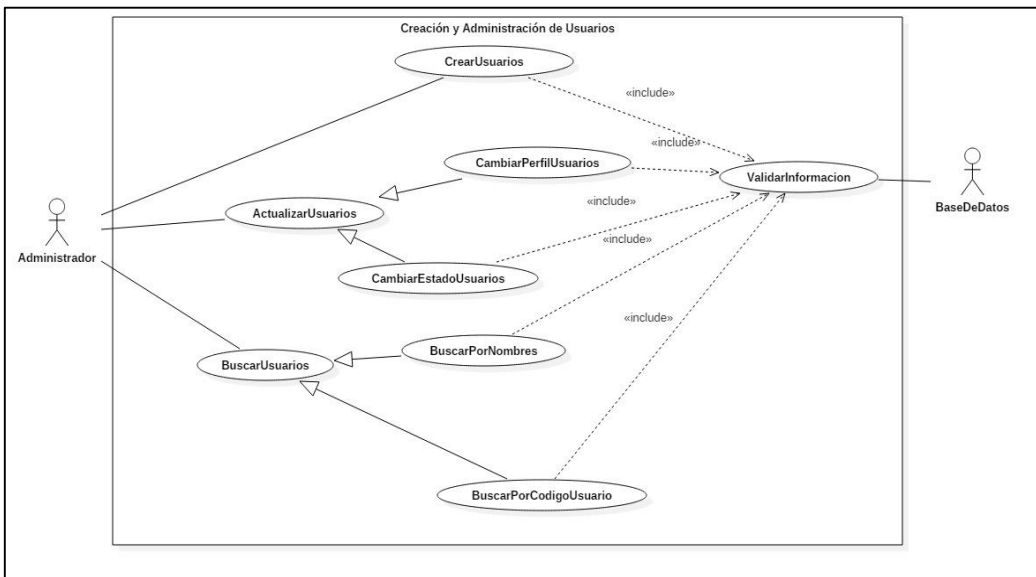


Grafico 6.- Diagrama de Casos de Uso: Administración de Usuarios

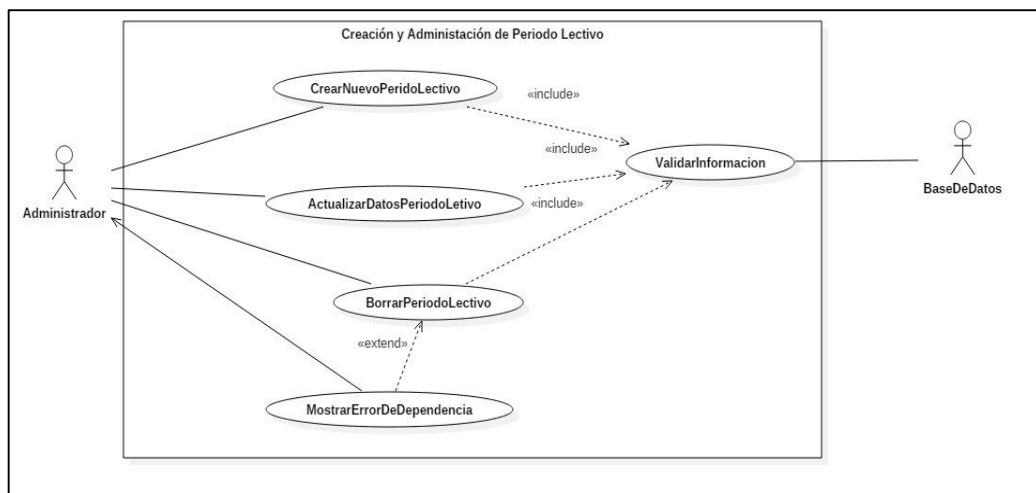


Grafico 7.- Diagrama de Casos de Uso: Administración de Periodo Lectivo

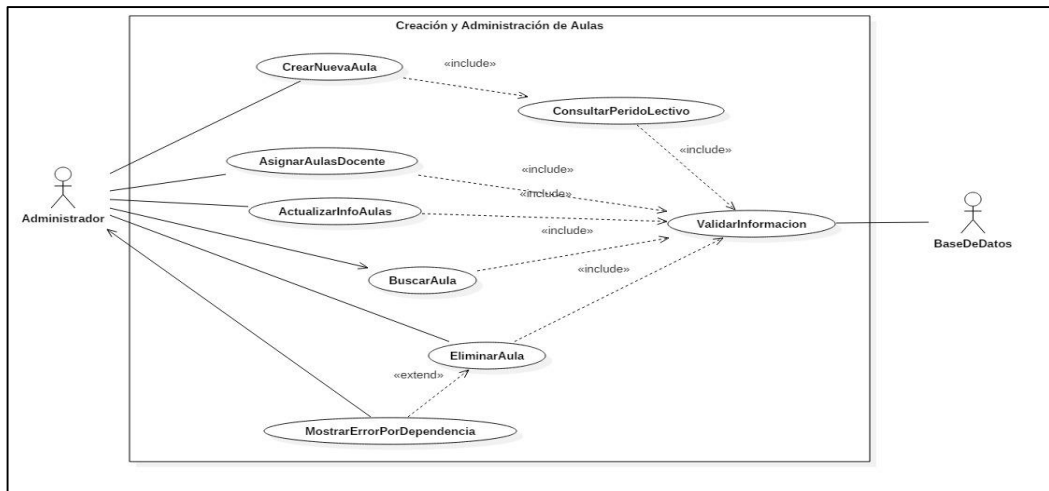


Grafico 8.- Diagrama de Casos de Uso: Administración de Aulas

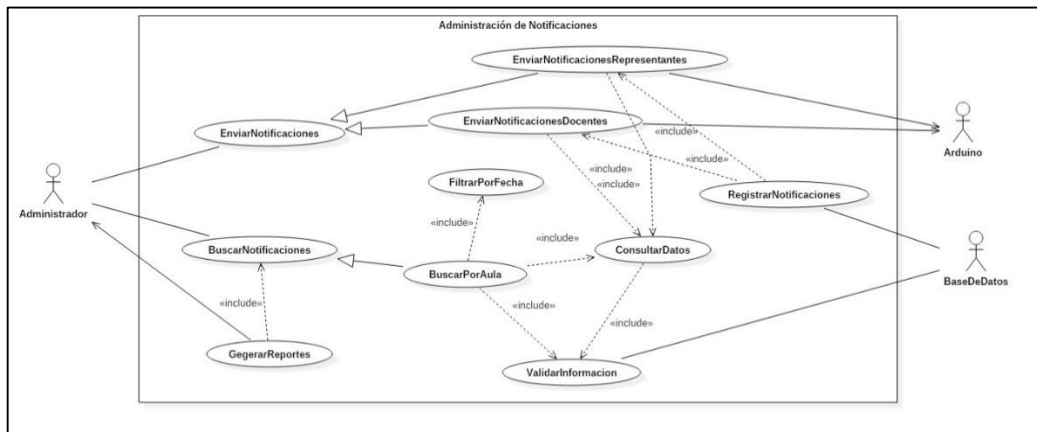


Grafico 9.- Diagrama de Casos de Uso: Administración de Notificaciones

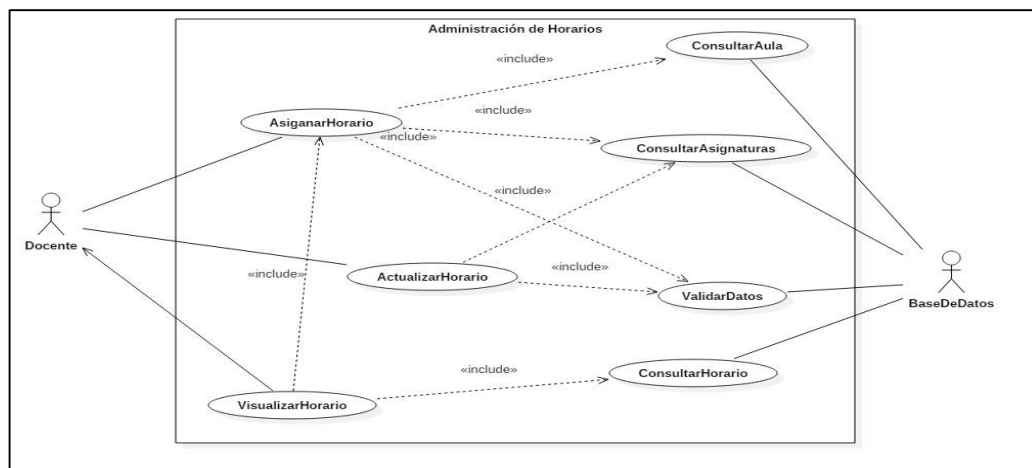


Grafico 10.- Diagrama de Casos de Uso: Administración de Horarios

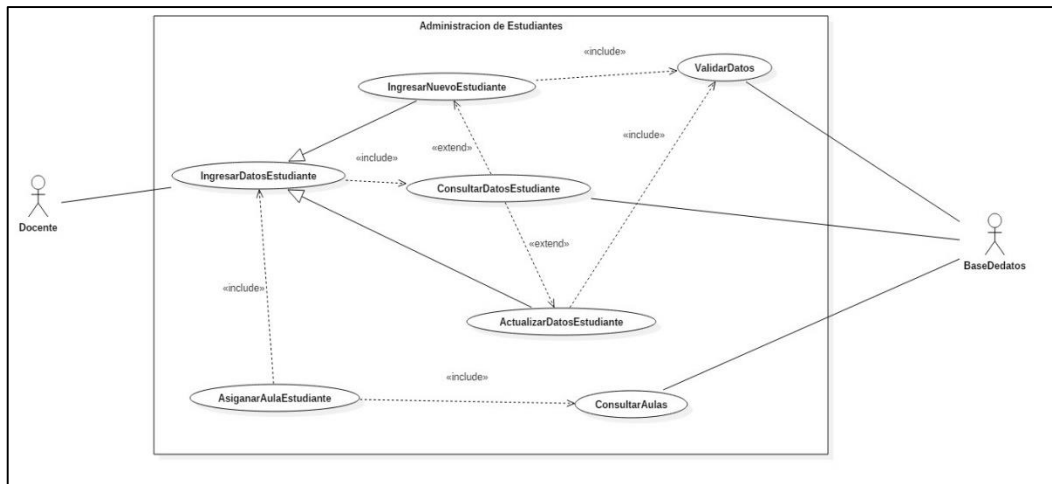


Grafico 11.- Diagrama de Casos de Uso: Administración de Estudiantes

