



ESCUELA SUPERIOR POLITECNICA DE CHIMBORAZO

FACULTAD DE INFORMATICA Y ELECTRÓNICA

ESCUELA DE INGENIERIA EN SISTEMAS

**“ESTUDIO COMPARATIVO DE LAS HERRAMIENTAS DE SERVICIO DE
NOTIFICACIONES DE EVENTOS VÍA MENSAJERÍA MÓVIL CASO PRÁCTICO:
SISTEMA ACADÉMICO ESPOCH”**

TESIS DE GRADO

Previa a la obtención del título de

INGENIERO EN SISTEMAS INFORMÁTICOS

Presentado por:

Hugo Javier Polo Saltos

Francisco Javier Gallegos Pillajo

RIOBAMBA – ECUADOR

2009

AGRADECIMIENTOS

De corazón agradecemos infinitamente a Dios por bendecirnos y nunca desampararnos en los momentos difíciles.

A la ESPOCH por permitirnos crecer intelectual y profesionalmente.

De una manera muy especial al Ing. Iván Menes, como director de tesis, de igual manera al Ing. Danilo Pastor como asesor de la misma; por su apoyo incondicional en el desarrollo de nuestra tesis.

A nuestros amigos por compartir cada uno de nuestros sueños y por contar con su valiosa amistad.

A Francisco, Rosario, Margarita y Fabián, mis padres y hermanos queridos, por su gran cariño y comprensión durante toda mi vida. A mi amor Pamela que me enseña a saber como ser mejor.

Francisco Javier Gallegos Pillaño.

A mi esposa e hija por ser el pilar fundamental de mi vida a madre Laurita por haber confiado en mí apoyarme en todos mis sueños junto a mi hermana Jessica y a mi padre Victor Hugo.

Hugo Javier Polo Saltos

DEDICATORIA

Dedico esta tesis a mis padres ya que ellos con sacrificio y amor incondicional me han dado el apoyo para luchar por mis sueños y me han sabido guiar por el camino de la fe y la esperanza como sólo ellos lo pueden reflejar en su diario vivir.

Francisco Javier Gallegos Pillajo

Dedico esta tesis al niño rey de reyes por haber dado la salud para concluir con éxito esta carrera, a mis padres que nunca me han hecha faltar sus bendiciones y sus consejos, a mi hermanita, esposa e hija por ser mi inspiración en mi vida para salir siempre adelante y darme las fuerzas para luchar.

Hugo Javier Polo Saltos

FIRMAS DE RESPONSABILIDAD

| NOMBRE | FIRMA | FECHA |
|--|-------|-------|
| Dr. Ms.C. Romeo Rodríguez DECANO DE LA FACULTAD DE INFORMÁTICA Y ELECTRÓNICA | | |
| Ing. Iván Menes DIRECTOR DE LA ESCUELA DE INGENIERÍA EN SISTEMAS | | |
| Ing. Iván Menes DIRECTOR DE TESIS | | |
| Ing. Danilo Pastor MIEMBRO DEL TRIBUNAL | | |
| Ing. Carlos Rodríguez DIRECTOR DEL CENTRO DE DOCUMENTACIÓN | | |
| NOTA DE LA TESIS | | |

“Nosotros, Francisco Javier Gallegos Pillajo y Hugo Javier Polo Saltos, somos responsables de las ideas, doctrinas y resultados expuestos en esta tesis; y, el patrimonio intelectual de la Tesis de Grado pertenece a la ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO.”

Francisco Javier Gallegos Pillajo

Hugo Javier Polo Saltos

ÍNDICE DE ABREVIATURAS

| ABREVIATURAS | SIGNIFICADO |
|--------------|---|
| API | Programa de interfaz de aplicaciones (Application Program Interface) |
| ESPOCH | Escuela Superior Politécnica de Chimborazo |
| HTTP | Protocolo de transferencia HiperTexto (HyperText Transfer Protocol) |
| IIS | Servicio de información internet (Internet Information Services) |
| IMAP | Protocolo de acceso de mensajes internet (Internet Message Access Protocol) |
| IP | Dirección de protocolo de Internet (Internet Protocol) |
| MIME | Extensiones de Correo Internet Multipropósito (Multipurpose Internet Mail Extensions) |
| OASIS | Sistema académico |
| PDA | Asistente digital personal (Personal Digital Assitant) |
| POP3 | Protocolo 3 de Correo (Post Office Protocol 3) |
| SMS | Servicio de mensajes cortos (Short Message Service) |
| SMTP | Protocolo de Transferencia Simple de Correo (Simple Mail Transfer Protocol) |
| SOAP | protocolo simple de acceso a objetos (Simple Object Acces Protocol) |
| SQL | Lenguaje estructurado de consulta (Structured Query Language) |
| TCP | Protocolo de control de transmisión (Transmision Control |

| | |
|------|---|
| | Protocol) |
| URL | Localizador Uniforme de Recursos (Uniform Resource Locator) |
| WSDL | Lenguaje de Descripción de Servicios Web (Web Services Description Language) |
| XML | Lenguaje de marca extensible (Extensible Markup Language) |
| XSLT | Lenguaje de estilos extendido transformado (Extensible Style Language Transformaciones) |

ÍNDICE

CAPÍTULO I

| | |
|--|----|
| MARCO DE REFERENCIA..... | 44 |
| 1.1. Título de la Investigación | 44 |
| 1.2. Problema de la Investigación..... | 45 |
| 1.2.1. Análisis | 45 |
| 1.2.2. Limitación | 45 |
| 1.3. Objetivos..... | 45 |
| 1.3.1. Objetivo General..... | 45 |
| 1.3.2. Objetivos Específicos | 45 |
| 1.4. Justificación de la Investigación | 46 |
| 1.4.1. Justificación Teórica | 46 |
| 1.4.2. Justificación Práctica..... | 46 |
| 1.5. Hipótesis | 47 |

CAPÍTULO II

| | |
|--------------------|----|
| MARCO TEORICO..... | 48 |
|--------------------|----|

| | |
|--|----|
| 2.1. Conceptos Generales de la Tecnología de Notificaciones..... | 48 |
| 2.1.1. Tecnología móvil..... | 49 |
| 2.1.2. Mensaje..... | 49 |
| 2.1.3. Notificación..... | 49 |
| 2.1.4. Sistema..... | 49 |
| 2.1.5. Aplicación..... | 50 |
| 2.1.6. Base de datos..... | 50 |
| 2.1.7. Evento..... | 50 |
| 2.1.8. Usuario del sistema..... | 50 |
| 2.1.9. Escenarios de uso..... | 51 |
| 2.1.10. Arquitectura..... | 51 |
| 2.1.11. Implementación..... | 51 |
| 2.1.12. Plataforma..... | 51 |
| 2.2. Introducción a los Servicios de Notificación..... | 51 |
| 2.3. Escenarios de uso de la tecnología de notificación..... | 52 |
| 2.3.1. Aplicaciones de consumidor..... | 52 |
| 2.3.2. Aplicaciones operativas..... | 53 |

| | |
|--|----|
| 2.3.3. Aplicaciones de Business Intelligence | 53 |
| 2.3.4. Comunicación con los empleados | 53 |
| 2.4. Arquitectura de la tecnología de notificación..... | 53 |
| 2.4.1. Datos del usuario | 54 |
| 2.4.2. Base de datos..... | 54 |
| 2.4.3. Manejador de notificaciones..... | 55 |
| 2.4.4. Notificación | 55 |
| 2.4.5. Dispositivos | 55 |
| CAPÍTULO III | |
| ESTUDIO COMPARATIVO DEL SERVICIO DE NOTIFICACIONES DE SQL SERVER Y ORACLE..... | |
| 3.1. Selección de Parámetros de Comparación..... | 56 |
| 3.1.1. Seguridad..... | 57 |
| 3.1.2. Confiabilidad..... | 57 |
| 3.1.3. Escalabilidad..... | 58 |
| 3.1.4. Compatibilidad..... | 59 |
| 3.1.5. Disponibilidad..... | 59 |
| 3.1.6. Eficiencia..... | 60 |

| | |
|--|----|
| 3.2. Tabla de valores para la comparación..... | 60 |
| 3.3. Ámbito de Estudio..... | 61 |
| 3.4. Descripción de las Herramientas..... | 61 |
| 3.4.1. Estudio del Servicio de Notificaciones de SQL Server..... | 62 |
| 3.4.1.1. Introducción a notification services | 62 |
| 3.4.1.2. Que es Notification Services..... | 64 |
| 3.4.1.2.1. ¿En qué consiste Notification Services?..... | 64 |
| 3.4.1.2.2. ¿Cómo funciona Notification Services?..... | 66 |
| 3.4.1.3. Arquitectura de Notification Services | 67 |
| 3.4.1.3.1. Arquitectura de administración de suscripciones..... | 68 |
| 3.4.1.3.1.1. Interfaces de administración de suscripciones | 68 |
| 3.4.1.3.1.2. Vistas de administración de suscripciones | 69 |
| 3.4.1.3.2. Arquitectura de recopilación de eventos | 69 |
| 3.4.1.3.2.1. Vistas de clase de evento..... | 71 |
| 3.4.1.3.2.2. Proveedores de eventos estándar y personalizados..... | 71 |
| 3.4.1.3.2.3. Proveedores de eventos alojados y no alojados | 71 |
| 3.4.1.3.3. Arquitectura de procesamiento de suscripciones | 72 |

| | |
|---|-----|
| 3.4.1.4.1. Seguridad | 76 |
| 3.4.1.4.2. Confiabilidad | 77 |
| 3.4.1.4.3. Escalabilidad..... | 78 |
| 3.4.1.4.4 .Disponibilidad..... | 79 |
| 3.4.2. Estudio del Servicio de Notificaciones de Oracle | 80 |
| 3.4.2.1. Introducción a push services | 80 |
| 3.4.2.2. Que es Push Services | 81 |
| 3.4.2.3. Arquitectura de Push Services..... | 86 |
| 3.5. Análisis Comparativo..... | 97 |
| 3.5.1. Seguridad..... | 98 |
| 3.5.2. Confiabilidad..... | 98 |
| 3.5.3. Escalabilidad..... | 99 |
| 3.5.4. Compatibilidad..... | 99 |
| 3.5.5. Disponibilidad..... | 100 |
| 3.5.6. Eficiencia..... | 100 |
| 3.6. Tabla General Comparativa | 100 |
| 3.7. Gráficos de pruebas de stress para el estudio comparativo..... | 102 |

| | |
|--|-----|
| 3.8. Resultados de la Comparación..... | 103 |
|--|-----|

CAPÍTULO IV

| | |
|---|-----|
| ESTUDIO DE LA INTERACCIÓN CON EL SISTEMA ACADÉMICO DE LA ESPOCH . | 104 |
|---|-----|

| | |
|--|-----|
| 4.1. Información contenida en la base de datos | 104 |
|--|-----|

| | |
|--|-----|
| 4.1.1. Definición del pensum de estudios | 104 |
|--|-----|

| | |
|---|-----|
| 4.1.2. Definición del periodo académico | 105 |
|---|-----|

| | |
|---------------------------------------|-----|
| 4.1.3. Proceso de matriculación | 105 |
|---------------------------------------|-----|

| | |
|------------------------------------|-----|
| 4.1.4. Proceso de evaluación | 105 |
|------------------------------------|-----|

| | |
|--|-----|
| 4.1.5. Proceso de egresamiento y graduación..... | 105 |
|--|-----|

| | |
|--|-----|
| 4.1.6. Administración y seguridad..... | 106 |
|--|-----|

| | |
|-----------------------------------|-----|
| 4.1.7. Estudiantes inscritos..... | 106 |
|-----------------------------------|-----|

| | |
|-------------------------|-----|
| 4.1.8. Estudiantes..... | 106 |
|-------------------------|-----|

| | |
|----------------------|-----|
| 4.1.9. Docentes..... | 106 |
|----------------------|-----|

| | |
|---------------------------|-----|
| 4.1.10. Institución | 106 |
|---------------------------|-----|

| | |
|--------------------------|-----|
| 4.1.11. Facultades | 107 |
|--------------------------|-----|

| | |
|-----------------------|-----|
| 4.1.12. Escuelas..... | 107 |
|-----------------------|-----|

| | |
|------------------------|-----|
| 4.1.13. Carreras | 107 |
|------------------------|-----|

| | |
|---|-----|
| 4.1.14. Materias | 107 |
| 4.1.15. Horario | 107 |
| 4.1.16. Niveles | 107 |
| 4.2. Usuarios que interactúan en el Sistema Académico | 108 |
| 4.2.1. Clientes Web..... | 108 |
| 4.2.1.1. Público General | 108 |
| 4.2.1.2. Estudiante | 108 |
| 4.2.1.3. Docente..... | 108 |
| 4.2.1.4. Autoridad..... | 108 |
| 4.2.2. Interfaces en Win32 | 109 |
| 4.2.2.1. Secretaría Académica General y Secretaría de Unidades Académicas | 109 |
| 4.3. Funcionalidad del Sistema Académico. | 109 |
| 4.3.1. OASis_BusinessServices | 109 |
| 4.3.2. Servicios Módulo de Inscripciones..... | 109 |
| 4.3.3. Servicios Módulo de matriculación | 109 |
| 4.3.4. Servicios Módulo de Educación y Evaluación | 110 |
| 4.3.5. Servicios de Egresamiento y Graduación | 111 |

| | |
|---|-----|
| 4.4. Selección de prestaciones del Sistema Académico para realizar notificaciones. | 112 |
| 4.4.1. Almacenamiento de notas de todos los estudiantes | 112 |
| 4.4.2. Cambio de notas en un período determinado | 113 |
| 4.4.3. Inscripciones y confirmación de inscripciones | 113 |
| 4.4.4. Matriculación y confirmación de matriculación | 113 |

CAPÍTULO V

| | |
|--|-----|
| DESARROLLO E IMPLEMENTACIÓN DEL SISTEMA DE NOTIFICACIONES | 114 |
| 5.1. Fase de Análisis | 114 |
| 5.1.1. Definir, refinar y diagramar los casos de uso esenciales en formato expandido | 114 |
| 5.1.2. Definir y refinar los diagramas de secuencia..... | 116 |
| 5.1.2.1. Operatividad del Servicio de Notificación del Sistema Académico. | 116 |
| 5.1.3. Definir y refinar los diagramas de estado | 116 |
| 5.2. Fase de Diseño | 117 |
| 5.2.1. Refinar los Casos de Uso | 117 |
| 5.2.1.1. Operatividad del Sistema de Notificación de Académicas..... | 117 |
| 5.2.2. Diseño de Base de Datos..... | 117 |
| 5.2.3. Modelado Físico | 118 |

5.2.4. Diagrama de Componentes.....118

5.3. FASE DE IMPLEMENTACIÓN.....118

5.3.1. Implementación.....118

CONCLUSIONES

RECOMENDACIONES

RESUMEN

SUMMARY

BIBLIOGRAFÍA

ANEXOS

ÍNDICE DE TABLAS

| | |
|---|----|
| Tabla III.1. . Tabla de valores para la comparación | 39 |
| Tabla III.2. Tabla de seguridad | 77 |
| Tabla III.3. . Tabla de confiabilidad | 77 |
| Tabla III.4. Tabla de escalabilidad | 78 |
| Tabla III.5. Tabla de compatibilidad | 79 |
| Tabla III.6. Tabla de disponibilidad | 79 |
| Tabla III.7. Tabla de eficiencia..... | 79 |
| Tabla III.8. Tabla general comparativa | 80 |

ÍNDICE DE FIGURAS

| | |
|--|----|
| Figura II.1. Arquitectura de la Tecnología de Notificación | 33 |
| Figura III.1. - Servicio de Notificaciones de SQL Server 2005 | 41 |
| Figura III.2. Funcionamiento básico de una aplicación Notification Services | 44 |
| Figura III.3. Arquitectura de Notification Services | 47 |
| Figura III.4. Interfaces de administración de suscripciones | 48 |
| Figura III.5. Arquitectura de recopilación de eventos | 49 |
| Figura III.6. Arquitectura de procesamiento de suscripciones | 52 |
| Figura III.7. Arquitectura de entrega y formato de notificaciones | 54 |
| Figura III.8. Servicio de Notificaciones de Oracle9iAS Wireless (Push Service) | 59 |
| Figura III.9. Servidor de Controladores de Mensajes | 61 |
| Figura III.10. Arquitectura del Sistema de Mensajería | 66 |
| Figura III.11. Pruebas de Stress de Push Services | 81 |
| Figura III.12. Pruebas de Stress de Notification Services | 82 |
| Figura IV.1. Módulos de OASIS | 91 |
| Figura V.1. Diagrama de Secuencia del Servicio de Notificaciones | 95 |
| Figura V.2. Diagrama de Estado del Servicio de Notificaciones | 95 |
| Figura V.3. Diagrama de Caso de Uso del Servicio de Notificaciones | 96 |
| Figura V.4. Modelo Relacional de la Base de Datos | 96 |
| Figura V.5. Modelo Físico del Servicio de Notificación | 97 |
| Figura V.6. Diagrama de Componentes del Servicio de Notificación | 97 |

INTRODUCCIÓN

En la actualidad la necesidad de encontrar medios para la comunicación y maneras de informar a diversos tipos de dispositivos que sirven de apoyo para las tareas de información y comunicación, ha llevado al desarrollo de la tecnología de notificación, que en el mundo cotidiano se presenta sino en todos en la mayoría de ámbitos que requieran de información y comunicación. Pero, ¿a que tipo de comunicación nos referimos? Pues a la comunicación remota, a diferentes sitios del planeta con tan solo un mensaje de texto al celular, mail, palm, etc.

La generación de confirmaciones o mensajes de notificación, resulta una tarea muy sencilla debido a herramientas que desarrollan este tipo de servicios, así como sencillo es para los clientes que reciben estos mensajes de una forma entendible y dinámica.

La tecnología de notificación nos sirve para la generación y envío de información a través de una confirmación o una modificación de datos en la base de datos la cual tiene que ser informada a los clientes de dicho servicio.

La siguiente tesis esta organizada a un estudio comparativo entre las herramientas para poder enviar mensajes móviles cuando se haga modificaciones en la base de datos es por eso que se a estudiado la arquitectura, sus ventajas, sus desventajas y todo cuanto sea necesario para obtener un buen resultado, para luego realizar un caso práctico en la Escuela Superior Politécnica de Chimborazo tomando en cuenta esta nueva tecnología para el desarrollo de notificación automática con respecto al sistema académico.

Para la obtención de los resultados realizaremos un estudio cualitativo de estas herramientas realizando un cuadro comparativo que nos ayudara a demostrar cual es la mejor opción para el desarrollo del caso practico de esta tesis.

ORGANIZACIÓN DEL DOCUMENTO

El Capítulo I presenta el marco de referencia en el que se detalla el título, el problema, los objetivos, la justificación y la hipótesis de la investigación.

El Capítulo II contiene el marco teórico en el cual se realiza una presentación de los conceptos generales, introducción, escenarios de uso, la arquitectura de la tecnología de notificación, un estudio del servicio de notificaciones de SQL Server, y el estudio del push servicios de Oracle.

El Capítulo III comprende el estudio comparativo del servicio de notificación de SQL Server y de Push Servicios de oracle, obtención de resultados y la toma de decisión de la herramienta para la implementación del caso práctico de la tesis.

El Capítulo IV se enfoca en estudio del sistema académico de la ESPOCH su información que genera, los usuarios que interactúan, su funcionalidad, y sus prestaciones que este tiene para luego de este estudio seleccionar cuales prestaciones se pueden implementar en nuestro sistema.

El Capítulo V Aquí se realiza la implementación del caso practico de la tesis es decir del sistema de notificación para el sistema académico tomando en cuenta todo el estudio anterior.

Por último se presentan las conclusiones y recomendaciones obtenidas a partir de la presente investigación.

CAPÍTULO I

MARCO DE REFERENCIA

1.1. Título de la Investigación

Estudio comparativo de las herramientas de servicio de notificaciones de eventos vía mensajería móvil. Caso práctico: Sistema Académico ESPOCH

1.2. Problema de la Investigación

En la actualidad la utilización de nuevas tecnologías para recibir la información se ha visto tan necesaria para el ahorro de tiempo que es de vital importancia para todo el mundo, por lo cual dichas tecnologías han creado nuevos métodos, servicios que facilitan la entrega de la información requerida; una de estas es por medio del Notification Services.

1.2.1. Análisis

Dentro del análisis del Servicio de Notificación realizaremos un estudio comparativo del mismo, dando a conocer su arquitectura, utilización, ventajas y la explotación que se le dará mediante un sistema software para la ESPOCH teniendo como base el Sistema Académico.

1.2.2. Limitación

La limitación es el registro de los datos necesarios de los estudiantes y los docentes de la ESPOCH para poder realizar la notificación de eventos en cuanto se presente alguna situación con respecto al Sistema Académico.

1.3. Objetivos

1.3.1. Objetivo General

Realizar el estudio comparativo de las herramientas que prestan el servicio de notificaciones vía mensajería móvil de SQL Server y Oracle e implementar en el Sistema Académico de la ESPOCH.

1.3.2. Objetivos Específicos

- Estudiar la tecnología de las notificaciones vía mensajería móvil.
- Realizar el estudio comparativo de las herramientas de servicio de notificaciones en SQL Server y Oracle.

- Establecer la utilización de la mejor herramienta que permita realizar la notificación de eventos vía mensajería móvil dentro de la Escuela Superior Politécnica De Chimborazo.
- Estudiar el sistema existente para la interacción con el nuevo sistema para la obtención de información y realizar la notificación de eventos de los estudiantes y docentes de la Escuela Superior Politécnica de Chimborazo.
- Desarrollar el sistema de Notificaciones de acuerdo a la herramienta más adecuada y a un método definido para la notificación de eventos a los estudiantes y docentes de la Escuela Superior Politécnica de Chimborazo.

1.4. Justificación de la Investigación

1.4.1. Justificación Teórica

Con el avance de la tecnología es necesario la creación de una nueva forma de poder notificar información primordial a los usuarios, es por eso que se ha visto la importancia de la creación de esta tesis que ayudaría a los usuarios del Sistema Académico estar al tanto de todos los eventos que se puedan suscitar durante un determinado período académico, todo esto teniendo como punto principal la base de datos del Sistema Académico.

Una de las ventajas importantes sería la información actualizada que se les impartiría a los estudiantes y profesores al momento de que una situación se presente tal es el caso de notas, exámenes, horarios, etc.

1.4.2. Justificación Práctica

Con el siguiente tema de tesis se va a resolver:

Estudiantes:

- Obtención de las notas finales de los estudiantes cuando sean ingresadas al Sistema Académico.

- Confirmación de cambio de notas.
- Confirmación de inscripción.
- Confirmación de matriculación.
- Confirmación de cambio de escuela o convalidación de materias correspondientes a un período académico.
- Información de retiro de una materia o carrera.
- Información cuando el estudiante ya esté listo para incorporarse.
- Información de horarios de clases y exámenes.
- Cambio de contraseña.

Docentes:

- Información de carga horaria y materias a dictar.
- Información cuando el docente no ingrese al Sistema Académico las notas.
- Información de la fecha de cierre de período académico.
- Cambio de contraseña.

Una de las restricciones que tendríamos es que al inicio los estudiantes y docentes deben conocer el servicio que se va a crear y que estos tienen que registrarse, el cual al momento de realizarlo se le confirmará con un mensaje de fallido o creado correctamente el nuevo usuario en la base de datos para obtener la información requerida para aplicar el envío de su respectiva información por medio de mensajería móvil.

1.5. Hipótesis

Con la implementación de la tecnología de notificación en el Sistema Académico de la ESPOCH, se mejorará la interacción de los estudiantes y docentes con el Sistema Académico.

CAPÍTULO II

MARCO TEORICO

2.1. Conceptos Generales de la Tecnología de Notificaciones

Para el mejor entendimiento del desarrollo de esta tesis se ha visto necesario el detalle de los principales conceptos de la tecnología de notificación, sus características y funcionamiento, revisemos los conceptos básicos que a esta envuelve.

2.1.1. Tecnología móvil

La telefonía móvil, también llamada telefonía celular, básicamente está formada por dos grandes partes: una red de comunicaciones (o red de telefonía móvil) y los terminales (o teléfonos móviles) que permiten el acceso a dicha red.

La red de telefonía móvil consiste en un sistema telefónico en el que mediante la combinación de una red de estaciones transmisoras-receptoras de radio (repetidores ó también llamados estaciones base) y una serie de centrales telefónicas de conmutación, se posibilita la comunicación entre terminales telefónicos portátiles (teléfonos móviles) o entre terminales portátiles y teléfonos de la red fija tradicional.

2.1.2. Mensaje

Está definido como la información que el emisor envía al receptor a través de un canal determinado o medio de comunicación (como el habla, la escritura, etc.); aunque el término también se aplica, dependiendo del contexto, a la presentación de dicha información; es decir, a los símbolos utilizados para transmitir el mensaje. Cualquiera que sea el caso, el mensaje es una parte fundamental en el proceso de intercambio de información.

2.1.3. Notificación

Es confirmar la información emitida por el trasmisor hacia el receptor de la información transmitida por medio de un canal y al ser entregada.

2.1.4. Sistema

Un sistema es un conjunto de elementos interrelacionados e interactuantes entre sí. El concepto tiene dos usos muy diferenciados, que se refieren respectivamente a los sistemas de conceptos y a los objetos reales más o menos complejos y dotados de organización.

2.1.5. Aplicación

Nombre que reciben los programas especializados en tareas concretas y de una cierta complejidad. En el mundo de los micros, los procesadores de textos y los sistemas de gestión de bases de datos son ejemplos de aplicaciones. El concepto de aplicación se opone al de software de base, por ejemplo, compuesto por sistemas operativos y lenguajes de programación.

Se define el término aplicación como una funcionalidad que tiene que ser desarrollada de acuerdo a las necesidades del usuario final y al tipo de notificación que se desee notificar.

2.1.6. Base de datos

Conjunto de datos organizados entre los cuales existe una correlación y que están almacenados con criterios independientes de los programas que los utilizan. La filosofía de las bases de datos es la de almacenar grandes cantidades de datos de una manera no redundante y que permita las posibles consultas de acuerdo a los derechos de acceso.

Es una colección de datos estructurada y organizada, para permitir el rápido acceso a la información de interés. Los elementos que la forman se denominan registros, los cuales, a su vez, están compuestos por campos.

2.1.7. Evento

Acontecimiento suceso imprevisto o contingente para emitir o enviar información referente al mismo. Cuando todas las actividades que llegan a un mismo nodo han sido terminadas.

2.1.8. Usuario del sistema

Destinatario que tiene el sistema al momento de generar una notificación con respecto a un evento sucedido que involucra al destinatario como tal para compartir información.

2.1.9. Escenarios de uso

Son todos y cada uno de los lugares, situaciones y ambientes en que una o varias aplicaciones informáticas se desarrollan y funcionan para el beneficio de una o grupo de personas.

