UNIVERSIDAD NACIONAL DE CHIMBORAZO

FACULTAD DE INGENIERÍA

ESCUELA DE INGENIERÍA EN SISTEMAS Y COMPUTACIÓN

"**Trabajo de Grado previo a la obtención del Título de Ingeniero en Sistemas y Computación**."

TRABAJO DE GRADUACIÓN

Título del proyecto

Propuesta metodológica para la integración de sistemas entre las arquitecturas J2EE y .NET. Caso práctico: Sistema de Gestión Académico de la Facultad de Ingeniería.

Autor: Johnny Danilo Latta Chavarrea

Director: Ing. Danny Velasco

Riobamba – Ecuador

2013

UNIVERSIDAD NACIONAL DE CHIMBORAZO

FACULTAD DE INGENIERÍA

ESCUELA DE INGENIERÍA EN SISTEMAS Y COMPUTACIÓN

Título del Proyecto

Propuesta metodológica para la integración de sistemas entre las arquitecturas J2EE y .NET. Caso práctico: Sistema de Gestión Académico de la Facultad de Ingeniería.

Los miembros del Tribunal de Graduación del proyecto de investigación de título: PROPUESTA METODOLÓGICA PARA LA INTEGRACIÓN DE SISTEMAS ENTRE LAS ARQUITECTURAS J2EE Y .NET. CASO PRÁCTICO: SISTEMA DE GESTIÓN ACADÉMICO DE LA FACULTAD DE INGENIERÍA presentado por: Johnny Danilo Latta Chavarrea y dirigida por: Ing. Danny Velasco.

Una vez escuchada la defensa oral y revisado el informe final del proyecto de investigación con fines de graduación escrito en la cual se ha constatado el cumplimiento de las observaciones realizadas, remite la presente para uso y custodia en la biblioteca de la Facultad de Ingeniería de la UNACH.

Para constancia de lo expuesto firman:

_____	_____
Presidente del Tribunal Ing. Fernando Molina	Firma
_____	_____
Miembro del Tribunal Ing. Danny Velasco	Firma
_____	_____
Miembro del Tribunal Ing. Lida Barba	Firma

PROPUESTA METODOLÓGICA PARA LA INTEGRACIÓN DE SISTEMAS ENTRE LAS ARQUITECTURAS J2EE Y .NET. CASO PRÁCTICO: SISTEMA DE GESTIÓN ACADÉMICO DE LA FACULTAD DE INGENIERÍA

AUTORÍA DE LA INVESTIGACIÓN

"La responsabilidad del contenido de este Proyecto de Graduación, nos corresponde exclusivamente a: Johnny Latta (autor) y del Ing. Danny Velasco (director); y el patrimonio intelectual de la misma a la Universidad Nacional de Chimborazo.

AGRADECIMIENTO

Agradezco a Dios por sus bendiciones impartidas en el desarrollo de este proyecto, a mi madre, a mi esposa e hijas, a mis hermanas por su apoyo y a los docentes por dotar sus conocimientos durante mi formación académica.

A mi familia y amigos por sus consejos para poder finalizar este trabajo.

Johnny Danilo Latta Chavarrea.

DEDICATORIA

Dedicado a mi madre Martha Chavarrea por ser la persona que me brindó todo su apoyo para culminar esta meta, a mi esposa e hijas por motivarme a la culminación de mi carrera, a mis hermanas Judith y Tamara, por su apoyo incondicional y a todas las personas que con su ayuda, guía o consejo hicieron posible la realización de esta meta.

ÍNDICE DE CONTENIDOS

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

RESUMEN

El presente trabajo describe en cada uno de los capítulos desarrollados los conceptos y definiciones necesarias para el desarrollo de una metodología que permitirá integrar tecnologías JAVA y .NET mediante el desarrollo de un sistema software implementado en la plataforma JAVA y haciendo uso Web Service implementado en la plataforma .NET.

Inicialmente, se presenta el marco teórico acerca de los conceptos sobre la integración de tecnologías, métodos de integración, se mencionan algunas de las estrategias utilizadas integrar tecnologías mediante el uso de web Service; así como también se expone un estudio introductorio a los web services, describiendo la historia de los mismos, cuando se los debe utilizar y las tecnologías que estos utilizan.

A continuación se presentan las formas de implementación de los web services en la plataforma .NET, conceptos necesarios para tener un conocimiento acera de su funcionamiento.

Posteriormente se define la captura de requerimientos, análisis y diseño del sistema GestionUnach que permitirá hacer uso de los web Services desarrollados en la plataforma .NET siguiendo la metodología Crain Larman, adicionalmente se detallan los métodos de integración asociados a JAVA y .NET, los mismos que permitirán evidenciar la forma do integrar estas tecnologías y evaluar su rendimiento debido a las respuestas que proporcione el web Service.

Finalmente se presentan como resultado de la evidencia de integración de las tecnologías y del análisis bibliográfico realizado a lo largo de la investigación, las conclusiones y recomendaciones, con la finalidad de construir un aporte significativo para el desarrollo de futuros proyectos que pretendan integrar tecnologías de gran nivel y trabajo técnico.