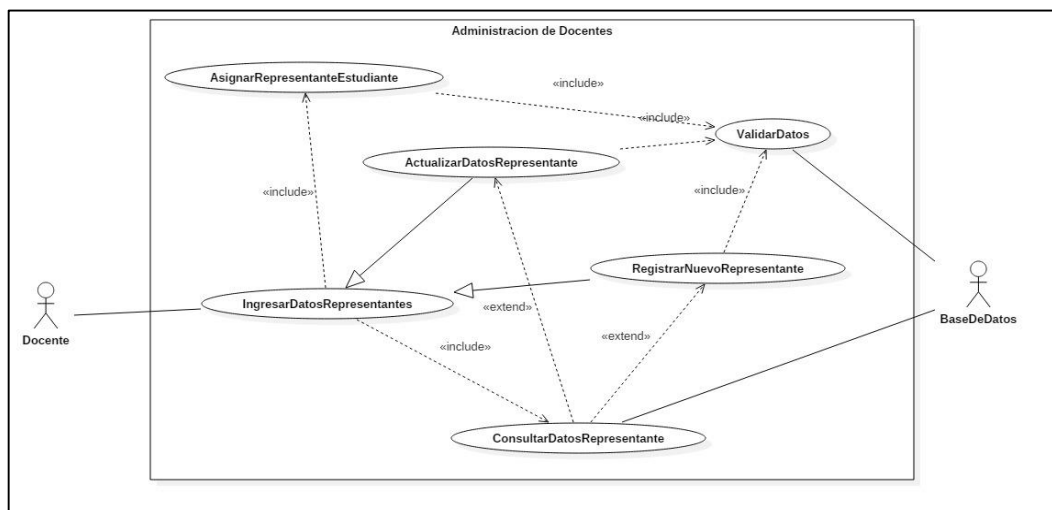


Grafico 12.- Diagrama de Casos de Uso: Administración de Docentes

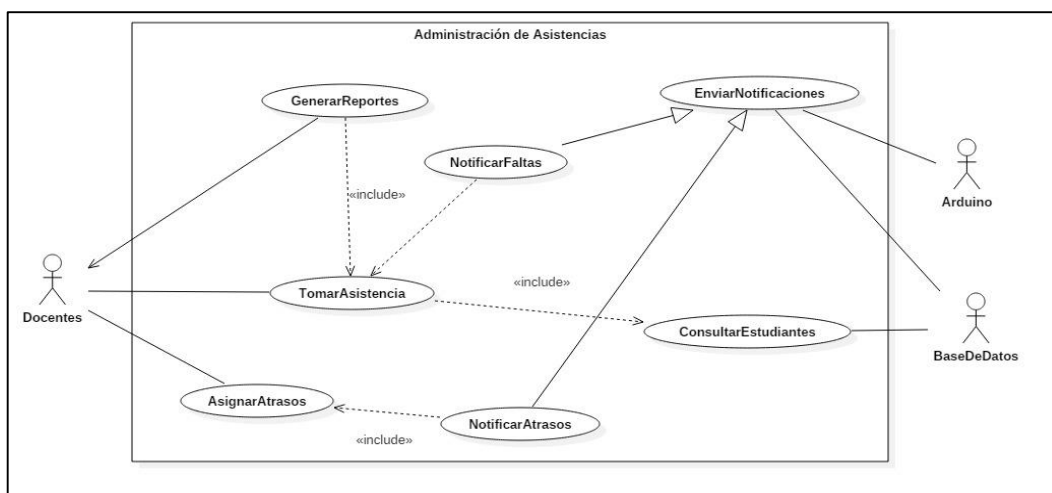


Grafico 13.- Diagrama de Casos de Uso: Administración de Asistencias

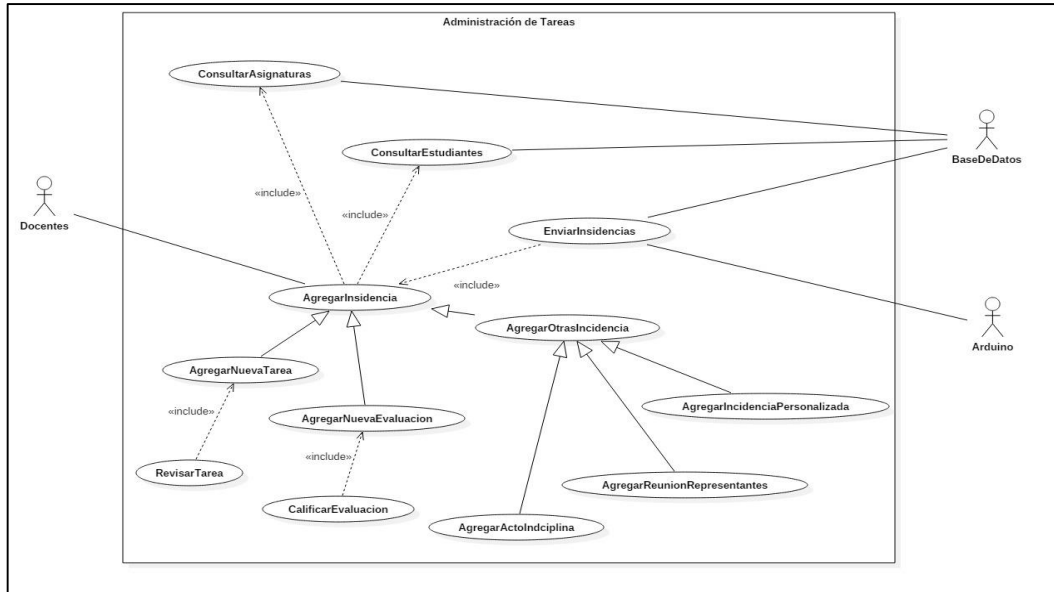


Grafico 14.- Diagrama de Casos de Uso: Administración de Tareas

### Diagramas de secuencias

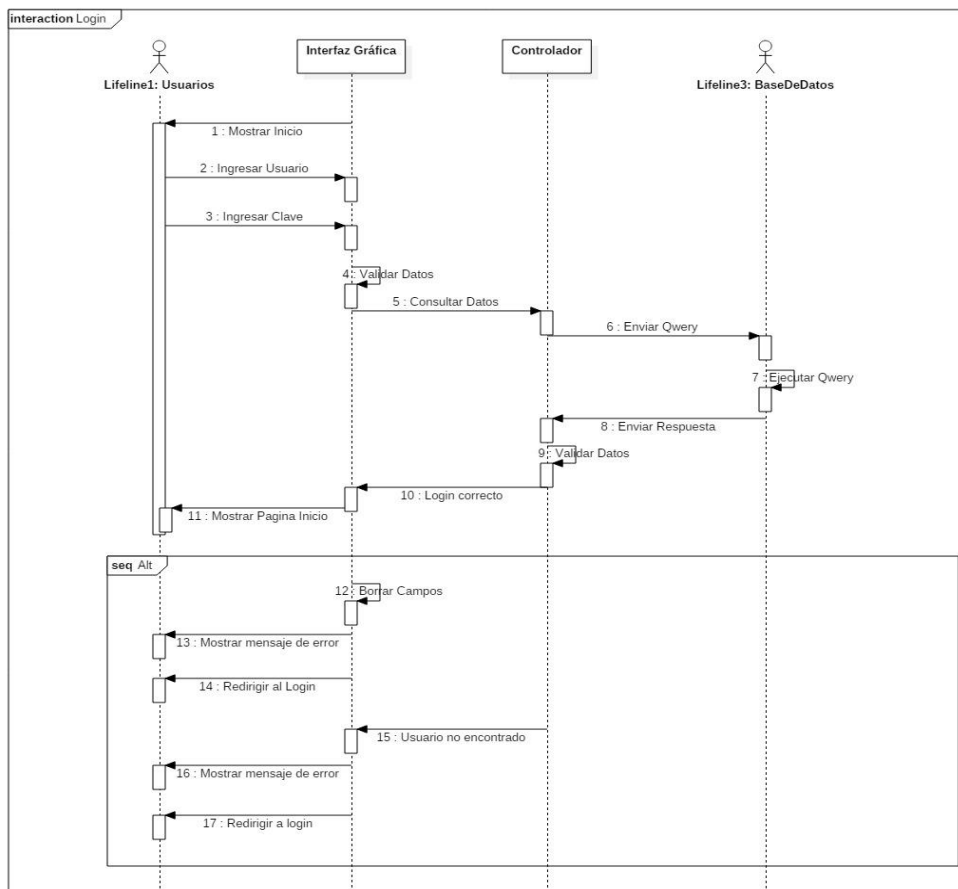


Grafico 15.- Diagrama de Secuencias: Inicio de Sesión al Sistema

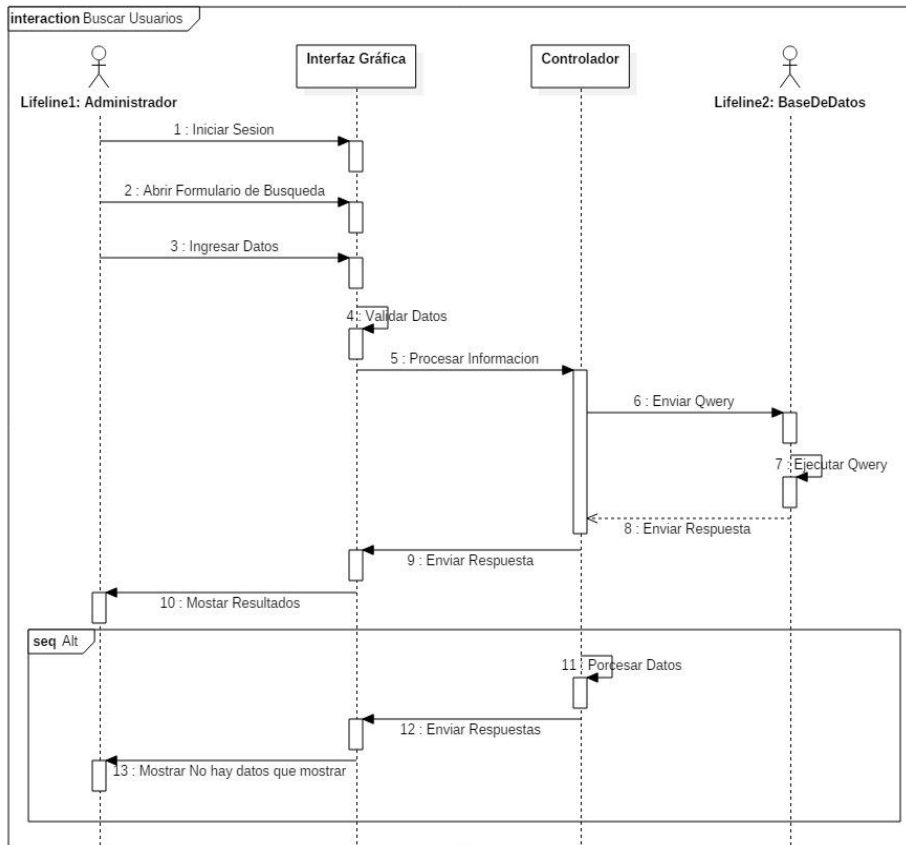


Grafico 16.- Diagrama de Secuencias: Buscar Usuarios

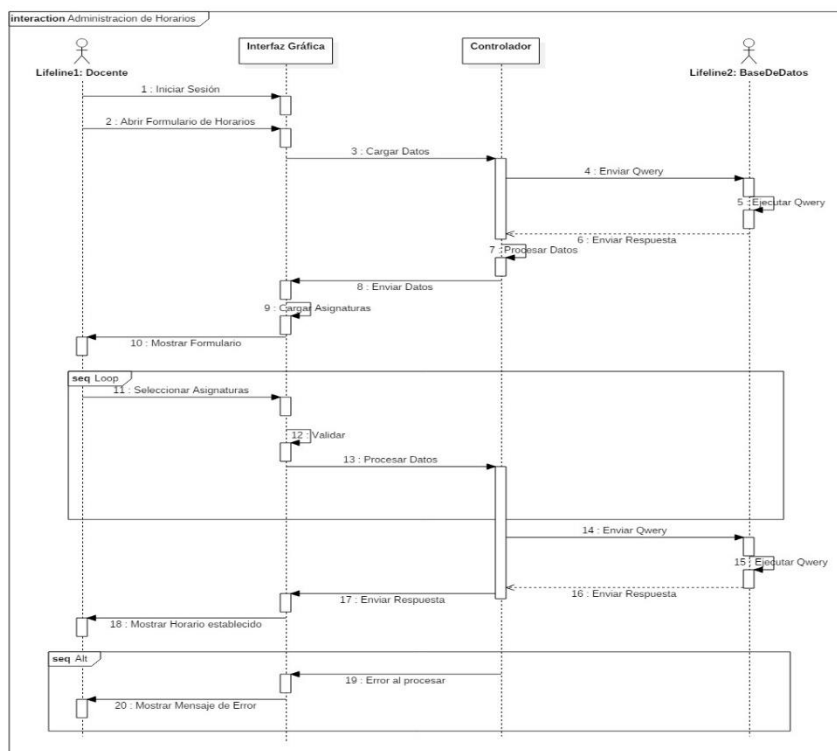


Grafico 17.- Diagrama de Secuencias:

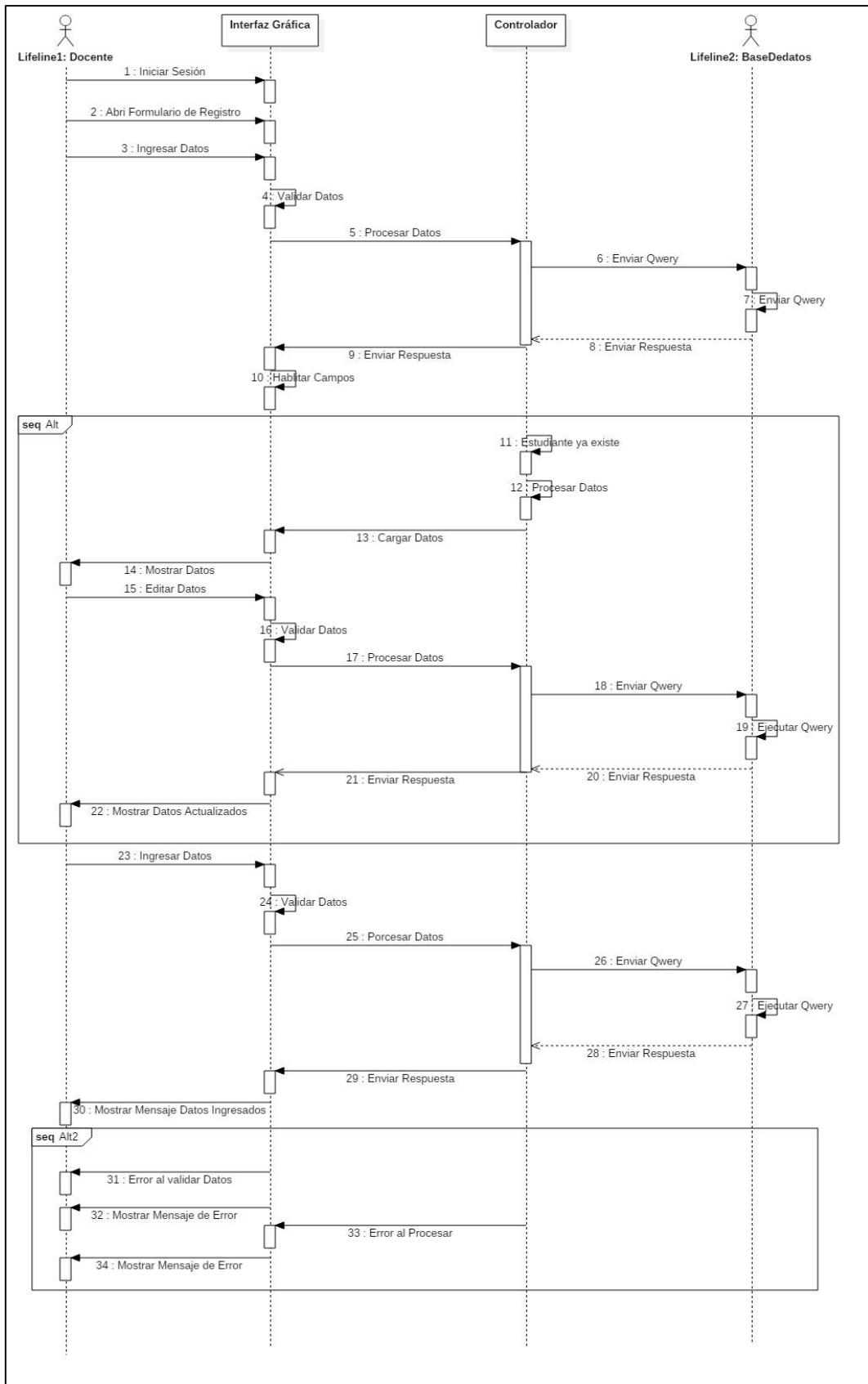


Grafico 18.- Diagrama de Secuencias: Registro de Usuarios

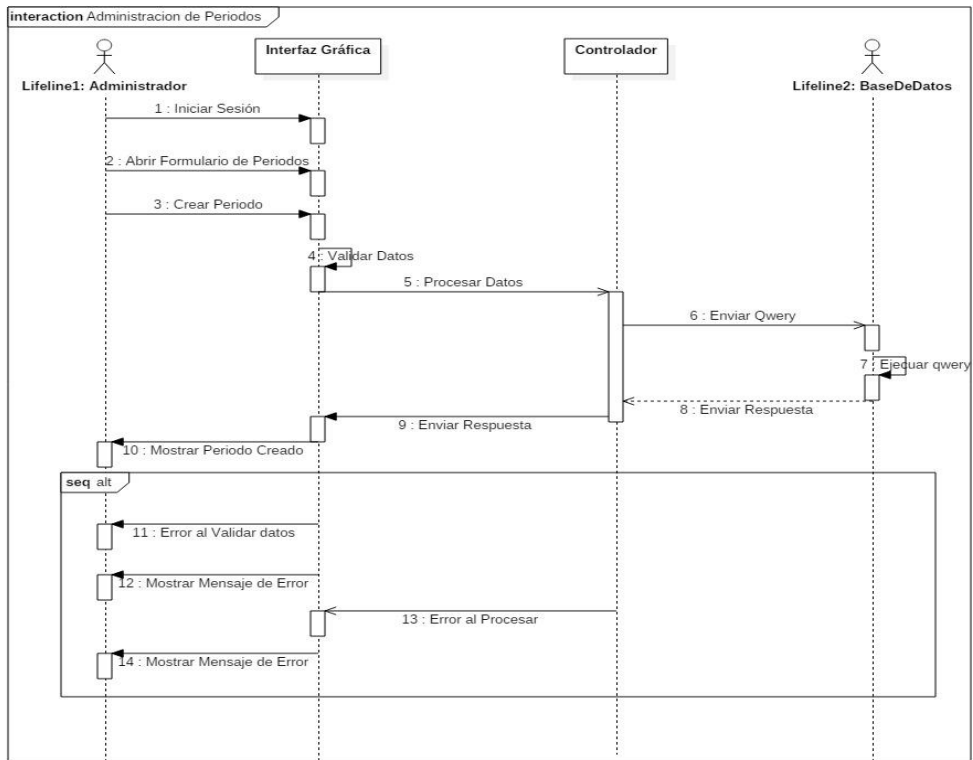


Grafico 19.- Diagrama de Secuencias: Administración de Periodos

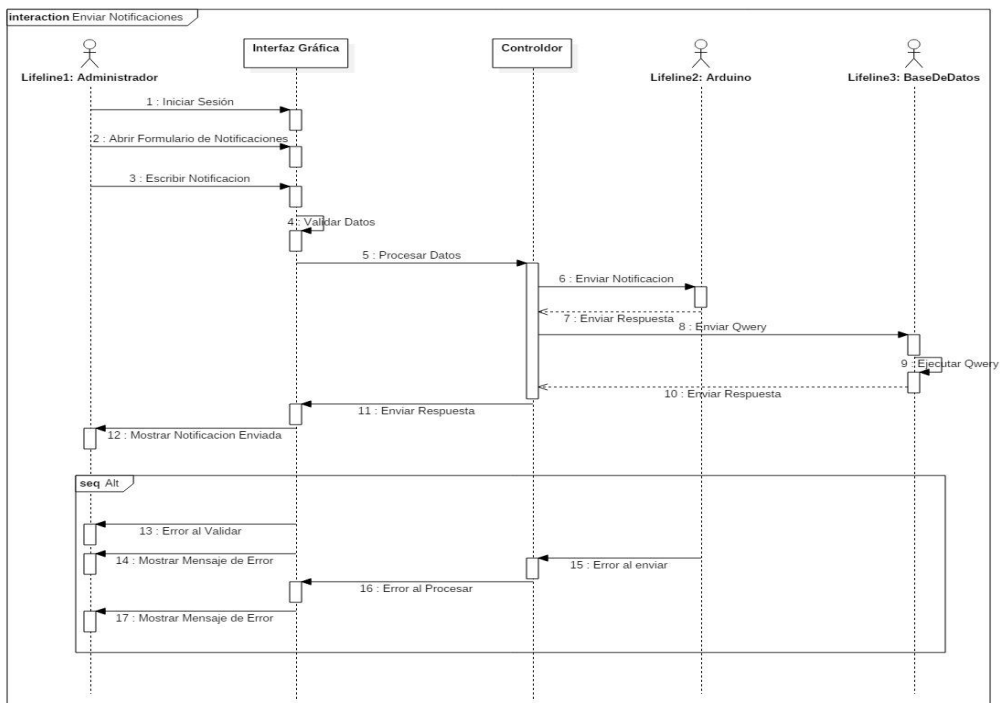


Grafico 20.- Diagrama de Secuencias: Envío de Notificaciones



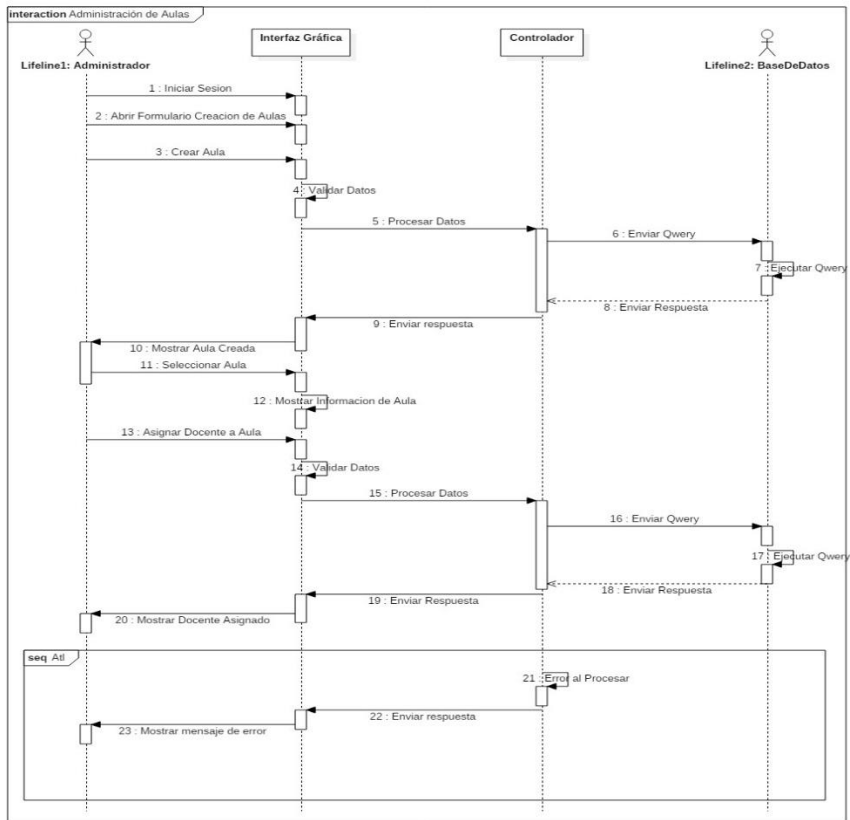


Grafico 21.- Diagrama de Secuencias: Administración de Aulas

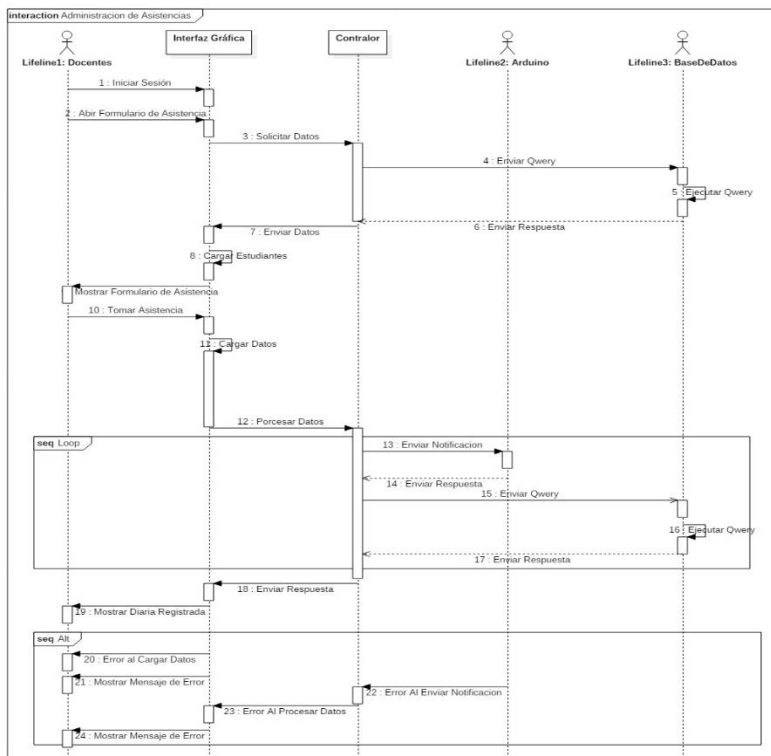


Grafico 22.- Diagrama de Secuencias: Administración de Asistencias

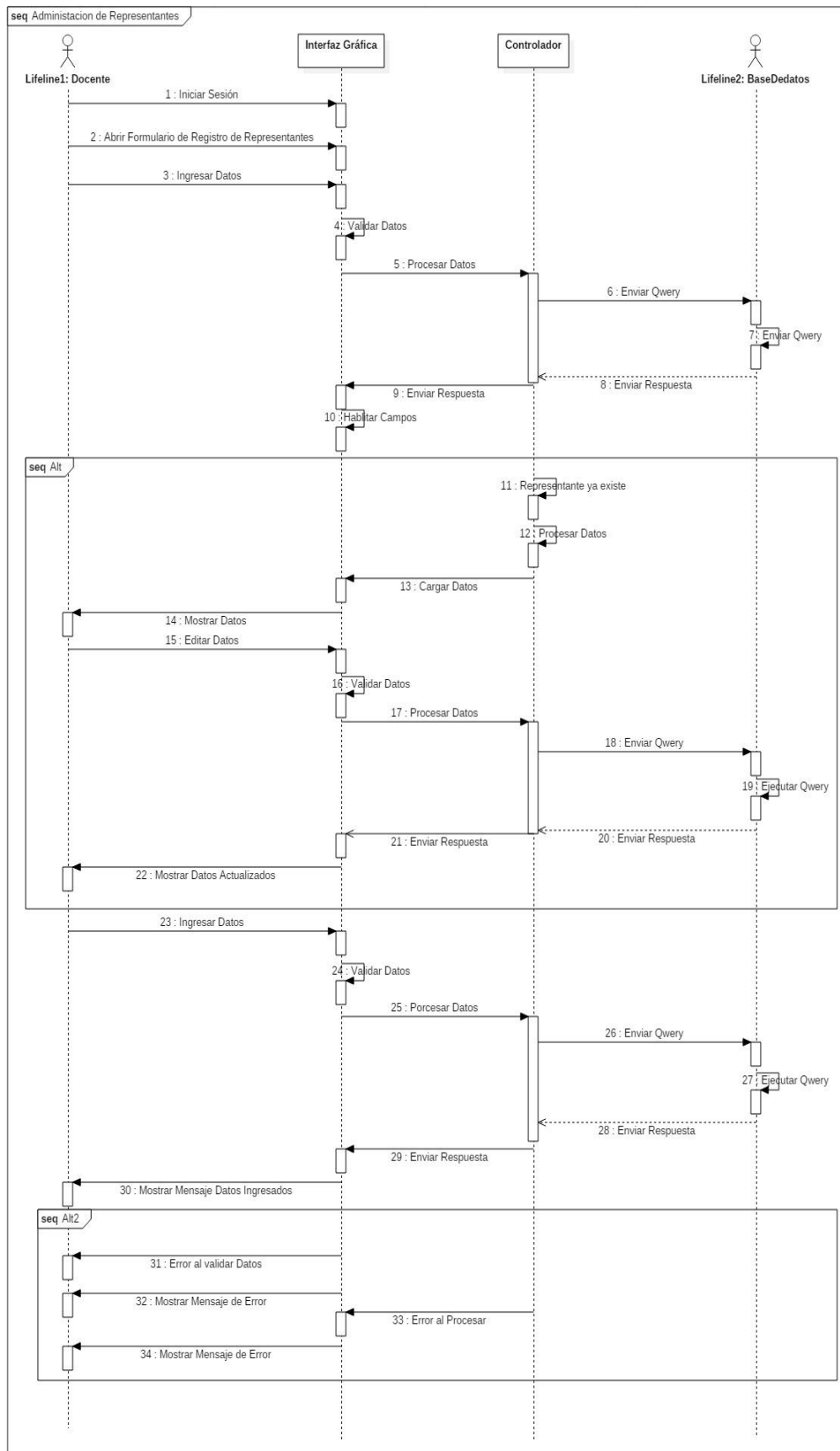


Grafico 23.- Diagrama de Secuencias: Administración de Representantes

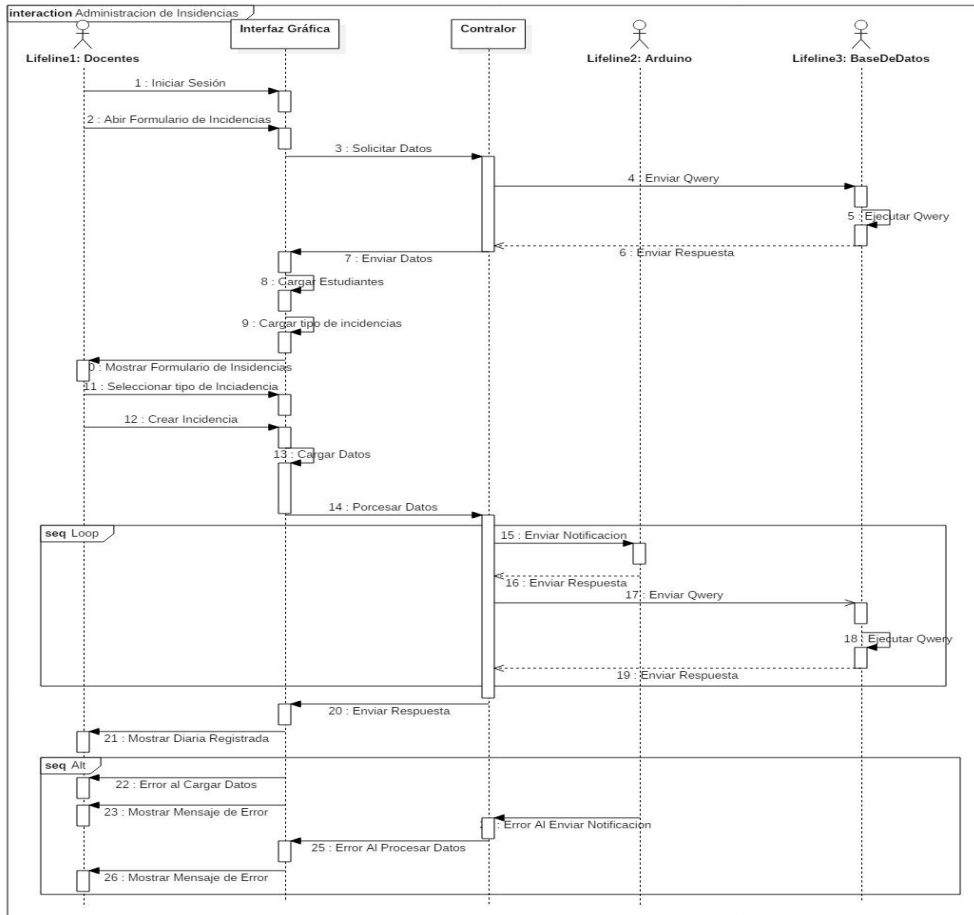


Grafico 24.- Diagrama de Secuencias: Administración de Incidencias

### Diagramas de Actividades

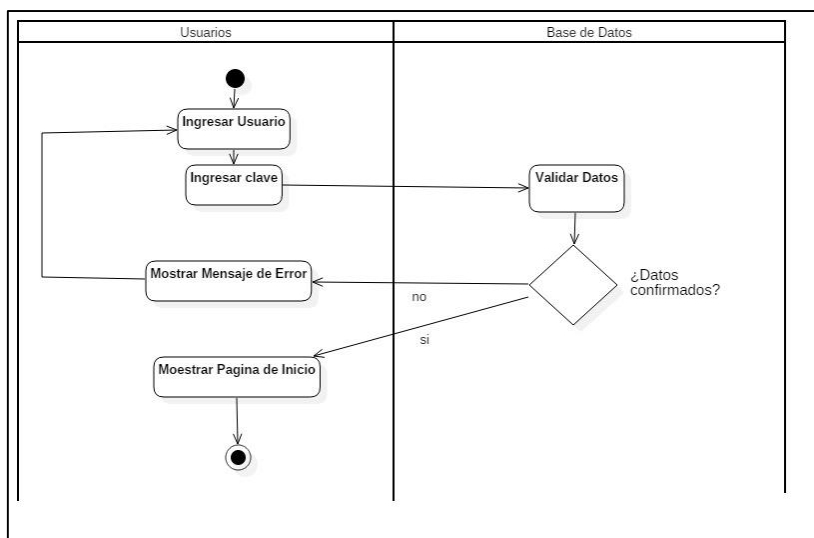


Grafico 25.- Diagrama de Actividades: Inicio de Sesión al Sistema

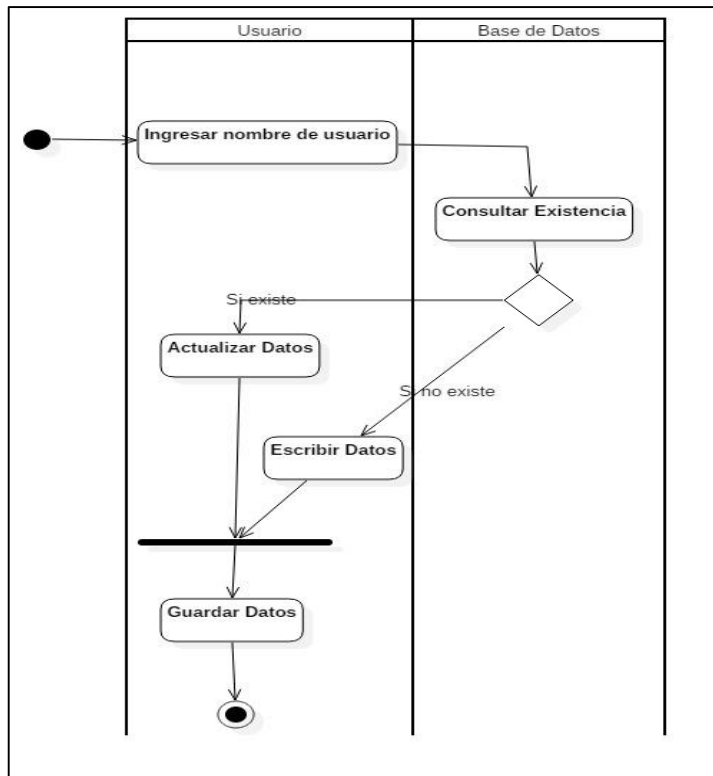


Grafico 26: Diagrama de Actividades: Administración de Usuarios

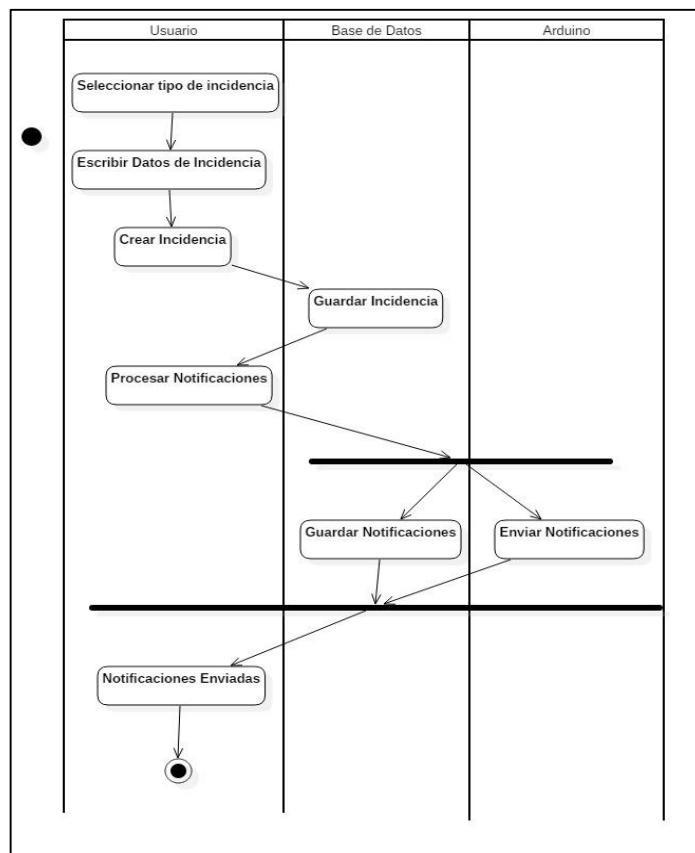


Grafico 27.- Diagrama de Actividades: Administración de Incidencias

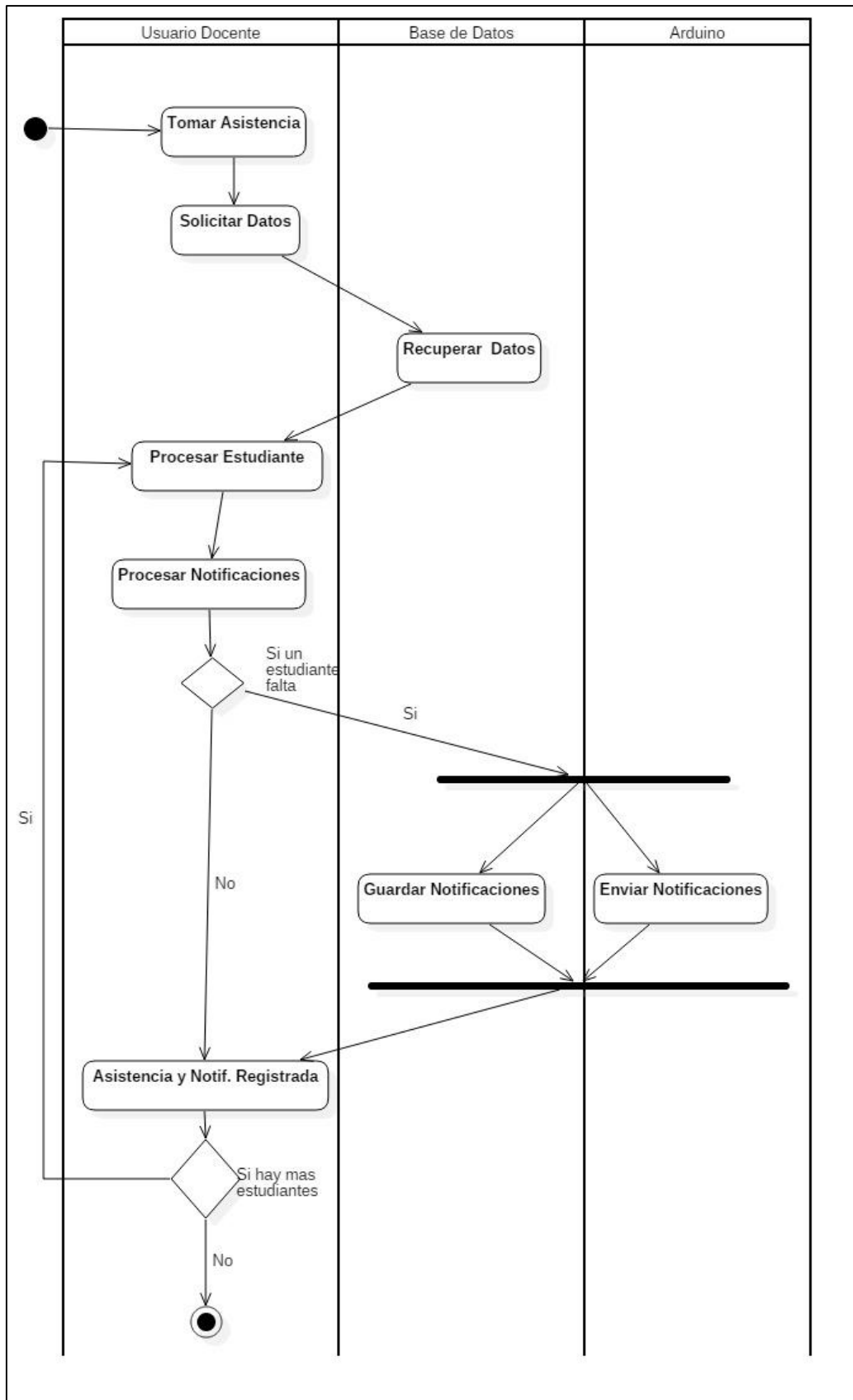


Grafico 28.- Diagrama de Actividades: Toma de Asistencias

## Modelo Conceptual

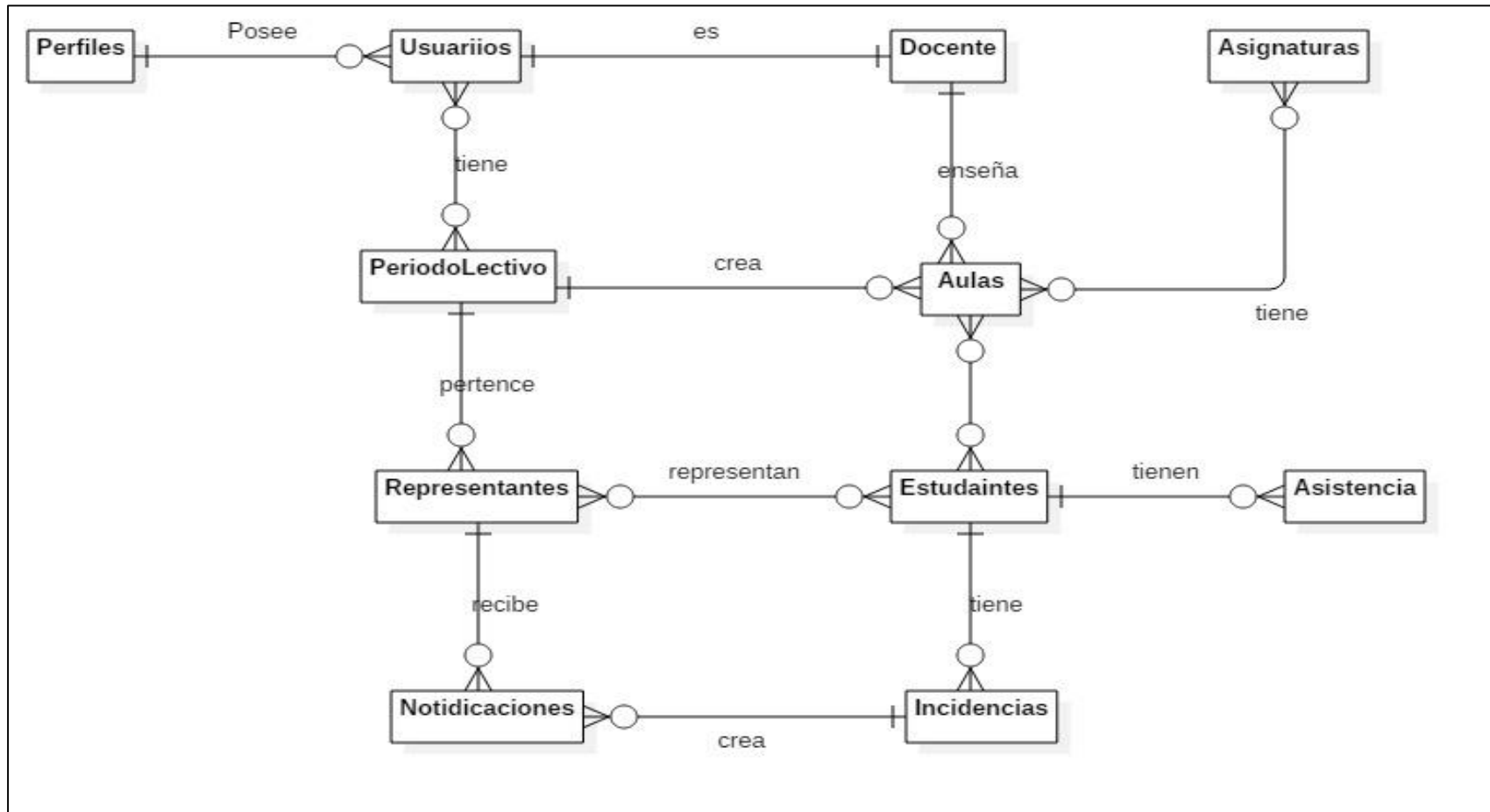


Grafico 29.- Diagrama Entidad Relación

# Modelo Lógico

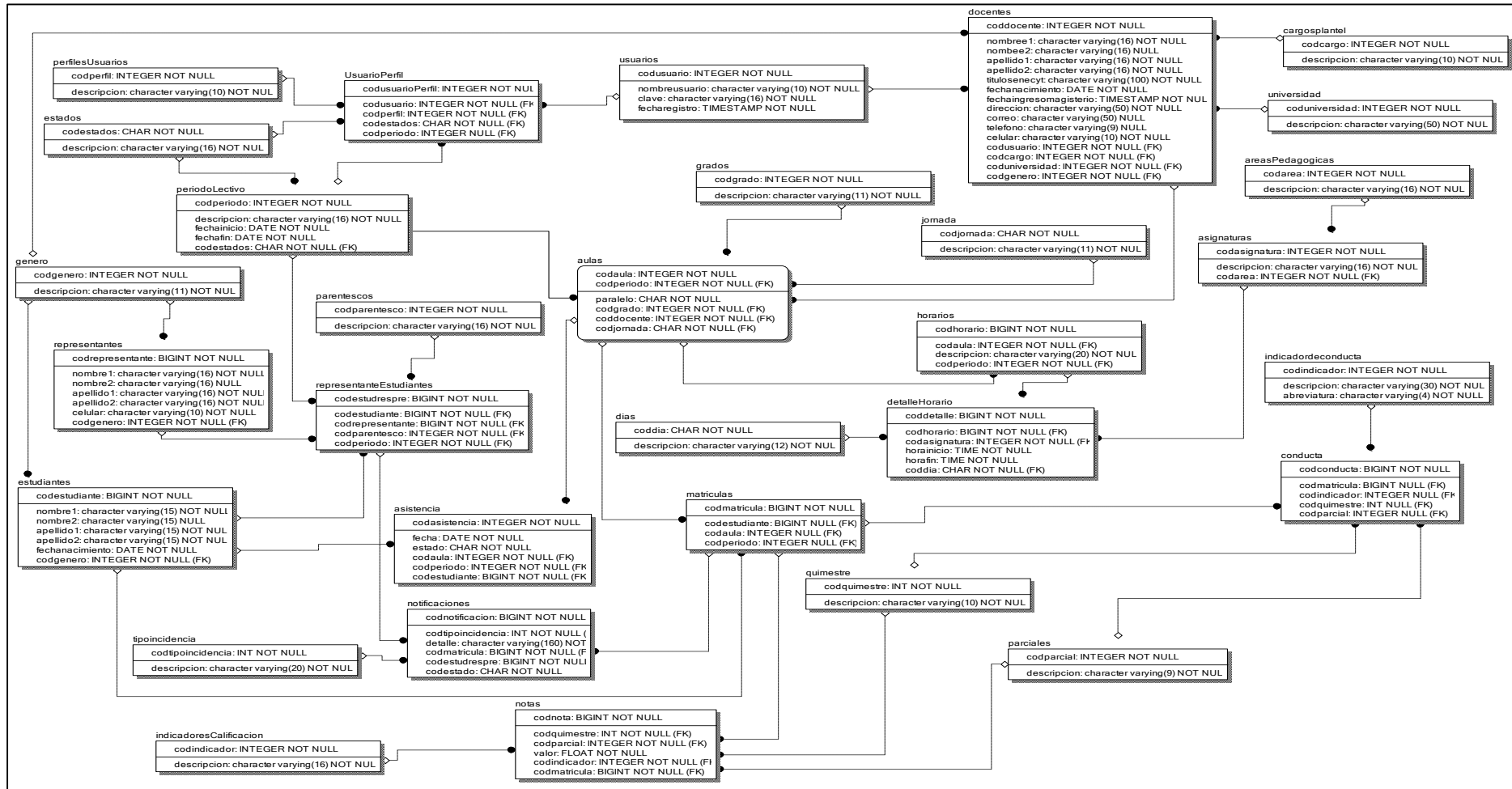


Grafico 30.- Diagrama Relacional

## Diccionario de Datos

### Tabla perfilesUsuarios

Columna	Tipo	Nulo	Primary key	Foreign key
codperfil	INTEGER	NOT NULL	Sí	
descripcion	CHARACTER VARYING(13)	NOT NULL		

*Tabla 1.- Perfiles de Usuarios*

### Tabla Estados

Columna	Tipo	Nulo	Primary key	Foreign key
codestados	CHAR	NOT NULL	Sí	
descripcion	CHARACTER VARYING(16)	NOT NULL		

*Tabla 2.- Estados*

### Tabla Usuarios

Columna	Tipo	Nulo	Primary key	Foreign key
codusuario	INTEGER	NOT NULL	Sí	
nombreusuario	CHARACTER VARYING(10)	NOT NULL		
clave	CHARACTER VARYING(16)	NOT NULL		
fecharegistro	TIMESTAMP	NOT NULL		

*Tabla 3.- Usuarios*



### Tabla periodoLectivo

Columna	Tipo	Nulo	Primary key	Foreign key
codperiodo	INTEGER	NOT NULL	Sí	
descripcion	CHARACTER VARYING(16)	NOT NULL		
fechainicio	DATE	NOT NULL		
fechafin	DATE	NOT NULL		
codestados	CHAR	NOT NULL		Sí

*Tabla 4.- Periodo Lectivo*

### Tabla UsuarioPerfil

Columna	Tipo	Nulo	Primary key	Foreign key
codusuarioPerfil	INTEGER	NOT NULL	Sí	
codusuario	INTEGER	NOT NULL		Sí
codperfil	INTEGER	NOT NULL		Sí
codestados	CHAR	NOT NULL		Sí

*Tabla 5.- Perfiles de Usuario Activos*

### Tabla Universidad

Columna	Tipo	Nulo	Primary key	Foreign key
coduniversidad	INTEGER	NOT NULL	Sí	
descripcion	CHARACTER VARYING(50)	NOT NULL		

*Tabla 6.- Universidad*

### Tabla Cargosplantel

Columna	Tipo	Nulo	Primary key	Foreign key
codcargo	INTEGER	NOT NULL	Sí	