2.1.10. Arquitectura

La arquitectura es el estudio de un sistema donde se determinan y se especifican las partes del mismo, las conexiones entre ellos, y las normas de interacción entre las partes del sistema haciendo uso de las conexiones especificadas.

2.1.11. Implementación

La implementación es la realización de las especificaciones que se necesita para proporciona toda la información para construir un sistema y ponerlo en funcionamiento.

2.1.12. Plataforma

Una plataforma es un conjunto de subsistemas y tecnologías que aportan un conjunto coherente de funcionalidades a través de interfaces y determinados patrones de uso, que cualquier aplicación que se construya para esa plataforma puede usar sin preocuparse por los detalles de la implementación o como se lleva a cabo la misma dentro de la plataforma.

2.2. Introducción a los Servicios de Notificación

Esta tecnología nace a raíz de la idea de notificar situaciones o acontecimientos que se produzcan en un repositorio de datos para lo cual las empresas desarrolladoras de manejadores de base de datos desarrollaron complementos que efectivicen dichas notificaciones.

El servicio de notificación es una tecnología para generar y enviar información a los usuarios de un sistema. Las notificaciones generadas son mensajes

personalizados y oportunos que pueden enviarse a una amplia gama de dispositivos y que reflejan las preferencias del suscriptor.

Las notificaciones pueden enviarse a una gran variedad de dispositivos. Por ejemplo, una notificación puede enviarse al teléfono móvil, a su asistente digital personal (PDA), o a su cuenta de correo electrónico.

Los clientes para obtener estos datos se puede generar a un solo cliente o a varios clientes dependiendo de las necesidades que se desee generar, así mejorando y fortaleciendo las relaciones con los clientes, ya que permiten ofrecerles servicios más personalizados en el momento oportuno con lo cual ayudando a incrementar los ingresos y la productividad, ya que facilitan el establecimiento de relaciones comerciales con los clientes.

2.3. Escenarios de uso de la tecnología de notificación

La tecnología de Notificación puede utilizarse en una gran variedad de aplicaciones y maneras por lo cual citaremos algunas de ellas.

2.3.1. Aplicaciones de consumidor

Se puede enviar notificaciones a los clientes:

- Una agencia de corredores de bolsa puede enviar precios de valores bursátiles y de fondos según los criterios definidos por el cliente. La notificación puede contener un vínculo que permita al cliente comprar o vender valores.
- Una institución financiera puede enviar notificaciones sobre balances bajos o pagos atrasados al dispositivo que elija el suscriptor, como el correo electrónico.
- Una compañía aérea puede enviar información sobre las llegadas de vuelos y los retrasos en las salidas.
- Un agente de la propiedad inmobiliaria puede enviar listados de casas que cumplan los criterios de un cliente.

2.3.2. Aplicaciones operativas

Se puede enviar notificaciones acerca de las operaciones de la empresa:

- Cuando una máquina de una línea de montaje necesita atención, el responsable de la línea de montaje recibe una notificación sobre la máquina que está inoperativa y el error de la misma.
- Cuando un registro de eventos del servidor de producción recibe un error importante, el técnico de soporte recibe una notificación que contiene el mensaje de error.

2.3.3. Aplicaciones de Business Intelligence

Se puede realizar un seguimiento de la información fundamental para la empresa:

- Una base de datos mantiene el inventario actual de todos los productos de una empresa. Cuando el inventario de tarjetas de red para la región del Sureste cae por debajo de las 50 unidades, la aplicación recibe una notificación para informarle de que envíe un pedido de tarjetas de red.

2.3.4. Comunicación con los empleados

Permite informar a los empleados acerca de sus proyectos:

- Usted y un colega colaboran en una propuesta importante. Cuando el colega actualice la propuesta, usted recibirá una notificación.
- Trabaja en un nuevo producto y necesita mantenerse informado sobre la investigación relacionada con el producto. Cuando se dispone de un nuevo informe en el archivo de la empresa, recibe un mensaje de correo electrónico que contiene un vínculo al informe.

2.4. Arquitectura de la tecnología de notificación

La arquitectura básica de un sistema de notificaciones se define con cinco elementos principales como se muestra en la Figura 1:



Figura II.1. Arquitectura de la Tecnología de Notificación

2.4.1. Datos del usuario

La información primordial que posee un sistema de notificación se resume en los datos del usuario final el cual recibe todos y cada uno de los avisos generados por el sistema, estos datos dependiendo del dispositivo a entregar pueden ser:

- Número de teléfono celular
- Dirección de correo electrónico
- Direcciones IP
- Nombres de Dominio

2.4.2. Base de datos

El sitio donde se almacena la información principal para que se generen los eventos y así los usuarios reciban la información necesaria o generada.

Los eventos que se van a generar aquí en la base de datos son:

- Inserciones
- Modificaciones
- Eliminaciones

Es decir aquí se generaran estos eventos y estos producirán las notificaciones de información a los usuarios.

2.4.3. Manejador de notificaciones

Organiza las acciones a realizarse después de detectar un evento. Del evento detectado busca la relación con los usuarios existentes, una vez hecho esto analiza la información de dichos usuarios, arma el mensaje o notificación a enviarse y ejecuta el envío de acuerdo al dispositivo registrado por el usuario en el sistema.

2.4.4. Notificación

Es la información o mensaje a ser enviado al usuario, dichos mensaje contendrán cada uno de los parámetros diseñados al momento del desarrollo del sistema de notificación.

2.4.5. Dispositivos

Destino de la información o el mensaje entre los dispositivos a los cuales se los puede enviar la notificación tales como:

- Número de teléfono celular
- Dirección de correo electrónico
- Direcciones IP
- Nombres de Dominio

CAPÍTULO III

ESTUDIO COMPARATIVO DEL SERVICIO DE NOTIFICACIONES DE SQL SERVER Y ORACLE.

3.1. Selección de Parámetros de Comparación.

Para realizar el estudio comparativo de las dos herramientas seleccionadas anteriormente, se ha escogido los siguientes parámetros de comparación que ayudarán a determinar la mejor herramienta de acuerdo a la valoración de los parámetros mencionados.

Los parámetros de comparación son:

3.1.1. Seguridad

Podemos entender como seguridad un estado de cualquier sistema (informático o no) que nos indica que ese sistema está libre de peligro, daño o riesgo. Se entiende como peligro o daño todo aquello que pueda afectar su funcionamiento directo o los resultados que se obtienen del mismo. Para la mayoría de los expertos el concepto de seguridad en la informática es utópico porque no existe un sistema 100% seguro.

Cada herramienta presenta elementos de seguridad que son importantes para el proceso de desarrollo de una aplicación de notificación entre los elementos se encuentran los siguientes:

Cifrado de datos.- la forma en que los datos se encriptan para el envío de mensajes de notificación

Protocolos Web.- los medios en los cuales son enviados cada uno de los mensajes

Creación personalizada de instancias/eventos.- personalización de los métodos de envío

Mecanismos propios del manejador.- la seguridad propia de cada manejador de base de datos según la herramienta

Cuentas de usuario.- los diferentes usuarios que tendrán acceso al sistema y los respectivos permisos de cada uno.

3.1.2. Confiabilidad

Los sistemas informáticos se consideran confiables cuando son predecibles, requieren poco mantenimiento y se ejecutan sin interrupciones para que los usuarios puedan utilizar los recursos que necesitan de manera oportuna. Dicho de otro modo, la confiabilidad es la medida del número de interrupciones críticas durante el tiempo que un programa o sistema está en funcionamiento. Para los

consumidores sin conocimientos técnicos, la confiabilidad significa que un equipo funciona del modo previsto siempre que se desea utilizarlo.

Servidor sólido.- permite registro de escritura anticipada o tiene mecanismos de conmutación por error.

Procesamiento de eventos.- tipo de procesamiento de eventos, suscripciones y notificaciones.

Entrega confiable.- mecanismos de repetición de entrega en el caso de que un error se haya detectado.

Estabilidad del sistema.- generación de proveedores de eventos personalizados y protocolos de entrega confiable.

3.1.3. Escalabilidad

La escalabilidad es la propiedad deseable de un sistema, una red o un proceso, que indica su habilidad para, o bien manejar el crecimiento continuo de trabajo de manera fluida, o bien para estar preparado para hacerse más grande sin perder calidad en los servicios ofrecidos. En general, también se podría definir como la capacidad del sistema informático de cambiar su tamaño o configuración para adaptarse a las circunstancias cambiantes.

Ejecución en paralelo.- la ejecución se puede realizar en paralelo con otras aplicaciones para la notificación.

Aplicación multiproceso.- se puede ampliar verticalmente con equipos que utilicen varios procesadores.

Servidores distintos.- implementación tanto de la base de datos como el servidor de notificaciones en servidores distintos.

Proveedores distribuidos.- distribución de los distintos proveedores de eventos en servidores diferentes.

División de instancias.- división de suscripciones en varias instancias para que funcionen cada una de forma independiente.

Plataformas.- medio en el que se puede ejecutar el sistema o aplicación.

3.1.4. Compatibilidad

La compatibilidad es la condición que hace que un programa y un sistema, arquitectura o aplicación logren comprenderse correctamente tanto directamente o indirectamente (mediante un algoritmo), a este algoritmo que hace que un programa logre ser comprendido por un sistema, arquitectura o aplicación se lo denomina emulador por el hecho de que es un intérprete entre el programa y el sistema, arquitectura o aplicación.

Varios dispositivos.- destinatarios del sistema de notificación que recibirán el mensaje.

Sistemas operativos.- compatibilidad con los sistemas operativos existentes en el medio.

Varios tipos de mensaje.- los distintos mensajes que se podrían enviar para realizar una notificación.

Distintos protocolos de entrega.- todos y cada uno de los mensajes necesita un medio de envío y son los protocolos.

3.1.5. Disponibilidad

Control de errores de hardware.- control automático de hardware en caso de existir algún error en el servidor.

Tecnología de disponibilidad.- medios por los cuales la disponibilidad se presenta en un alto nivel.

Réplica transaccional.- diferentes réplicas de la base de datos para asegurar la disponibilidad.

3.1.6. Eficiencia

Tiempos de respuesta.- tiempo en que el usuario final recibe una notificación desde que se produjo un evento.

Volumen de tráfico.- la cantidad de información que debe ser entregada en un plazo determinado.

Notificaciones familiares.- notificaciones entendibles y personalizables.

3.2. Tabla de valores para la comparación

Para el análisis de comparación de las herramientas de notificación se ha determinado los diferentes parámetros, los cuales serán evaluados con los siguientes valores cuantitativos y cualitativos.

| Nombre de Valor de Ponderación | Equivalencia de Ponderación |
|--------------------------------|-----------------------------|
| Si | 5 |
| No | 0 |
| Alto | 5 |
| Medio | 3 |
| Bajo | 1 |
| Siempre | 5 |
| Casi Siempre | 3 |
| A veces | 2 |
| Nunca | 0 |
| Avanzado | 5 |
| Normal | 3 |
| Simple | 1 |
| Mucho | 5 |
| Poco | 3 |
| Nada | 0 |

Tabla III.1. Tabla de valores para la comparación

3.3. Ámbito de Estudio

El siguiente estudio se centraliza en hacer una comparación entre herramientas que proporcionen el servicio de notificación de eventos, relacionados todos estos con una base de datos y acciones a tomar de acuerdo a los sucesos ocurridos en ella, para lo cual se ha llegado a encontrar las siguientes herramientas que prestan este servicio: DB2 de IBM posee un sistema de notificación de una base de datos para inteligencia de negocios, MySQL posee una herramienta capaz de hacer cualquier tipo de notificaciones, Microsoft con la herramienta SQL Server 2005 que presta el servicio de notificación de la manera más sencilla, Oracle con la herramienta Push Services que se basa en la utilización de servicios web para realizar las notificaciones, todas estas herramientas suman un total de cuatro de las cuales procederemos a escoger dos.

La primera herramienta escogida es la de Microsoft con el Notification Service, por aspectos de negocio y de importancia, ya que, es una herramienta en la cual la mayoría de empresas nacionales e internacionales desarrollan su ámbito de trabajo por lo cual la hemos definido como una de las herramientas adecuadas para el estudio.

La siguiente herramienta es Oracle con el Push Services, por ser una de las más grandes empresas a nivel nacional e internacional que ofrecen un manejador de bases de datos de gran calidad. Por ende ha tenido un desarrollo en la utilización de esta en diferentes empresas del país; es por eso que se ha escogido a esta herramienta.

3.4. Descripción de las Herramientas.

Para el estudio a realizar se ha seleccionado dos herramientas, las mismas que en nuestro medio son las más utilizadas como gestores de base de datos a niveles institucionales y empresariales. SQL Server y Oracle.

3.4.1. Estudio del Servicio de Notificaciones de SQL Server



Figura III.1. Servicio de Notificaciones de SQL Server 2005

3.4.1.1. Introducción a notification services

Microsoft SQL Server Notification Services es una plataforma para desarrollar e implementar aplicaciones que generan y envían notificaciones a los suscriptores. Las notificaciones generadas son mensajes personalizados y oportunos que pueden enviarse a una amplia gama de dispositivos y que reflejan las preferencias del suscriptor.

Los suscriptores crean suscripciones para las aplicaciones de notificación. Una suscripción es un interés manifiesto en un tipo determinado de evento. Por ejemplo, las suscripciones pueden expresar preferencias como ésta: "avisarme cuando el precio de los valores alcance los \$70,00" o "avisarme cuando se actualice el documento de estrategia que está redactando mi equipo".

Una notificación puede generarse y enviarse al suscriptor tan pronto como se produzca un evento que la desencadene. O bien, puede generarse y enviarse según una programación predeterminada especificada por el suscriptor.

Las notificaciones pueden enviarse a una gran variedad de dispositivos. Por ejemplo, una notificación puede enviarse al teléfono móvil de un suscriptor, a su asistente digital personal (PDA), a Microsoft Windows Messenger o a su cuenta de correo electrónico. Dado que el suscriptor suele llevar consigo estos dispositivos, las notificaciones son el medio ideal para enviar información importante.

Las aplicaciones de notificación son valiosas por numerosas razones, entre las que se incluyen:

- Permiten enviar información importante a clientes, socios y empleados. Las notificaciones pueden contener vínculos a un sitio Web para obtener más información o para confirmar la recepción de la información.
- Mejoran y fortalecen las relaciones con los clientes, ya que permiten ofrecerles servicios más personalizados en el momento oportuno.
- Ayudan a incrementar los ingresos, ya que facilitan el establecimiento de relaciones comerciales con los clientes.
- Ayudan a aumentar la productividad de los empleados, ya que les proporcionan la información que necesitan en el momento y lugar oportunos.
- Permiten la comunicación con suscriptores móviles a través de una gran variedad de dispositivos.

Notification Services consta de:

- Un marco de programación de Notification Services sencillo a la vez que eficaz, que permite crear e implementar rápidamente aplicaciones de notificaciones. Permite desarrollar aplicaciones mediante XML u objetos de administración de Notification Services (NMO).
- Un motor confiable, de alto rendimiento y escalable que ejecuta las aplicaciones de notificaciones. El motor de Notification Services se basa en Microsoft .NET Framework y SQL Server 2005.

3.4.1.2. Que es Notification Services

Notification Services es un marco de programación para crear aplicaciones que generan y envían notificaciones, además de una plataforma para alojar dichas aplicaciones. El marco de programación permite crear con rapidez aplicaciones que generan y envían notificaciones a suscriptores. Tras crear la aplicación, se puede implementar en la plataforma Notification Services.

3.4.1.2.1. ¿En qué consiste Notification Services?

Una aplicación de Notification Services genera y envía mensajes a usuarios y otras aplicaciones que se hayan suscrito a la aplicación. Para comprender las aplicaciones de notificaciones, es importante definir algunos términos clave:

- Un *suscriptor* es una persona o aplicación que se suscribe y recibe notificaciones.
- Una suscripción es una solicitud de información específica, por ejemplo, el precio de un valor bursátil o el resultado de un partido, que se entrega a un dispositivo concreto, como una dirección de correo electrónico o un teléfono celular.
- Un evento es una información que interesa a los suscriptores y que reciben de la aplicación de notificaciones. El precio de un valor bursátil en un momento determinado es un evento, como lo es también un resultado deportivo o un mensaje de entrega de producto. Prácticamente cualquier acontecimiento del mundo real puede expresarse como un evento de Notification Services.
- Una notificación es un mensaje que contiene información relacionada con una suscripción. Puede contener un mensaje acerca de un nuevo valor al alza de un valor bursátil determinado o el resultado final de un partido.

En la ilustración siguiente se muestra el funcionamiento básico de una aplicación de Notification Services.

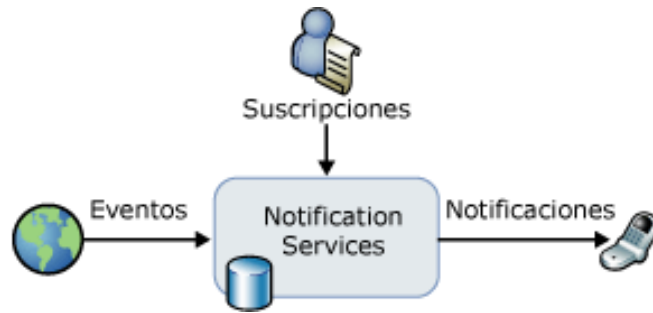


Figura III.2. Funcionamiento básico de una aplicación Notification Services

La aplicación produce notificaciones del siguiente modo:

1. Los suscriptores crean suscripciones relacionadas con la aplicación.
2. La aplicación recopila eventos.
3. Notification Services hace corresponder las suscripciones con los eventos.
4. Cuando un evento y una suscripción coinciden, Notification Services genera una notificación.
5. Notification Services aplica formato a la notificación y la envía al dispositivo especificado en la suscripción.

Un buen ejemplo de una aplicación de notificaciones es una aplicación de valores bursátiles. Un suscriptor podría crear una suscripción para recibir notificaciones en su teléfono celular sobre los valores de ADCP si el precio supera los 70 dólares.

La aplicación de notificaciones recopila datos de los valores a partir de un tablero de cotizaciones. Cada vez que se recibe un "evento" de un lote de precios de valores, Notification Services compara las suscripciones con el conjunto de eventos actual. Si el precio del valor de ADCP supera los 70 dólares, Notification Services crea una notificación, le aplica formato y la envía a un servicio de entrega que reenvía el mensaje al teléfono celular del suscriptor.

Notification Services incluye características para evitar el envío de notificaciones por duplicado. Por ejemplo, puede registrar que un suscriptor determinado ya ha recibido una notificación sobre la cotización de ADCP. Si el siguiente valor bursátil de ADCP también es superior a los 70 dólares, la lógica de notificación cancelará el envío de más notificaciones al suscriptor.

3.4.1.2.2. ¿Cómo funciona Notification Services?

Una aplicación de Notification Services recopila eventos y suscripciones, genera notificaciones y, a continuación, las distribuye a servicios de entrega externos como, por ejemplo, un servidor de protocolo simple de transferencia de correo (SMTP). Notification Services lleva a cabo estas tareas tal como se indica a continuación:

1. Notification Services almacena los datos del suscriptor y de la suscripción en bases de datos del servidor SQL Server. Mediante los *objetos de administración de suscripciones*, que forman parte de la API de Notification Services, se crea una aplicación de administración de suscripciones personalizada que permite administrar datos de suscriptor y de suscripción.
2. Mediante los proveedores de eventos, Notification Services recopila datos de eventos y los almacena en la base de datos de la aplicación. El proveedor de eventos permite incluir archivos de eventos XML en el directorio; a continuación, el proveedor de eventos lee los eventos XML y los envía a la base de datos de la aplicación. Los proveedores de eventos pueden ejecutarse en el componente host del proveedor de eventos, o bien de forma independiente de Notification Services.
3. El generador establece coincidencias entre las suscripciones y los eventos, y genera notificaciones. El generador se ejecuta en un intervalo definido para la aplicación, lo que garantiza que las suscripciones se evalúen regularmente. El programador de

aplicaciones escribe consultas Transact-SQL, para las que los usuarios especifican parámetros o condiciones, que permiten determinar el modo en que se evalúan las suscripciones y la información que se incluye en las notificaciones.

4. El distribuidor aplica formato a las notificaciones y las envía a los suscriptores utilizando uno o varios servicios de entrega. El programador de aplicaciones especifica la transformación de los datos sin procesar a una notificación con formato utilizando un formateador de contenido, como el formateador integrado XSLT.

El motor de Notification Services ejecuta el host de proveedor de eventos, los generadores y los distribuidores. Este motor puede alojarse en un servicio *NS\$instanceName* de Windows, que puede instalarse al registrar la instancia, o en una aplicación o proceso personalizados.

3.4.1.3. Arquitectura de Notification Services

La plataforma Notification Services proporciona una interfaz para recopilar datos de suscriptor y de suscripción, y contiene componentes que recopilan eventos y generan y distribuyen notificaciones, después de darles formato. Además de los componentes integrados, Notification Services permite crear componentes personalizados para recopilar eventos y aplicar formato a las notificaciones.

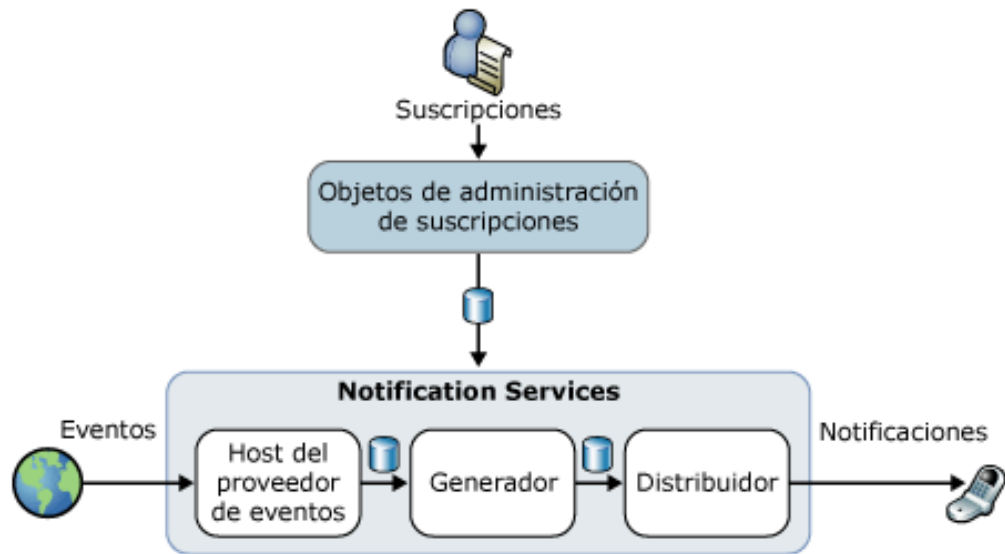


Figura III.3. Arquitectura de Notification Services

3.4.1.3.1. Arquitectura de administración de suscripciones

Para enviar notificaciones, una aplicación de Notification Services debe tener información acerca de los suscriptores, información de interés para los suscriptores e información sobre adónde enviar la información. La administración de suscripciones es el proceso de administrar los datos de suscriptor, de suscripción y de dispositivo de suscriptor.

3.4.1.3.1.1. Interfaces de administración de suscripciones

La administración de suscripciones se controla mediante interfaces de administración de suscripciones personalizadas. Estas interfaces, que pueden ser aplicaciones Web, aplicaciones estándar de Microsoft Windows o incluso aplicaciones de consola para sistemas pequeños internos, se encargan de escribir los datos del suscriptor, de la suscripción y del dispositivo del suscriptor en las bases de datos correspondientes. El programador utiliza los objetos de administración de suscripciones suministrados con Notification Services para simplificar el proceso de desarrollo de la aplicación.

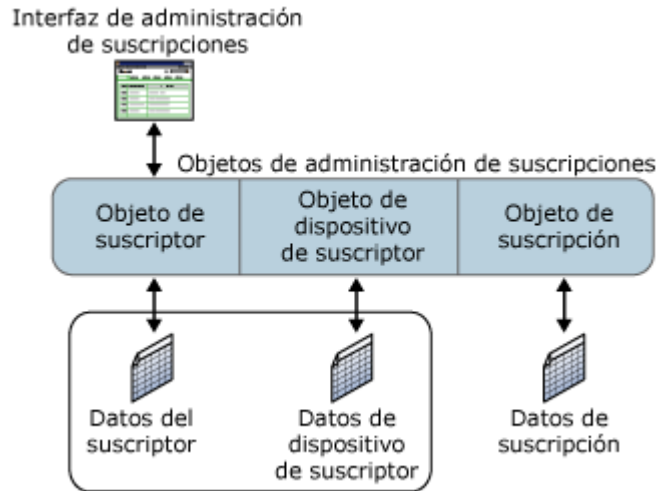


Figura III.4. Interfaces de administración de suscripciones

3.4.1.3.1.2. Vistas de administración de suscripciones

Al crear una instancia de Notification Services, Notification Services agrega tres vistas para administrar el suscriptor, el dispositivo del suscriptor y los datos de suscripción:

- **NSSubscriberView**, que encontrará en la base de datos y el esquema de la instancia, para agregar, modificar y eliminar datos del suscriptor.
- **NSSubscriberDeviceView**, que encontrará en la base de datos y el esquema de la instancia, para agregar, modificar y eliminar dispositivos del suscriptor.
- **NSSubscriptionClassNameView**, que encontrará en la base de datos y el esquema de la instancia, para ver, agregar, modificar y eliminar suscripciones basadas en parámetros. Esta vista no permite administrar suscripciones basadas en condiciones.

3.4.1.3.2. Arquitectura de recopilación de eventos

La recopilación de eventos es el proceso de recopilar datos de eventos de uno o varios orígenes, como, por ejemplo, archivos XML, aplicaciones o bases de datos, y enviar esta información a una

aplicación de notificaciones. Los proveedores de eventos se encargan de ello.

Cada aplicación utiliza uno o varios proveedores de eventos para recopilar eventos. Cada proveedor envía datos a la aplicación mediante una de las tres API de eventos: una API de objetos de evento, una API de XML o una API de SQL Server. En la siguiente Figura 7 se muestra una vista de alto nivel sobre el funcionamiento de estas API.