SUMMARY

This research describes each of developed chapters about necessary concept and definitions inorder to develop a methodology to integrate technologies JAVA and. NET through development of a software system implemented in the JAVA platform and using Web Service implemented in the platform .NET.

Initially, we introduce the theoretical concepts about technology and methods of integration, and used strategies to integrate technologies using Web Service, and also expose an introductory study of web services, describing the history of them, when they should be used and the technologies that they use.

Below there are ways of implementing web services on the platform.NET concepts necessary to have knowledge of its working operation.

Then we define the requirements capture, analysis and Gestion Unach system design that it will allow the use of Web Services developed on the platform. NET Crain Larman following the additionally detailed methodology in the integration methods associated with Java and. NET, the same that will allow evidence the form to integrate these technologies for assessing its performance due to the answers given by the web Service.

Finally, we introduce evidence resulting from the integration of technologies and the checked conducted bibliography about research, findings and recommendations, in order to build a significant contribution to the development of future projects seeking to integrate technologies of large level and technical work.

INTRODUCCIÓN

Aunque muchas tecnologías utilizadas en sistemas de información institucionales pudieran ser consideradas innovadoras, la integración de sus componentes para ser habilitados en ambientes de internet es llevado a cabo a través de un grupo selecto de plataformas. Las de mayor importancia por ahora, que seguramente prevalecerán en los próximos 5 o 10 años son: Java y .Net.

La mayoría de los procesos institucionales han cambiado en estos últimos años, necesitando nuevas características, tareas y aplicaciones tales como: flexibilidad, interconectividad y autonomía debido a las condiciones del mercado, los nuevos modelos institucionales y a los escenarios de uso de los sistemas de información.

Una persona que está familiarizada con el desarrollo de aplicaciones web sabe que el desarrollo web no es una tarea simple, ya que mientras un modelo de programación para aplicaciones de uso común está muy bien establecido y soportado por un gran número de lenguajes, herramientas de desarrollo; la programación web es una mezcla de varios lenguajes de etiquetas, un gran uso de lenguajes de script y plataformas de servidor. Entonces en este contexto, el entorno web está cambiando la forma en la que se ofrecen los servicios de gestión en instituciones, los servicios en la sociedad y la forma en la que estos servicios interoperan. Esta tendencia conlleva a sistemas de información conectados e integrados a través de una infraestructura para redes, tanto en internet como también en una intranet. Es por eso que se introduce un nuevo entorno para aplicaciones orientadas a la web. La estrategia o método de integración de sistemas llamado web services proporcionan la plataforma tecnológica ideal para conseguir la completa integración de los procesos de negocio de una institución (educativa, gubernamental, privada), empresa o cualquier entidad. Pero como se conoce que las aplicaciones de software se encuentran desarrolladas en diferentes plataformas, ha sido necesario proponer metodologías a seguir para comunicar las diferentes aplicaciones o componentes de éstas, de forma estándar a través de protocolos comunes y de manera independiente al lenguaje de programación, plataforma de implementación, formato de presentación o sistema operativo.

El resultado es un sistema software funcional, minimiza instancias de los datos, actividades manuales sin redundancia, costos más bajos y respuestas eficientes para los administradores del sistema.

En la actualidad la mayoría de sistemas informáticos que se utilizan en varias empresas, se adhieren al mundo de las tecnologías que predominan en el mercado como JAVA, .NET, etc, para satisfacer las necesidades y los objetivos de una empresa también se adhieren a una metodología de sistemas y a la vez de integración entre tecnologías, las estrategias de un sistema se basa en las funcionalidades, facilidades y procesos que brinda un sistema informático.

Las instituciones probablemente ya han identificado un requisito de negocio que sus aplicaciones trabajen juntas y exista interoperabilidad entre tecnologías, de la misma forma en que los colaboradores tienen que trabajar juntos para alcanzar los objetivos de negocio y acoplarse a las funciones del sistema en sí definido, las aplicaciones necesitan hacer lo mismo; de aquí la necesidad de plantear estrategias para integrar tecnologías y lo que es más para la construcción de las mismas es una tarea difícil y compleja, para lo cual se pone a consideración el presente trabajo que incluye un estudio completo y desarrollo del diseño e implementación de una metodología de integración y de un sistema software que hará uso de una estrategia de integración de sistemas como es Web Service implementado en la plataforma .Net, exclusivamente se hará uso en la Secretaría de la Facultad de Ingenieria de la Universidad Nacional de Chimborazo.

CAPÍTULO I

1. MARCO REFERENCIAL

1.1. PLANTEAMIENTO DEL PROBLEMA

La Secretaría de la Facultad de Ingeniería de la Universidad Nacional de Chimborazo "UNACH" se encuentra trabajando de forma manual en la emisión de reportes de varios procesos de los estudiantes, desaprovechando el uso de sistemas informáticos y plataformas que pueden ser integrables al actual sistema académico institucional.

Las diferentes tareas, procesos y solicitudes que se presentan por parte de los estudiantes y docentes de la Facultad de Ingeniería, ha presentado demora al momento de emitir la información, no existe un sistema capaz de realizar varios procesos en dicho departamento, que emita reportes con mayor seguridad y aplicado un orden jerárquico, Los procesos realizados manualmente dificulta guardar los datos con mayor seguridad, y esto a su vez no permite trabajar de manera eficiente.