descripcion	CHARACTER VARYING(10)	NOT NULL		

*Tabla 7.- Cargos del Plantel*

### Tabla Genero

Columna	Tipo	Nulo	Primary key	Foreign key
codgenero	INTEGER	NOT NULL	Sí	
descripcion	CHARACTER VARYING(11)	NOT NULL		

*Tabla 8.- Género*

### Tabla areasPedagogicas

Columna	Tipo	Nulo	Primary key	Foreign key
codarea	INTEGER	NOT NULL	Sí	
descripcion	CHARACTER VARYING(16)	NOT NULL		

*Tabla 9.- Áreas Pedagógicas*

### Tabla Grados

Columna	Tipo	Nulo	Primary key	Foreign key
codgrado	INTEGER	NOT NULL	Sí	
descripcion	CHARACTER VARYING(11)	NOT NULL		

*Tabla 10.- Grados*

## Tabla Docentes

Columna	Tipo	Nulo	Primary key	Foreign key
coddocente	INTEGER	NOT NULL	Sí	
nombre1	CHARACTER VARYING(16)	NOT NULL		
nombre2	CHARACTER VARYING(16)	NULL		
apellido1	CHARACTER VARYING(16)	NOT NULL		
apellido2	CHARACTER VARYING(16)	NOT NULL		
fechanacimiento	DATE	NOT NULL		
direccion	CHARACTER VARYING(50)	NOT NULL		
correo	CHARACTER VARYING(50)	NULL		
telefono	CHARACTER VARYING(9)	NULL		
celular	CHARACTER VARYING(10)	NOT NULL		
fechaingresomagisterio	TIMESTAMP	NOT NULL		
titulosenecyt	CHARACTER VARYING(100)	NOT NULL		
codusuario	INTEGER	NOT NULL		Sí
codcargo	INTEGER	NOT NULL		Sí
coduniversidad	INTEGER	NOT NULL		Sí
codgenero	INTEGER	NOT NULL		Sí

Tabla 11.- Docentes

### Tabla Estudiantes

Columna	Tipo	Nulo	Primary key	Foreign key
codestudiante	BIGINT	NOT NULL	Sí	
nombre1	CHARACTER VARYING(16)	NOT NULL		
nombre2	CHARACTER VARYING(16)	NULL		
apellido1	CHARACTER VARYING(16)	NOT NULL		
apellido2	CHARACTER VARYING(16)	NOT NULL		
fechanacimiento	DATE	NOT NULL		
codgenero	INTEGER	NOT NULL		Sí

*Tabla 12.- Estudiantes*

### Tabla Representantes

Columna	Tipo	Nulo	Primary key	Foreign key
codrepresentante	BIGINT	NOT NULL	Sí	
nombre1	CHARACTER VARYING(16)	NOT NULL		
nombre2	CHARACTER VARYING(16)	NULL		
apellido2	CHARACTER VARYING(16)	NOT NULL		
celular	CHARACTER VARYING(10)	NULL		
codgenero	INTEGER	NOT NULL		Sí

*Tabla 13.- Representantes*

### Tabla Parentescos

Columna	Tipo	Nulo	Primary key	Foreign key
codparentesco	INTEGER	NOT NULL	Sí	
descripcion	CHARACTER VARYING(16)	NOT NULL		

*Tabla 14.- Parentescos*

### Tabla representanteEstudiantes

Columna	Tipo	Nulo	Primary key	Foreign key
codestudrespre	BIGINT	NOT NULL	Sí	
codestudiante	BIGINT	NOT NULL		Sí
codrepresentante	BIGINT	NOT NULL		Sí
codparentesco	INTEGER	NOT NULL		Sí
codperiodo	INTEGER	NOT NULL		Sí

*Tabla 15.- Representantes y Estudiantes*

### Tabla Asignaturas

Columna	Tipo	Nulo	Primary key	Foreign key
codasignatura	INTEGER	NOT NULL	Sí	
descripcion	CHARACTER VARYING(16)	NOT NULL		
codarea	INTEGER	NOT NULL		Sí

*Tabla 16.- Asignaturas*

### Tabla Jornada

Columna	Tipo	Nulo	Primary key	Foreign key
codjornada	CHAR	NOT NULL	Sí	
descripcion	CHARACTER VARYING(11)	NOT NULL		

*Tabla 17.- Jornada*

### Tabla Aulas

Columna	Tipo	Nulo	Primary key	Foreign key