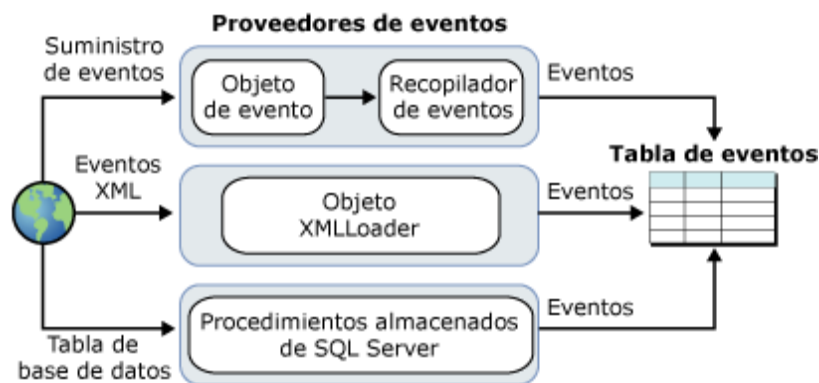


Figura III.5. Arquitectura de recopilación de eventos

- La API de objetos de evento utiliza los objetos Event y EventCollector para enviar eventos individuales. Utilizando los nombres de los campos de una tabla de eventos, una aplicación envía un objeto **Event** al recopilador de eventos, que, a continuación, escribe los datos en la tabla de eventos.
- La API de XML proporciona una forma de carga masiva de datos XML. El proveedor de eventos XML recopila un documento o secuencia XML de un origen de eventos y envía los datos a XML EventLoader, que, a continuación, escribe los eventos en la tabla de eventos.
- La API de SQL Server utiliza procedimientos almacenados para cargar datos de eventos de objetos de base de datos. Dos formas habituales de utilizar el proveedor de eventos de SQL Server son invocar al proveedor mediante un procedimiento almacenado y

ejecutar una consulta según una programación. El proveedor de eventos recibe un conjunto de resultados y lo escribe en la tabla de eventos utilizando los procedimientos almacenados de la API.

3.4.1.3.2.1. Vistas de clase de evento

Al crear una instancia de Notification Services, Notification Services agrega una vista que tiene el mismo nombre que la clase de evento. Esta vista es el origen de eventos para las consultas de generación de notificaciones.

SQL Server 2005 Notification Services también admite la inserción de eventos en esta vista. Cuando se insertan eventos en esta vista, Notification Services crea y cierra un lote de eventos para cada instrucción de inserción.

3.4.1.3.2.2. Proveedores de eventos estándar y personalizados

Los programadores de aplicaciones de Notification Services pueden escribir sus propios proveedores de eventos personalizados mediante cualquiera de las API mencionadas anteriormente, o bien utilizar uno de los proveedores de eventos estándar suministrados con Notification Services. Los proveedores de eventos estándar pueden obtener datos XML de una carpeta de inspección y realizar consultas a las bases de datos de SQL Server y a los cubos de Analysis Services.

3.4.1.3.2.3. Proveedores de eventos alojados y no alojados

Los proveedores de eventos pueden estar alojados o no alojados.

Los proveedores de eventos alojados se ejecutan en Notification Services. Pueden ejecutarse de forma continua o invocarse según una programación definida en la definición de la aplicación. Estos

proveedores de eventos se ejecutan en un componente de Notification Services denominado host del proveedor de eventos. Los proveedores de eventos no alojados se ejecutan como aplicaciones externas y envían eventos según una programación propia. Por ejemplo, un proveedor de eventos alojado en los Servicios de Internet Information Server (IIS) y que expone un método Web para enviar eventos es un proveedor de eventos no alojado.

3.4.1.3.2.4. Escritura de eventos por lotes

Los proveedores de eventos escriben eventos por lotes. La escritura de eventos por lotes permite que el generador combine el conjunto actual de suscripciones con todos los eventos de un lote de eventos a la vez. Este procesamiento orientado a lotes mejora el rendimiento de la aplicación.

3.4.1.3.3. Arquitectura de procesamiento de suscripciones

Una vez obtenidos los eventos, Notification Services ya puede procesar las suscripciones y generar notificaciones. La función del generador es evaluar las suscripciones según los eventos.

Para generar notificaciones, el programador de la aplicación crea una o varias reglas para la aplicación. Estas reglas se escriben como consultas **Transact-SQL** que especifican cómo se relacionan los eventos y las suscripciones, así como cualquier otra condición que deba cumplirse para generar una notificación.

En una aplicación sencilla, cuando el generador activa una regla, la aplicación evalúa todas las suscripciones disponibles y las compara con el lote actual de eventos visible desde una vista de eventos. Cuando un evento coincide con una suscripción, el generador de notificaciones crea una notificación. Esta notificación contiene datos acerca del evento. También hace referencia a datos del suscriptor, del

dispositivo del suscriptor y, en ocasiones, a la configuración local del suscriptor, además de otra información necesaria para la distribución.



Figura III.6. Arquitectura de procesamiento de suscripciones

La notificación no se envía nada más crearla. En su lugar, el generador escribe la notificación en una tabla de notificaciones interna. Cuando un lote de notificaciones está preparado, el distribuidor aplica formato a las notificaciones y las distribuye.

3.4.1.3.3.1. Tipos de regla

Las reglas definidas en la aplicación controlan el funcionamiento del generador. Puede crear los siguientes tipos de reglas:

- Las *reglas de crónica de eventos* almacenan o actualizan el historial de eventos en tablas de crónica definidas por el programador de la aplicación. Cada vez que se ejecuta el generador, activa este tipo de regla en primer lugar.
- Las reglas de evento generan notificaciones para las suscripciones por eventos. Este tipo de regla se ejecuta después de la regla de crónica de eventos si se encuentra disponible un lote de eventos asociado. Este tipo de regla también puede administrar tablas de crónica.
- Las reglas programadas generan notificaciones de suscripciones programadas. Este tipo de regla se ejecuta después de la regla de crónica de eventos para cualquier suscripción relacionada que deba

procesarse. Este tipo de regla también puede administrar tablas de crónica.

3.4.1.3.3.2. Tipos de acción de regla

Las reglas de evento y programadas especifican la acción que debe ejecutarse al desencadenarse la regla. Cada acción es una consulta Transact-SQL que define una unidad de trabajo que debe realizar el generador. Las reglas de evento y programadas utilizan tanto acciones simples basadas en parámetros como acciones de condición más flexibles:

- Las *acciones simples* son consultas Transact-SQL que definen por completo la consulta de generación de notificación, incluidas las cláusulas WHERE. Las acciones simples obtienen las expresiones de la cláusula WHERE de los datos de suscripción y los datos de evento.
- Cuando una regla utiliza una acción simple, los suscriptores proporcionan parámetros para la consulta tales como el nombre de la ciudad.
- Las acciones de condición permiten a los suscriptores definir por completo las condiciones para la búsqueda de la consulta. La interfaz de administración de suscripciones simplifica la escritura de las condiciones de búsqueda hasta tal punto que basta con seleccionar columnas y operadores de los cuadros de lista y especificar los valores en los cuadros de texto.

Las acciones simples generan un conjunto limitado de condiciones de búsqueda que debe evaluar el generador, por lo que su rendimiento suele ser mejor que el de las acciones de condición. Las acciones de condición, por su parte, son más eficaces, pero deben evaluar más condiciones de búsqueda para la regla de evento o regla programada.

3.4.1.3.4. Arquitectura de entrega y formato de notificaciones

En Notification Services, el formato y la distribución de notificaciones son tarea del distribuidor. Una vez que el generador crea un lote de notificaciones, el distribuidor separa el lote en elementos de trabajo y destina cada uno de estos al mismo canal de entrega. A continuación, el distribuidor envía el elemento de trabajo al formateador de contenido. Cuando finaliza el formato, el distribuidor envía las notificaciones al extremo de entrega que haya especificado el canal de entrega.

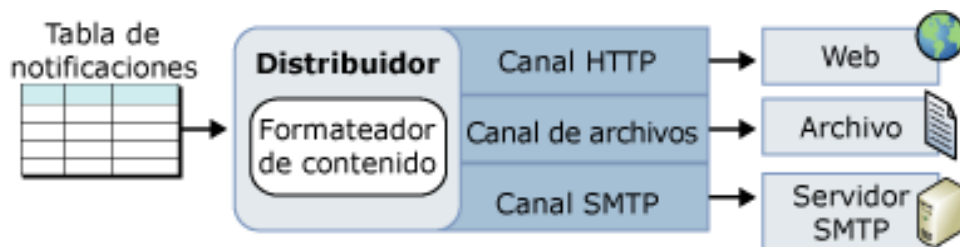


Figura III.7. Arquitectura de entrega y formato de notificaciones

3.4.1.3.4.1. Formato de mensajes

Gran parte del proceso de creación de una aplicación consiste en definir la transformación de los datos de notificación sin procesar en mensajes legibles. Notification Services incluye un formateador de contenido XSLT (Extensible Stylesheet Language Transformation) estándar, que requiere un archivo XSLT para cada tipo de dispositivo y configuración regional compatible con la aplicación.

3.4.1.3.4.2. Canales de entrega

Notification Services no controla la entrega final de las notificaciones. En cambio, Notification Services utiliza canales de entrega, cada canal de entrega utiliza un protocolo de entrega específico para cada

extremo de entrega concreto. Notification Services incluye los siguientes protocolos más habituales:

- El protocolo SMTP para enviar notificaciones al servidor Microsoft Exchange u otros servidores SMTP.
- El protocolo HTTP extensible, que puede utilizarse con el protocolo simple de acceso a objetos (SOAP), el servicio de mensajes cortos (SMS), .NET Alerts y otros protocolos relacionados con HTTP.
- El protocolo File para enviar notificaciones a archivos del sistema operativo. Se utiliza principalmente para depurar aplicaciones.

Si es necesario, puede definir varios canales de entrega para un único protocolo de entrega.

3.4.1.4. Seguridad, confiabilidad, escalabilidad y disponibilidad de Notification Services

Un sistema empresarial debe ser seguro, confiable, escalable y debe estar disponible. Microsoft SQL Server Notification Services hace frente a estos requisitos en el marco de programación y en la arquitectura del servidor que ejecuta Notification Services.

3.4.1.4.1. Seguridad

Notification Services, Microsoft .NET Framework y SQL Server incorporan características de diseño que permiten a los programadores de aplicaciones y a los administradores reforzar la seguridad de un extremo a otro de las aplicaciones y la implementación global:

- Notification Services utiliza modelos de seguridad Web estándar en el sector para comunicarse a través de Internet mediante dispositivos con cables e inalámbricos. Utiliza mecanismos de

seguridad estándar de Microsoft Windows y mecanismos de seguridad basados en SQL Server 2005 para proteger el servidor.

- Las instancias individuales de Notification Services se aíslan entre sí mediante mecanismos de seguridad basados en SQL Server 2005. De este modo, se garantiza la posibilidad de alojar varias instancias en un mismo equipo sin poner en peligro la seguridad de cada una de las instancias. Si desea aislar las instancias por completo, no coloque varias instancias en la misma base de datos porque comparten funciones de base de datos.
- Los suscriptores individuales se aíslan entre sí debido al método utilizado para generar y distribuir las notificaciones. Como resultado, cada suscripción se procesa como si fuera la única suscripción en el sistema.
- Las acciones de condición, compatibles con consultas definidas por el usuario en conjuntos de datos definidos, requieren cuentas de usuario de base de datos independientes que garanticen que los usuarios no puedan tener acceso a los datos del servidor.
- Los programadores de aplicaciones pueden ampliar Notification Services definiendo sus propios proveedores de eventos. Estos proveedores se ejecutan como código administrado compatible con .NET Framework, lo que garantiza que no se ponga en peligro la seguridad del servidor.

3.4.1.4.2. Confiabilidad

El SQL Server 2005 Database Engine (Motor de base de datos de SQL Server 2005) almacena suscripciones de datos y procesos con una gran confiabilidad, y Notification Services aprovecha esta característica:

- Los datos de suscripciones que administra Notification Services se almacenan en SQL Server 2005, un servidor de bases de datos muy sólido que admite el registro de escritura anticipada y el clúster de conmutación por error.

- El procesamiento de eventos, suscripciones y notificaciones en Notification Services es completamente transaccional, por lo que ofrece garantías de confiabilidad adicionales. Tenga en cuenta que algunos protocolos de entrega (por ejemplo, el Protocolo simple de transferencia de correo o SMTP) no ofrecen una entrega garantizada de los mensajes; sin embargo, la plataforma Notification Services tiene una lógica de reintentos que permite que una aplicación vuelva a enviar mensajes para intentar de nuevo la entrega.

Asimismo, el uso de código administrado en los componentes personalizados por parte de Notification Services garantiza que los componentes personalizados no pongan en peligro la estabilidad del sistema. Los programadores de aplicaciones pueden generar proveedores de eventos y protocolos de entrega confiables, por ejemplo, basados en Message Queuing (también denominado MSMQ), logrando de este modo una confiabilidad completa de un extremo a otro.

3.4.1.4.3. Escalabilidad

Notification Services admite aplicaciones de gran volumen con miles de eventos entrantes por minuto, millones de suscripciones coincidentes y miles de notificaciones entregadas por minuto en un servidor Windows con varios procesadores. Las aplicaciones pueden escalarse de varias formas:

- Las interfaces de administración de suscripciones pueden ejecutarse en paralelo, lo que permite obtener una mayor eficacia de la ampliación en vertical en el SQL Server Database Engine (Motor de base de datos de SQL Server) subyacente.
- Notification Services es una aplicación multiproceso y se amplía verticalmente cuando se ejecuta en un equipo con varios procesadores.
- El servicio Notification Services puede implementarse en un servidor y la base de datos de Notification Services en otro

distinto. Esta configuración es muy eficaz ya que la generación de notificaciones se ejecuta en el servidor de bases de datos, mientras que el formato de notificaciones se ejecuta en el servidor de Notification Services.

- Una única instancia puede admitir varios proveedores de eventos y varios distribuidores. Es posible distribuir los proveedores de eventos, el generador y los distribuidores en varios servidores. La capacidad para ampliar en horizontal es importante en las implementaciones con un volumen elevado de notificaciones y con sobrecargas importantes de formato y entrega.
- Para una mayor ampliación en horizontal, se pueden dividir las suscripciones en varias instancias de Notification Services, cada una de las cuales funciona de forma independiente. Puesto que el modelo de aplicación crea suscripciones independientes entre sí, dicha partición es natural y hace que las aplicaciones de notificaciones sean muy escalables. Es necesario implementar la partición de suscripciones en la aplicación porque Notification Services no lleva a cabo esta tarea automáticamente.

3.4.1.4.4 .Disponibilidad

En una instancia de Notification Services, el estado y todos los datos se mantienen en el servidor de bases de datos de SQL Server. Sólo los datos transitorios se mantienen en el proveedor de eventos, el generador y el distribuidor.

SQL Server 2005, donde se almacenan y procesan los datos, es un servidor de bases de datos confiable con una gran variedad de tecnologías de alta disponibilidad, incluido el clúster de conmutación por error, el trasvase de registros y la réplica transaccional. Se puede configurar el servidor de bases de datos para que controle automáticamente los errores de hardware.

3.4.2. Estudio del Servicio de Notificaciones de Oracle

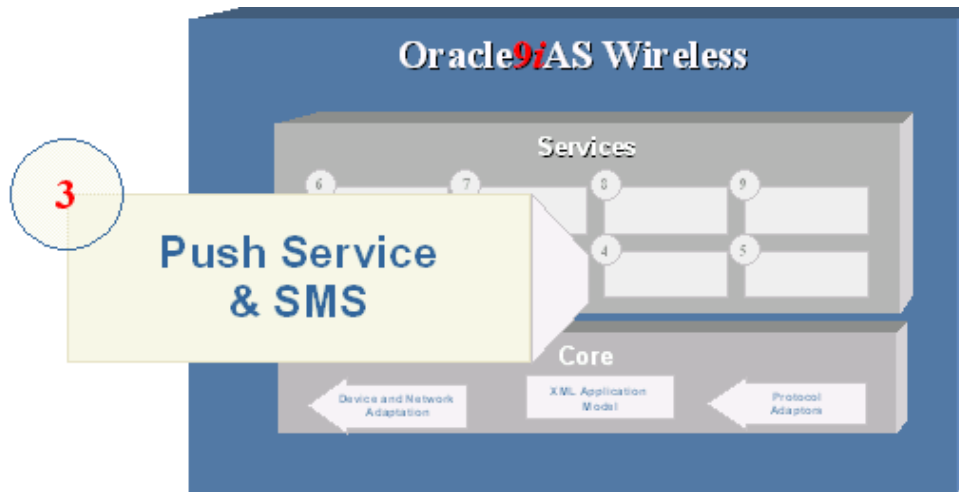


Figura III.8. Servicio de Notificaciones de Oracle9iAS Wireless (Push Service)

3.4.2.1. Introducción a push services

Push Services en Oracle9iAS Wireless es una plataforma escalable que puede manejar grandes volúmenes de mensajes a muchos dispositivos. El Push Services está basado en una arquitectura extensible y un diseño que puede ser extendido para soportar una variedad de dispositivos y protocolos Push. Los mensajes Push SOAP son manejados por un subsistema de traspaso de mensajes de manera electrónica en el servidor Oracle9iAS Wireless. Este subsistema soporta una arquitectura basada en controladores. Los controladores son componentes en el sistema de mensajería inalámbrica que manejan todas las especificaciones de dispositivos o especificadas rutinas de protocolos de comunicación. El subsistema de mensajería, basada en las direcciones de dispositivos y tipos de transporte (SMS, Voz, E-mail), despacha el mensaje a la implementación del controlador transporte/protocolo apropiado. La interfaz del controlador entrega el mensaje a los dispositivos en el protocolo nativo del dispositivo. El Subsistema de mensajería puede soportar múltiples controladores en una instancia simple y puede soportar múltiples controladores.

Push services son un componente importante que mejora las características de tu aplicación móvil soportando alertas a tus usuarios móviles. Oracle 9iAS Wireless Push services provee un mecanismo altamente escalable para entregar mensajes a todos los dispositivos móviles en un protocolo que es nativo para el dispositivo, por ejemplo vía SMS a un teléfono móvil, como un e-mail a un buscapersonas de dos vías, como un mensaje de audio a un teléfono normal o como fax a una máquina de fax.

Push Services en Oracle9iAS Wireless están implementados como un Servicio Web (WSDL) y utilizan SOAP sobre HTTP. El servicio SOAP permite a las aplicaciones invocar métodos objeto sobre el protocolo HTTP. Esto permite a las aplicaciones invocar a push service desde cualquier parte del Internet y usar cualquier modelo de programación. Oracle9iAS Wireless Push Services permite a las aplicaciones especificar tanto al mensaje como al recipiente o recipientes del mensaje. La aplicación se comunica al Push Service en Oracle9iAS Wireless usando SOAP y HTTP. Oracle9iAS Wireless recibe el mensaje y entrega los mensajes a los dispositivos móviles usando protocolos apropiados tales como SMS, E-mail, Voz, y otros.

3.4.2.2. Que es Push Services

Push Services es para crear aplicaciones que generan y envían notificaciones, a través de una plataforma escalable. Por medio de estas aplicaciones que generan y envían notificaciones a suscriptores. Tras crear la aplicación, se puede implementar en la plataforma Push Services.

3.4.2.2.1. ¿En qué consiste Push Services?

El servidor Oracle9iAS Wireless envía una implementación especial de controlador que permitiendo a tu instancia inalámbrica (o instala) actuar como un cliente a otra instalación Oracle9iAS Wireless o cualquier servicio que respete la interfaz del servicio Web definida por Oracle9iAS Wireless. Este controlador especificado usa la interfaz SOAP (con el Push APIs) para el envío de mensajes. En el producto por defecto, este

controlador está configurado para actuar como el cliente push a una instancia de Oracle9iAS Wireless hospedada, en Internet, por Oracle. Los administradores de instancia pueden cambiar esta configuración por defecto para apuntar a cualquier servidor que respete la interfaz PUSH WSDL definida por Oracle9iAS Wireless.

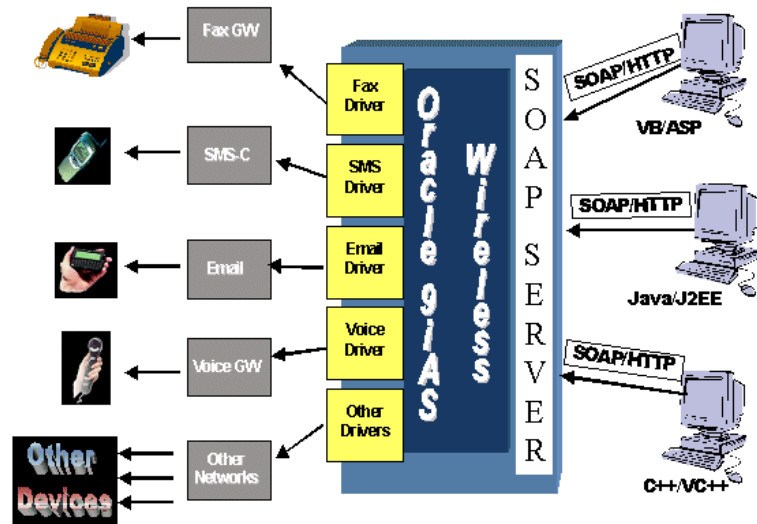


Figura III.9. Servidor de Controladores de Mensajes

Oracle9iAS Wireless puede soportar contenido Push de diferentes tipos MIME como documentos de Microsoft Word o tonos. Un mensaje sólo puede consistir en texto o puede ser un mensaje del multipartes complejo. Oracle9iAS Wireless identifica los tipos de mensaje basados en el MIME, por lo tanto entregar un documento como MS-Word o PDF es posible si el dispositivo designado soporta el tipo de mensaje MIME. Oracle9iAS Wireless proporciona el Java Push API que soporta mensaje de solo texto (PushLite) y un Push API avanzado que soporta mensajes de cualquier tipo MIME (Push).

Las siguientes son las dos interfaz para enviar un Mensaje Push:

- Push Lite
- Push

3.4.2.2.1.1. PushLite

PushLite proporciona las habilidades de mensajería de texto. Es ligero y muy fácil usar.

Manda un mensaje de texto (sin el asunto) a múltiples destinatarios de múltiples transportes. Codificar el mensaje es "texto/plano". Este método proporciona la manera más fácil de mandar los mensajes de texto. Otros métodos sobrecargados send() pueden usarse para poner el asunto, conteste a, la codificación del tipo de contenido (el tipo MIME) y parámetros importantes asociados.

Los remitentes – es una serie de las direcciones de remitentes. La dirección de un remitente tiene un tipo de transporte y dirección, separada por dos puntos (:). Un remitente por el tipo de transporte. El último remitente del mismo tipo de transporte atrópellará a los remitentes más tempranos del mismo tipo de transporte en la serie.

Los destinatarios - las direcciones de destinatarios (Direcciones de correo electrónico o números telefónicos del Oracle9iAS Wireless)

El formato de direcciones del destinatario:

```
<transport>:<recipient address 1>[,recipient address 2 ...]
```

Pueden separarse destinatarios del mismo transporte en líneas múltiples, Pero, estas líneas no pueden separarse por los destinatarios de otras líneas de tipo de transporte. Una excepción se lanzará si se ha descubierto una.

El mensaje - el cuerpo de mensaje.

Manda un mensaje de texto con el asunto, conteste a las direcciones, llaves asociadas y codificación del tipo de contenido.

El asunto - el asunto del mensaje (opcional).

El mensaje - el cuerpo del mensaje.

Codificar - el tipo MIME con codificación opcional charset del mensaje. Por ejemplo: "text/plain", "text/plain; charset=us-ascii" y "text/html"

3.4.2.2.1.2. Push

Envía el paquete de mensaje.

PKT - El paquete de mensaje a ser entregado.

La clase del paquete representa un mensaje genérico en el mundo real. (Por ejemplo: el correo electrónico) puede tener un asunto, un cuerpo o un conjunto de cuerpos del mensaje (multiparte). El mismo mensaje puede entregarse a los destinatarios múltiples de tipos de transporte múltiples (tipos de entrega).

Por ejemplo: el mismo mensaje puede entregarse a 2 destinatarios de correo electrónico, 3 destinatarios de SMS y 4 máquinas de fax en el mismo paquete.

Cada tipo de transporte puede tener un remitente, una contestación a dirección alterna y un grupo de destinatarios. El paquete podría tener un conjunto de instrucciones de entrega opcionales, como la prioridad, registrado, etc.

Para lograr esto, primero se construye una instancia de paquete vacía. Entonces se pone él, la información del mensaje, el remitente, contestación a y destinatarios del paquete.

3.4.2.2.2. ¿Cómo funciona Push Services?

El servicio Push de Oracle9iAS Wireless es destacado como un Servicio Web (WSDL) usando SOAP con HTTP como la capa de transporte. WSDL (Web Services Definition Language) es un estándar de la interfaz XML que define una aplicación de Servicio Web. Con WSDL claramente definido, los desarrolladores pueden construir aplicaciones en cualquier lenguaje de programación tal como Java y Visual Basic, y comunicarse con la interfaz del subsistema de mensajería del Oracle9iAS Wireless sobre Internet. El desarrollador puede usar cualquier herramienta de WSDL para implementar rápidamente una aplicación Push y enviar mensajes a dispositivos móviles usando cualquier instancia Oracle9iAS Wireless en el Internet.

Oracle9iAS Wireless soporta un simple Push API, en Java, que resumen cualquier implementación de protocolo específico (SOAP) desde el código de la aplicación Java. El Java Push API es una interfaz en el lenguaje de programación Java. Es el API preferido para los desarrolladores de aplicaciones que necesiten una clara y simple interfaz de entrega de mensajes. Las API Java Push usan SOAP sobre HTTP para comunicar una instancia del servidor Oracle9iAS Wireless.

El API Push soporta una interfaz uniforme para la entrega de mensajes a cualquier controlador de dispositivos como SMS, Voz, Correo electrónico, y Fax. El API permite a las aplicaciones especificar múltiples destinatarios para un solo mensaje usando sólo una petición de entrega. Además, las direcciones de destino de mensaje pueden tener dispositivos usando diferentes canales de comunicación, por ejemplo, una aplicación de petición de entrega de mensaje simple puede enviar mensajes E-mail así como a las máquinas de fax. Las aplicaciones pueden hacer una petición de entrega y enviar los mensajes a una lista de usuarios con los dispositivos de SMS, clientes del Correo electrónico y dispositivos de Voz.

Oracle9iAS Wireless soporta diferentes tipos de contenido para la entrega. Un mensaje simplemente puede consistir de sólo caracteres de texto, o puede ser tan complejo como un mensaje de multipartes. Se identifican los tipos del mensaje basado en el MIME, por lo tanto entregar documentos tales como Microsoft Word o Adobe PDF es posible si el dispositivo destino soporta el mensaje de tipo MIME.

El API Push proporciona los métodos para poner las propiedades de un mensaje y despachar a la instancia Oracle9iAS Wireless. Para enviar un mensaje push se necesitará proporcionar a lo siguiente:

- El servidor Oracle9iAS Wireless dónde el Servicio Web Push se está ejecutando. Se incluye el username/password y el proxy HTTP requeridos para acceder al Servicio Web remoto Oracle9iAS Wireless (a menos que la aplicación esté corriendo en un Oracle9iAS Wireless VM).
- El mensaje actual para ser enviado y el tipo de contenido (MIME) del mensaje.
- La dirección de los dispositivos destino como la dirección de correo electrónico, número de teléfono etc.
- Cómo el mensaje debe entregarse, por ejemplo, como mensaje de audio sobre voz, mensaje SMS sobre voz o como mensaje de correo electrónico.
- La dirección del remitente de este mensaje. El API permite poner la dirección del remitente sobre una base de tipo de transporte, como una dirección de remitente para SMS, uno para correo electrónico y uno para voz.