En la mayoría de sistemas y proyectos informáticos que se han desarrollado en diversas partes del país y del mundo, se han aplicado el uso de metodologías para la integración de sistemas, llevando a cabo procesos ordenados, organizados y sistemáticos, a la vez ofreciendo las herramientas y técnicas suficientes como para cubrir todos los aspectos que se pueden encontrar en un sistema para que los resultados sean alcanzables y eficientes.

1.2. FORMULACIÓN DEL PROBLEMA

¿Permitirá la propuesta metodológica integrar las arquitecturas J2EE y .NET?

1.3. OBJETIVOS

1.3.1. OBJETIVO GENERAL

Proponer una metodología para integrar sistemas entre las arquitecturas J2EE y .NET en la Secretaria de la Facultad de Ingeniería de la UNACH.

1.3.2. OBJETIVOS ESPECÍFICOS

- Estudiar los métodos de integración de sistemas informáticos entre tecnologías JAVA y .NET.
- Establecer una metodología de integración de sistemas informáticos entre tecnologías JAVA y .NET.
- Desarrollar un sistema de Gestión Académica integrable a la plataforma informática universitaria.
- Analizar los resultados de la aplicación de la metodología para integrar tecnologías Java y .Net.

1.3.3. JUSTIFICACIÓN

En la actualidad el uso de sistemas y aplicaciones informáticas en diversas instituciones públicas y privadas del país se ha incrementado en su totalidad gracias a las funcionalidades, facilidades y procesos que se pueden ejecutar dentro de ellos, estos sistemas ofrecen fluidez y eficiencia que van acorde a las tareas encomendadas en determinado departamento.

El uso de metodologías en proyectos o sistemas informáticos es indispensable y necesario porque se requiere una manera sistemática, controlada, empírica y crítica para llevarla a cabo y es necesario transformar los planteamientos iniciales en forma más precisa y estructurada, para obtener resultados que se acoplen de acuerdo a los objetivos planteados.

Hoy en día la integración de sistemas son una manera eficiente de cumplir con los objetivos de un negocio ya sea por las aplicaciones que se ejecuten previo a un orden jerárquico resultado de una metodología, o los avances tecnológicos en cada una de las paltaformas objeto de estudio, siendo una solución muy importante a la hora de escoger formas de desarrollar aplicaciones.

La razón más significativa en la realización del sistema de gestión académico es para evitar la pérdida y duplicidad de datos en la Secretaría de la Facultad, evitar demora al momento de entregar y emitir reportes de los procesos de los estudiantes, no colapsar la información y realizar varias peticiones al mismo tiempo con mayor rapidez y eficiencia.

En la Secretaría de la Facultad de Ingeniería el sistema de gestión académico propuesto se pretende integrar al sistema académico actual “SICOA” que utiliza la arquitectura .NET administrado por el departamento llamado UTECA, utilizando la tecnología JAVA con la arquitectura J2EE puesto que es una arquitectura orientada a servicios y se deberá analizar y escoger un método de integración para las arquitecturas antes mencionadas.

CAPÍTULO II

2. MARCO TEÓRICO

2.1. MÉTODOS DE INTEGRACIÓN DE SISTEMAS INFORMÁTICOS ENTRE TECNOLOGÍAS J2EE Y .NET

La incesante expansión de los negocios incrementa la complejidad y el riesgo inherente a la administración de sistemas como éstos. El valor real se encuentra en la interacción de los sistemas dentro de una red homogénea o heterogénea de servicios agregables, modulares y confiables.

Para agravar la magnitud del desafío, dos tecnologías líderes – Java y .NET – han divido el mercado en dos bandos, intensificando aún más las divergencias. Como muchas de las decisiones sobre la infraestructura de software se toman más a nivel departamental que a nivel empresarial, las compañías han visto cómo sus inversiones están asociadas a ambas tecnologías.

La situación obliga a las organizaciones de IT a dominar un alto grado de heterogeneidad para responder satisfactoriamente a los requerimientos fijados por sus respectivas compañías.

2.2. INTRODUCCIÓN A LA INTEGRACIÓN DE SISTEMAS

En las empresas más allá de lo estrictamente técnico, existen variadas razones que provocan la existencia de ambientes heterogéneos, ya que se combinan distintos sistemas operativos, bases de datos y plataformas de aplicaciones.

En la mayor parte de ellas el problema a resolver es establecer una conexión entre la lógica desarrollada en J2EE de tal forma que pueda ser accedida desde .Net o viceversa.

2.2.1. Conceptos de la integración:

- La integración de sistemas se refiere a la creación de uniones más firmes entre los diferentes sistemas de información.[1]

- La integración de sistemas se exige normalmente para lograr la integración de los negocios.[2] (Mendoza, 2002)

En algunos casos no es posible lograr los objetivos del negocio integrando sistemas, porque los sistemas individuales no contienen los datos necesarios o no están en un mismo formato para permitir el análisis deseado.

Lo ideal para una organización es tener sistemas integrados, donde se garantice que los sistemas de información sean capaces de ejecutarse en un amplio rango de plataformas de hardware y software, además que se manejen estándares tecnológicos para las bases de datos y las comunicaciones y sobretodo que haya capacidad para soportar en tiempo real los procesos claves del negocio a través de toda la organización y sin consideraciones de espacio.