codaula	INTEGER	NOT NULL	Sí	
codperiodo	INTEGER	NOT NULL	Sí	
codgrado	INTEGER	NOT NULL		Sí
coddocente	INTEGER	NOT NULL		Sí
codjornada	CHAR	NOT NULL		Sí
paralelo	CHAR	NOT NULL		

*Tabla 18.- Aulas*

### Tabla Horarios

Columna	Tipo	Nulo	Primary key	Foreign key
codhorario	BIGINT	NOT NULL	Sí	
descripcion	CHARACTER VARYING(20)	NOT NULL		
codaula	INTEGER	NOT NULL		Sí
codperiodo	INTEGER	NOT NULL		Sí

*Tabla 19.- Horarios*

### Tabla Dias

Columna	Tipo	Nulo	Primary key	Foreign key
coddia	CHAR		Sí	
descripcion	CHARACTER VARYING(12)			

*Tabla 20.- Días*

### Tabla detalleHorario

Columna	Tipo	Nulo	Primary key	Foreign key
coddetalle	BIGINT	NOT NULL	Sí	
codhorario	BIGINT	NOT NULL		Sí
codasignatura	INTEGER	NOT NULL		Sí
horainicio	TIME	NOT NULL		
horafin	TIME	NOT NULL		
coddia	CHAR	NOT NULL		Sí

*Tabla 21.- Detalle de Horario*

### Tabla Matriculas

Columna	Tipo	Nulo	Primary key	Foreign key
codmatricula	BIGINT	NOT NULL	Sí	
codestudiante	BIGINT	NOT NULL		Sí
codaula	INTEGER	NOT NULL		
codperiodo	INTEGER	NOT NULL		

*Tabla 22.- Matriculas*

### Tabla indicadordeconducta

Columna	Tipo	Nulo	Primary key	Foreign key
codindicador	INTEGER	NOT NULL	Sí	
descripcion	CHARACTER VARYING(30)	NOT NULL		
abreviatura	CHARACTER VARYING(4)	NOT NULL		

*Tabla 23.- Indicadores de Conducta*

### Tabla Parcial

Columna	Tipo	Nulo	Primary key	Foreign key
codparcial	INTEGER	NOT NULL	Sí	
descripcion	CHARACTER VARYING(9)	NOT NULL		

*Tabla 24.- Parcial*

### Tabla Quimestre

Columna	Tipo	Nulo	Primary key	Foreign key
codquimestre	INT	NOT NULL	Sí	
descripcion	CHARACTER VARYING(10)	NOT NULL		

*Tabla 25.- Quimestre*



### Tabla Conducta

Columna	Tipo	Nulo	Primary key	Foreign key
codconducta	BIGINT	NOT NULL	Sí	
codindicador	INTEGER	NOT NULL		Sí
codmatricula	BIGINT	NOT NULL		Sí
codquimestre	INTEGER	NOT NULL		Sí
codparcial	INTEGER	NOT NULL		Sí

*Tabla 26.- Conducta*

### Tabla indicadoresCalificacion

Columna	Tipo	Nulo	Primary key	Foreign key
codindicador	INTEGER	NOT NULL	Sí	
descripcion	CHARACTER VARYING(16)	NOT NULL		

*Tabla 27.- Indicadores de Calificación*

### Tabla Notas

Columna	Tipo	Nulo	Primary key	Foreign key
codnota	BIGINT	NOT NULL	Sí	
codparcial	INTEGER	NOT NULL		Sí
codindicador	INTEGER	NOT NULL		Sí
codquimestre	INTEGER	NOT NULL		Sí
codmatricula	BIGINT	NOT NULL		Sí
valor	FLOAT	NOT NULL		

*Tabla 28.- Notas*

### Tabla Tipoincidencia

Columna	Tipo	Nulo	Primary key	Foreign key
codtipoincidencia	INTEGER	NOT NULL	Sí	
descripcion	CHARACTER VARYING(20)	NOT NULL		

*Tabla 29.- Tipo de Incidencias*

### Tabla Notificaciones

Columna	Tipo	Nulo	Primary key	Foreign key
codnotificacion	BIGINT	NOT NULL	Sí	
codtipoincidencia	INTEGER	NOT NULL		Sí
codmatricula	BIGINT	NOT NULL		
codestudrespre	BIGINT	NOT NULL		
codestado	CHAR	NOT NULL		
detalle	CHARACTER VARYING(160)	NOT NULL		