3.4.2.3. Arquitectura de Push Services

Oracle9iAS Wireless contiene un subsistema de mensajería que maneja los mensajes de envío y recepción a y desde los dispositivos, y todas las funciones de ruteo del mensaje. El controlador, un componente de este subsistema, implementa la memoria del protocolo de comunicación actual

mientras el sistema de mensajería soporta el API de transporte que proporciona la abstracción para las aplicaciones para comunicar con el subsistema de mensajería. El API de Transporte (así como en el Push API) permite a las aplicaciones enviar mensajes a dispositivos móviles. Además de enviar mensajes el API de Transporte permite a las aplicaciones recibir mensajes desde dispositivos cliente y también permite a las aplicaciones cambiar mensajes de ruteo en tiempo de ejecución.

La aplicación del API de Transporte debe estar corriendo en el mismo Java VM como la instancia Oracle9iAS Wireless para usar el API de Transporte (diferente del Push API el cual no tiene este requisito). Las aplicaciones internas Oracle9iAS Wireless, como el servidor Async y el Motor de alarma, usan el API de Transporte para entregar los mensajes a los dispositivos móviles.

Muestra la arquitectura del subsistema de mensajería. Para recibir los mensajes del dispositivo, las aplicaciones deben implementar una interfaz del oyente y deben registrar la instancia del oyente con el subsistema de mensajería.

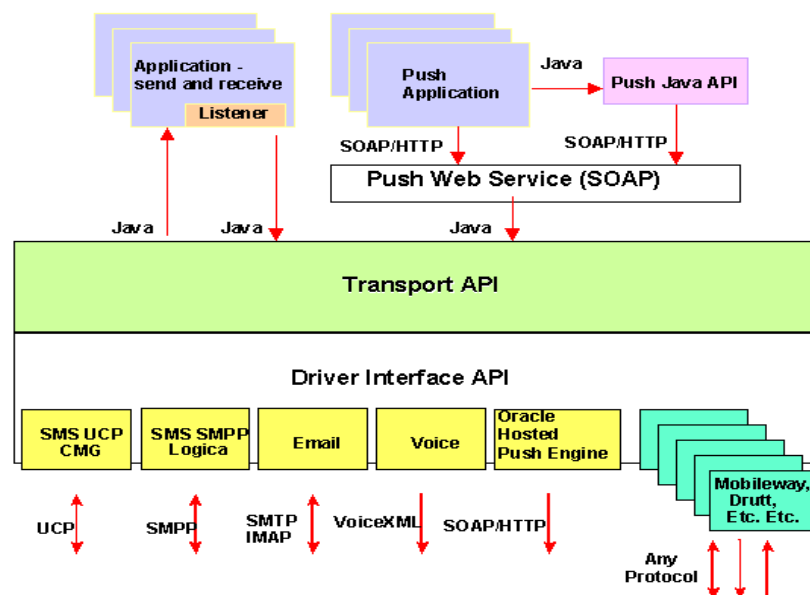


Figura III.10. Arquitectura del Sistema de Mensajería

Los API de transporte son independientes de los protocolos fundamentales de la red requeridos para comunicar con los dispositivos móviles. Los protocolos fundamentales de la red están implementados por el controlador. Los controladores implementan las memorias de mensaje específico de comunicación dispositivo-específico e inalámbrica. Para entregar los mensajes, el sistema de mensajería usa al controlador apropiado para entregar los mensajes al dispositivo. El API de Transporte soporta operaciones que permite a las aplicaciones enviar y recibir los mensajes de los dispositivos.

El API de Transporte proporciona a las aplicaciones Push la abstracción necesaria de los protocolos fundamentales inalámbricos y permite a la aplicación Push ser el agnóstico del dispositivo/red. El API de Transporte permite a las aplicaciones, si se requiere, controlar mensaje que rutean con los controladores disponibles.

El sistema de mensajería apoya a los controladores múltiples para diferentes protocolos inalámbricos y también permite a los controladores múltiples que soporten un protocolo de red dado. El sistema de la mensajería maneja todas las funciones de ruteo de mensajes, técnicas para aumentar rendimiento, tolerancia de falla, y soporta un ambiente altamente escalable. El API les permite a diseñadores personalizar varias funciones del sistema de transporte.

Los controladores son componentes interconectables en el sistema de mensajería; ellos se ocupan de todos los requisitos específicos de red/protocolo. El sistema de Mensajería define una Interfaz de Controlador que proporciona el desacoplamiento necesario para lograr la abstracción requerida. El Controlador API proporciona una interfaz extensible para la red y los controladores protocolo-específico para ser conectado en el sistema de mensajería Oracle9iAS Wireless. Una implementación de controlador envuelve rutinas de protocolo de red requerido por ejemplo, un controlador de correo electrónico puede

implementar una interfaz de IMAP/SMTP o puede usar una interfaz de MAPI. Los Controladores deben implementar la interfaz definida por el sistema de mensajería.

3.4.2.3.1. Servicio Web Push SOAP

El Servicio Web Push es un Servlet que corre como parte del VM para el tiempo de ejecución del Oracle9iAS Wireless, ningún proceso especial o independiente se empieza para él.

3.4.2.3.2. Servidores de mensajería

Los procesos de servidor de mensajería son contenedores en tiempo de ejecución para los controladores de protocolo y proveen el ambiente necesario para los controladores. El servidor de mensajería ejecuta las instancias del controlador y maneja el ciclo de vida de las instancias del controlador. Oracle9iAS Wireless puede ejecutar los servidores de la mensajería múltiples en máquinas diferentes.

3.4.2.3.3. Controlador e Instancia de Controlador

El Controlador es el software que implementa la Interfaz del Controlador de Transporte Oracle9iAS Wireless. Una instancia de Controlador es una ejemplificación en tiempo de ejecución de un Controlador que corre en un proceso.

Cada controlador identifica el tipo de protocolo que maneja como correo electrónico o Fax. Un solo controlador puede manejar más de un protocolo. La clase de implementación y el parámetro de configuración de controladores son especificados usando el Oracle9iAS Wireless Webtool.

Los controladores se configuran como una propiedad de nivel del sitio para una instalación Oracle9iAS Wireless y pueden registrarse

con uno o más servidores de mensajería. Tal como una configuración les permite a administradores cargar el equilibrio y manejar recursos del hardware basados en el uso de controladores individuales.

3.4.2.3.4. El API de Transporte

Esta sección detalla el API de Transporte, explicando las estructuras mayores y funcionalidad disponible para personalizar el Sistema de Transporte.

El API de transporte es la interfaz de mensajería del lado del cliente. El API de transporte es un conjunto rico de APIs que pueden usarse para enviar y recibir. Por lo que se refiere a enviar, el API de transporte proporciona algunos rasgos extra valorados, como el ruteo de mensajería y rastreo de estado.

Para recibir los mensajes, la aplicación debe registrar escuchando los puntos de fin y un mensaje de llamada de retorno del oyente al sistema de transporte. Un end point está esencialmente en el formulario de una dirección como un número de teléfono. Identifica al sistema de transporte cómo el mensaje debe despacharse. Cuando un mensaje se recibe por una dirección apuntada, se despacha al oyente asociado con un end point con una dirección correspondiente.

Cuando un mensaje de petición de entrega se somete, el sistema de transporte realiza análisis de los destinatarios y encamina el mensaje a los controladores de protocolos apropiados para la entrega.

3.4.2.3.4.1. Análisis de destino

Un solo mensaje puede entregarse a los destinatarios múltiples de diferentes protocolos de comunicación. Por ejemplo, uno puede enviar un recordatorio de reunión a algunas personas que usan

SMS, y algunas otras personas a sus direcciones de correo electrónico. Así antes de encaminar los mensajes a controladores, el transporte debe analizar y agrupar a los destinatarios por su categoría de entrega. Típicamente, el sistema de transporte empieza su proceso interno analizando todos los destinos y se los agrupa respectivamente.

3.4.2.3.4.2 Controladores pre-construidos Oracle9iAS Wireless

Como se explico antes, los controladores son componentes interconectables a Oracle9iAS Wireless que extienden el soporte de un protocolo específico del sistema. Oracle9iAS Wireless envía con algunos controladores pre-construidos que soportan la mayoría de protocolos que han sido aceptados como estándares de la industria.

Los controladores pre-construidos manejan protocolos de comunicación tales como SMS (SMPP y UCP), E-mail, Voz, Fax. La siguiente sección provee un breve vistazo de estos controladores pre-construidos y las propiedades de configuración requeridas para estos controladores. Los controladores implementan solamente un subconjunto especificaciones de protocolos de comunicación que son suficientes para construir, entregar y recibir un mensaje. Los controladores pre-construidos Oracle9iAS Wireless típicamente no implementan la especificación completa de los protocolos de comunicación.

3.4.2.3.4.2.1. El Controlador PushClient

Este controlador utiliza un servicio Hosted Push y por defecto utiliza el servicio hospedado por Oracle Corporation. Este controlador esencialmente actúa como

un cliente Push a un servidor Oracle9iAS Wireless albergado en el Internet y puede ser configurado para apuntar a cualquier servicio que soporte el Servicio Web Push de Oracle9iAS Wireless. El controlador PushClient utiliza un protocolo especial, SOAP sobre HTTP (el Servicio Web Push de Oracle9iAS Wireless). De hecho, el servidor push de Oracle no requiere ninguna cuenta para el acceso. Si no se ha registrado, se puede utilizar "" como username y password. El URL para esto es:

<http://messenger.us.oracle.com/push/webservices>

3.4.2.3.4.2.2. El controlador E-mail

El controlador e-mail soporta SMTP en la entrega de mensajes, e IMAP o POP3 en la recepción de mensajes. Este controlador puede manejar envío y recepción de mensajes. Los protocolos IMAP y POP3 son soportados para la recepción de mensajes.

3.4.2.3.4.2.3. Controlador de Voz

El controlador de voz soporta el protocolo Out Bound Call soportado por VoiceGenie. Actualmente, ha sido probado para trabajar sólo con una puerta VoiceGenie. Este controlador maneja mensajes de envío solamente. Aunque el controlador solo puede enviar mensajes, debe ser configurado para capacidades de envío y recepción para que el controlador trabaje.

Este controlador puede manejar texto de tipo de contenido

`ContentTypes.MOBILE_XML_URL`

`ContentTypes.MOBILE_XML_URL_REMOTE`

ContentTypes.MOBILE_XML

ContentTypes.URL (solamente si el tipo de contenido del recurso URL es uno de los anteriores)

Para otros tipos de contenido, el controlador lanzará una excepción.

3.4.2.3.4.2.4. El controlador UCP

UCP (Universal Communication Protocol) es uno de los más populares protocolos GSM SMS. El Servidor Oracle9iAS Wireless se manda con una implementación pre construida del controlador UCP que es capaz de enviar y recibir.

Este controlador maneja los siguientes tipos de contenido:

ContentTypes.RING_TONE

ContentTypes.GRAPHICS

ContentTypes.WAP_SETTINGS

ContentTypes.URL (solamente si el tipo de contenido del recurso relacionado es uno de los mencionados)

Para otros tipos de contenido, el controlador lanzará una excepción.

3.4.2.3.4.2.5. El controlador SMPP

SMPP (Short Message Peer to Peer) es uno de los más populares protocolos GSM SMS. El Servidor Oracle9iAS Wireless se manda con una implementación pre construida del controlador SMPP que es capaz de enviar y recibir. El

controlador abre una conexión TCP al SMSC como un receptor transmisor, por lo tanto solamente una conexión (iniciada por el controlador) se necesita para todas las comunicaciones entre el controlador y el SMSC.

Este controlador puede manejar tipo de contenido texto ContentTypes.URL (solamente si el tipo de contenido del recurso relacionado es texto). Para otros tipos de contenido, el controlador lanzará una excepción.

3.4.2.3.4.2.6. El controlador de Fax (RightFax)

Este es un controlador que soporta mensajes de Fax y soporta RightFax (by Captaris) protocolo Fax. El controlador depende del paquete de software RightFax y la disponibilidad del servidor RightFax Fax para entregar mensajes Fax. Este controlador es capaz solamente de enviar mensajes.

Este controlador puede manejar cualquier tipo de contenido. Particularmente reconoce los siguientes tipos MIME:

text/xml

application/msword

application/msexcel

application/msppt

application/postscript

application/octet-stream

En el caso del ContentTypes.URL, el controlador recuperara el contenido de la URL especificada. El contenido y el tipo MIME retornados por esta operación se hará el contenido y el tipo MIME enviado al servidor fax.

3.4.2.4. Seguridad, confiabilidad, escalabilidad y disponibilidad de Push Services

Un sistema empresarial debe ser seguro, confiable, escalable y debe estar disponible. Oracle9iAS Wireless Push Services debe cumplir hace frente a estos requisitos en el marco de programación y en la arquitectura del servidor que ejecuta Push Services.

3.4.2.4.1. Seguridad

Push Services, Oracle9iAS Wireless incorporan características de diseño que permiten a los programadores de aplicaciones y a los administradores reforzar la seguridad de un extremo a otro de las aplicaciones y la implementación global:

- Para acceder a los datos en una BD Oracle, se debe tener acceso a una cuenta en esa BD. Cada cuenta debe tener una palabra clave o *password* asociada. Una cuenta en una BD puede estar ligada con una cuenta de sistema operativo. Los *passwords* son fijados cuando se crea un usuario y pueden ser alterados por el DBA o por el usuario mismo. La BD almacena una versión encriptada del *password* en una tabla del diccionario llamada *dba_users*. Si la cuenta en la BD está asociada a una cuenta del sistema operativo puede evitarse la comprobación del *password*, dándose por válida la comprobación de la identidad del usuario realizada por el SO.
- El acceso a los objetos de la BD se realiza via privilegios. Estos permiten que determinados comandos sean utilizados contra

determinados objetos de la BD. Esto se especifica con el comando GRANT, conceder. Los privilegios se pueden agrupar formando lo que se conoce por roles. La utilización de los roles simplifica la administración de los privilegios cuando tenemos muchos usuarios. Los roles pueden ser protegidos con passwords, y pueden activarse y desactivarse dinámicamente, con lo que constituyen una capa más de seguridad en el sistema.

3.4.2.4.2. Confiabilidad

El Oracle9iAS Wireless Database Engine (Motor de base de datos de Oracle9iAS Wireless) almacena suscripciones de datos y procesos con una gran confiabilidad, y Push Services aprovecha esta característica:

- Los datos de suscripciones que administra Push Services se almacenan en Oracle9iAS Wireless, un servidor de bases de datos muy sólido que admite el registro de escritura.
- Tenga en cuenta que algunos protocolos de entrega no ofrecen una entrega garantizada de los mensajes; sin embargo, la plataforma Push Services tiene una lógica de reintentos que permite que una aplicación vuelva a enviar mensajes para intentar de nuevo la entrega.

3.4.2.4.3. Escalabilidad

Push Services admite aplicaciones de eventos entrantes por minuto, millones de suscripciones coincidentes y miles de notificaciones entregadas por minuto en un servidor Windows con varios procesadores. Las aplicaciones pueden escalarse de varias formas:

- Push Services es una aplicación multiproceso y se amplía verticalmente cuando se ejecuta en un equipo con varios procesadores.

- Una única instancia puede admitir varios proveedores de eventos y varios distribuidores. Es posible distribuir los proveedores de eventos, el generador y los distribuidores en varios servidores. La capacidad para ampliar en horizontal es importante en las implementaciones con un volumen elevado de notificaciones y con sobrecargas importantes de formato y entrega.
- Para una mayor ampliación en horizontal, se pueden dividir las suscripciones en varias instancias de Push Services, cada una de las cuales funciona de forma independiente. Puesto que el modelo de aplicación crea suscripciones independientes entre sí, dicha partición es natural y hace que las aplicaciones de notificaciones sean muy escalables. Es necesario implementar la partición de suscripciones en la aplicación porque Push Services no lleva a cabo esta tarea automáticamente.

3.4.2.4.4 .Disponibilidad

Oracle9iAS Wireless Unified Messaging brinda una plataforma unificada para la administración de correo electrónico, mensajes de voz y de fax en el altamente escalable y confiable Oracle Database Oracle9iAS Wireless.

Las características mejoradas de Oracle9iAS Wireless Unified Messaging incluyen el nuevo Oracle Web Access, que acopla un diseño de fácil uso con abundantes características que ofrecen al usuario una experiencia similar a la de un escritorio en un browser. Al apoyar estándares tales como Push-IMAP protocol, Oracle9iAS Wireless Unified Messaging permite el envío de correos electrónicos a una variedad de dispositivos inalámbricos de redes públicas, reduciendo el costo de despliegue de esta solución.

3.5. Análisis Comparativo.

Para el análisis comparativo de las herramientas seleccionadas tendremos encuentra los parámetros y la tabla de los valores.

3.5.1. Seguridad

| Parámetro | Herramientas | |
|--|-----------------------|---------------|
| | Notification Services | Push Services |
| Cifrado de datos | Si | Si |
| Protocolos Web seguros | Si | Si |
| Creación personalizada de instancias y eventos | Si | Si |
| Mecanismos de seguridad propios del manejador | Alto | Medio |
| Cuentas de usuario | Siempre | A veces |

Tabla III.2. Tabla de seguridad

Al analizar los parámetros de seguridad se encuentra que tanto Notification Services como Push Services poseen cifrado de datos así como protocolos seguros para el envío de notificaciones y creación de instancias y eventos; muy diferente en mecanismos de seguridad propios de la base de datos que en el caso de Notification Service se presenta con un mayor volumen que Push Service. Se denota que una de las vías de seguridad más importantes se encuentra en el acceso de usuarios por medio de cuentas que en el caso de la primera herramienta se presenta siempre que sea necesario el acceso, en cambio en la otra las cuentas de usuario no se presentan siempre.

3.5.2. Confiabilidad

| Parámetro | Herramientas | |
|--------------------------|-----------------------|---------------|
| | Notofication Services | Push Services |
| Servidor Solido | Alto | Alto |
| Procedimiento de eventos | Alto | Medio |
| Entrega confiable | Alto | Alto |
| Estabilidad del sistema | Alto | Medio |

Tabla III.3. Tabla de confiabilidad

En el análisis de confiabilidad se presentan las dos herramientas como servidores sólidos en un nivel alto de confiabilidad, por el lado de los procedimientos de

eventos Notification Services posee un mejor manejo que Push Services ya que el proveedor de eventos ofrece mejor rendimiento. En cuanto a la entrega confiable los dos tienen un índice alto de confiabilidad, y por la estabilidad del sistema la primera en mejor por poseer métodos de control de errores.

3.5.3. Escalabilidad

| Parámetro | Herramientas | |
|--------------------------|-----------------------|---------------|
| | Notofication Services | Push Services |
| Ejecución en paralelo | Si | No |
| Aplicación multiproceso | Si | Si |
| Servidores distintos | Si | No |
| Proveedores distribuidos | Si | Si |
| División de instancias | Si | No |
| Plataformas | Simple | Avanzado |

Tabla III.4. Tabla de escalabilidad

Para la escalabilidad los parámetros muestran que Notification Services posee ejecución en paralelo de las interfaces de administración mientras que Push Services no, ambas herramientas son aplicaciones multiprocesos lo que significa que pueden atender varias notificaciones a la vez, la primera herramienta puede tener su distribución en distintos servidores mientras que la otra no, ambas pueden tener proveedores distribuidos para facilitar la ejecución, las dos herramientas ejecutan división de instancias, con respecto a la plataforma Notification Services solo se puede ejecutar bajo un entorno Windows mientras que Push Services se puede ejecutar en cualquier plataforma dependiendo claro está del lenguaje de programación utilizado.

3.5.4. Compatibilidad

| Parámetro | Herramientas | |
|--------------------------|-----------------------|---------------|
| | Notofication Services | Push Services |
| Varios Dispositivos | Muchos | Muchos |
| Sistemas Operativos | Muchos | Muchos |
| Varios tipos de mensajes | Muchos | Muchos |

| | | |
|---------------------------------|--------|--------|
| Distintos protocolos de entrega | Muchos | Muchos |
|---------------------------------|--------|--------|

Tabla III.5. Tabla de compatibilidad

El análisis de compatibilidad muestra que las dos herramientas son iguales es decir compatibles a dispositivos, tipos de mensajes, protocolos de entrega y sistemas operativos de los dispositivos que se pueden entregar notificaciones

3.5.5. Disponibilidad

| Parámetro | Herramientas | |
|--------------------------------|-----------------------|---------------|
| | Notofication Services | Push Services |
| Control de errores de hardware | Si | Si |
| Tecnología de disponibilidad | Si | Si |
| Réplica transaccional | Si | Si |

Tabla III.6. Tabla de disponibilidad

La disponibilidad de las herramientas según el análisis desarrollado es el mismo, es decir, siempre estará disponible la aplicación para cualquier requerimiento del usuario, ya que poseen métodos de control y tecnología de disponibilidad.

3.5.6. Eficiencia

| Parámetro | Herramientas | |
|---------------------------|-----------------------|---------------|
| | Notofication Services | Push Services |
| Tiempos de respuesta | Altos | Altos |
| Volumen de trafico | Medio | Medio |
| Notificaciones familiares | Avanzado | Simple |

Tabla III.7. Tabla de eficiencia

3.6. Tabla General Comparativa

| Parámetro | | Herramientas | |
|-----------|---------------------------|-----------------------|---------------|
| | | Notofication Services | Push Services |
| Seguridad | Cifrado | 5 | 5 |
| | Protocolo | 5 | 5 |
| | Creación personalizada de | 5 | 5 |

| | | | |
|-----------------------|----------------------------------|------------|-----------|
| | instancias y eventos | | |
| | Mecanismos propios del manejador | 5 | 3 |
| | Cuentas de usuario | 5 | 2 |
| Confiabilidad | Servidor Solido | 5 | 5 |
| | Procedimiento de eventos | 5 | 3 |
| | Entrega confiable | 5 | 5 |
| | Estabilidad del sistema | 5 | 3 |
| Escalabilidad | Ejecución en paralelo | 5 | - |
| | Aplicación multiproceso | 5 | 5 |
| | Servidores distintos | 5 | - |
| | Proveedores distribuidos | 5 | 5 |
| | División de instancias | 5 | - |
| | Plataformas | 1 | 5 |
| Compatibilidad | Varios Dispositivos | 5 | 5 |
| | Sistemas Operativos | 5 | 5 |
| | Varios tipos de mensajes | 5 | 5 |
| | Distintos protocolos de entrega | 5 | 5 |
| Disponibilidad | Control de errores de hardware | 5 | 5 |
| | Tecnología de disponibilidad | 5 | 5 |
| | Réplica transaccional | 5 | 5 |
| Eficiencia | Tiempos de respuesta | 5 | 5 |
| | Volumen de trafico | 3 | 3 |
| | Notificaciones familiares | 5 | 1 |
| TOTAL | | 119 | 90 |

Tabla III.8. Tabla general comparativa

3.7. Gráficos de pruebas de stress para el estudio comparativo.

Para la realización de las pruebas de stress para la comparación de rendimientos y tiempos se utilizo as herramienta llamada **QACenter Enterprise Edition de Compuware**

QACenter Enterprise Edition analiza cómo se distribuye el tiempo y permite a los equipos de calidad tomar decisiones estudiadas acerca del cambio de prioridades, cuánto tiempo adicional es necesario o cuántos recursos adicionales se requieren para cumplir los plazos establecidos ideal para este estudio.

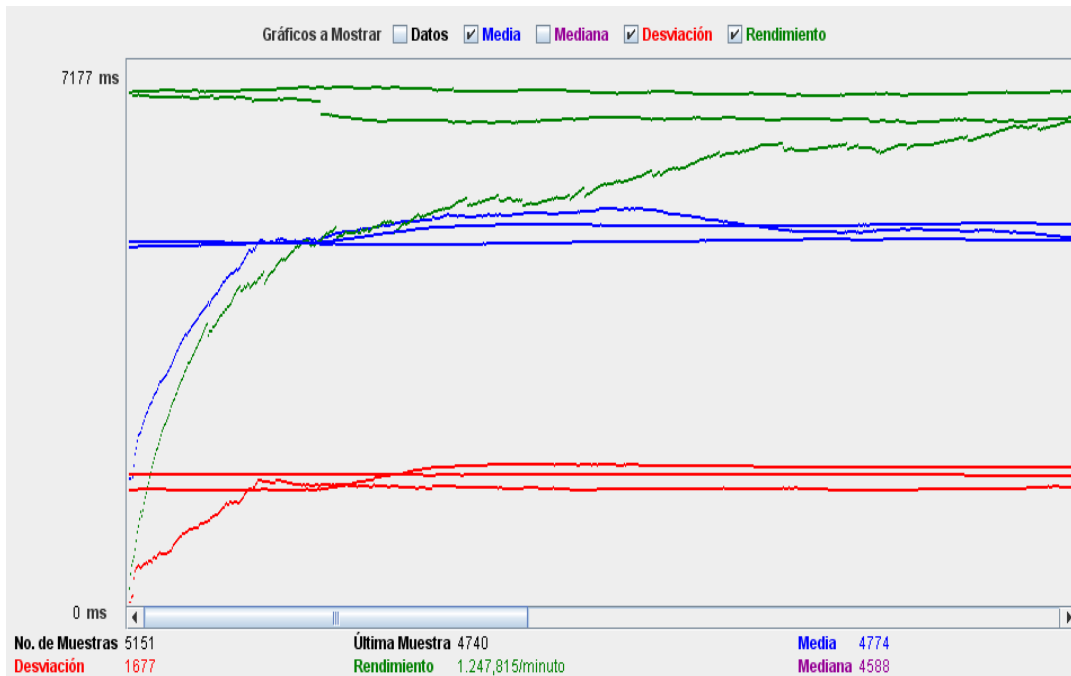


Figura III.11. Pruebas de Stress de Push Services



Figura III.12. Pruebas de Stress de Notification Services

3.8. Resultados de la Comparación

Notification Services proporciona un marco de programación que facilita y agiliza el desarrollo de aplicaciones de notificación profesionales e incluye además una plataforma para ejecutar dichas aplicaciones en un entorno eficaz y escalable según los resultados obtenidos en el análisis comparativo de las dos herramientas de SQL Server con Notification Services y de Oracle 9i con Push Services.