2.3. ESTRATEGIAS DE INTEGRACIÓN DE TECNOLOGÍAS

Existen varias estrategias para la integración de sistemas, de los cuales se han analizado las siguientes alternativas para la integración de las plataformas J2EE (Java 2 Enterprise Edition) de Oracle y .Net de Microsoft.

La más sencilla es implementar una base de datos de uso compartido, donde las aplicaciones J2EE y .Net puedan acceder a través de JDBC y ADO .Net respectivamente para escribir y leer información, la integración mediante este método se puede desarrollar siempre y cuando los componentes de ambas aplicaciones se encuentren en la capa de negocios.

---

[1] *Sistemas de información y Tecnologías de Información. Mendoza, Luis. Pag. 4*
[2] *Sistemas de información y Tecnologías de Integración. Mendoza, Luis. Pag. 4*

Para interoperar componentes que se encuentran en capas diferentes, el mecanismo de integración dependerá del punto de interconexión que se esté considerando, (Figura 1).

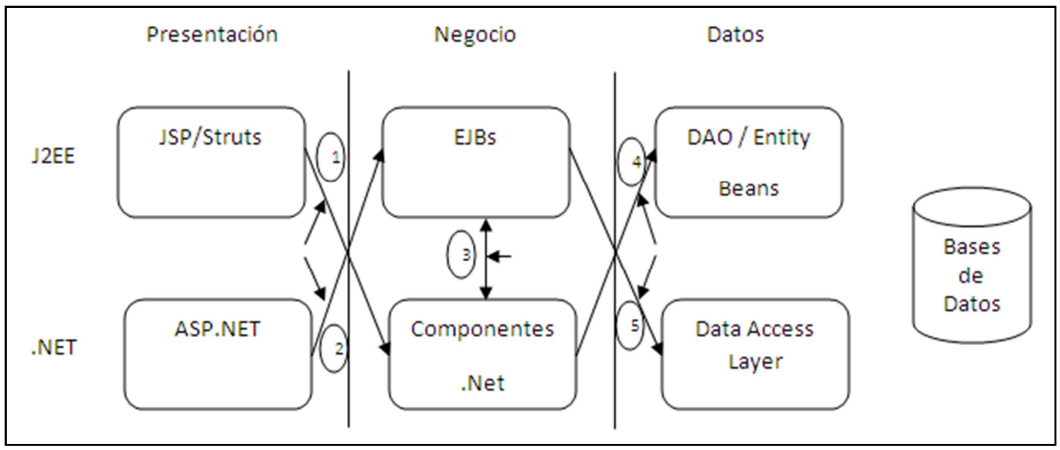

Figura 1: *Puntos de interconexión para aplicaciones multi-capa*
Fuente: *Microsoft (2013). Interoperabilidad entre .Net y J2EE. Disponible en: http://msdn.microsoft.com/es-es/library/bb972252.aspx*

En el caso de que el punto de interconexión se extienda a través de la intranet, la solución es el uso de Web Services, la ventaja del uso de Web Services es que al ser estándares abiertos permiten conservar la compatibilidad con otras aplicaciones, y la desventaja es estar limitada por la velocidad del enlace y el costo de emplear XML para la transmisión de mensajes, es decir que si no disponemos de un enlace permanente y confiable se debería utilizar servicios de mensajería asíncrona los mismos que se emplean entre las capas de negocios y de datos.

Lo que se hace es implementar colas de mensajes en cada una de las plataformas e interconectarlas a través de un bridge (cada aplicación produce y consume mensajes a través de la API habitual). Por otra parte, si podemos sacrificar el soporte de transacciones y callbacks, es posible generar un wrapper que permita el acceso a través de un Web Service.

Cuando se busca acceder a nivel de clases y métodos, la mejor opción es un runtime bridge, los mismos que se agregan en cada plataforma para administrar la interacción y comunicación entre objetos de diferente origen.

Se puede optar por una solución de más bajo nivel y llegar a eliminar el nivel de la comunicación remota aprovechando los métodos de interacción con código nativo que tiene cada plataforma.

Por último, si se está encarando una migración masiva a gran escala, lo que seguramente resulte más conveniente sea emplear herramientas de conversión de plataforma, las cuales se hacen cargo del mapeo de tipos y de proporcionar las APIs correspondientes.

2.3.1. WEB SERVICES

2.3.1.1. HISTORIA DE LOS WEB SERVICES

Desde la década de los 90 y aún más con la aparición de internet, las empresas desarrolladoras de software han tenido la necesidad e inquietud de buscar o contar con algún método para lograr integrar sistemas software y hardware. Para esto muchas compañías empezaron a crear de forma individual la mejor manera de lograr esta integración, el tiempo pasaba y la competencia entre ellas cada vez era más fuerte buscando tecnología integradora. El término Web Service como tal nace aproximadamente en el año 2000 por iniciativa de MS e IBM, Surgen como una necesidad de la industria en las áreas: o Enterprise Application Integration (EAI) o Business to Business (B2B). [3]

Debido al crecimiento del internet a niveles altos y el impacto causado por la tecnología de la información en las últimas dos décadas, la manera de hacer negocios y la comunicación entre las personas y las empresas cambió rotundamente. Bajo este contexto se hacía cada vez mayor la necesidad de integrar y compartir información entre distintas plataformas tanto de software como hardware.