*Tabla 30.- Notificaciones*

### Tabla Asistencia

Columna	Tipo	Nulo	Primary key	Foreign key
codasistencia	INTEGER	NOT NULL	Sí	
fecha	DATE	NOT NULL		
estado	CHAR	NOT NULL		Sí
codaula	INTEGER	NOT NULL		Sí
codperiodo	INTEGER	NOT NULL		Sí

*Tabla 31.- Asistencia*

## Tabla DetalleAsistencia

Columna	Tipo	Nulo	Primary key	Foreign key
coddetalle	BIGINT	NOT NULL	Sí	
codasistencia	INTEGER	NOT NULL		Sí
estado	CHAR	NOT NULL		Sí
codestudiante	INTEGER	NOT NULL		Sí

*Tabla 32.- Detalle de Asistencia*

### Modelo físico

#### Script de la base de datos

```
CREATE TABLE perfilesUsuarios (  
codperfil INTEGER NOT NULL PRIMARY KEY,  
descripcion character varying(13) NOT NULL  
);
```

```
CREATE TABLE estados (  
codestados CHAR NOT NULL PRIMARY KEY,  
descripcion character varying(16) NOT NULL );
```

```
CREATE TABLE usuarios (  
codusuario INTEGER NOT NULL PRIMARY KEY,  
nombreusuario character varying(10) NOT NULL ,  
clave character varying(16) NOT NULL ,  
fecharegistro TIMESTAMP NOT NULL  
);
```

```
CREATE TABLE periodoLectivo (  
codperiodo INTEGER NOT NULL PRIMARY KEY,  
descripcion character varying(16) NOT NULL ,  
fechainicio DATE NOT NULL ,  
fechafin DATE NOT NULL ,  
codestados CHAR NOT NULL  
);
```

```
CREATE TABLE UsuarioPerfil (  
codusuarioPerfil INTEGER NOT NULL PRIMARY KEY,  
codusuario INTEGER NOT NULL  
references usuarios on update cascade on delete restrict,  
codperfil INTEGER NOT NULL  
references perfilesUsuarios on update cascade on delete restrict,  
codestados CHAR NOT NULL  
references estados on update cascade on delete restrict,  
codperiodo INTEGER NOT NULL  
references periodoLectivo on update cascade on delete restrict  
);
```

```
CREATE TABLE universidad (  
coduniversidad INTEGER NOT NULL PRIMARY KEY,  
descripcion character varying(50) NOT NULL  
);
```

```
CREATE TABLE cargosplantel (  
codcargo INTEGER NOT NULL PRIMARY KEY,  
descripcion character varying(10) NOT NULL  
);
```

```
CREATE TABLE genero (  
codgenero INTEGER NOT NULL PRIMARY KEY,  
descripcion character varying(11) NOT NULL  
);
```

```
CREATE TABLE docentes (  
coddocente INTEGER NOT NULL PRIMARY KEY,  
nombre1 character varying(16) NOT NULL ,  
nombre2 character varying(16) ,  
apellido1 character varying(16) NOT NULL ,  
apellido2 character varying(16) NOT NULL ,  
fechanacimiento DATE NOT NULL ,  
direccion character varying(50) NOT NULL ,  
correo character varying(50) NOT NULL ,  
telefono character varying(9) ,  
celular character varying(10) ,  
fechaingresomagisterio TIMESTAMP NOT NULL ,  
titulosenecyt character varying(100) NOT NULL ,  
codusuario INTEGER  
references usuarios on delete restrict on update cascade,
```

```
codcargo INTEGER
references cargosplantel on delete restrict on update cascade,
coduniversidad INTEGER
references universidad on delete restrict on update cascade,
codgenero INTEGER
references genero on delete restrict on update cascade
);
```

```
CREATE TABLE estudiantes (
codestudiante BIGINT NOT NULL PRIMARY KEY,
nombre1 character varying(15) NOT NULL ,
nombre2 character varying(15) ,
apellido1 character varying(15) NOT NULL ,
apellido2 character varying(15) NOT NULL ,
fechanacimiento DATE NOT NULL ,
codgenero INTEGER NOT NULL
references genero on delete restrict on update cascade);
```

```
CREATE TABLE representantes (
codrepresentante BIGINT NOT NULL PRIMARY KEY,
nombre1 character varying(16) NOT NULL ,
nombre2 character varying(16) ,
apellido1 character varying(16) ,
apellido2 character varying(16) NOT NULL ,
celular character varying(10) NOT NULL ,
```

```
codgenero INTEGER NOT NULL  
references genero on delete restrict on update cascade  
);
```

```
CREATE TABLE grados (  
codgrado INTEGER NOT NULL PRIMARY KEY,  
descripcion character varying(11) NOT NULL  
);
```

```
CREATE TABLE jornada (  
codjornada CHAR NOT NULL PRIMARY KEY,  
descripcion character varying(11) NOT NULL  
);
```

```
CREATE TABLE aulas (  
codaula INTEGER NOT NULL ,  
paralelo CHAR NOT NULL ,  
codgrado INTEGER NOT NULL  
references grados on delete restrict on update cascade,  
coddocente INTEGER NOT NULL  
references docentes on delete restrict on update cascade,  
codjornada CHAR NOT NULL references jornada on delete restrict on update cascade,  
codperiodo INTEGER NOT NULL  
references periodoLectivo on delete restrict on update cascade,  
constraint pk_aula primary key (codaula,codperiodo));
```

```
CREATE TABLE parentescos (  
codparentesco INTEGER NOT NULL PRIMARY KEY,  
descripcion character varying(16) NOT NULL  
);
```

```
CREATE TABLE representanteEstudiantes (  
codestudiante BIGINT PRIMARY KEY,  
codestudrespre BIGINT NOT NULL ,  
codrepresentante BIGINT  
references estudiantes on delete restrict on update cascade,  
codparentesco INTEGER  
references parentescos on delete restrict on update cascade,  
codperiodo INTEGER  
references periodoLectivo on delete restrict on update cascade);
```

```
CREATE TABLE areasPedagogicas (  
codarea INTEGER NOT NULL PRIMARY KEY,  
descripcion character varying(16)  
);
```

```
CREATE TABLE asignaturas (  
codasignatura INTEGER NOT NULL PRIMARY KEY,  
descripcion character varying(16) NOT NULL ,  
codarea INTEGER NOT NULL  
references areasPedagogicas on delete restrict on update cascade);
```



```
CREATE TABLE horarios (  
codhorario BIGINT NOT NULL PRIMARY KEY,  
codaula INTEGER NOT NULL ,  
codperiodo INTEGER NOT NULL,  
constraint fk_horario_aula foreign key (codaula,codperiodo)  
references aulas (codaula,codperiodo) on delete restrict on update cascade  
);
```

```
CREATE TABLE dias (  
coddia CHAR NOT NULL PRIMARY KEY,  
descripcion character varying(12) NOT NULL  
);
```

```
CREATE TABLE detalleHorario (  
coddetalle BIGINT NOT NULL PRIMARY KEY,  
codhorario BIGINT NOT null  
references horarios on delete restrict on update cascade,  
codasignatura INTEGER  
references asignaturas on delete restrict on update cascade,  
horainicio TIME ,  
horafin TIME ,  
coddia CHAR  
references dias on delete restrict on update cascade  
);
```

```
CREATE TABLE matriculas (  
codmatricula BIGINT NOT NULL PRIMARY KEY,  
codestudiante BIGINT  
references estudiantes on delete restrict on update cascade,  
codaula INTEGER ,  
codperiodo INTEGER ,  
constraint fk_matricula_aula foreign key(codaula,codperiodo)  
references aulas(codaula,codperiodo) on delete restrict on update cascade  
);
```

```
CREATE TABLE indicadordeconducta (  
codindicador INTEGER NOT NULL PRIMARY KEY,  
descripcion character varying(30) NOT NULL ,  
abreviatura character varying(4) NOT NULL  
);
```

```
CREATE TABLE quimestre (  
codquimestre INT NOT NULL PRIMARY KEY,  
descripcion character varying(10) NOT NULL  
);
```

```
CREATE TABLE parcial (  
codparcial INTEGER NOT NULL PRIMARY KEY,  
descripcion character varying(9) NOT NULL  
);
```

```
CREATE TABLE conducta (  
codconducta BIGINT NOT NULL PRIMARY KEY,  
codmatricula BIGINT ,  
codindicador INTEGER  
references indicadordeconducta on delete restrict on update cascade,  
codquimestre INT  
references quimestre on delete restrict on update cascade,  
codparcial INTEGER  
references parciales on delete restrict on update cascade  
);
```

```
CREATE TABLE indicadoresCalificacion (  
codindicador INTEGER NOT NULL PRIMARY KEY,  
descripcion character varying(16) NOT NULL  
);
```

```
CREATE TABLE notas (  
codnota BIGINT NOT NULL PRIMARY KEY,  
codquimestre INT NOT NULL  
references quimestre on delete restrict on update cascade,  
codparcial INTEGER NOT NULL  
references parciales on delete restrict on update cascade,  
codindicador INTEGER NOT NULL  
references indicadoresCalificacion on delete restrict on update cascade,  
codmatricula BIGINT NOT NULL
```

references matriculas on delete restrict on update cascade,

valor FLOAT NOT NULL

);

CREATE TABLE tipoincidencia(

codtipoincidencia INT NOT NULL PRIMARY KEY,

descripcion character varying(20) NOT NULL

);

CREATE TABLE notificaciones(

codnotificacion BIGINT NOT NULL PRIMARY KEY,

codtipoincidencia INT

references indicadoresCalificacion on delete restrict on update cascade,

codmatricula BIGINT

references matriculas on delete restrict on update cascade,

codestudrespre BIGINT

references representanteEstudiantes on delete restrict on update cascade,

codestado CHAR NOT NULL

references estados on delete restrict on update cascade

);

CREATE TABLE asistencia (

codasistencia INTEGER NOT NULL PRIMARY KEY,

fecha DATE NOT NULL,

estado CHAR NOT NULL

```

references estados on delete restrict on update cascade,
codaula INTEGER NOT NULL ,
codperiodo INTEGER NOT NULL ,
codestudiante BIGINT NOT NULL
references estudiantes on delete restrict on update cascade,
constraint fk_asistencia_aula foreign key(codaula,codperiodo) references
aulas(codaula,codperiodo) on delete restrict on update cascade
);

```

```

create table detalleasistencia(
coddetalle bigint primary key not null default nextval('sig_detalle'),
codasistencia BIGINT NOT NULL
references asistencia on delete restrict on update cascade,
estado CHAR NOT NULL references estados on delete restrict on update cascade,
codestudiante BIGINT NOT NULL references estudiantes on delete restrict on update
cascade
);

```

### 2.4.3. Fase de desarrollo

Las herramientas de desarrollo que se utilizan que se utilizan durante esta fase son las siguientes:

**Netbeans IDE** (Entorno de Desarrollo Integrado) un poderoso gestor de código open source, en cual se puede desarrollar aplicaciones escritas en JAVA. También soporta números lenguajes de programación haciéndolo una herramienta muy utilizada por los desarrolladores.

Para su funcionamiento se requiere el **JDK** (Kit de Desarrollo de JAVA), pieza fundamental para el funcionamiento del presente prototipo en el lado del servidor. Este Kit provee al desarrollador una variedad de herramientas para crear aplicaciones tanto web como de escritorio.

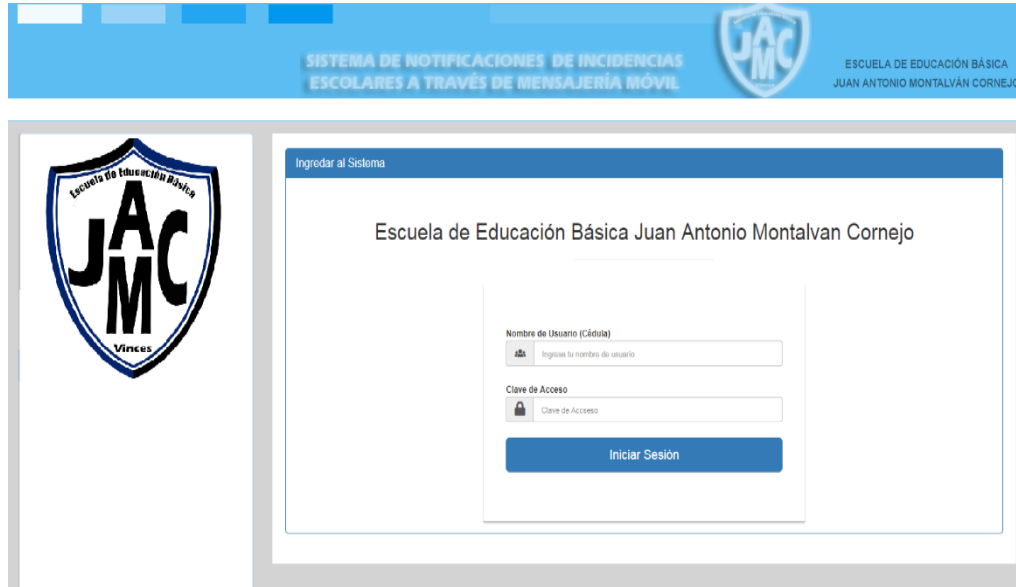
**TomCat Apache** es el servidor de aplicaciones escritas en JAVA, se lo instaló para realizar las pruebas correspondientes durante el desarrollo de prototipo. Se lo puede obtener en la página oficial de Apache, aunque viene incluido en la versión completa de Netbeans.

**PgAdmin III** se utilizó la gestión de la base de datos del prototipo. Esta herramienta provee administración visual para el Sistema Manejador de Bases de Datos PostgreSQL.

**Arduino IDE** es el entorno para programar las placas de código abierto de Arduino. La placa de Arduino para realizar las correspondientes pruebas de envío de mensajes de texto a teléfonos móviles fue Arduino UNO junto con el módulo GPRS/SIM900.

## 2.5. Ejecución del prototipo tecnológico.

Al ingresar al sistema el docente y administrador visualizarán el formulario de inicio de sesión del sistema.