Por tal motivo hemos elegido para la parte practica la implementación de la herramienta de SQL Server Notification Services de Microsoft

CAPÍTULO IV

ESTUDIO DE LA INTERACCIÓN CON EL SISTEMA ACADÉMICO DE LA ESPOCH

4.1. Información contenida en la base de datos

4.1.1. Definición del pensum de estudios

La siguiente información denota la distribución de todas y cada una de las materias según la carrera, niveles de la Escuela Superior Politécnica de Chimborazo representados por los campos Código del pensum, nombre y estado.

4.1.2. Definición del periodo académico

Se contempla todos los periodos que han transcurrido desde la implantación del sistema existente identificado mediante un ID del periodo, descripción, fecha de inicio y finalización, fecha de tope de matrícula ordinaria, extraordinaria, provisional y de retiro, correspondiente al mismo.

4.1.3. Proceso de matriculación

En esta sección se almacena la información de cada matrícula de los estudiantes correspondiente a un periodo determinado, registrando código de matrícula, periodo, código del estudiante, nivel y estado.

4.1.4. Proceso de evaluación

Para el proceso de evaluación se tiene los siguientes campos para el almacenamiento de los estudiantes que tienen evaluación en las diferentes materias todo esto registrado en los siguientes campos, código del estudiante, fecha de la matrícula, periodo, nivel, código de matrícula, Tipo de matrícula, primera, segunda y tercera nota parcial, nota suspensión, promedio total, equivalencia y asistencia.

4.1.5. Proceso de egresamiento y graduación

Al haber cumplido con el pensum de estudios correspondiente los estudiantes pasan a un proceso de egresamiento y graduación para lo cual se almacena en la base de datos los siguientes campos para el egresamiento Código de egresamiento, código estudiante, código del título, promedio de notas, numero de créditos, fecha de aprobación, código de resolución, fecha de asentamiento. Para la graduación los siguientes campos código de graduación, código de egresamiento, código de proyecto de grado, numero de acta, promedio notas, nota de tesis escrita, nota de tesis oral, promedio de nota de grado, numero de créditos, fecha de grado.

4.1.6. Administración y seguridad

Para la utilización de los diferentes sistemas implantados en la politécnica los usuarios poseen un identificador de usuario de permisos con su respectiva contraseña para el ingreso a estos.

4.1.7. Estudiantes inscritos

Para que los estudiantes estén inscritos correctamente estos deben cumplir con unos requerimientos de los cuales se almacenan datos principales comprendidos en los siguientes campos código de inscripción, cedula del estudiante, código de la carrera, código del periodo, fecha de inscripción, confirmación de inscripción, código de formulario de inscripción y observaciones.

4.1.8. Estudiantes

Para una correcta información de los estudiantes se ha tomado como datos principales los siguientes para el almacenamiento en la base de datos código del estudiante, cedula, nombres, apellidos, cedula militar, fecha de nacimiento, fecha de ingreso, email, nacionalidad, documentación, sexo, código título, forma de inscripción.

4.1.9. Docentes

Para los docentes se recopila la siguiente información para la base de datos código de docente, cedula, nombres, apellidos, cedula militar, carnet del IESS, fecha de nacimiento, fecha de ingreso, dirección, teléfono, email, pagina WEB, foto, fecha de la última salida e ingreso, títulos, cargos, lugar de nacimiento, nacionalidad, código tipo de título, sexo, estado, código tipo docente, estado civil, código horas trabajo, código de categoría.

4.1.10. Institución

La institución cuenta con los siguientes datos de configuración código de la institución, nombre, rector, cedula del rector, vicerrector académico, cedula del vicerrector académico, vicerrector administrativo, cedula del vicerrector

administrativo, secretario académico, cedula secretario académico, dirección, teléfono y logo.

4.1.11. Facultades

Las diferentes facultades tienen la diferente información en la base de datos código de la facultad, nombre de la facultad, decano, cedula del decano, subdecano, cedula del subdecano, fecha de creación, ubicación y logo.

4.1.12. Escuelas

Las escuelas tienen la siguiente información para el almacenamiento de la base de datos, código de la escuela, nombre de la escuela, director, cedula del director, fecha de creación, ubicación y logo.

4.1.13. Carreras

Las escuelas tienen la siguiente información para el almacenamiento de la base de datos, código de la carrera, nombre de la carrera, código de estado, nombre de la base de datos, código de escuela, código del tipo de identidad.

4.1.14. Materias

Las materias tienen la siguiente información para el almacenamiento de la base de datos, código de materia, nombre de la materia, fecha de creación, fecha de eliminación y actividad.

4.1.15. Horario

Los horarios tienen la siguiente información para el almacenamiento de la base de datos, día y hora.

4.1.16. Niveles

Los niveles tienen la siguiente información para el almacenamiento de la base de datos, código del nivel y descripción.

4.2. Usuarios que interactúan en el Sistema Académico

4.2.1. Clientes Web

4.2.1.1. Público General

Mediante la página web de la Escuela Superior Politécnica de Chimborazo todo el público puede tener acceso a la información general de la conformación y organización de la universidad, así como a los diferentes sistemas que se encuentran implementados en el Departamento de Sistemas y Telemática.

4.2.1.2. Estudiante

Todos y cada uno de los estudiantes de la politécnica pueden acceder mediante un explorador de internet a todos los sistemas que les compete ya sea para información académica de sus notas como información personal.

4.2.1.3. Docente

Al igual que los estudiantes los docentes pueden acceder a los diversos sistemas que para su facilidad están desarrollados en la politécnica, todos estos utilitarios ayudan a los docentes en tareas tales como: ingreso de notas, información personal, material académico, organización de dictado de materias, etc.

4.2.1.4. Autoridad

Todas las entidades que componen la Escuela Superior Politécnica de Chimborazo tienen acceso a la tecnología desarrollada en la misma, este es el caso de las Autoridades que con sus respectivas credenciales politécnicas pueden ingresar a sus respectivos sistemas.

4.2.2. Interfaces en Win32

4.2.2.1. Secretaría Académica General y Secretaría de Unidades Académicas

Todo el sistema de control de evaluación está disponible para las secretarías tanto general como de facultad o unidades académicas, con lo cual se lleva a cabo el proceso de asignación de notas a cada uno de los estudiantes. El proceso se realiza mediante una aplicación de escritorio.

4.3. Funcionalidad del Sistema Académico.

4.3.1. OASis_BusinessServices

Es la fachada de un conjunto de servicios que proporcionarán los datos para los reportes y el soporte para todos los procesos que realice el Sitio Web.

4.3.2. Servicios Módulo de Inscripciones

Este proceso comprende aquellas actividades propias para la admisión de los aspirantes a la ESPOCH. El proceso incluye lo siguiente:

1. Inscripción de los aspirantes manualmente y/o mediante Internet. Manualmente el aspirante llena la Ficha de Inscripción en la ESPOCH, el otro caso se da cuando la ficha sea llenada por una página Web.
2. Riguroso seguimiento de los aspirantes, rastreando aquellos estudiantes que quieran ingresar como nuevos habiendo estado matriculados anteriormente.
3. Emitir reportes con posibilidad de exportar los mismos a diferentes formatos.
4. Control de cambio de Escuela/Carrera para los aspirantes ya inscritos.

4.3.3. Servicios Módulo de matriculación

Este proceso se realiza principalmente en cada una de las carreras, permitiendo a los estudiantes recibir las materias del semestre / año que le corresponda. Incluye lo siguiente:

1. Control de cambio de carrera haciendo cumplir el reglamento vigente permitiendo verificar si el estudiante no ha perdido un nivel con tercera matrícula.
2. Asignación de materias para los estudiantes que ya pertenecen a una carrera. Para los estudiantes sin arrastres y estudiantes que ingresan al Curso de Ajuste Básico pueden asignarse las materias mediante Internet.
3. Consulta de *horarios de clases* por estudiante, materia y nivel.
4. Emisión de *órdenes de pago* de matrícula.
5. Realización de un seguimiento de matrículas provisionales y registro de sus causas.
6. Emisión de la *Matrícula* del estudiante luego del pago del respectivo derecho.
7. Control del número de matrículas a nivel institucional, conociendo la trayectoria del estudiante por la ESPOCH.
8. Emisión de *Certificaciones de Matrícula*.
9. Control y seguimiento de los retiros de matrícula según el reglamento.
10. Seguimiento de los estudiantes que se retiran y que reingresan a las diferentes carreras de la ESPOCH.
11. Emisión de reportes para la Secretaría Académica General.
12. Emisión de reportes para las Secretarías de las Escuelas / Carreras

4.3.4. Servicios Módulo de Educación y Evaluación

Este proceso maneja las etapas propias de la educación y evaluación académica, realizando los controles y validaciones respectivas en cada una, esto es, desde que el estudiante ha tomado las materias hasta generar su historia académica. El proceso incluye lo siguiente:

1. Emisión de *Actas de Notas para exámenes principales* utilizando formatos pre impresos o con formatos generados por el sistema.
2. Ingreso de notas de evaluación acumulada, examen principal y porcentaje de asistencia.

3. Emisión de *Actas de Notas para exámenes de suspensión* utilizando formatos pre impresos o con formatos generados por el sistema.
4. Ingreso de notas de examen de suspensión.
5. Registro de exámenes atrasados (principal y/o) suspensión, recalificaciones y rectificación de notas. Emisión del acta correspondiente.
6. Registro de convalidaciones y exámenes de suficiencia, emisión de un acta de notas para asentar la nota en caso de exámenes de suficiencia.
7. Llevar una bitácora para el asentamiento y rectificaciones de notas como mecanismo de control y seguridad.
8. Publicación por Internet de notas individuales del estudiante.
9. Emisión de *Certificados de Promoción* de los estudiantes.
10. Emisión de *Récord Académicos* de los estudiantes
11. Generación de escalafones estudiantiles en orden descendente por el promedio de cada estudiante. Establecer el mejor estudiante por nivel y por Escuela / Carrera.
12. Permitir el ingreso de notas atrasadas después del cierre del período.

4.3.5. Servicios de Egresamiento y Graduación

Este proceso se produce una vez que el estudiante ha completado su pensum de estudios. Mediante el sistema se realiza el seguimiento y control del récord académico del estudiante para que se le acredite como egresado y en consecuencia pueda realizar los trámites pertinentes para su graduación. El proceso incluye lo siguiente:

1. Emisión del *Currículum Académico*.
2. Registro de datos de legalización del Egresamiento.
3. Registro de notas y fechas de evaluaciones escrita y oral de proyectos de graduación. (Tesis, memorias, etc.)
4. Registro de datos de legalización de la graduación.
5. Emisión de *actas de calificación* de graduación.
6. Emisión de *Actas de Grado* con numeración automática.
7. Determinación de mejores egresados por escuela/carrera.

8. Emisión de reportes para Secretaría Académica General.
9. Emisión de reportes para las Secretarías de las Escuelas / Carreras.

Estos servicios no pueden actuar por si solos en relación a la obtención de la información contenida en las bases de datos de cada uno de los procesos antes descritos, por lo que es necesaria la implementación de servicio de datos de los cuales citamos los siguientes:

- Servicios de Datos Módulo de Inscripciones
- Servicios de Datos Módulo de Matriculación
- Servicios de Datos Módulo de Evaluación
- Servicios de Datos Módulo de Egresamiento y Graduación
- Servicios de Datos Comunes

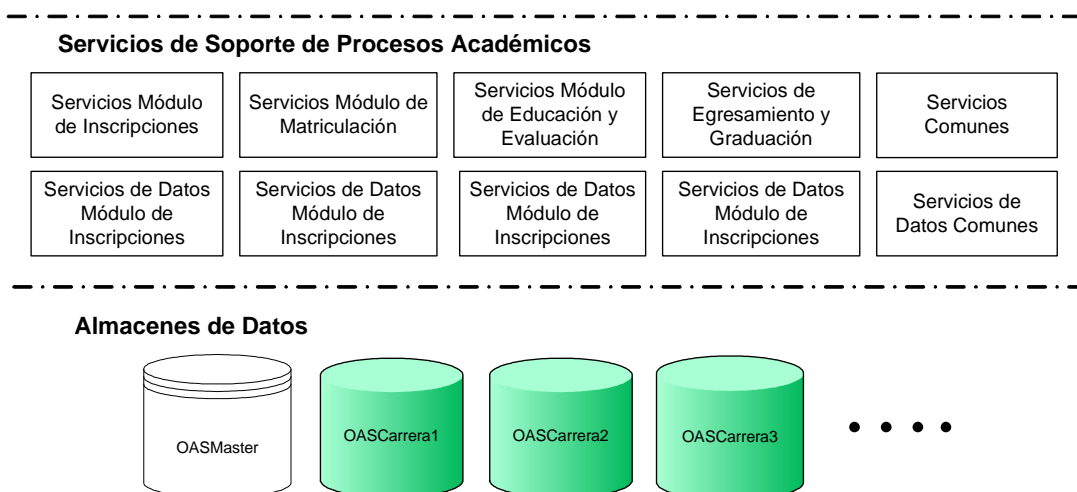


Figura IV.1. Módulos de OASIS

4.4. Selección de prestaciones del Sistema Académico para realizar notificaciones.

4.4.1. Almacenamiento de notas de todos los estudiantes

Para el almacenamiento de las notas se lo realiza en una base de datos Maestra la cual se divide en sub-bases de datos las cuales dependen de las carreras teniendo

en cuanto al estudiante y las materias en las cuales los alumnos se hayan matriculado.

4.4.2. Cambio de notas en un período determinado

Para el cambio de las notas de unas materias determinadas de alumno se debe realizar una modificación en la subbase correspondiente a la carrera que este se encuentre matriculado con las debidas autorizaciones que se deban tramitar para la realización de estos cambios.

4.4.3. Inscripciones y confirmación de inscripciones

Para que un alumno este correctamente inscrito este debe cumplir con varios requisitos los cuales se irán validando desacuerdo a lo que las diferentes carreras lo necesiten y al momento que estos sean cumplidos a la inscripción que este realice se le ha de confirmar su respectiva inscripción.

4.4.4. Matriculación y confirmación de matriculación

Para que un alumno quede totalmente matriculado, éste debe realizar un trámite pertinente, el cual confirme en qué materias y en qué carrera se inscribe correctamente para su matriculación, así al cumplir con este requisito, se le confirmará su matriculación.

CAPÍTULO V

DESARROLLO E IMPLEMENTACIÓN DEL SISTEMA DE NOTIFICACIONES

5.1. Fase de Análisis

Con los conceptos y definiciones sobre esta tesis, empezaremos con la implementación de la fase de Análisis. En dicha fase se ha definido el problema, sobre los conceptos relacionados con el subconjunto de casos de uso que se está tratando, ampliándolos con criterios que analizaremos en el ciclo de desarrollo actual.

5.1.1. Definir, refinar y diagramar los casos de uso esenciales en formato expandido

En esta sección se presentan casos de uso en formato expandido (nivel más detallado) y sus respectivos diagramas los mismos que nos servirán para establecer las iteraciones existentes entre los actores y el sistema propuesto.

CASOS DE USO

| | | |
|---|---|--------|
| Caso de Uso: | Operatividad del Sistema de Notificación de Académicas. | CUE_02 |
| Actores: | Sistema, Usuario | |
| Tipo: | Prioridad Alta | |
| Descripción: | Para la generación de la información lo primero que se tiene que esperar es la información que genere el sistema académico al realizar esta se deberá esperar la ejecución de eventos que estarán ya programados al terminar esta ejecución se verificaran las reglas y se escribirán las notificaciones para ser enviadas a los destinatarios pertinentes que estén suscritos y activados el servicio. | |
| Propósito | Generar la notificación a los clientes suscritos y activados el servicio | |
| Curso típico de eventos | | |
| Actores | Sistema | |
| 1. Generación de la información del sistema académico | 2. Ejecución de eventos | |
| | 3. Verificación de las reglas para la ejecución de eventos | |
| | 4. Generación de notificación para los usuarios registrados en el sistema de notificación académica | |
| | 5. Envío de mensajes a los celulares de los usuarios según la ejecución de los eventos | |
| 6. Recepción de los mensajes a los usuarios | | |

5.1.2. Definir y refinar los diagramas de secuencia

5.1.2.1. Operatividad del Servicio de Notificación del Sistema Académico.

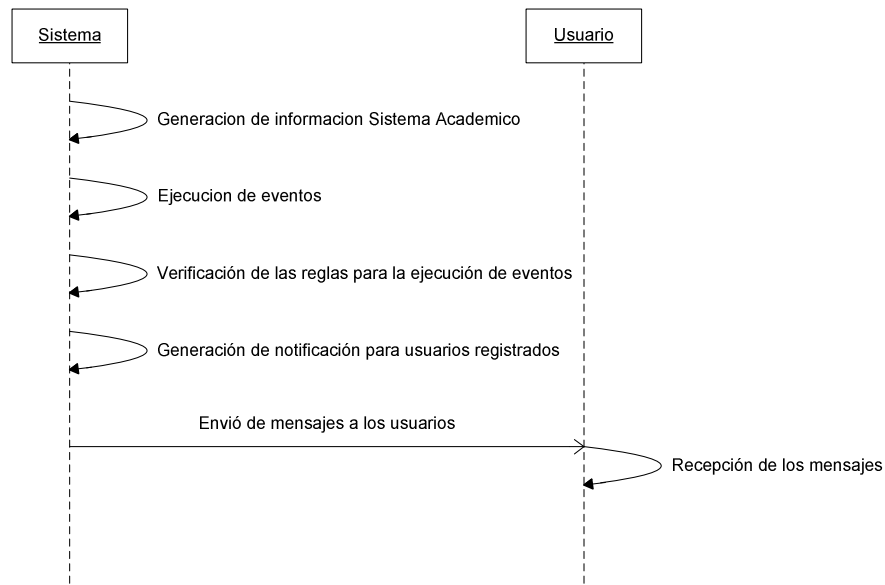


Figura V.1. Diagrama de Secuencia del Servicio de Notificaciones

5.1.3. Definir y refinar los diagramas de estado

5.1.3.1. Operatividad del Sistema de Notificación de Académicas.

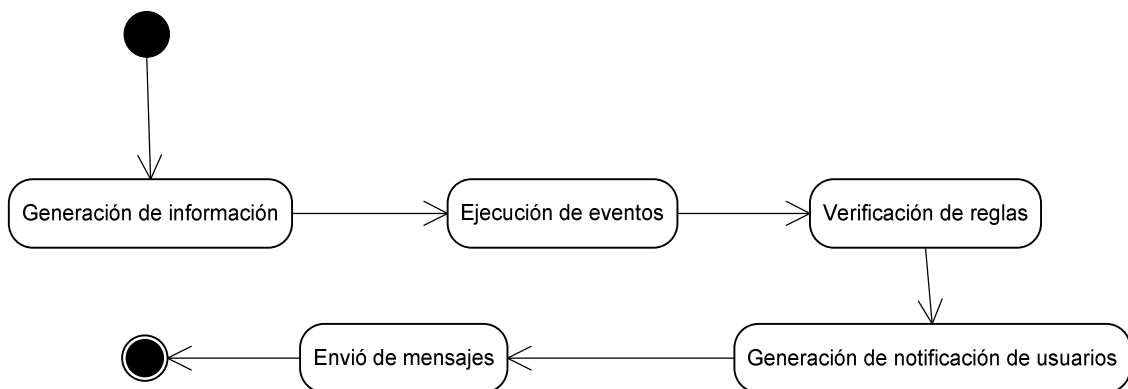


Figura V.2. Diagrama de Estado del Servicio de Notificaciones

5.2. Fase de Diseño

5.2.1. Refinar los Casos de Uso

A continuación se describe los casos de uso reales enfocados a la programación definiendo las variables y los tipos de datos que intervienen en los diferentes métodos a ser implementados.

5.2.1.1. Operatividad del Sistema de Notificación de Académicas.

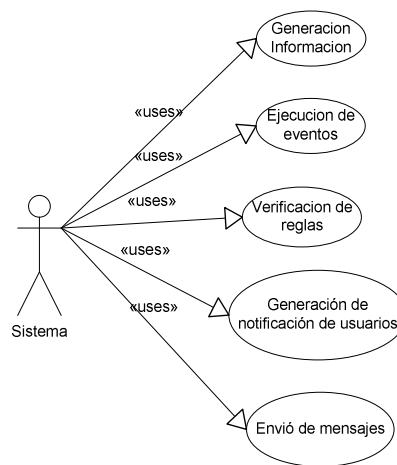


Figura V.3. Diagrama de Caso de Uso del Servicio de Notificaciones

5.2.2. Diseño de Base de Datos

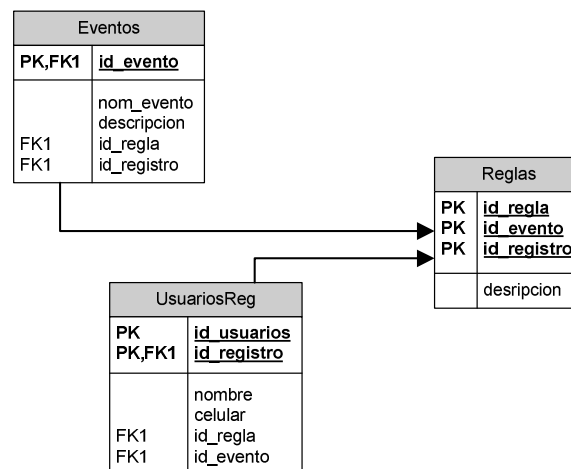


Figura V.4. Modelo Relacional de la Base de Datos.

5.2.3. Modelado Físico

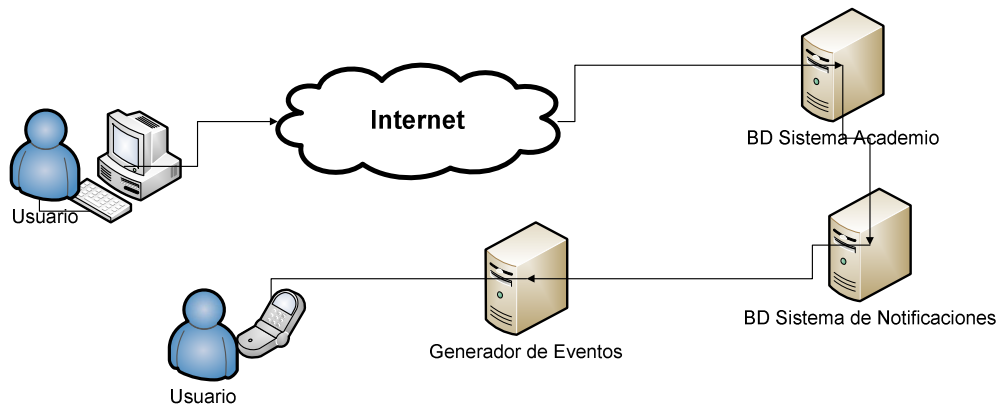


Figura V.5. Modelo Físico del Servicio de Notificación.

5.2.4. Diagrama de Componentes

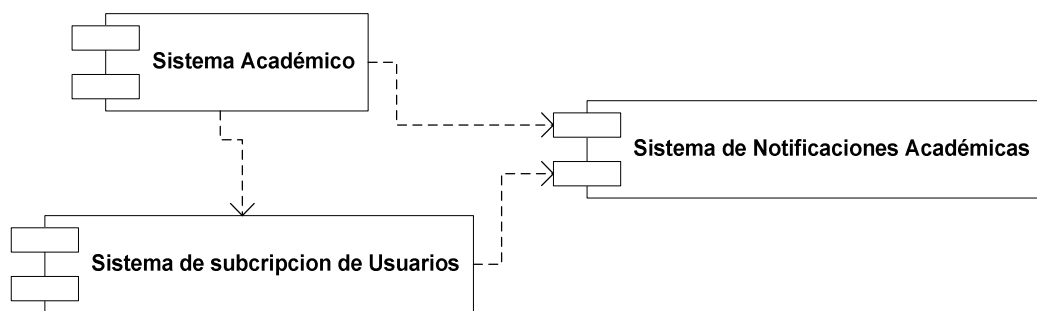


Figura V.6. Diagrama de Componentes del Servicio de Notificación.

5.3. FASE DE IMPLEMENTACIÓN

5.3.1. Implementación

Los siguientes procesos se dan al momento de ingresar una información determinada a la base de datos como son notas, inscripciones, matriculas, etc., esta información es conocida como suscripción, el usuario final que recibe la información es llamado, suscriptor y el mensaje que se recibe es la notificación.

En cada uno de los procesos se necesita generar un archivo de configuración, dicho archivo esta en formato XML, y posee la definición de la base de datos

necesaria para la notificación. En los anexos se adjunta toda la codificación de mencionados archivos en cada uno de los procesos.

Estudiantes:

- Obtención de las notas finales de los estudiantes cuando sean ingresadas al Sistema Académico.
- Confirmación de cambio de notas.
- Confirmación de inscripción y matriculación.
- Confirmación de cambio de escuela o convalidación de materias correspondientes a un período académico.
- Información de retiro de una materia o carrera.
- Información cuando el estudiante ya esté listo para incorporarse.
- Información de horarios de clases y exámenes.
- Cambio de contraseña.

Docentes:

- Información de carga horaria y materias a dictar.
- Información de la fecha de cierre de período académico.
- Cambio de contraseña.

CONCLUSIONES

- De acuerdo a los parámetros de comparación, se eligió la mejor herramienta para la implementación de la tesis, la cual es Microsoft Notification Services, la que brinda mejores prestaciones que las otras existentes en el mercado.
- La plataforma en donde reside el Sistema Académico de la ESPOCH, es totalmente compatible con la herramienta que se seleccionó en el estudio comparativo, siendo factible su utilización.
- La interacción de los estudiantes y docentes con el Sistema Académico de la ESPOCH, alcanza una nueva mejora, dando a sus clientes un método más sencillo y ágil de obtener la información en un tiempo determinado.

RECOMENDACIONES

- Para un mejor estudio, los parámetros de comparación que se deben escoger deben ser características fundamentales en las herramientas que serán parte del estudio comparativo; de esta forma el resultado se obtiene de una forma sencilla y veraz.
- Las notificaciones a dispositivos móviles vía SMS, presentan un costo mucho mayor a las que se realizan vía MAIL, ya que depende del costo que las operadoras celulares ofrecen para este servicio, tanto en el envío como en la recepción del mensaje. En el segundo caso, mediante el uso de la web, cualquier destinatario de correo electrónico recibirá su notificación sin costo alguno directamente a su cuenta de correo.
- Se debe definir correctamente el alcance de la solución con respecto a los eventos que son posibles implementar y enviar una notificación, ya que en algunos casos los eventos dependen de lapsos de tiempos que la base de datos no registra o no define explícitamente, dificultando el proceso de selección de información para la notificación.
- Para un correcto funcionamiento del proceso de notificación, todos y cada uno de los estudiantes y docentes inmersos en la solución, deberán tener registrada su dirección mail o número de dispositivo móvil, ya que de esto depende que la notificación sea entregada.