Las empresas se dieron cuenta que era casi imposible crear una plataforma integrando de forma individual, es así que deciden atacar el problema de raíz. Para esto decidieron que en lugar de crear la mejor plataforma integradora, era

[3] *Web Services. Laboratorio de Integración de Sistemas. LINS. Pág. 3-5*

muchomejor buscar un lenguaje común de intercambio de información aprovechando los estándares existentes en el mercado. Es de esta forma que nacen los Web Services.

2.3.1.2. USO DE LOS WEB SERVICES

Para resolver un problema en cualquier organización, primero se necesita tener un claro entendimiento del negocio y sus necesidades, y de las herramientas que se pueden utilizar para resolver tal problema. Los Web Services son una solución poderosa, pero no siempre son la solución para todos los problemas del negocio. Los Web Services tienen su gran ventaja en la interoperabilidad; existen también situaciones en las que los clientes y proveedores poseen una plataforma en común que pueden proveer una solución optimizada para esa plataforma. Para elegir usar Web Services se debe conocer la infraestructura actual y además considerar cualquier cambio a futuro.

Los Web Services son comúnmente conocidos como una tecnología de internet, por el funcionamiento y su forma de trabajar colaborativamente con estándares y protocolos, es por tal razón que los Web Services son una solución para internet.

En otros asuntos tenemos que considerar algunos de los problemas típicos que se presenta en el internet, es así que si el problema presenta una transferencia para una gran cantidad de datos, los Web Services no serían la mejor opción. Entonces una de las principales consideraciones debería ser el ancho de banda que puede manejar la aplicación.

Una oportunidad ideal para usar Web Services es proveer información a disposición de compañías y personas externas, y no solamente esto, sino también un Web Service debe proveer funcionalidad como puede ser objetos y métodos que provean algún servicio.

2.3.1.3. VENTAJAS DE USAR WEB SERVICES

Los Web Services operan utilizando estándares abiertos, lo que permite la comunicación entre aplicaciones escritas en diferentes lenguajes y entre diferentes plataformas. Un Web Service puede ser utilizado para la comunicación con múltiples compañías. Comparativamente con otras soluciones, los Web Services son fáciles y no muy costosos de implementar, esto porque utiliza la infraestructura existente.[4]

Los Web Services pueden reducir significativamente los costos de comunicaciones negocio a negocio e integración de aplicaciones ofreciendo así un tangible retorne de inversión. La mayor ventaja de usar estándares abiertos, Word Wide Web Consortion (W3C) es una organización que define tecnologías para el Web, está encargada de asegurar que tales especificaciones se mantengan abiertas a cualquier vendedor. Además de Microsoft e IBM están promoviendo la interoperabilidad entre implementaciones de web services y son dos de los miembros fundadores de Web Services Interoperability Organization.

Los Web Services es la revolución informática de la nueva generación de aplicaciones que trabajan colaborativamente en las cuales el software está distribuido en diferentes servidores. Es un conjunto de protocolos cuya utilidad específica es permitir el intercambio de información entre aplicaciones ya sea que estas estén programadas en diferentes tecnologías con lenguajes diferente.

La integración de las aplicaciones mediante Web Services se realiza mediante el uso de los protocolos de Internet XML, SOAP, WSDL y UDDI, donde XML es usado para describir los datos, SOAP se ocupa de la transferencia de los datos, WSDL permite describir los servicios disponibles y UDDI permite conocer los servicios disponible.

---

[4] *Construcción de un Brechmark para realizar un estudio comparativo entre web services en plataformas JAVA y .NET. Marmol Panami Miguel Patricio, Peñaherrera Pacheco Deysi de las Mercedes, Pág. 9-10*

Los Web Services permiten el intercambio de datos sin necesidad de conocer los detalles de sus respectivos sistemas de información, a diferencia de los modelos Cliente/Servidor como páginas Web, los Web Services no proveen al usuario de una interfaz gráfica en lugar de ello comparten la lógica de negocios, los datos y los procesos por medio de una interfaz de programas a través de la red, (Figura 2). Es decir se conectan con programas no interactúan directamente con el usuario.