The screenshot shows the login interface. At the top, there is a blue header with the system name and the school logo (JACM). The main content area is titled 'Ingresar al Sistema' and 'Escuela de Educación Básica Juan Antonio Montalvan Cornejo'. It contains two input fields: 'Nombre de Usuario (Cédula)' with a placeholder 'Ingresa tu nombre de usuario' and 'Clave de Acceso' with a placeholder 'Clave de Acceso'. Below the fields is a blue button labeled 'Iniciar Sesión'.

*Ilustración 1 Formulario de Inicio de Sesión*

Al iniciar sesión se habilitarán las funciones según el perfil de usuario asignado.



The screenshot shows the user dashboard. At the top, there is a blue header with the system name and the school logo. Below the header is a navigation bar with five items: 'ENVIAR NOTIFICACIÓN', 'VER AULAS', 'CONSULTAR HORARIOS', 'TOMAR ASISTENCIA', and 'CERRAR SESIÓN'. The main content area is titled 'BIENVENIDO/A AL SISTEMA' and 'SISTEMA DE NOTIFICACIONES DE INCIDENCIAS ESCOLARES A TRAVÉS DE MENSAJERÍA MÓVIL'. It features a user profile card for 'MARIA ELENA MARQUEZ DIRECTORA' with a circular avatar. Below the profile card is a vertical menu with items: 'Inicio', 'Usuarios', 'Docentes', 'Aulas', 'Estudiantes', 'Representantes', 'Incidencias', and 'Notificaciones'. The main content area also contains two cards: 'Tomar Asistencia' (with a clipboard icon) and 'Registrar Incidencia Escolar' (with a calendar icon). Each card has a brief description and a link to the respective function.

*Ilustración 2 Pantalla de Bienvenida*

Formulario de incidencias podrán acceder tanto administradores como docentes

The screenshot shows the 'REGISTRAR NUEVA INCIDENCIA' (Register New Incident) form. The header includes the system name 'SISTEMA DE NOTIFICACIONES DE INCIDENCIAS ESCOLARES A TRAVÉS DE MENSAJERÍA MÓVIL' and the school name 'ESCUELA DE EDUCACIÓN BÁSICA JUAN ANTONIO MONTALVÁN CORNEJO'. The left sidebar shows the user profile 'MARIA ELENA MARQUEZ DIRECTORA' and a menu with options like 'Inicio', 'Usuarios', 'Docentes', 'Aulas', 'Estudiantes', 'Representantes', 'Incidencias', and 'Notificaciones'. The main form area has a title 'REGISTRAR NUEVA INCIDENCIA' and contains the following fields: 'AULA' (dropdown menu with 'SEGUNDO B VESPERTINO'), 'TIPO INCIDENCIA' (dropdown menu with 'TAREAS'), and a text area labeled 'Escribir texto'. A blue 'ENVIAR' button is at the bottom.

*Ilustración 3 Registro de Incidencias*

Módulo administrador en el cual se registran los docentes

The screenshot shows the 'Registro de Docentes' (Teacher Registration) form. The header is the same as in the previous screenshot. The left sidebar is also the same. The main form area has a title 'Registro de Docentes' and contains the following fields: 'Cedula de Ciudadanía' (text input), 'Primer Nombre', 'Segundo Nombre', 'Primer Apellido', and 'Segundo Apellido' (text inputs), 'Genero' (dropdown menu with 'FEMENINO'), 'Cargo en la Institución' (dropdown menu with 'DOCENTE'), 'Universidad en la que Estudió' (dropdown menu with 'UNIVERSIDAD TÉCNICA DE BABAHOYO'), 'Dirección' (text input), 'Fecha de Nacimiento' (text input with '30/05/1991'), 'Título de Senecyt' (text input), 'Fecha de Ingreso al Magisterio' (text input with '30/05/1991'), 'Teléfono Convencional', 'Teléfono Celular', and 'Correo' (text inputs). A blue 'Registrar' button is at the bottom.

*Ilustración 4 Formulario de Registro de Docentes*



## Módulo de administrador, para la creación de aulas del periodo lectivo

The screenshot shows the 'Agregar Aulas' (Add Classrooms) form in the administrator module. The header includes the system name 'SISTEMA DE NOTIFICACIONES DE INCIDENCIAS ESCOLARES A TRAVÉS DE MENSAJERÍA MÓVIL' and the school name 'ESCUELA DE EDUCACIÓN BÁSICA JUAN ANTONIO MONTALVÁN CORNEJO'. The navigation bar contains 'ENVIAR NOTIFICACIÓN', 'VER AULAS', 'CONSULTAR HORARIOS', 'TOMAR ASISTENCIA', and 'CERRAR SESIÓN'. The user profile on the left is for 'MARÍA ELENA MARQUEZ DIRECTORA'. The form fields are: 'Grado' (PRIMERO), 'Paralelo' (PARALELO A), 'Docente' (with a search icon), and 'Jornada' (MATUTINO). A 'Crear Aula' button is at the bottom. Below the form is a table with columns: AULA, PARALELO, JORNADA, DOCENTE, and ACCIÓN.

AULA	PARALELO	JORNADA	DOCENTE	ACCIÓN
PRIMERO	B	VESPERTINO	JOSÉ MANUEL ROSADO ANZULES	editar borrar

*Ilustración 5 Formulario de Registro de Aulas*

## Módulo de docente, para el registro de nuevos estudiantes

The screenshot shows the 'Registro de Estudiantes' (Student Registration) form in the teacher module. The header is identical to the administrator module. The navigation bar highlights 'CONSULTAR HORARIOS'. The user profile on the left is for 'JOSE BRAVO TROUA DOCENTE'. The form fields are: 'Cédula de ciudadanía', 'Primer nombre', 'Segundo Nombre', 'Apellido Paterno', 'Apellido Materno', 'Fecha Nacimiento' (dd/mm/aaaa), and 'Genero' (FEMENINO). A 'Registrar Estudiante' button is at the bottom.

*Ilustración 6 Formulario de Registro de Estudiantes*

## Módulo de docente, para el registro de padres de familia

The screenshot shows the 'Registro de Representantes' form. At the top, there is a navigation bar with the school logo 'JMC' and the text 'SISTEMA DE NOTIFICACIONES DE INCIDENCIAS ESCOLARES A TRAVÉS DE MENSAJERÍA MÓVIL' and 'ESCUELA DE EDUCACIÓN BÁSICA JUAN ANTONIO MONTALVÁN CORNEJO'. Below the navigation bar are buttons for 'ENVIAR NOTIFICACIÓN', 'VER AULAS', 'CONSULTAR HORARIOS', 'TOMAR ASISTENCIA', and 'CERRAR SESIÓN'. On the left, there is a sidebar with a user profile for 'JOSE BRAVO AVILES DOCENTE' and a menu with options: 'Inicio', 'Usuarios', 'Docentes', 'Aulas', 'Estudiantes', 'Representantes', 'Incidencias', and 'Notificaciones'. The main form area is titled 'Registro de Representantes' and contains the following fields: 'Cédula de ciudadanía', 'Primer nombre', 'Segundo Nombre', 'Apellido Paterno', 'Apellido Materno', 'Celular', and 'Genero' (with a dropdown menu showing 'FEMENINO'). A 'Registrar Representante' button is located at the bottom of the form.

*Ilustración 7 Formulario de Registro de Representantes*

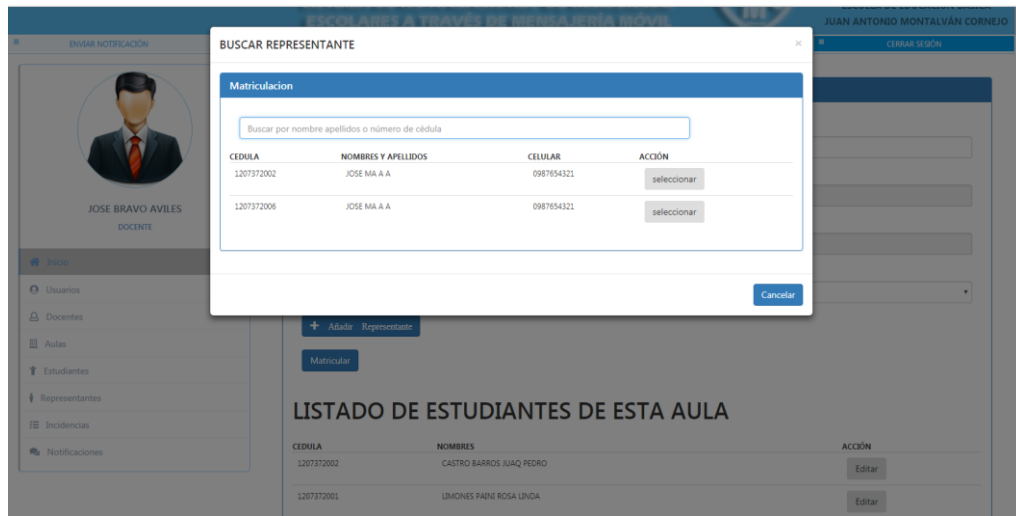
## Módulo de docente, para la matriculación de estudiantes.

The screenshot shows the 'Matriculación' form and a student list. The navigation bar and sidebar are identical to the previous screenshot. The main form area is titled 'Matriculación' and contains the following fields: 'Cedula', 'Primer nombre', 'Segundo Nombre', 'Apellido Paterno', 'Apellido Materno', 'Genero', and 'Aula' (with a dropdown menu showing 'PRIMERO B VESPERTINO'). There are two buttons: '+ Añadir Representante' and 'Matricular'. Below the form is a table titled 'LISTADO DE ESTUDIANTES DE ESTA AULA' with the following data:

CEDULA	NOMBRES	ACCIÓN
1207372002	CASTRO BARRIOS JUAQ PEDRO	Editar
1207372001	LIMONES PAINI ROSA LINDA	Editar

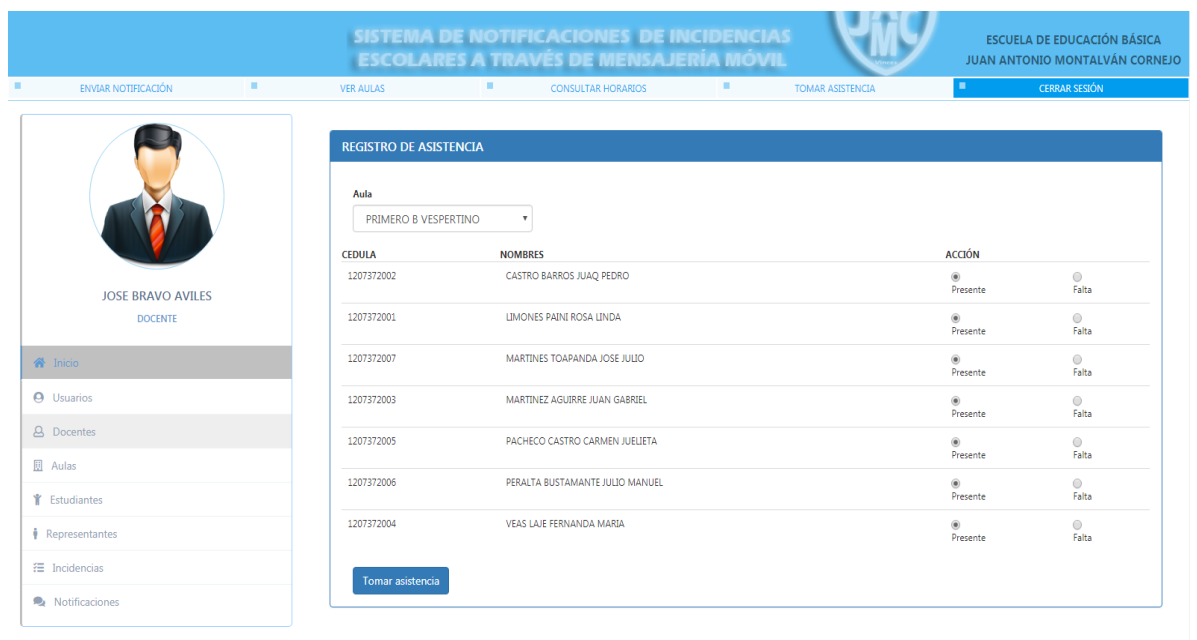
*Ilustración 8 Formulario de Registro de Matriculas*

## Módulo de docente, búsqueda de padres de familia



*Ilustración 9 Formulario de Búsqueda de Representantes*

Módulo de docente, para el registro de asistencias de los estudiantes. En este módulo se envía un mensaje automático al representante en caso el estudiante falte a clases.



*Ilustración 10 Formulario de Registro de Asistencia*

Módulo de docente, para visualizar el reporte de asistencia diario.

**REGISTRO DE ASISTENCIA**

Aula: PRIMERO B VESPERTINO

CEDULA	NOMBRES	ESTADO
1207372002	CASTRO BARRIOS JUAQ PEDRO	PRESENTE
1207372001	LIMONES PAINI ROSA LINDA	PRESENTE
1207372007	MARTINES TOAPANDA JOSE JULIO	FALTA
1207372003	MARTINEZ AGUIRRE JUAN GABRIEL	PRESENTE
1207372005	PACHECO CASTRO CARMEN JUELIETA	FALTA
1207372006	PERALTA BUSTAMANTE JULIO MANUEL	PRESENTE
1207372004	VEAS LAIE FERNANDA MARIA	FALTA

Tomar asistencia

*Ilustración 11 Reporte de Asistencia*

Formulario de búsqueda de docentes

**BUSCAR DOCENTES**

Selecciones un docente

Selecciona un docente

Buscar por nombre apellidos o número de cédula

CEDULA	NOMBRES Y APELLIDOS	TELÉFONO	CELULAR	TITULO TERCER NIVEL	ACCIÓN
1207372003	JOSÉ MANUEL ROSADO ANZULES	052912344	0991327413	LICENCIADO EN EDUCACIÓN PRIMARIA	seleccionar
1207372002	JOSÉ MANUEL ANZULES ROSADO	052912344	0991327413	LICENCIADO EN EDUCACIÓN PRIMARIA	seleccionar

Cancelar

**ESCUELA DE EDUCACIÓN BÁSICA JUAN ANTONIO MONTALVÁN CORNEJO**

Director: MARIA ELENA MARQUEZ

AULA	PARALELO	JORNADA	DOCENTE	ACCIÓN
PRIMERO	B	VESPERTINO	JOSÉ MANUEL ROSADO ANZULES	editar borrar

*Ilustración 12 Formulario de búsqueda de docentes.*

## CAPÍTULO III

### 3. Evaluación del prototipo

#### 3.1. Plan de evaluación.

<b>Cronograma de Actividades del desarrollo del Prototipo</b>	
<b>Fecha</b>	<b>Actividad</b>
17/10/2017	Entrevista con la directora de la escuela.
18/10/2017	Recopilación de requerimientos del sistema.
20/10/2017	Instalación de las herramientas necesarias para el desarrollo del prototipo.
21/10/2017	Realización de Base de Datos. Utilizando los Requerimientos del Prototipo.
26/10/2017	Análisis de requerimientos mediante la construcción de diagramas UML.
28/10/2017	Creación del modelo físico de la base de datos en el gestor de postgresQL.
30/10/2017	Creación de las relaciones entre tablas. Inserción de datos a tablas estáticas.
03/11/2017	Creación de procedimientos almacenados y funciones en postgresQL
05/11/2017	Comprobar que la Base de Datos no tenga problema realizando pruebas básicas.
15/11/2017	Desarrollo del Prototipo. Prueba de conexión entre JAVA y la Base de Datos con el Sistema.
16/11/2017	Desarrollo del Prototipo. Elaboración formulario de Inicio de Sesión.