RESUMEN

Se realizó el estudio comparativo de las herramientas de notificación de eventos vía mensajería móvil aplicado al Sistema Académico de la Escuela Superior Politécnica de Chimborazo, con el objetivo de mejorar la interacción de los estudiantes y docentes con el sistema.

Se utilizó SQL Server 2005 como manejador de base de datos, Notification Services y Push Services como herramientas de notificación y XML como archivos de configuración. Se configura el manejador de eventos en el XML, se verifica un evento en la base de datos, se envía una notificación al suscriptor de acuerdo a las reglas establecidas para la suscripción y el evento que generó la notificación; busca en la base de datos el dispositivo y dirección registrados, arma el mensaje con los datos del evento y envía la notificación a su destinatario.

Se midió en una escala del uno a cinco los parámetros y sub-parámetros de seguridad, escalabilidad, confiabilidad, compatibilidad, disponibilidad y eficiencia de las herramientas, para cada uno de estos parámetros se obtuvo un valor de acuerdo a las características de cada herramienta, obteniendo que para Notification Services el valor fue de 119 puntos, mientras que para Push Services fue de 90, descubriendo que la primera opción era más efectiva en el proceso.

Se concluyó que la mejor herramienta de notificaciones es Notification Services y los mensajes que generó de forma rápida y automática ayudaron a la interacción de los estudiantes y docentes con el sistema académico, por lo que se recomienda la implementación del sistema de notificaciones.

SUMMARY

It was made the comparative study of events notification tools via mobile messaging applied to Academic System of ESPOCH, with the goal of improve the interaction among students, teachers and the system.

It was used SQL Server 2005 as database manager, Notification Services and Push Services as notification tools and XML as configuration files. It is configured the event manager in the XML file, It is verified a event in database, it is sent a notification to the subscriber according to established rules for the subscription and the event that generate the notification; it searches in database the registered device and address, builds the message with data from event and sends the notification to the recipient.

It was measured in a scale of one to five parameter and sub-parameter of security, scalability, reliability, compatibility, availability and efficiency of tools, for each parameters it was gotten a value according to the tool settings, getting for Notification Services a score of 119 points, meanwhile, for Push Services a score of 90, discovering that first option was the more effective in the process.

It was concluded that the best notification tool is Notification Services and messages generated in a fast and automatic way helped on the interaction among students, teachers and academic system, so the implementation of the notification system.

BIBLIOGRAFÍA

LIBROS

- 1.- ALBA, M. Supporting Handheld Collaboration through COMAL. Sixth International Workshop on Groupware, CRIWG'2000. Portugal: IEEE Computer Press, 2000. pp. 52-59.
- 2.- MORÁN, L. Document Presence Notification Services for Collaborative Writing. Seventh International Workshop on Groupware, CRIWG'2001. Germany: IEEE Computer Press, 2001. pp. 125-133.
- 3.- PATHER, S. Microsoft SQL Server 2005 Notification Services. USA: Sams, 2006. pp. 9-77, 117-160, 393-415
- 4.- WEBB, J. The Rational Guide to SQL Server Notification Services. USA: Mann Publishing Group, 2004. pp. 45-63, 65-75, 77-87, 89-102, 157-171.
- 5.- YEUNG, A. Oracle9i Mobile. USA: McGraw-Hill/Osborne, 2002. pp. 33-53, 107-132.

DIRECCIONES WEB

- 1.- BROMBERG, P. Using SQL Server Notification Services.

<http://www.eggheadcafe.com/articles/20021122.asp>

20080115

- 2.- C# ONLINE.NET. Notification Services

http://en.csharp-online.net/Notification_Services

20080606

3.- MICROSOFT. Implementar Notification Services.

<http://msdn.microsoft.com/es-es/library/ms171302%28SQL.90%29.aspx>

20070822

4.- ORACLE. Creating Messaging Applications.

http://download.oracle.com/docs/cd/B14099_19/wireless.1012/b13819/push.htm

20080219

5.- ORACLE. Oracle Mobile Push Mail, Troubleshooting and FAQs.

http://www.oracle.com/technology/products/cs/user_info/omobile_collaboration/mobile_push_mail_index.html

20080121

6.- ROSEN, C. Concur, Oracle Push B2B Portals.

<http://www.allbusiness.com/transportation-communications/transportation-services/4150810-1.html>

20080115

7.- WALSH, N. A Technical Introduction to XML.

<http://www.xml.com/pub/a/98/10/guide0.html>

20070822

ANEXOS

- Obtención de las notas finales de los estudiantes cuando sean ingresadas al Sistema Académico.

```

<?xml version="1.0" encoding="utf-8" ?>
<Application xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xmlns="http://www.microsoft.com/MicrosoftNotificationServices/ApplicationD
efinitionFileSchema">

  <!-- Version -->

  <!-- Database Definition -->

  <!-- Event Classes -->
<EventClasses>
  <EventClass>
    <EventClassName>NotasData</EventClassName>
    <Schema>
      <Field>
        <FieldName>Materia</FieldName>
        <FieldType>nvarchar(35)</FieldType>
        <FieldTypeMods>not null</FieldTypeMods>
      </Field>
      <Field>
        <FieldName>Fecha</FieldName>
        <FieldType>datetime</FieldType>
        <FieldTypeMods>not null</FieldTypeMods>
      </Field>
      <Field>
        <FieldName>Acumulado</FieldName>
        <FieldType>float</FieldType>
        <FieldTypeMods>not null</FieldTypeMods>
      </Field>
      <Field>
        <FieldName>Final</FieldName>
        <FieldType>float</FieldType>
        <FieldTypeMods>not null</FieldTypeMods>
      </Field>
      <Field>
        <FieldName>Descripcion</FieldName>
        <FieldType>nvarchar(3500)</FieldType>
        <FieldTypeMods>not null</FieldTypeMods>
      </Field>
    </Schema>
    <IndexSqlSchema>
      <SqlStatement>
        CREATE INDEX myIndex
        ON NotasData ( Materia );
      </SqlStatement>

```

```

    </IndexSqlSchema>
  </EventClass>
</EventClasses>

<!-- Subscription Classes -->
<SubscriptionClasses>
  <SubscriptionClass>
    <SubscriptionClassName>NotasMateria</SubscriptionClassName>
    <Schema>
      <Field>
        <FieldName>DeviceName</FieldName>
        <FieldType>nvarchar(255)</FieldType>
        <FieldTypeMods>not null</FieldTypeMods>
      </Field>
      <Field>
        <FieldName>SubscriberLocale</FieldName>
        <FieldType>nvarchar(10)</FieldType>
        <FieldTypeMods>not null</FieldTypeMods>
      </Field>
      <Field>
        <FieldName>Materia</FieldName>
        <FieldType>nvarchar(40)</FieldType>
        <FieldTypeMods>not null</FieldTypeMods>
      </Field>
    </Schema>
    <EventRules>
      <EventRule>
        <RuleName>NotasEventRule</RuleName>
        <EventClassName>NotasData</EventClassName>
        <Action>
          INSERT INTO NotasAlerts(SubscriberId,
            DeviceName, SubscriberLocale, Materia, Fecha, Acumulado, Final,
            Descripcion)
          SELECT s.SubscriberId, s.DeviceName, s.SubscriberLocale,
            e.Materia, e.Fecha, e.Acumulado, e.Final, e.Descripcion
          FROM NotasData e, NotasMateria s
          WHERE e.Materia = s.Materia;
        </Action>
      </EventRule>
    </EventRules>
  </SubscriptionClass>
</SubscriptionClasses>

<!-- Notification Classes -->
<NotificationClasses>
  <NotificationClass>
    <NotificationClassName>
      NotasAlerts
    </NotificationClassName>
    <Schema>
      <Fields>
        <Field>
          <FieldName>Materia</FieldName>

```

```

    <FieldType>nvarchar(35)</FieldType>
</Field>
<Field>
    <FieldName>Fecha</FieldName>
    <FieldType>datetime</FieldType>
</Field>
<Field>
    <FieldName>Acumulado</FieldName>
    <FieldType>float</FieldType>
</Field>
<Field>
    <FieldName>Final</FieldName>
    <FieldType>float</FieldType>
</Field>
<Field>
    <FieldName>Descripcion</FieldName>
    <FieldType>nvarchar(3500)</FieldType>
</Field>
</Fields>
</Schema>
<ContentFormatter>
    <ClassName>XsltFormatter</ClassName>
    <Arguments>
        <Argument>
            <Name>XsltBaseDirectoryPath</Name>
            <Value>%_AppPath_%</Value>
        </Argument>
        <Argument>
            <Name>XsltFileName</Name>
            <Value>NotasTransform.xslt</Value>
        </Argument>
    </Arguments>
</ContentFormatter>
<Protocols>
    <Protocol>
        <ProtocolName>File</ProtocolName>
    </Protocol>
    <Protocol>
        <ProtocolName>SMTP</ProtocolName>
    </Protocol>
</Protocols>
<Fields>
    <Field>
        <FieldName>Subject</FieldName>
        <SqlExpression>'Notas'</SqlExpression>
    </Field>
    <Field>
        <FieldName>BodyFormat</FieldName>
        <SqlExpression>'html'</SqlExpression>
    </Field>
    <Field>
        <FieldName>From</FieldName>
        <SqlExpression>'sender@francisco'</SqlExpression>
    </Field>
    <Field>

```

```

        <FieldName>Priority</FieldName>
        <SqlExpression>'Normal'</SqlExpression>
    </Field>
    <Field>
        <FieldName>To</FieldName>
        <SqlExpression>DeviceAddress</SqlExpression>
    </Field>
</Fields>
</Protocol>
</Protocols>
</NotificationClass>
</NotificationClasses>

<!-- Event Providers -->
<Providers>
    <NonHostedProvider>
        <ProviderName>NotasSPEventProvider</ProviderName>
    </NonHostedProvider>
</Providers>

<!-- Generator -->
<Generator>
    <SystemName>%_NSServer_%</SystemName>
</Generator>

<!-- Distributors -->
<Distributors>
    <Distributor>
        <SystemName>%_NSServer_%</SystemName>
        <QuantumDuration>PT15S</QuantumDuration>
    </Distributor>
</Distributors>

<!-- ApplicationExecutionSettings -->
<ApplicationExecutionSettings>
    <QuantumDuration>PT15S</QuantumDuration>
    <DistributorLogging>
        <LogBeforeDeliveryAttempts>>false</LogBeforeDeliveryAttempts>
        <LogStatusInfo>>false</LogStatusInfo>
        <LogNotificationText>>false</LogNotificationText>
    </DistributorLogging>
    <Vacuum>
        <RetentionAge>P1D</RetentionAge>
        <VacuumSchedule>
            <Schedule>
                <StartTime>23:00:00</StartTime>
                <Duration>P0DT02H00M00S</Duration>
            </Schedule>
        </VacuumSchedule>
    </Vacuum>
</ApplicationExecutionSettings>

</Application>

```

- Confirmación de cambio de notas.

```
<?xml version="1.0" encoding="utf-8" ?>
<Application xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
```

```
  xmlns="http://www.microsoft.com/MicrosoftNotificationServices/ApplicationDefin
  itionFileSchema">
```

```
<!-- Version -->
```

```
<!-- Database Definition -->
```

```
<!-- Event Classes -->
```

```
<EventClasses>
```

```
<EventClass>
```

```
<EventClassName>NotasData</EventClassName>
```

```
<Schema>
```

```
<Field>
```

```
<FieldName>Materia</FieldName>
```

```
<FieldType>nvarchar(35)</FieldType>
```

```
<FieldTypeMods>not null</FieldTypeMods>
```

```
</Field>
```

```
<Field>
```

```
<FieldName>Fecha</FieldName>
```

```
<FieldType>datetime</FieldType>
```

```
<FieldTypeMods>not null</FieldTypeMods>
```

```
</Field>
```

```
<Field>
```

```
<FieldName>Acumulado</FieldName>
```

```
<FieldType>float</FieldType>
```

```
<FieldTypeMods>not null</FieldTypeMods>
```

```
</Field>
```

```
<Field>
```

```
<FieldName>Final</FieldName>
```

```
<FieldType>float</FieldType>
```

```
<FieldTypeMods>not null</FieldTypeMods>
```

```
</Field>
```

```
<Field>
```

```
<FieldName>Descripcion</FieldName>
```

```
<FieldType>nvarchar(3500)</FieldType>
```

```
<FieldTypeMods>not null</FieldTypeMods>
```

```
</Field>
```

```
</Schema>
```

```
<IndexSqlSchema>
```

```
<SqlStatement>
```

```
CREATE INDEX myIndex
```

```
ON NotasData ( Materia );
```

```
</SqlStatement>
```

```
</IndexSqlSchema>
```

```
</EventClass>
```

```

</EventClasses>

<!-- Subscription Classes -->
<SubscriptionClasses>
  <SubscriptionClass>
    <SubscriptionClassName>NotasMateria</SubscriptionClassName>
    <Schema>
      <Field>
        <FieldName>DeviceName</FieldName>
        <FieldType>nvarchar(255)</FieldType>
        <FieldTypeMods>not null</FieldTypeMods>
      </Field>
      <Field>
        <FieldName>SubscriberLocale</FieldName>
        <FieldType>nvarchar(10)</FieldType>
        <FieldTypeMods>not null</FieldTypeMods>
      </Field>
      <Field>
        <FieldName>Materia</FieldName>
        <FieldType>nvarchar(40)</FieldType>
        <FieldTypeMods>not null</FieldTypeMods>
      </Field>
    </Schema>
    <EventRules>
      <EventRule>
        <RuleName>NotasEventRule</RuleName>
        <EventClassName>NotasData</EventClassName>
        <Action>
          INSERT INTO NotasAlerts(SubscriberId,
            DeviceName, SubscriberLocale, Materia, Fecha, Acumulado, Final,
            Descripcion)
          SELECT s.SubscriberId, s.DeviceName, s.SubscriberLocale,
            e.Materia, e.Fecha, e.Acumulado, e.Final, e.Descripcion
          FROM NotasData e, NotasMateria s
          WHERE e.Materia = s.Materia;
        </Action>
      </EventRule>
    </EventRules>
  </SubscriptionClass>
</SubscriptionClasses>

<!-- Notification Classes -->
<NotificationClasses>
  <NotificationClass>
    <NotificationClassName>
      NotasAlerts
    </NotificationClassName>
    <Schema>
      <Fields>
        <Field>
          <FieldName>Materia</FieldName>
          <FieldType>nvarchar(35)</FieldType>
        </Field>
      </Fields>
    </Schema>
  </NotificationClass>
</NotificationClasses>

```

```

<Field>
  <FieldName>Fecha</FieldName>
  <FieldType>datetime</FieldType>
</Field>
<Field>
  <FieldName>Acumulado</FieldName>
  <FieldType>float</FieldType>
</Field>
<Field>
  <FieldName>Final</FieldName>
  <FieldType>float</FieldType>
</Field>
<Field>
  <FieldName>Descripcion</FieldName>
  <FieldType>nvarchar(3500)</FieldType>
</Field>
</Fields>
</Schema>
<ContentFormatter>
  <ClassName>XsltFormatter</ClassName>
  <Arguments>
    <Argument>
      <Name>XsltBaseDirectoryPath</Name>
      <Value>%_AppPath_%</Value>
    </Argument>
    <Argument>
      <Name>XsltFileName</Name>
      <Value>NotasTransform.xslt</Value>
    </Argument>
  </Arguments>
</ContentFormatter>
<Protocols>
  <Protocol>
    <ProtocolName>File</ProtocolName>
  </Protocol>
  <Protocol>
    <ProtocolName>SMTP</ProtocolName>
    <Fields>
      <Field>
        <FieldName>Subject</FieldName>
        <SqlExpression>'Notas'</SqlExpression>
      </Field>
      <Field>
        <FieldName>BodyFormat</FieldName>
        <SqlExpression>'html'</SqlExpression>
      </Field>
      <Field>
        <FieldName>From</FieldName>
        <SqlExpression>'sender@francisco'</SqlExpression>
      </Field>
      <Field>
        <FieldName>Priority</FieldName>
        <SqlExpression>'Normal'</SqlExpression>
      </Field>
    </Fields>
  </Protocol>
</Protocols>

```

```

        </Field>
        <Field>
            <FieldName>To</FieldName>
            <SqlExpression>DeviceAddress</SqlExpression>
        </Field>
    </Fields>
</Protocol>
</Protocols>
</NotificationClass>
</NotificationClasses>

<!-- Event Providers -->
<Providers>
    <NonHostedProvider>
        <ProviderName>NotasSPEventProvider</ProviderName>
    </NonHostedProvider>
</Providers>

<!-- Generator -->
<Generator>
    <SystemName>%_NSServer_%</SystemName>
</Generator>

<!-- Distributors -->
<Distributors>
    <Distributor>
        <SystemName>%_NSServer_%</SystemName>
        <QuantumDuration>PT15S</QuantumDuration>
    </Distributor>
</Distributors>

<!-- ApplicationExecutionSettings -->
<ApplicationExecutionSettings>
    <QuantumDuration>PT15S</QuantumDuration>
    <DistributorLogging>
        <LogBeforeDeliveryAttempts>>false</LogBeforeDeliveryAttempts>
        <LogStatusInfo>>false</LogStatusInfo>
        <LogNotificationText>>false</LogNotificationText>
    </DistributorLogging>
    <Vacuum>
        <RetentionAge>P1D</RetentionAge>
        <VacuumSchedule>
            <Schedule>
                <StartTime>23:00:00</StartTime>
                <Duration>P0DT02H00M00S</Duration>
            </Schedule>
        </VacuumSchedule>
    </Vacuum>
</ApplicationExecutionSettings>

</Application>

```


- Confirmación de inscripción.

```
<?xml version="1.0" encoding="utf-8" ?>
<Application xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
```

```
  xmlns="http://www.microsoft.com/MicrosoftNotificationServices/ApplicationDefin
  itionFileSchema">
```

```
<!-- Version -->
```

```
<!-- Database Definition -->
```

```
<!-- Event Classes -->
```

```
<EventClasses>
```

```
<EventClass>
```

```
<EventClassName>InscripcionData</EventClassName>
```

```
<Schema>
```

```
<Field>
```

```
<FieldName>CodigoAlumno</FieldName>
```

```
<FieldType>varchar(13)</FieldType>
```

```
<FieldTypeMods>not null</FieldTypeMods>
```

```
</Field>
```

```
<Field>
```

```
<FieldName>Escuela</FieldName>
```

```
<FieldType>nvarchar(100)</FieldType>
```

```
<FieldTypeMods>not null</FieldTypeMods>
```

```
</Field>
```

```
<Field>
```

```
<FieldName>Periodo</FieldName>
```

```
<FieldType>nvarchar(100)</FieldType>
```

```
<FieldTypeMods>not null</FieldTypeMods>
```

```
</Field>
```

```
<Field>
```

```
<FieldName>Fecha</FieldName>
```

```
<FieldType>datetime</FieldType>
```

```
<FieldTypeMods>not null</FieldTypeMods>
```

```
</Field>
```

```
</Schema>
```

```
<IndexSqlSchema>
```

```
<SqlStatement>
```

```
CREATE INDEX myIndex
```

```
ON InscripcionData ( CodigoAlumno );
```

```
</SqlStatement>
```

```
</IndexSqlSchema>
```

```
</EventClass>
```

```
</EventClasses>
```

```
<!-- Subscription Classes -->
```

```
<SubscriptionClasses>
```

```
<SubscriptionClass>
```

```

<SubscriptionClassName>InscripcionAlumno</SubscriptionClassName>
<Schema>
  <Field>
    <FieldName>DeviceName</FieldName>
    <FieldType>nvarchar(255)</FieldType>
    <FieldTypeMods>not null</FieldTypeMods>
  </Field>
  <Field>
    <FieldName>SubscriberLocale</FieldName>
    <FieldType>nvarchar(10)</FieldType>
    <FieldTypeMods>not null</FieldTypeMods>
  </Field>
  <Field>
    <FieldName>CodigoAlumno</FieldName>
    <FieldType>nvarchar(13)</FieldType>
    <FieldTypeMods>not null</FieldTypeMods>
  </Field>
</Schema>
<EventRules>
  <EventRule>
    <RuleName>InscripcionEventRule</RuleName>
    <EventClassName>InscripcionData</EventClassName>
    <Action>
      INSERT INTO InscripcionAlerts(SubscriberId,
      DeviceName, SubscriberLocale, Escuela, Periodo, Fecha)
      SELECT s.SubscriberId, s.DeviceName, s.SubscriberLocale,
      e.Escuela, e.Periodo, e.Fecha
      FROM InscripcionData e, InscripcionAlumno s
      WHERE e.CodigoAlumno = s.CodigoAlumno;
    </Action>
  </EventRule>
</EventRules>
</SubscriptionClass>
</SubscriptionClasses>

```

```

<!-- Notification Classes -->

```

```

<NotificationClasses>
  <NotificationClass>
    <NotificationClassName>
      InscripcionAlerts
    </NotificationClassName>
    <Schema>
      <Fields>
        <Field>
          <FieldName>Escuela</FieldName>
          <FieldType>nvarchar(100)</FieldType>
        </Field>
        <Field>
          <FieldName>Periodo</FieldName>
          <FieldType>nvarchar(100)</FieldType>
        </Field>
        <Field>
          <FieldName>Fecha</FieldName>

```

```

        <FieldType>datetime</FieldType>
      </Field>
    </Fields>
  </Schema>
  <ContentFormatter>
    <ClassName>XsltFormatter</ClassName>
    <Arguments>
      <Argument>
        <Name>XsltBaseDirectoryPath</Name>
        <Value>%_AppPath_%</Value>
      </Argument>
      <Argument>
        <Name>XsltFileName</Name>
        <Value>InscripcionTransform.xslt</Value>
      </Argument>
    </Arguments>
  </ContentFormatter>
  <Protocols>
    <Protocol>
      <ProtocolName>File</ProtocolName>
    </Protocol>
    <Protocol>
      <ProtocolName>SMTP</ProtocolName>
      <Fields>
        <Field>
          <FieldName>Subject</FieldName>
          <SqlExpression>'Confirmacion
Inscripcion'</SqlExpression>
        </Field>
        <Field>
          <FieldName>BodyFormat</FieldName>
          <SqlExpression>'html'</SqlExpression>
        </Field>
        <Field>
          <FieldName>From</FieldName>
          <SqlExpression>'webmaster@epoch.edu.ec'</SqlExpression>
        </Field>
        <Field>
          <FieldName>Priority</FieldName>
          <SqlExpression>'Normal'</SqlExpression>
        </Field>
        <Field>
          <FieldName>To</FieldName>
          <SqlExpression>DeviceAddress</SqlExpression>
        </Field>
      </Fields>
    </Protocol>
  </Protocols>
</NotificationClass>
</NotificationClasses>

<!-- Event Providers -->

```

```

<Providers>
  <NonHostedProvider>
    <ProviderName>InscripcionSPEventProvider</ProviderName>
  </NonHostedProvider>
</Providers>

<!-- Generator -->
<Generator>
  <SystemName>%_NSServer_%</SystemName>
</Generator>

<!-- Distributors -->
<Distributors>
  <Distributor>
    <SystemName>%_NSServer_%</SystemName>
    <QuantumDuration>PT15S</QuantumDuration>
  </Distributor>
</Distributors>

<!-- ApplicationExecutionSettings -->
<ApplicationExecutionSettings>
  <QuantumDuration>PT15S</QuantumDuration>
  <DistributorLogging>
    <LogBeforeDeliveryAttempts>>false</LogBeforeDeliveryAttempts>
    <LogStatusInfo>>false</LogStatusInfo>
    <LogNotificationText>>false</LogNotificationText>
  </DistributorLogging>
  <Vacuum>
    <RetentionAge>P1D</RetentionAge>
    <VacuumSchedule>
      <Schedule>
        <StartTime>23:00:00</StartTime>
        <Duration>P0DT02H00M00S</Duration>
      </Schedule>
    </VacuumSchedule>
  </Vacuum>
</ApplicationExecutionSettings>

</Application>

```

- Confirmación de matriculación.

```

<?xml version="1.0" encoding="utf-8" ?>
<Application xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

  xmlns="http://www.microsoft.com/MicrosoftNotificationServices/ApplicationD
efinitionFileSchema">

  <!-- Version -->

```

<!-- Database Definition -->

<!-- Event Classes -->

<EventClasses>

<EventClass>

<EventClassName>MatriculacionesData</EventClassName>

<Schema>

<Field>

<FieldName>Semestre</FieldName>

<FieldType>nvarchar(35)</FieldType>

<FieldTypeMods>not null</FieldTypeMods>

</Field>

<Field>

<FieldName>Fecha</FieldName>

<FieldType>datetime</FieldType>

<FieldTypeMods>not null</FieldTypeMods>

</Field>

<Field>

<FieldName>Descripcion</FieldName>

<FieldType>nvarchar(3500)</FieldType>

<FieldTypeMods>not null</FieldTypeMods>

</Field>

</Schema>

<IndexSqlSchema>

<SqlStatement>

CREATE INDEX myIndex

ON MatriculacionesData (Semestre);

</SqlStatement>

</IndexSqlSchema>

</EventClass>

</EventClasses>

<!-- Subscription Classes -->

<SubscriptionClasses>

<SubscriptionClass>

<SubscriptionClassName>MatriculacionesSemestre</SubscriptionClassName>

<Schema>

<Field>

<FieldName>DeviceName</FieldName>

<FieldType>nvarchar(255)</FieldType>

<FieldTypeMods>not null</FieldTypeMods>

</Field>

<Field>

<FieldName>SubscriberLocale</FieldName>

<FieldType>nvarchar(10)</FieldType>

<FieldTypeMods>not null</FieldTypeMods>

</Field>

<Field>

<FieldName>Semestre</FieldName>

<FieldType>nvarchar(40)</FieldType>

<FieldTypeMods>not null</FieldTypeMods>

```

</Field>
</Schema>
<EventRules>
  <EventRule>
    <RuleName>MEventRule</RuleName>
    <EventClassName>MatriculacionesData</EventClassName>
    <Action>
      INSERT INTO MatriculacionesAlerts(SubscriberId,
      DeviceName, SubscriberLocale, Semestre, Fecha, Descripcion)
      SELECT s.SubscriberId, s.DeviceName, s.SubscriberLocale,
      e.Semestre, e.Fecha, e.Descripcion
      FROM MatriculacionesData e, MatriculacionesSemestre s
      WHERE e.Semestre = s.Semestre;
    </Action>
  </EventRule>
</EventRules>
</SubscriptionClass>
</SubscriptionClasses>

<!-- Notification Classes -->
<NotificationClasses>
  <NotificationClass>
    <NotificationClassName>
      MatriculacionesAlerts
    </NotificationClassName>
    <Schema>
      <Fields>
        <Field>
          <FieldName>Semestre</FieldName>
          <FieldType>nvarchar(35)</FieldType>
        </Field>
        <Field>
          <FieldName>Fecha</FieldName>
          <FieldType>datetime</FieldType>
        </Field>
        <Field>
          <FieldName>Descripcion</FieldName>
          <FieldType>nvarchar(3500)</FieldType>
        </Field>
      </Fields>
    </Schema>
    <ContentFormatter>
      <ClassName>XsltFormatter</ClassName>
      <Arguments>
        <Argument>
          <Name>XsltBaseDirectoryPath</Name>
          <Value>%_AppPath_%</Value>
        </Argument>
        <Argument>
          <Name>XsltFileName</Name>
          <Value>MatriculacionTransform.xslt</Value>
        </Argument>
      </Arguments>
    </ContentFormatter>
  </NotificationClass>
</NotificationClasses>