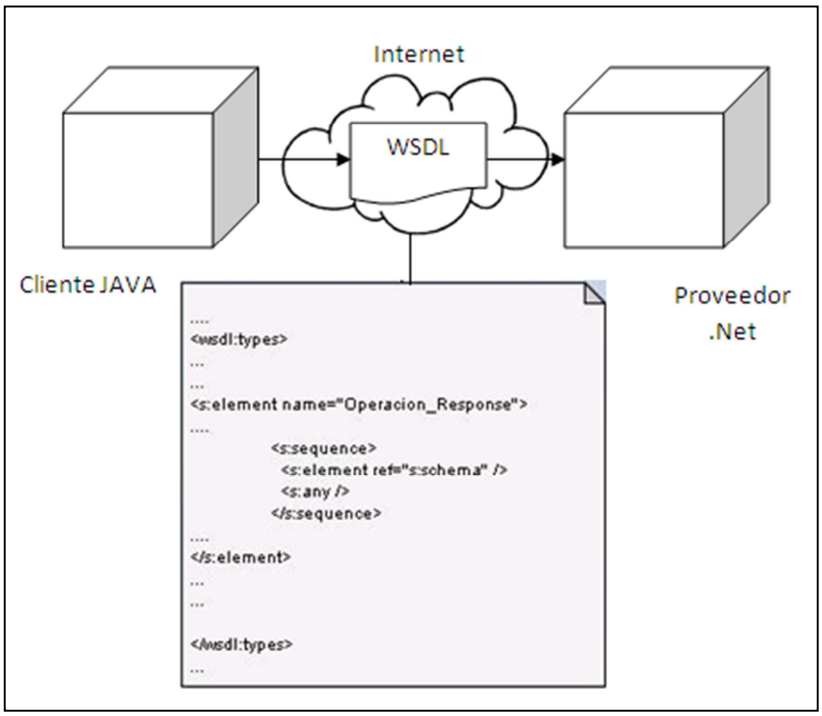

Figura 2: *Web service enre Java y .Net*
Autor: *Johnny Latta*

2.3.1.4. TECNOLOGÍA WEB SERVICES

Los Web Services están construidos con varias tecnologías que trabajan conjuntamente con los estándares permitiendo seguridad y operatibilidad, de tal manera que el uso combinado de varios Web Services sea independiente de la o las empresas que los proveen. A continuación se describen brevemente estos estándares.

El XML (Extensible MarkupLanguage) que permite a los desarrolladores crear sus propios tags, que les permiten habilitar definiciones, transmisiones,

validaciones, e interpretaciones de los datos entre aplicaciones y entre organizaciones.[5]

Soap es un protocolo estándar que define cómo dos objetos en diferentes procesos pueden comunicarse por medio de intercambio de datos XML.[6] Este protocolo deriva de un protocolo creado por David Winer en 1998, llamadoXML-RPC. SOAP fue creado por Microsoft, IBM y otros y está actualmente bajo el auspicio de la W3C. Es uno de los protocolos utilizados en los servicios Web.

Simple Object Access Protocol (SOAP) es un protocolo ligero basado en XML para intercambiar información estructurada y con tipo. SOAP es el que permite usar la funcionalidad de máquinas remotas sin necesidad de saber algo específico sobre la máquina. XML es usado en los Web Services para representar datos, y los datos necesitan tener un esquema en común, para esto SOAP es el que define el esquema.

La especificación SOAP divide a un mensaje SOAP en cuatro partes, de ésta solo uno es obligatorio y es la envoltura SOAP que encapsula el mensaje. La segunda parte describe los tipos de datos definidos en la aplicación, la tercera parte describe un patrón de intercambio de mensaje de petición o respuesta llamado de procedimiento remoto (RPC remote Procedure Call) y la cuarta parte del mensaje SOAP define una conexión entre SOAP y HTTP.

Cada mensaje SOAP contiene un elemento envoltura compuesto por un elemento opcional, la cabecera y un elemento requerido cuerpo. La cabecera puede contener información sobre el mensaje, instrucciones de conversión para nodos que reciben el mensaje de información de seguridad. El cuerpo describe el propósito del mensaje SOAP y contiene la carga del mensaje tal como datos o instrucciones para la aplicación receptora.

Un mensaje de intercambio SOAP se produce de la siguiente forma:

---

[5] *Herramienta Didáctica para el Aprendizaje y Esfuerzo de verbos en la lengua francófona a través de un web service en Java. Valbuena, Leonardo. Pág. 39*
[6] *SOAP. Hernadez, Ricardo. Pag. 32*

La aplicación del consumidor crea un mensaje SOAP que es una petición para invocar al web Service provisto por el proveedor de servicio. El documento XML en el cuerpo del mensaje puede ser una petición SOAP RPC como es indicado en la descripción del servicio. El cliente SOAP interactúa con el protocolo de red apropiado como por ejemplo HTTP para enviar el mensaje de red.

Uno de los propósitos del mensaje SOAP es ejecutar un procedimiento a través de la red, es por eso que un mensaje SOAP enviado por la red hacia un web Service es un RPC. Cuando se entrega el mensaje al proveedor de servicio, el servidor SOAP enruta el mensaje hacia el web Service y es el responsable de convertir el mensaje XML en objetos comprensibles por el lenguaje de programación utilizado donde se utiliza las reglas de codificación incluidas en el mensaje SOAP.

El web Service es el responsable de procesar el mensaje y conceder una respuesta que es también un mensaje SOAP e igualmente se envía el mensaje por la red con el protocolo establecido.

Cuando la respuesta llega, el mensaje XML es convertido en objetos específicos del lenguaje de programación del consumidor y es presentado a la aplicación correspondiente.

2.3.1.5. ALTERNATIVA DE SOAP

Algunas compañías han usado otra tecnología XML RPC en lugar de SOAP para algunas implementaciones. Según IBM el protocolo XML reemplazará a SOAP como el estándar en la industria como protocolo de mensajería, para esto el W3C tiene que lanzar el protocolo XML como estándar. Además de utilizar SOAP o XML, algunas empresas pueden desarrollar sus propias alternativas basadas en XML para la comunicación de mensajes, pero la desventaja es que se requiere gran experiencia en esos desarrollos y los mayores vendedores de software han adoptado productos SOAP que ofrecen mejores características y funcionalidad.