20/11/2017	Desarrollo del Prototipo. Elaboración del Menú del Sistema.
25/11/2017	Desarrollo del Prototipo. Edición de las páginas para ver los formularios de registro, ingresar, editar, y buscar datos de los estudiantes, padres de familia, aulas y docentes.
30/11/2017	Desarrollo del Prototipo. Conexión entre java y Arduino.
01/12/2017	Desarrollo del Prototipo. Elaboración de validaciones
15/12/2017	Pruebas y testeó.
18/12/2017	Prueba de Caja blanca.

### **3.2. Resultados de la evaluación.**

El sistema informático desarrollado y testeado en la Escuela de Educación Básica Juan Antonio Montalván Cornejo. Se realizaron pruebas en tiempo real y tanto los docentes y representantes se mostraron muy satisfechos durante la realización de las pruebas. Se comprobó que el sistema cumple con los factores de eficacia, validez se reconoció que se cumplieron y se superaron las expectativas de los requerimientos planteados sobre el la construcción del prototipo que se esperaba.

Se enviaron mensajes de texto de las incidencias escolares que se iban presentando, cumpliendo a cabalidad los el objetivo de mejorar la comunicación entre representantes y docentes de todas las actividades que realiza el estudiante en la institución.

## CONCLUSIONES

El sistema informático obtuvo resultados eficientes por la Escuela de educación básica Juan Antonio Montalván Cornejo, al cumplir todos los objetivos planteados. El sistema informático para la notificación de las actividades de las actividades académicas de los estudiantes ayudará de forma eficiente a dar a conocer a los representantes sobre el estado de sus hijos.

El sistema informático está diseñado en base a la metodología orientada a objetos metodología de gran notoriedad utilizada en la elaboración de software y el patrón de diseño MVC (Modelo Vista Controlador), muy conveniente para futuras actualizaciones y fácil mantenimiento. Se utilizó el lenguaje UML (Lenguaje de Modelado Unificado) para visualizar, especificar, construir y evidenciar el funcionamiento del sistema.

Para desarrollar el sistema informático se utilizó la JAVA, que va muy acorde con la metodología de desarrollo escogida. La programación orientada a objetos encapsula la información procesada brindando mayor seguridad a los datos.

Se logró poner en funcionamiento el sistema Informático de manera local. Este sistema está diseñado para funcionar una intranet o una red de área local. El sistema de notificaciones funciona eficientemente como se lo propuso.

Aunque se ha utilizado en su mayor parte software open source para la construcción de este prototipo, se debe pagar el servicio de mensajería SMS a una determinada operadora; para la realización de las pruebas se utilizó una tarjeta SIM de la red móvil de CLARO.

## **RECOMENDACIONES**

El servidor que se usará para alojar y ejecutar el sistema no debe utilizarse para otros fines, y además no debe ser utilizado por personas no autorizadas a la escuela. Además el servidor debe estar en línea al menos durante la jornada académica de la escuela.

En el caso del cliente, se debe tener cuidado de no entregar su contraseña a otra persona que se encuentre en la institución educativa, para mantener la integridad de la información que se manipula en el sistema.

El docente y representante son el eje fundamental en la formación académica del estudiante de primaria, desde luego ambos deben estar constantemente comunicados. Se recomienda seguir mejorando este prototipo para garantizar la seguridad y mejorar la educación de los niños.



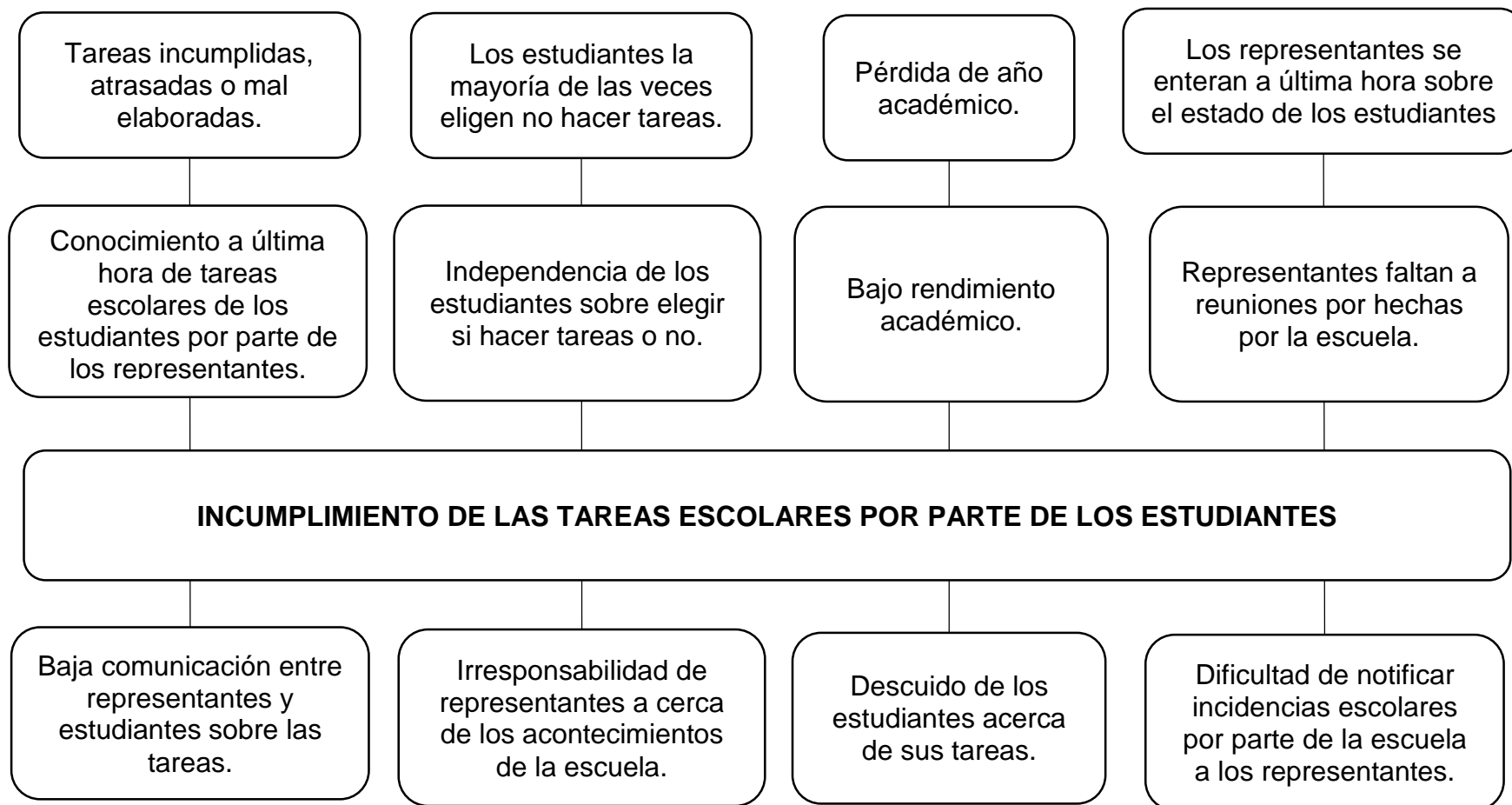
## BIBLIOGRAFÍA

- Alibi, M., & Roy, B. (2016). *Mastering CentOS 7 Linux Server*. Birmingham: Packt Publishing Ltd.
- Arias, Á. (2014). *Aprende a Programar con Java*. Vigo: IT Campus Academy.
- Bodic, G. L. (2005). *Mobile Messaging Technologies and Services: SMS, EMS and MMS*. Hoboken: John Wiley & Sons.
- Bruegge, B., & Dutoit, A. H. (2002). *Ingeniería de software orientado a objetos*. México: Pearson Educación.
- Cánepa, G. (2016). *CentOS High Performance*. Birmingham: Packt Publishing Ltd.
- Castro, J. R. (2015). *Building a Home Security System with Arduino*. Birmingham: Packt Publishing Ltd.
- DeBolt, V. (2007). *Mastering Integrated HTML and CSS*. Hoboken: Wiley Publishing.
- Hillebrand, F. (2010 ). *Short Message Service (SMS): The Creation of Personal Global Text Messaging*. Hoboken: John Wiley & Sons.
- Hobbs, L. (1999). *Diseñar su propia página web*. Barcelona: Marcombo S.A.
- Jamsa, K. (2013 ). *Introduction to Web Development Using HTML 5*. Burlington: Jones & Bartlett Publishers.
- Khare, T. (2012). *Apache Tomcat 7 Essentials*. Birmingham: Packt Publishing Ltd.
- Krishna Hari K, K., N, R., & Salim BOUHLEL, M. (2013). *Proceedings of International Conference on Human Machine Interaction 2013*. Chennai: Techno Forum R&d Centre.
- Kulkarni, U. G. (2017). *Arduino: A Begineer's Guide*.
- Larman, C. (2004). *UML y Patronos 2da Edición*. Madrid: PRENTICE HALL.
- Ley Orgánica de Educación Intercultural. (25 de Agosto de 2015). *Registro Oficial No. 572*. Quito, Ecuador.
- López Gutiérrez, A. (2017). *Técnicos Especialistas en Radiodiagnóstico*. Madrid: EDITORIAL CEP S.L.
- Lyons, D. K. (2013). *Instant PostgreSQL Starter*. Birmingham: Packt Publishing Ltd.
- McFarland, D. S. (2015). *CSS: The Missing Manual*. Sebastopol: O'Reilly Media, Inc.
- Membrey, P., Verhoeven, T., & Angenendt , R. (2009). *The Definitive Guide to CentOS*. New York: Springer.
- MINISTERIO DE EDUCACIÓN. (2017). *PORTAL EDUCAR EDUADOR*. Obtenido de <https://www.educarecuador.gob.ec/index.php/home/que-servicios-ofrece>
- Mueller, J. P. (2015 ). *Security for Web Developers: Using JavaScript, HTML, and CSS*. Sebastopol: O'Reilly Media, Inc.

- Newton, D. (2009). *Apache Struts 2 Web Application Development*. Birmingham: Packt Publishing Ltd.
- Obe, R., & Hsu, L. (2012). *PostgreSQL: Up and Running*. Sebastopol: O'Reilly Media, Inc.
- Padmanabhan, P. (2018). *Java EE 8 and Angular*. Birmingham: Packt Publishing Ltd.
- Paxton, J., Resig, J., & Ferguson, R. (2015). *Pro JavaScript Techniques*. New York: Apress.
- Pitt, C. (2012). *Pro PHP MVC*. New York: Apress L. P.
- Raible, M., Li, S., Johnson, D., Jepp, D., Dalton, S., & Brown, C. (2008). *Pro JSP*. New York: Apress.
- Ramos Martín, A., & Ramos Martín, M. J. (2014 ). *Aplicaciones Web*. Madrid: Ediciones Paraninfo, S.A.
- Román Zamitiz, C. A. (2007). *Páginas Personales de Profesores* .
- Román Zamitiz, C. A. (2017). *Páginas Personales de Profesores*. Obtenido de [http://profesores.fi-b.unam.mx/carlos/aydoo/conceptos\\_oo.html](http://profesores.fi-b.unam.mx/carlos/aydoo/conceptos_oo.html)
- Roughley, I. (2008 ). *Practical Apache Struts 2 Web 2.0 Projects*. New York: Apress.
- Ruiz, P. (13 de Agosto de 2013). <http://somebooks.es/>. Obtenido de <http://somebooks.es/ventajas-e-inconvenientes-de-la-arquitectura-clienteservidor/>
- Sznajdleder, P. A. (2013). *Java a fondo*. Buenos Aires: Alfaomega Grupo Editor Argentino S.A.

# ANEXOS

## ÁRBOL DE PROBLEMAS



## Urkund Analysis Result

**Analysed Document:** Rosado\_Anzules\_José\_Manuel\_Ingeniería\_en\_Sistemas\_2018.docx  
(D36517308)  
**Submitted:** 3/14/2018 3:11:00 PM  
**Submitted By:** josama.14@gmail.com  
**Significance:** 1 %

### Sources included in the report:

Tesis de Grado.pdf (D26679519)  
propuesta tecnolog inescuvi para urkund.docx (D24219981)

### Instances where selected sources appear:

2



Jose Rosado <josama.14@gmail.com>

---

## APROBACION

1 mensaje

---

**Unidad de Titulacion FAFI** <unidaddetitulacionfafi@utb.edu.ec>

14 de marzo de 2018, 12:41

Para: Jose Rosado <josama.14@gmail.com>

Aprobado, 1% felicitaciones

--

CPA. Julio Mora Aristega, MAE

COORDINADOR DE LA UNIDAD DE TITULACIÓN FAFI - UTB



**Urkund Report - Rosado\_Anzules\_José\_Manuel\_Ingeniería\_en\_Sistemas\_2018.docx**

**(D36517308).pdf**

159K