```

```

</ContentFormatter>
<Protocols>
  <Protocol>
    <ProtocolName>File</ProtocolName>
  </Protocol>
<Protocol>
  <ProtocolName>SMTP</ProtocolName>
<Fields>
  <Field>
    <FieldName>Subject</FieldName>
    <SqlExpression>'Matricula'</SqlExpression>
  </Field>
  <Field>
    <FieldName>BodyFormat</FieldName>
    <SqlExpression>'html'</SqlExpression>
  </Field>
  <Field>
    <FieldName>From</FieldName>
    <SqlExpression>'sender@francisco'</SqlExpression>
  </Field>
  <Field>
    <FieldName>Priority</FieldName>
    <SqlExpression>'Normal'</SqlExpression>
  </Field>
  <Field>
    <FieldName>To</FieldName>
    <SqlExpression>DeviceAddress</SqlExpression>
  </Field>
</Fields>
</Protocol>
</Protocols>
</NotificationClass>
</NotificationClasses>

<!-- Event Providers -->
<Providers>
  <NonHostedProvider>
    <ProviderName>MatriculacionSPEventProvider</ProviderName>
  </NonHostedProvider>
</Providers>

<!-- Generator -->
<Generator>
  <SystemName>%_NSServer_%</SystemName>
</Generator>

<!-- Distributors -->
<Distributors>
  <Distributor>
    <SystemName>%_NSServer_%</SystemName>
    <QuantumDuration>PT15S</QuantumDuration>
  </Distributor>
</Distributors>

```

```

<!-- ApplicationExecutionSettings -->
<ApplicationExecutionSettings>
  <QuantumDuration>PT15S</QuantumDuration>
  <DistributorLogging>
    <LogBeforeDeliveryAttempts>>false</LogBeforeDeliveryAttempts>
    <LogStatusInfo>>false</LogStatusInfo>
    <LogNotificationText>>false</LogNotificationText>
  </DistributorLogging>
  <Vacuum>
    <RetentionAge>P1D</RetentionAge>
    <VacuumSchedule>
      <Schedule>
        <StartTime>23:00:00</StartTime>
        <Duration>P0DT02H00M00S</Duration>
      </Schedule>
    </VacuumSchedule>
  </Vacuum>
</ApplicationExecutionSettings>

</Application>

```

- Confirmación de cambio de escuela o convalidación de materias correspondientes a un período académico.

```

<?xml version="1.0" encoding="utf-8" ?>
<Application xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

```

```

  xmlns="http://www.microsoft.com/MicrosoftNotificationServices/ApplicationDef
  initionFileSchema">

```

```

<!-- Version -->

```

```

<!-- Database Definition -->

```

```

<!-- Event Classes -->

```

```

<EventClasses>
  <EventClass>
    <EventClassName>CambioData</EventClassName>
    <Schema>
      <Field>
        <FieldName>CodigoAlumno</FieldName>
        <FieldType>varchar(13)</FieldType>
        <FieldTypeMods>not null</FieldTypeMods>
      </Field>
      <Field>
        <FieldName>Escuela</FieldName>
        <FieldType>nvarchar(100)</FieldType>
        <FieldTypeMods>not null</FieldTypeMods>
      </Field>
    </Schema>
  </EventClass>
</EventClasses>

```



```

    </Field>
  <Field>
    <FieldName>Periodo</FieldName>
    <FieldType>nvarchar(100)</FieldType>
    <FieldTypeMods>not null</FieldTypeMods>
  </Field>
  <Field>
    <FieldName>Fecha</FieldName>
    <FieldType>datetime</FieldType>
    <FieldTypeMods>not null</FieldTypeMods>
  </Field>
</Schema>
<IndexSqlSchema>
  <SqlStatement>
    CREATE INDEX myIndex
    ON CambioData (CodigoAlumno );
  </SqlStatement>
</IndexSqlSchema>
</EventClass>
</EventClasses>

<!-- Subscription Classes -->
<SubscriptionClasses>
  <SubscriptionClass>
    <SubscriptionClassName>CambioAlumno</SubscriptionClassName>
    <Schema>
      <Field>
        <FieldName>DeviceName</FieldName>
        <FieldType>nvarchar(255)</FieldType>
        <FieldTypeMods>not null</FieldTypeMods>
      </Field>
      <Field>
        <FieldName>SubscriberLocale</FieldName>
        <FieldType>nvarchar(10)</FieldType>
        <FieldTypeMods>not null</FieldTypeMods>
      </Field>
      <Field>
        <FieldName>CodigoAlumno</FieldName>
        <FieldType>nvarchar(13)</FieldType>
        <FieldTypeMods>not null</FieldTypeMods>
      </Field>
    </Schema>
    <EventRules>
      <EventRule>
        <RuleName>CambioEventRule</RuleName>
        <EventClassName>CambioData</EventClassName>
        <Action>
          INSERT INTO CambioAlerts(SubscriberId,
            DeviceName, SubscriberLocale, Escuela, Periodo, Fecha)
          SELECT s.SubscriberId, s.DeviceName, s.SubscriberLocale,
            e.Escuela, e.Periodo, e.Fecha
          FROM CambioData e, CambioAlumno s
          WHERE e.CodigoAlumno = s.CodigoAlumno;
        </Action>
      </EventRule>
    </EventRules>
  </SubscriptionClass>
</SubscriptionClasses>

```

```

        </Action>
    </EventRule>
</EventRules>
</SubscriptionClass>
</SubscriptionClasses>

<!-- Notification Classes -->
<NotificationClasses>
    <NotificationClass>
        <NotificationClassName>
            CambioAlerts
        </NotificationClassName>
        <Schema>
            <Fields>
                <Field>
                    <FieldName>Escuela</FieldName>
                    <FieldType>nvarchar(100)</FieldType>
                </Field>
                <Field>
                    <FieldName>Periodo</FieldName>
                    <FieldType>nvarchar(100)</FieldType>
                </Field>
                <Field>
                    <FieldName>Fecha</FieldName>
                    <FieldType>datetime</FieldType>
                </Field>
            </Fields>
        </Schema>
        <ContentFormatter>
            <ClassName>XsltFormatter</ClassName>
            <Arguments>
                <Argument>
                    <Name>XsltBaseDirectoryPath</Name>
                    <Value>%_AppPath_%</Value>
                </Argument>
                <Argument>
                    <Name>XsltFileName</Name>
                    <Value>CambioTransform.xslt</Value>
                </Argument>
            </Arguments>
        </ContentFormatter>
        <Protocols>
            <Protocol>
                <ProtocolName>File</ProtocolName>
            </Protocol>
            <Protocol>
                <ProtocolName>SMTP</ProtocolName>
            </Protocol>
        </Protocols>
        <Fields>
            <Field>
                <FieldName>Subject</FieldName>
                <SqlExpression>'Confirmacion
Escuela'</SqlExpression>
            </Field>
        </Fields>
    </NotificationClass>
</NotificationClasses>

```

Cambio

```

        <Field>
            <FieldName>BodyFormat</FieldName>
            <SqlExpression>'html'</SqlExpression>
        </Field>
        <Field>
            <FieldName>From</FieldName>

            <SqlExpression>'webmaster@epoch.edu.ec'</SqlExpression>
        </Field>
        <Field>
            <FieldName>Priority</FieldName>
            <SqlExpression>'Normal'</SqlExpression>
        </Field>
        <Field>
            <FieldName>To</FieldName>
            <SqlExpression>DeviceAddress</SqlExpression>
        </Field>
    </Fields>
</Protocol>
</Protocols>
</NotificationClass>
</NotificationClasses>

<!-- Event Providers -->
<Providers>
    <NonHostedProvider>
        <ProviderName>CambioSPEventProvider</ProviderName>
    </NonHostedProvider>
</Providers>

<!-- Generator -->
<Generator>
    <SystemName>%_NSServer_</SystemName>
</Generator>

<!-- Distributors -->
<Distributors>
    <Distributor>
        <SystemName>%_NSServer_</SystemName>
        <QuantumDuration>PT15S</QuantumDuration>
    </Distributor>
</Distributors>

<!-- ApplicationExecutionSettings -->
<ApplicationExecutionSettings>
    <QuantumDuration>PT15S</QuantumDuration>
    <DistributorLogging>
        <LogBeforeDeliveryAttempts>>false</LogBeforeDeliveryAttempts>
        <LogStatusInfo>>false</LogStatusInfo>
        <LogNotificationText>>false</LogNotificationText>
    </DistributorLogging>
    <Vacuum>
        <RetentionAge>P1D</RetentionAge>

```

```

    <VacuumSchedule>
      <Schedule>
        <StartTime>23:00:00</StartTime>
        <Duration>P0DT02H00M00S</Duration>
      </Schedule>
    </VacuumSchedule>
  </Vacuum>
</ApplicationExecutionSettings>

</Application>

```

- Información de retiro de una materia o carrera.

```

<?xml version="1.0" encoding="utf-8" ?>
<Application xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

  xmlns="http://www.microsoft.com/MicrosoftNotificationServices/ApplicationDef
initionFileSchema">

  <!-- Version -->

  <!-- Database Definition -->

  <!-- Event Classes -->
  <EventClasses>
    <EventClass>
      <EventClassName>RetirosData</EventClassName>
      <Schema>
        <Field>
          <FieldName>Materia</FieldName>
          <FieldType>nvarchar(35)</FieldType>
          <FieldTypeMods>not null</FieldTypeMods>
        </Field>
        <Field>
          <FieldName>Fecha</FieldName>
          <FieldType>datetime</FieldType>
          <FieldTypeMods>not null</FieldTypeMods>
        </Field>
        <FieldName>Descripcion</FieldName>
        <FieldType>nvarchar(3500)</FieldType>
        <FieldTypeMods>not null</FieldTypeMods>
      </Field>
    </Schema>
    <IndexSqlSchema>
      <SqlStatement>
        CREATE INDEX myIndex
        ON RetirosData ( Materia );
      </SqlStatement>
    </IndexSqlSchema>
  </EventClass>

```

```

</EventClasses>

<!-- Subscription Classes -->
<SubscriptionClasses>
  <SubscriptionClass>
    <SubscriptionClassName>RetirosMateria</SubscriptionClassName>
    <Schema>
      <Field>
        <FieldName>DeviceName</FieldName>
        <FieldType>nvarchar(255)</FieldType>
        <FieldTypeMods>not null</FieldTypeMods>
      </Field>
      <Field>
        <FieldName>SubscriberLocale</FieldName>
        <FieldType>nvarchar(10)</FieldType>
        <FieldTypeMods>not null</FieldTypeMods>
      </Field>
      <Field>
        <FieldName>Materia</FieldName>
        <FieldType>nvarchar(40)</FieldType>
        <FieldTypeMods>not null</FieldTypeMods>
      </Field>
    </Schema>
    <EventRules>
      <EventRule>
        <RuleName>RetirosEventRule</RuleName>
        <EventClassName>RetirosData</EventClassName>
        <Action>
          INSERT INTO RetirosAlerts(SubscriberId,
            DeviceName, SubscriberLocale, Materia, Fecha, Descripcion)
          SELECT s.SubscriberId, s.DeviceName, s.SubscriberLocale,
            e.Materia, e.Fecha, e.Descripcion
          FROM RetirosData e, RetirosMateria s
          WHERE e.Materia = s.Materia;
        </Action>
      </EventRule>
    </EventRules>
  </SubscriptionClass>
</SubscriptionClasses>

<!-- Notification Classes -->
<NotificationClasses>
  <NotificationClass>
    <NotificationClassName>
      RetirosAlerts
    </NotificationClassName>
    <Schema>
      <Fields>
        <Field>
          <FieldName>Materia</FieldName>
          <FieldType>nvarchar(35)</FieldType>
        </Field>
        <Field>

```

```

        <FieldName>Fecha</FieldName>
        <FieldType>datetime</FieldType>
    </Field>
    <FieldName>Descripcion</FieldName>
    <FieldType>nvarchar(3500)</FieldType>
</Field>
</Fields>
</Schema>
<ContentFormatter>
    <ClassName>XsltFormatter</ClassName>
    <Arguments>
        <Argument>
            <Name>XsltBaseDirectoryPath</Name>
            <Value>%_AppPath_%</Value>
        </Argument>
        <Argument>
            <Name>XsltFileName</Name>
            <Value>RetirosTransform.xslt</Value>
        </Argument>
    </Arguments>
</ContentFormatter>
<Protocols>
    <Protocol>
        <ProtocolName>File</ProtocolName>
    </Protocol>
    <Protocol>
        <ProtocolName>SMTP</ProtocolName>
    </Protocol>
</Protocols>
<Fields>
    <Field>
        <FieldName>Subject</FieldName>
        <SqlExpression>'Retiro'</SqlExpression>
    </Field>
    <Field>
        <FieldName>BodyFormat</FieldName>
        <SqlExpression>'html'</SqlExpression>
    </Field>
    <Field>
        <FieldName>From</FieldName>
        <SqlExpression>'sender@francisco'</SqlExpression>
    </Field>
    <Field>
        <FieldName>Priority</FieldName>
        <SqlExpression>'Normal'</SqlExpression>
    </Field>
    <Field>
        <FieldName>To</FieldName>
        <SqlExpression>DeviceAddress</SqlExpression>
    </Field>
</Fields>
</NotificationClass>
</NotificationClasses>

```

```

<!-- Event Providers -->
<Providers>
  <NonHostedProvider>
    <ProviderName>RetirosSPEventProvider</ProviderName>
  </NonHostedProvider>
</Providers>

<!-- Generator -->
<Generator>
  <SystemName>%_NSServer_%</SystemName>
</Generator>

<!-- Distributors -->
<Distributors>
  <Distributor>
    <SystemName>%_NSServer_%</SystemName>
    <QuantumDuration>PT15S</QuantumDuration>
  </Distributor>
</Distributors>

<!-- ApplicationExecutionSettings -->
<ApplicationExecutionSettings>
  <QuantumDuration>PT15S</QuantumDuration>
  <DistributorLogging>
    <LogBeforeDeliveryAttempts>>false</LogBeforeDeliveryAttempts>
    <LogStatusInfo>>false</LogStatusInfo>
    <LogNotificationText>>false</LogNotificationText>
  </DistributorLogging>
  <Vacuum>
    <RetentionAge>P1D</RetentionAge>
    <VacuumSchedule>
      <Schedule>
        <StartTime>23:00:00</StartTime>
        <Duration>P0DT02H00M00S</Duration>
      </Schedule>
    </VacuumSchedule>
  </Vacuum>
</ApplicationExecutionSettings>

</Application>

```

- Información cuando el estudiante ya esté listo para incorporarse.

```

<?xml version="1.0" encoding="utf-8" ?>
<Application xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

```

```
xmlns="http://www.microsoft.com/MicrosoftNotificationServices/ApplicationD
efinitionFileSchema">
```

```
<!-- Version -->
```

```
<!-- Database Definition -->
```

```
<!-- Event Classes -->
```

```
<EventClasses>
```

```
<EventClass>
```

```
<EventClassName>GraduadoData</EventClassName>
```

```
<Schema>
```

```
<Field>
```

```
<FieldName>CodigoAlumno</FieldName>
```

```
<FieldType>int</FieldType>
```

```
<FieldTypeMods>not null</FieldTypeMods>
```

```
</Field>
```

```
<Field>
```

```
<FieldName>Acta</FieldName>
```

```
<FieldType>varchar(20)</FieldType>
```

```
<FieldTypeMods>not null</FieldTypeMods>
```

```
</Field>
```

```
<Field>
```

```
<FieldName>NotasPromedio</FieldName>
```

```
<FieldType>float</FieldType>
```

```
<FieldTypeMods>not null</FieldTypeMods>
```

```
</Field>
```

```
<Field>
```

```
<FieldName>PromedioEscrito</FieldName>
```

```
<FieldType>float</FieldType>
```

```
<FieldTypeMods>not null</FieldTypeMods>
```

```
</Field>
```

```
<Field>
```

```
<FieldName>PromedioOral</FieldName>
```

```
<FieldType>float</FieldType>
```

```
<FieldTypeMods>not null</FieldTypeMods>
```

```
</Field>
```

```
<Field>
```

```
<FieldName>PromedioGrado</FieldName>
```

```
<FieldType>float</FieldType>
```

```
<FieldTypeMods>not null</FieldTypeMods>
```

```
</Field>
```

```
<Field>
```

```
<FieldName>Fecha</FieldName>
```

```
<FieldType>datetime</FieldType>
```

```
<FieldTypeMods>not null</FieldTypeMods>
```

```
</Field>
```

```
</Schema>
```

```
<IndexSqlSchema>
```

```
<SqlStatement>
```

```
CREATE INDEX myIndex
```

```
ON GraduadoData ( CodigoAlumno );
```



```

        </SqlStatement>
    </IndexSqlSchema>
</EventClass>
</EventClasses>

<!-- Subscription Classes -->
<SubscriptionClasses>
  <SubscriptionClass>
    <SubscriptionClassName>GraduadoAlumno</SubscriptionClassName>
    <Schema>
      <Field>
        <FieldName>DeviceName</FieldName>
        <FieldType>nvarchar(255)</FieldType>
        <FieldTypeMods>not null</FieldTypeMods>
      </Field>
      <Field>
        <FieldName>SubscriberLocale</FieldName>
        <FieldType>nvarchar(10)</FieldType>
        <FieldTypeMods>not null</FieldTypeMods>
      </Field>
      <Field>
        <FieldName>CodigoAlumno</FieldName>
        <FieldType>int</FieldType>
        <FieldTypeMods>not null</FieldTypeMods>
      </Field>
    </Schema>
    <EventRules>
      <EventRule>
        <RuleName>GraduadoEventRule</RuleName>
        <EventClassName>GraduadoData</EventClassName>
        <Action>
          INSERT INTO GraduadoAlerts(SubscriberId,
            DeviceName, SubscriberLocale, Acta, NotasPromedio,
            PromedioEscrito, PromedioOral, PromedioGrado, Fecha)
          SELECT s.SubscriberId, s.DeviceName, s.SubscriberLocale,
            e.Acta, e.NotasPromedio, e.PromedioEscrito, e.PromedioOral,
            e.PromedioGrado, e.Fecha
          FROM GraduadoData e, GraduadoAlumno s
          WHERE e.CodigoAlumno = s.CodigoAlumno;
        </Action>
      </EventRule>
    </EventRules>
  </SubscriptionClass>
</SubscriptionClasses>

<!-- Notification Classes -->
<NotificationClasses>
  <NotificationClass>
    <NotificationClassName>
      GraduadoAlerts
    </NotificationClassName>
    <Schema>
      <Fields>

```

```

    <Field>
      <FieldName>Acta</FieldName>
      <FieldType>varchar(20)</FieldType>
    </Field>
    <Field>
      <FieldName>NotasPromedio</FieldName>
      <FieldType>float</FieldType>
    </Field>
    <Field>
      <FieldName>PromedioEscrito</FieldName>
      <FieldType>float</FieldType>
    </Field>
    <Field>
      <FieldName>PromedioOral</FieldName>
      <FieldType>float</FieldType>
    </Field>
    <Field>
      <FieldName>PromedioGrado</FieldName>
      <FieldType>float</FieldType>
    </Field>
    <Field>
      <FieldName>Fecha</FieldName>
      <FieldType>datetime</FieldType>
    </Field>
  </Fields>
</Schema>
<ContentFormatter>
  <ClassName>XsltFormatter</ClassName>
  <Arguments>
    <Argument>
      <Name>XsltBaseDirectoryPath</Name>
      <Value>%_AppPath_%</Value>
    </Argument>
    <Argument>
      <Name>XsltFileName</Name>
      <Value>GraduadoTransform.xslt</Value>
    </Argument>
  </Arguments>
</ContentFormatter>
<Protocols>
  <Protocol>
    <ProtocolName>File</ProtocolName>
  </Protocol>
  <Protocol>
    <ProtocolName>SMTP</ProtocolName>
  </Protocol>
</Protocols>
<Fields>
  <Field>
    <FieldName>Subject</FieldName>
    <SqlExpression>'Confirmacion
Graduacion'</SqlExpression>
  </Field>
  <Field>
    <FieldName>BodyFormat</FieldName>

```

```

        <SqlExpression>'html'</SqlExpression>
    </Field>
    <Field>
        <FieldName>From</FieldName>

    <SqlExpression>'webmaster@epoch.edu.ec'</SqlExpression>
    </Field>
    <Field>
        <FieldName>Priority</FieldName>
        <SqlExpression>'Normal'</SqlExpression>
    </Field>
    <Field>
        <FieldName>To</FieldName>
        <SqlExpression>DeviceAddress</SqlExpression>
    </Field>
    </Fields>
</Protocol>
</Protocols>
</NotificationClass>
</NotificationClasses>

<!-- Event Providers -->
<Providers>
    <NonHostedProvider>
        <ProviderName>GraduadoSPEventProvider</ProviderName>
    </NonHostedProvider>
</Providers>

<!-- Generator -->
<Generator>
    <SystemName>%_NSServer_%</SystemName>
</Generator>

<!-- Distributors -->
<Distributors>
    <Distributor>
        <SystemName>%_NSServer_%</SystemName>
        <QuantumDuration>PT15S</QuantumDuration>
    </Distributor>
</Distributors>

<!-- ApplicationExecutionSettings -->
<ApplicationExecutionSettings>
    <QuantumDuration>PT15S</QuantumDuration>
    <DistributorLogging>
        <LogBeforeDeliveryAttempts>>false</LogBeforeDeliveryAttempts>
        <LogStatusInfo>>false</LogStatusInfo>
        <LogNotificationText>>false</LogNotificationText>
    </DistributorLogging>
    <Vacuum>
        <RetentionAge>P1D</RetentionAge>
        <VacuumSchedule>
            <Schedule>

```

```

        <StartTime>23:00:00</StartTime>
        <Duration>P0DT02H00M00S</Duration>
    </Schedule>
</VacuumSchedule>
</Vacuum>
</ApplicationExecutionSettings>

</Application>

```

- Información de horarios de clases y exámenes.

```

<?xml version="1.0" encoding="utf-8" ?>
<Application xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

  xmlns="http://www.microsoft.com/MicrosoftNotificationServices/ApplicationD
efinitionFileSchema">

  <!-- Version -->

  <!-- Database Definition -->

  <!-- Event Classes -->
<EventClasses>
  <EventClass>
    <EventClassName>HorarioClaseData</EventClassName>
    <Schema>
      <Field>
        <FieldName>CodigoAlumno</FieldName>
        <FieldType>int</FieldType>
        <FieldTypeMods>not null</FieldTypeMods>
      </Field>
      <Field>
        <FieldName>Nivel</FieldName>
        <FieldType>nvarchar(20)</FieldType>
        <FieldTypeMods>not null</FieldTypeMods>
      </Field>
      <Field>
        <FieldName>Paralelo</FieldName>
        <FieldType>nvarchar(20)</FieldType>
        <FieldTypeMods>not null</FieldTypeMods>
      </Field>
      <Field>
        <FieldName>Materia</FieldName>
        <FieldType>nvarchar(50)</FieldType>
        <FieldTypeMods>not null</FieldTypeMods>
      </Field>
      <Field>
        <FieldName>Periodo</FieldName>
        <FieldType>nvarchar(50)</FieldType>
        <FieldTypeMods>not null</FieldTypeMods>
    </Schema>
  </EventClass>
</EventClasses>

```

```

        </Field>
    <Field>
        <FieldName>Dia</FieldName>
        <FieldType>nvarchar(20)</FieldType>
        <FieldTypeMods>not null</FieldTypeMods>
    </Field>
    <Field>
        <FieldName>Hora</FieldName>
        <FieldType>nvarchar(20)</FieldType>
        <FieldTypeMods>not null</FieldTypeMods>
    </Field>
</Schema>
<IndexSqlSchema>
    <SqlStatement>
        CREATE INDEX myIndex
        ON HorarioClaseData ( CodigoAlumno );
    </SqlStatement>
</IndexSqlSchema>
</EventClass>
</EventClasses>

<!-- Subscription Classes -->
<SubscriptionClasses>
    <SubscriptionClass>
        <SubscriptionClassName>ClaseAlumno</SubscriptionClassName>
        <Schema>
            <Field>
                <FieldName>DeviceName</FieldName>
                <FieldType>nvarchar(255)</FieldType>
                <FieldTypeMods>not null</FieldTypeMods>
            </Field>
            <Field>
                <FieldName>SubscriberLocale</FieldName>
                <FieldType>nvarchar(10)</FieldType>
                <FieldTypeMods>not null</FieldTypeMods>
            </Field>
            <Field>
                <FieldName>CodigoAlumno</FieldName>
                <FieldType>int</FieldType>
                <FieldTypeMods>not null</FieldTypeMods>
            </Field>
        </Schema>
        <EventRules>
            <EventRule>
                <RuleName>HorarioClaseEventRule</RuleName>
                <EventClassName>HorarioClaseData</EventClassName>
                <Action>
                    INSERT INTO HorarioClaseAlerts(SubscriberId,
                    DeviceName, SubscriberLocale, Nivel, Paralelo, Materia, Periodo,
Dia, Hora)
                    SELECT s.SubscriberId, s.DeviceName, s.SubscriberLocale,
                    e.Nivel, e.Paralelo, e.Materia, e.Periodo, e.Dia, e.Hora
                    FROM HorarioClaseData e, ClaseAlumno s

```

```

        WHERE e.CodigoAlumno = s.CodigoAlumno;
    </Action>
</EventRule>
</EventRules>
</SubscriptionClass>
</SubscriptionClasses>

<!-- Notification Classes -->
<NotificationClasses>
    <NotificationClass>
        <NotificationClassName>
            HorarioClaseAlerts
        </NotificationClassName>
        <Schema>
            <Fields>
                <Field>
                    <FieldName>Nivel</FieldName>
                    <FieldType>nvarchar(20)</FieldType>
                </Field>
                <Field>
                    <FieldName>Paralelo</FieldName>
                    <FieldType>nvarchar(20)</FieldType>
                </Field>
                <Field>
                    <FieldName>Materia</FieldName>
                    <FieldType>nvarchar(50)</FieldType>
                </Field>
                <Field>
                    <FieldName>Periodo</FieldName>
                    <FieldType>nvarchar(50)</FieldType>
                </Field>
                <Field>
                    <FieldName>Dia</FieldName>
                    <FieldType>nvarchar(20)</FieldType>
                </Field>
                <Field>
                    <FieldName>Hora</FieldName>
                    <FieldType>nvarchar(20)</FieldType>
                </Field>
            </Fields>
        </Schema>
        <ContentFormatter>
            <ClassName>XsltFormatter</ClassName>
            <Arguments>
                <Argument>
                    <Name>XsltBaseDirectoryPath</Name>
                    <Value>%_AppPath_%</Value>
                </Argument>
                <Argument>
                    <Name>XsltFileName</Name>
                    <Value>HorarioClaseTransform.xslt</Value>
                </Argument>
            </Arguments>
        </ContentFormatter>
    </NotificationClass>
</NotificationClasses>