Algunas aplicaciones para implementar SOAP en sistemas empresariales son: SOAP Toolkit de Microsoft, Web Services Toolkit de IBM y Axis de Apache.

SOAP Web ServicesDescriptionLanguage es un lenguaje específico de XML que define a los Web Services como puntos de comunicación capaces de intercambiar mensajes.[7]

Es por la descripción del servicio que un proveedor de servicio comunica todas las especificaciones para que el consumidor del servicio invoque al web Service. Esta descripción del servicio es parte clave en la estructura de los web Service poco acoplada. Web Services Description Languaje (WSDL) es un lenguaje basado en XML que es reutilizado por la aplicación cliente para obtener información técnica del web Service con la cual se puede comunicar.

Muchos web services publican el documento WSDL por el internet que contiene definiciones que describen al web Service en formato XML. El documento WSDL especifica las capacidades que tiene el servicio, su ubicación en la web e instrucciones sobre como acceder al web Service. Un documento WSDL define cual va ser la estructura de los mensajes que el web Service puede enviar o recibir.

Cuando el web Service es publicado, el administrador publica un enlace hacia el documento WSDL del web Service en alguno de los registros XML u otro repositorio, entonces, el archivo WSDL está disponible cuando aún aplicación busca el registro y logra ubicar el Web Service. El cliente accede al documento WSDL para obtener la información sobre el web Service y para crear los mensajes SOAP con la correcta estructura. Utilizando esta información, la aplicación cliente invoca al Web Service.

El documento WSDL puede estar publicado ya sea un repositorio o en cualquier ya que es un URL que redirecciona hacia este documento.

UDDI Es un protocolo basado en XML que describe los accesos al Web Service. Se dice que es el manual de operación del mismo, porque indica cuáles son las interfaces que provee el Servicio web y los tipos de datos necesarios para su utilización.[8]

7SCS – Sistemas Cliente/Servidor. FJRP, FMBR. Pág. 12 - 14

8Web Services con PHP. Brea, Orlando. Pág. 4 – 5.

Universal Description, Discovery and Integration (UDDI) permite la creación de registros web service para ser buscados por consumidores. Las especificaciones UDDI definen la manera para publicar y descubrir información sobre los web services. Descubrimiento (Discovery) es el proceso de ubicar un web Service por medio de registros. Se usa un registro, una empresa puede localizar todos los servicios disponibles requeridos y puede comparar entre los servicios ofrecidos para ubicar el que llene sus expectativas.

La información de UDDI se aloja en nodos de operador, empresas que se han comprometido a ejecutar un nodo publico conforme a la especificación que rige el consorcio UDDI.org. En la actualidad existen dos nodos públicos que se ajustan a la versión 1 de la especificación UDDI: Microsoft aloja uno e IBM otro. Hewlett Packard se ha comprometido a alojar un nodo bajo la versión 2 de la especificación. Los operadores del host deben replicar datos entre ellos a través de un canal seguir, para conseguir la redundancia de la información en el registro UDDI. Se pueden publicar los datos en un nodo y descubrirlos en otro tras la réplica. Actualmente, la réplica se produce cada 24 horas. En el futuro, este intervalo entre replicas se reducirá, ya que habrá más aplicaciones que dependan de los datos de UDDI.

2.3.1.6. OTRAS TECNOLOGÍAS

EbXML, define su propia estructura de registro por medio del cual los consumidores de servicios pueden acceder a documentos XML que contienen información sobre los proveedores de servicio. El registro permite que las compañías inicien acuerdos y realicen transacciones.

Ws-Inspection, es una tecnología desarrollada por Microsoft e IBM que define como una aplicación cliente puede localizar descripciones de WS que residen en un servidor SW. Un documento WS-Inspection es mantenido por el proveedor de servicio y contiene referencias hacia los documentos de descripción de WS como WSDL en el servicio. WS Inspecion entonces sirve cuando desarrolladores conocen el servicio que contiene los WS que pueden utilizar.

2.3.2. JAVA CONNECTOR ARCHITECTURE (JCA)

J2EE ConnectorArchitecture es un estándar de desarrollo de Java CommunityProcess,basada en Java, para permitir la integración entre servidores de aplicaciones y EIS (Enterprise Information Servers), el objetivo principal de JCA es que los vendedores de servidores de aplicaciones extiendan sus productos para soportar la especificación JCA y los vendedores de EIS proveer de los llamados resourceadapters para cada uno de sus EIS, los mismos que se registran en un servidor de aplicaciones, quedando disponibles para ser utilizados por los componentes que residen en el mismo, mediante un mecanismo homogéneo API llamado CommonClient Interface o CCI.

Así, un servidor que da soporte a JCA asegura la conectividad con múltiples sistemas y productos de diferentes vendedores, y un producto que provee de un adaptador de recursos tiene la capacidad de ser utilizado por cualquier servidor que de soporte a JCA. (Barrios, 2003)

La arquitectura de JCA tiene tres componentes principales, los contratos de nivel de sistema entre el adaptador de recursos y el servidor de aplicaciones, la interfaz de cliente común que proporciona una API de cliente para las aplicaciones Java y herramientas de desarrollo para acceder al adaptador de recursos, y un embalaje estándar y la facilidad de implementación para los adaptadores de recursos (Figura 3).