```

```

</ContentFormatter>
<Protocols>
  <Protocol>
    <ProtocolName>File</ProtocolName>
  </Protocol>
  <Protocol>
    <ProtocolName>SMTP</ProtocolName>
  </Protocol>
  <Fields>
    <Field>
      <FieldName>Subject</FieldName>
      <SqlExpression>'Confirmacion de Horario de
Clases'</SqlExpression>
    </Field>
    <Field>
      <FieldName>BodyFormat</FieldName>
      <SqlExpression>'html'</SqlExpression>
    </Field>
    <Field>
      <FieldName>From</FieldName>
      <SqlExpression>'webmaster@epoch.edu.ec'</SqlExpression>
    </Field>
    <Field>
      <FieldName>Priority</FieldName>
      <SqlExpression>'Normal'</SqlExpression>
    </Field>
    <Field>
      <FieldName>To</FieldName>
      <SqlExpression>DeviceAddress</SqlExpression>
    </Field>
  </Fields>
</Protocol>
</Protocols>
</NotificationClass>
</NotificationClasses>

<!-- Event Providers -->
<Providers>
  <NonHostedProvider>
    <ProviderName>HorarioClaseSPEventProvider</ProviderName>
  </NonHostedProvider>
</Providers>

<!-- Generator -->
<Generator>
  <SystemName>%_NSServer_%</SystemName>
</Generator>

<!-- Distributors -->
<Distributors>
  <Distributor>
    <SystemName>%_NSServer_%</SystemName>
    <QuantumDuration>PT15S</QuantumDuration>
  </Distributor>
</Distributors>

```

```

</Distributor>
</Distributors>

<!-- ApplicationExecutionSettings -->
<ApplicationExecutionSettings>
  <QuantumDuration>PT15S</QuantumDuration>
  <DistributorLogging>
    <LogBeforeDeliveryAttempts>>false</LogBeforeDeliveryAttempts>
    <LogStatusInfo>>false</LogStatusInfo>
    <LogNotificationText>>false</LogNotificationText>
  </DistributorLogging>
  <Vacuum>
    <RetentionAge>P1D</RetentionAge>
    <VacuumSchedule>
      <Schedule>
        <StartTime>23:00:00</StartTime>
        <Duration>P0DT02H00M00S</Duration>
      </Schedule>
    </VacuumSchedule>
  </Vacuum>
</ApplicationExecutionSettings>

</Application>

```

- Cambio de contraseña.

```

<?xml version="1.0" encoding="utf-8" ?>
<Application xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

  xmlns="http://www.microsoft.com/MicrosoftNotificationServices/ApplicationD
efinitionFileSchema">

  <!-- Version -->

  <!-- Database Definition -->

  <!-- Event Classes -->
  <EventClasses>
    <EventClass>
      <EventClassName>CambioClaveEstudianteData</EventClassName>
      <Schema>
        <Field>
          <FieldName>Cedula</FieldName>
          <FieldType>nvarchar(13)</FieldType>
          <FieldTypeMods>not null</FieldTypeMods>
        </Field>
        <Field>
          <FieldName>Clave</FieldName>
          <FieldType>nvarchar(10)</FieldType>
          <FieldTypeMods>not null</FieldTypeMods>

```



```

    </Field>
  </Schema>
  <IndexSqlSchema>
    <SqlStatement>
      CREATE INDEX myIndex
      ON CambioClaveEstudianteData ( Cedula );
    </SqlStatement>
  </IndexSqlSchema>
</EventClass>
</EventClasses>

<!-- Subscription Classes -->
<SubscriptionClasses>
  <SubscriptionClass>
    <SubscriptionClassName>CambioClave</SubscriptionClassName>
    <Schema>
      <Field>
        <FieldName>DeviceName</FieldName>
        <FieldType>nvarchar(255)</FieldType>
        <FieldTypeMods>not null</FieldTypeMods>
      </Field>
      <Field>
        <FieldName>SubscriberLocale</FieldName>
        <FieldType>nvarchar(10)</FieldType>
        <FieldTypeMods>not null</FieldTypeMods>
      </Field>
      <Field>
        <FieldName>Cedula</FieldName>
        <FieldType>nvarchar(13)</FieldType>
        <FieldTypeMods>not null</FieldTypeMods>
      </Field>
    </Schema>
    <EventRules>
      <EventRule>
        <RuleName>CambioClaveEstudianteEventRule</RuleName>
        <EventClassName>CambioClaveEstudianteData</EventClassName>
        <Action>
          INSERT INTO CambioClaveEstudianteAlerts(SubscriberId,
          DeviceName, SubscriberLocale, Clave)
          SELECT s.SubscriberId, s.DeviceName, s.SubscriberLocale,
          e.Clave
          FROM CambioClaveEstudianteData e, CambioClave s
          WHERE e.Cedula = s.Cedula;
        </Action>
      </EventRule>
    </EventRules>
  </SubscriptionClass>
</SubscriptionClasses>

<!-- Notification Classes -->
<NotificationClasses>
  <NotificationClass>
    <NotificationClassName>

```

```

CambioClaveEstudianteAlerts
</NotificationClassName>
<Schema>
  <Fields>
    <Field>
      <FieldName>Clave</FieldName>
      <FieldType>nvarchar(10)</FieldType>
    </Field>
  </Fields>
</Schema>
<ContentFormatter>
  <ClassName>XsltFormatter</ClassName>
  <Arguments>
    <Argument>
      <Name>XsltBaseDirectoryPath</Name>
      <Value>%_AppPath_%</Value>
    </Argument>
    <Argument>
      <Name>XsltFileName</Name>
      <Value>CambioClaveEstTransform.xslt</Value>
    </Argument>
  </Arguments>
</ContentFormatter>
<Protocols>
  <Protocol>
    <ProtocolName>File</ProtocolName>
  </Protocol>
  <Protocol>
    <ProtocolName>SMTP</ProtocolName>
  </Protocol>
  <Fields>
    <Field>
      <FieldName>Subject</FieldName>
      <SqlExpression>'Confirmacion      Cambio      de
Clave'</SqlExpression>
    </Field>
    <Field>
      <FieldName>BodyFormat</FieldName>
      <SqlExpression>'html'</SqlExpression>
    </Field>
    <Field>
      <FieldName>From</FieldName>
      <SqlExpression>'webmaster@epoch.edu.ec'</SqlExpression>
    </Field>
    <Field>
      <FieldName>Priority</FieldName>
      <SqlExpression>'Normal'</SqlExpression>
    </Field>
    <Field>
      <FieldName>To</FieldName>
      <SqlExpression>DeviceAddress</SqlExpression>
    </Field>
  </Fields>

```

```

        </Protocol>
    </Protocols>
    </NotificationClass>
</NotificationClasses>

<!-- Event Providers -->
<Providers>
    <NonHostedProvider>

<ProviderName>CambioClaveEstudianteSPEventProvider</ProviderName>
    </NonHostedProvider>
</Providers>

<!-- Generator -->
<Generator>
    <SystemName>%_NSServer_%</SystemName>
</Generator>

<!-- Distributors -->
<Distributors>
    <Distributor>
        <SystemName>%_NSServer_%</SystemName>
        <QuantumDuration>PT15S</QuantumDuration>
    </Distributor>
</Distributors>

<!-- ApplicationExecutionSettings -->
<ApplicationExecutionSettings>
    <QuantumDuration>PT15S</QuantumDuration>
    <DistributorLogging>
        <LogBeforeDeliveryAttempts>>false</LogBeforeDeliveryAttempts>
        <LogStatusInfo>>false</LogStatusInfo>
        <LogNotificationText>>false</LogNotificationText>
    </DistributorLogging>
    <Vacuum>
        <RetentionAge>P1D</RetentionAge>
        <VacuumSchedule>
            <Schedule>
                <StartTime>23:00:00</StartTime>
                <Duration>P0DT02H00M00S</Duration>
            </Schedule>
        </VacuumSchedule>
    </Vacuum>
</ApplicationExecutionSettings>

</Application>

```

Docentes:

- Información de carga horaria y materias a dictar.

```

<?xml version="1.0" encoding="utf-8" ?>
<Application xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xmlns="http://www.microsoft.com/MicrosoftNotificationServices/ApplicationD
efinitionFileSchema">

  <!-- Version -->

  <!-- Database Definition -->

  <!-- Event Classes -->
  <EventClasses>
    <EventClass>
      <EventClassName>MateriasDocenteData</EventClassName>
      <Schema>
        <Field>
          <FieldName>CedDocente</FieldName>
          <FieldType>nvarchar(11)</FieldType>
          <FieldTypeMods>not null</FieldTypeMods>
        </Field>
        <Field>
          <FieldName>Nivel</FieldName>
          <FieldType>nvarchar(20)</FieldType>
          <FieldTypeMods>not null</FieldTypeMods>
        </Field>
        <Field>
          <FieldName>Paralelo</FieldName>
          <FieldType>nvarchar(20)</FieldType>
          <FieldTypeMods>not null</FieldTypeMods>
        </Field>
        <Field>
          <FieldName>Materia</FieldName>
          <FieldType>nvarchar(50)</FieldType>
          <FieldTypeMods>not null</FieldTypeMods>
        </Field>
        <Field>
          <FieldName>Periodo</FieldName>
          <FieldType>nvarchar(50)</FieldType>
          <FieldTypeMods>not null</FieldTypeMods>
        </Field>
      </Schema>
      <IndexSqlSchema>
        <SqlStatement>
          CREATE INDEX myIndex
          ON MateriasDocenteData ( CedDocente );
        </SqlStatement>
      </IndexSqlSchema>
    </EventClass>
  </EventClasses>

  <!-- Subscription Classes -->
  <SubscriptionClasses>

```

```

<SubscriptionClass>
  <SubscriptionClassName>MateriaDocente</SubscriptionClassName>
  <Schema>
    <Field>
      <FieldName>DeviceName</FieldName>
      <FieldType>nvarchar(255)</FieldType>
      <FieldTypeMods>not null</FieldTypeMods>
    </Field>
    <Field>
      <FieldName>SubscriberLocale</FieldName>
      <FieldType>nvarchar(10)</FieldType>
      <FieldTypeMods>not null</FieldTypeMods>
    </Field>
    <Field>
      <FieldName>CedDocente</FieldName>
      <FieldType>nvarchar(11)</FieldType>
      <FieldTypeMods>not null</FieldTypeMods>
    </Field>
  </Schema>
  <EventRules>
    <EventRule>
      <RuleName>MateriasDocenteEventRule</RuleName>
      <EventClassName>MateriasDocenteData</EventClassName>
      <Action>
        INSERT INTO MateriasDocenteAlerts(SubscriberId,
        DeviceName, SubscriberLocale, Nivel, Paralelo, Materia, Periodo)
        SELECT s.SubscriberId, s.DeviceName, s.SubscriberLocale,
        e.Nivel, e.Paralelo, e.Materia, e.Periodo
        FROM MateriasDocenteData e, MateriaDocente s
        WHERE e.CedDocente = s.CedDocente;
      </Action>
    </EventRule>
  </EventRules>
</SubscriptionClass>
</SubscriptionClasses>

```

```

<!-- Notification Classes -->
<NotificationClasses>
  <NotificationClass>
    <NotificationClassName>
      MateriasDocenteAlerts
    </NotificationClassName>
    <Schema>
      <Fields>
        <Field>
          <FieldName>Nivel</FieldName>
          <FieldType>nvarchar(20)</FieldType>
        </Field>
        <Field>
          <FieldName>Paralelo</FieldName>
          <FieldType>nvarchar(20)</FieldType>
        </Field>
      </Fields>
    </Schema>
  </NotificationClass>
</NotificationClasses>

```

```

    <FieldName>Materia</FieldName>
    <FieldType>nvarchar(50)</FieldType>
  </Field>
  <Field>
    <FieldName>Periodo</FieldName>
    <FieldType>nvarchar(50)</FieldType>
  </Field>
</Fields>
</Schema>
<ContentFormatter>
  <ClassName>XsltFormatter</ClassName>
  <Arguments>
    <Argument>
      <Name>XsltBaseDirectoryPath</Name>
      <Value>%_AppPath_%</Value>
    </Argument>
    <Argument>
      <Name>XsltFileName</Name>
      <Value>MateriasDocentesTransform.xslt</Value>
    </Argument>
  </Arguments>
</ContentFormatter>
<Protocols>
  <Protocol>
    <ProtocolName>File</ProtocolName>
  </Protocol>
  <Protocol>
    <ProtocolName>SMTP</ProtocolName>
  </Protocol>
  <Fields>
    <Field>
      <FieldName>Subject</FieldName>
      <SqlExpression>'Confirmacion de Materias a
Dictar'</SqlExpression>
    </Field>
    <Field>
      <FieldName>BodyFormat</FieldName>
      <SqlExpression>'html'</SqlExpression>
    </Field>
    <Field>
      <FieldName>From</FieldName>
      <SqlExpression>'webmaster@epoch.edu.ec'</SqlExpression>
    </Field>
    <Field>
      <FieldName>Priority</FieldName>
      <SqlExpression>'Normal'</SqlExpression>
    </Field>
    <Field>
      <FieldName>To</FieldName>
      <SqlExpression>DeviceAddress</SqlExpression>
    </Field>
  </Fields>
</Protocol>

```

```

    </Protocols>
  </NotificationClass>
</NotificationClasses>

<!-- Event Providers -->
<Providers>
  <NonHostedProvider>
    <ProviderName>MateriasDocenteSPEventProvider</ProviderName>
  </NonHostedProvider>
</Providers>

<!-- Generator -->
<Generator>
  <SystemName>%_NSServer_%</SystemName>
</Generator>

<!-- Distributors -->
<Distributors>
  <Distributor>
    <SystemName>%_NSServer_%</SystemName>
    <QuantumDuration>PT15S</QuantumDuration>
  </Distributor>
</Distributors>

<!-- ApplicationExecutionSettings -->
<ApplicationExecutionSettings>
  <QuantumDuration>PT15S</QuantumDuration>
  <DistributorLogging>
    <LogBeforeDeliveryAttempts>>false</LogBeforeDeliveryAttempts>
    <LogStatusInfo>>false</LogStatusInfo>
    <LogNotificationText>>false</LogNotificationText>
  </DistributorLogging>
  <Vacuum>
    <RetentionAge>P1D</RetentionAge>
    <VacuumSchedule>
      <Schedule>
        <StartTime>23:00:00</StartTime>
        <Duration>P0DT02H00M00S</Duration>
      </Schedule>
    </VacuumSchedule>
  </Vacuum>
</ApplicationExecutionSettings>

</Application>

```

- Información de la fecha de cierre de período académico.

```

<?xml version="1.0" encoding="utf-8" ?>
<Application xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

```

```
xmlns="http://www.microsoft.com/MicrosoftNotificationServices/ApplicationDefinitionFileSchema">
```

```
<!-- Version -->
```

```
<!-- Database Definition -->
```

```
<!-- Event Classes -->
```

```
<EventClasses>
```

```
<EventClass>
```

```
<EventClassName>CierriePeriodo</EventClassName>
```

```
<Schema>
```

```
<Field>
```

```
<FieldName>Codigo_Periodo</FieldName>
```

```
<FieldType>nvarchar(35)</FieldType>
```

```
<FieldTypeMods>not null</FieldTypeMods>
```

```
</Field>
```

```
<Field>
```

```
<FieldName>Fecha</FieldName>
```

```
<FieldType>datetime</FieldType>
```

```
<FieldTypeMods>not null</FieldTypeMods>
```

```
</Field>
```

```
</Schema>
```

```
<IndexSqlSchema>
```

```
<SqlStatement>
```

```
CREATE INDEX myIndex
```

```
ON CierriePeriodo ( Codigo_Periodo );
```

```
</SqlStatement>
```

```
</IndexSqlSchema>
```

```
</EventClass>
```

```
</EventClasses>
```

```
<!-- Subscription Classes -->
```

```
<SubscriptionClasses>
```

```
<SubscriptionClass>
```

```
<SubscriptionClassName>C_Periodo</SubscriptionClassName>
```

```
<Schema>
```

```
<Field>
```

```
<FieldName>DeviceName</FieldName>
```

```
<FieldType>nvarchar(255)</FieldType>
```

```
<FieldTypeMods>not null</FieldTypeMods>
```

```
</Field>
```

```
<Field>
```

```
<FieldName>SubscriberLocale</FieldName>
```

```
<FieldType>nvarchar(10)</FieldType>
```

```
<FieldTypeMods>not null</FieldTypeMods>
```

```
</Field>
```

```
<Field>
```

```
<FieldName>Codigo_Periodo</FieldName>
```

```
<FieldType>nvarchar(40)</FieldType>
```

```
<FieldTypeMods>not null</FieldTypeMods>
```

```
</Field>
```



```

</Schema>
<EventRules>
  <EventRule>
    <RuleName>CierrePeriodoEventRule</RuleName>
    <EventClassName>CierriePeriodo</EventClassName>
    <Action>
      INSERT INTO CierrePeriodoAlerts(SubscriberId,
      DeviceName, SubscriberLocale,Codigo_Periodo, Fecha)
      SELECT s.SubscriberId, s.DeviceName, s.SubscriberLocale,
      e.Codigo_Periodo, e.Fecha
      FROM CierriePeriodo e, C_Periodo s
      WHERE e.Codigo_Periodo = s.Codigo_Periodo;
    </Action>
  </EventRule>
</EventRules>
</SubscriptionClass>
</SubscriptionClasses>

<!-- Notification Classes -->
<NotificationClasses>
  <NotificationClass>
    <NotificationClassName>
      CierrePeriodoAlerts
    </NotificationClassName>
    <Schema>
      <Fields>
        <Field>
          <FieldName>Codigo_Periodo</FieldName>
          <FieldType>nvarchar(35)</FieldType>
        </Field>
        <Field>
          <FieldName>Fecha</FieldName>
          <FieldType>datetime</FieldType>
        </Field>
      </Fields>
    </Schema>
    <ContentFormatter>
      <ClassName>XsltFormatter</ClassName>
      <Arguments>
        <Argument>
          <Name>XsltBaseDirectoryPath</Name>
          <Value>%_AppPath_%</Value>
        </Argument>
        <Argument>
          <Name>XsltFileName</Name>
          <Value>CierrePeriodoTransform.xslt</Value>
        </Argument>
      </Arguments>
    </ContentFormatter>
  </NotificationClass>
</NotificationClasses>
<Protocols>
  <Protocol>
    <ProtocolName>File</ProtocolName>
  </Protocol>

```

```

<Protocol>
  <ProtocolName>SMTP</ProtocolName>
<Fields>
  <Field>
    <FieldName>Subject</FieldName>
    <SqlExpression>'Cierre Periodo'</SqlExpression>
  </Field>
  <Field>
    <FieldName>BodyFormat</FieldName>
    <SqlExpression>'html'</SqlExpression>
  </Field>
  <Field>
    <FieldName>From</FieldName>
    <SqlExpression>'sender@francisco'</SqlExpression>
  </Field>
  <Field>
    <FieldName>Priority</FieldName>
    <SqlExpression>'Normal'</SqlExpression>
  </Field>
  <Field>
    <FieldName>To</FieldName>
    <SqlExpression>DeviceAddress</SqlExpression>
  </Field>
</Fields>
</Protocol>
</Protocols>
</NotificationClass>
</NotificationClasses>

<!-- Event Providers -->
<Providers>
  <NonHostedProvider>
    <ProviderName>CierrePeriodoSPEventProvider</ProviderName>
  </NonHostedProvider>
</Providers>

<!-- Generator -->
<Generator>
  <SystemName>%_NSServer_%</SystemName>
</Generator>

<!-- Distributors -->
<Distributors>
  <Distributor>
    <SystemName>%_NSServer_%</SystemName>
    <QuantumDuration>PT15S</QuantumDuration>
  </Distributor>
</Distributors>

<!-- ApplicationExecutionSettings -->
<ApplicationExecutionSettings>
  <QuantumDuration>PT15S</QuantumDuration>
  <DistributorLogging>

```

```

    <LogBeforeDeliveryAttempts>>false</LogBeforeDeliveryAttempts>
    <LogStatusInfo>>false</LogStatusInfo>
    <LogNotificationText>>false</LogNotificationText>
  </DistributorLogging>
  <Vacuum>
    <RetentionAge>P1D</RetentionAge>
    <VacuumSchedule>
      <Schedule>
        <StartTime>23:00:00</StartTime>
        <Duration>P0DT02H00M00S</Duration>
      </Schedule>
    </VacuumSchedule>
  </Vacuum>
</ApplicationExecutionSettings>

</Application>

```

- Cambio de contraseña.

```

<?xml version="1.0" encoding="utf-8" ?>
<Application xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

  xmlns="http://www.microsoft.com/MicrosoftNotificationServices/ApplicationDef
initionFileSchema">

  <!-- Version -->

  <!-- Database Definition -->

  <!-- Event Classes -->
  <EventClasses>
    <EventClass>
      <EventClassName>CambioClaveDocenteData</EventClassName>
      <Schema>
        <Field>
          <FieldName>Cedula</FieldName>
          <FieldType>nvarchar(11)</FieldType>
          <FieldTypeMods>not null</FieldTypeMods>
        </Field>
        <Field>
          <FieldName>Clave</FieldName>
          <FieldType>nvarchar(10)</FieldType>
          <FieldTypeMods>not null</FieldTypeMods>
        </Field>
      </Schema>
      <IndexSqlSchema>
        <SqlStatement>
          CREATE INDEX myIndex
          ON CambioClaveDocenteData ( Cedula );
        </SqlStatement>

```

```

    </IndexSqlSchema>
  </EventClass>
</EventClasses>

<!-- Subscription Classes -->
<SubscriptionClasses>
  <SubscriptionClass>
    <SubscriptionClassName>CambioClave</SubscriptionClassName>
    <Schema>
      <Field>
        <FieldName>DeviceName</FieldName>
        <FieldType>nvarchar(255)</FieldType>
        <FieldTypeMods>not null</FieldTypeMods>
      </Field>
      <Field>
        <FieldName>SubscriberLocale</FieldName>
        <FieldType>nvarchar(10)</FieldType>
        <FieldTypeMods>not null</FieldTypeMods>
      </Field>
      <Field>
        <FieldName>Cedula</FieldName>
        <FieldType>nvarchar(11)</FieldType>
        <FieldTypeMods>not null</FieldTypeMods>
      </Field>
    </Schema>
    <EventRules>
      <EventRule>
        <RuleName>CambioClaveDocenteEventRule</RuleName>
        <EventClassName>CambioClaveDocenteData</EventClassName>
        <Action>
          INSERT INTO CambioClaveDocenteAlerts(SubscriberId,
          DeviceName, SubscriberLocale, Clave)
          SELECT s.SubscriberId, s.DeviceName, s.SubscriberLocale,
          e.Clave
          FROM CambioClaveDocenteData e, CambioClave s
          WHERE e.Cedula = s.Cedula;
        </Action>
      </EventRule>
    </EventRules>
  </SubscriptionClass>
</SubscriptionClasses>

<!-- Notification Classes -->
<NotificationClasses>
  <NotificationClass>
    <NotificationClassName>
      CambioClaveDocenteAlerts
    </NotificationClassName>
    <Schema>
      <Fields>
        <Field>
          <FieldName>Clave</FieldName>
          <FieldType>nvarchar(10)</FieldType>
        </Field>
      </Fields>
    </Schema>
  </NotificationClass>
</NotificationClasses>

```

```

        </Field>
      </Fields>
    </Schema>
    <ContentFormatter>
      <ClassName>XsltFormatter</ClassName>
      <Arguments>
        <Argument>
          <Name>XsltBaseDirectoryPath</Name>
          <Value>%_AppPath_%</Value>
        </Argument>
        <Argument>
          <Name>XsltFileName</Name>
          <Value>CambioClaveDocTransform.xslt</Value>
        </Argument>
      </Arguments>
    </ContentFormatter>
    <Protocols>
      <Protocol>
        <ProtocolName>File</ProtocolName>
      </Protocol>
      <Protocol>
        <ProtocolName>SMTP</ProtocolName>
      </Protocol>
    </Protocols>
    <Fields>
      <Field>
        <FieldName>Subject</FieldName>
        <SqlExpression>'Confirmacion de Cambio de
Clave'</SqlExpression>
      </Field>
      <Field>
        <FieldName>BodyFormat</FieldName>
        <SqlExpression>'html'</SqlExpression>
      </Field>
      <Field>
        <FieldName>From</FieldName>
        <SqlExpression>'webmaster@esepoch.edu.ec'</SqlExpression>
      </Field>
      <Field>
        <FieldName>Priority</FieldName>
        <SqlExpression>'Normal'</SqlExpression>
      </Field>
      <Field>
        <FieldName>To</FieldName>
        <SqlExpression>DeviceAddress</SqlExpression>
      </Field>
    </Fields>
  </Protocol>
</Protocols>
</NotificationClass>
</NotificationClasses>

<!-- Event Providers -->
<Providers>

```

```
<NonHostedProvider>
<ProviderName>CambioClaveDocenteSPEventProvider</ProviderName>
  </NonHostedProvider>
</Providers>

<!-- Generator -->
<Generator>
  <SystemName>%_NSServer_%</SystemName>
</Generator>

<!-- Distributors -->
<Distributors>
  <Distributor>
    <SystemName>%_NSServer_%</SystemName>
    <QuantumDuration>PT15S</QuantumDuration>
  </Distributor>
</Distributors>

<!-- ApplicationExecutionSettings -->
<ApplicationExecutionSettings>
  <QuantumDuration>PT15S</QuantumDuration>
  <DistributorLogging>
    <LogBeforeDeliveryAttempts>>false</LogBeforeDeliveryAttempts>
    <LogStatusInfo>>false</LogStatusInfo>
    <LogNotificationText>>false</LogNotificationText>
  </DistributorLogging>
  <Vacuum>
    <RetentionAge>P1D</RetentionAge>
    <VacuumSchedule>
      <Schedule>
        <StartTime>23:00:00</StartTime>
        <Duration>P0DT02H00M00S</Duration>
      </Schedule>
    </VacuumSchedule>
  </Vacuum>
</ApplicationExecutionSettings>

</Application>
```