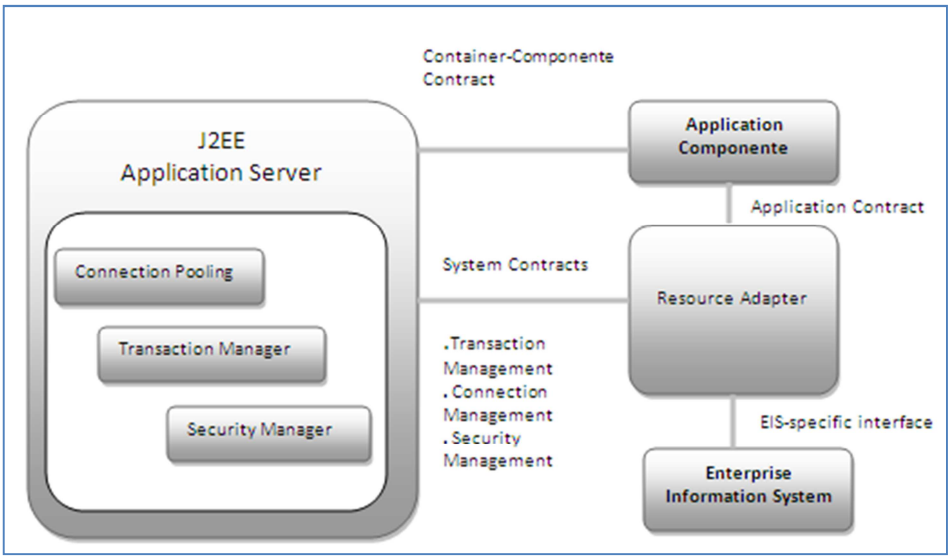

Figura 3: *Arquitectura de J2EE ConnectorArchitecture*
Fuente: *Desarrollo de aplicaciones con J2EE. Disponible en: ftp://ftp.etsii.ull.es/pub/INGSOF3/tema5.PDF*

Los servidores de aplicaciones que implementan JCA deben cumplir una serie de contratos, como los resourceadapters de cada EIS para definir mecanismos de escalabilidad, seguridad y permitir manejos transaccionales entre los mismos. Los tres tipos de contratos que se establecen son:
Manejo de Conexiones: Permite el manejo homogéneo de las conexiones, provee una interfaz común para el manejo de la conexión y un pool de conexiones permitiendo escalabilidad de la solución.

Seguridad: Extiende los contratos de manejo de conexiones con detalles específicos a la seguridad. Como permitir pasar credenciales desde el servidor de aplicaciones al resourceadapter. (Rodríguez, Vignaga, Zipitría).

Manejo de transacciones: Permite establecer un ambiente transaccional al trabajar con distintas EIS de una forma completamente transparente.

JCA está basada en la existencia de resourceadapters específicos de cada EIS, los cuales son componentes de software que implementan los contratos por el lado de los EIS, sirviendo de intermediarios entre el servidor de aplicaciones y los EIS,

mientras el servidor de aplicaciones se extiende para soportar JCA, implementando los contratos.

Servidores de Aplicaciones con soporte JCA

Existen en la actualidad varios servidores de aplicaciones J2EE con soporte para J2EE ConnectorArchitecture estos son:

Tabla 1: *Servidores de Aplicaciones con soporte JCA*

Vendedor/Producto	Versión JCA
BEA WEBLogicPreview 7.0	1.0
Borland Enterprise Server 5.0	1.0
IBM WebSphere Technology for Developers	1.0
MacomediaJRun Server TecnologyPreview	1.0
Pramati Server 3.0	1.0
SilverStreameXtend App Server 4.0 Beta	1.0
SybaseEAServer 4.1	1.0
TriforkApplication Server 3.0	1.0

Autor: *Johnny Latta*

2.3.3. JA.NET

Ja.Net es un producto creado por la empresa Intrinsyc Software que provee un bridge (puente) entre Java y Microsoft.Net, permite el acceso a EJBs desde cualquier lenguaje provisto por .Net y el acceso a componentes .Net desde cualquier aplicación Java, inclusive EJBs.

Se basa en la comunicación de managed componentes (CLR) que estén en diferentes dominios a través de .Net remoting en la plataforma .Net, ya que al hacer uso de .Net remoting Ja.Net permite que los componentes Java aparezcan como componentes CLR y que los componentes CLR aparezcan como componentes Java.

2.3.4. ACTIVEX BRIDGE

ActiveX Bridge es una tecnología agregada al servidor de aplicaciones IBM WebSphere a partir de la versión 4.0, que permiten que aplicaciones COM puedan acceder a componentes EJB y otros servicios disponibles de la plataforma J2EE que se encuentren disponibles en el servidor.

Dicho proceso se realiza mediante el uso de un JVM (Java Virtual Machine) que corre dentro del proceso ActiveX la misma que se comunica con el servidor, mientras el COM a través del ActiveX bridge realiza las invocaciones correspondientes a la JVM de la siguiente manera, (Figura 4):

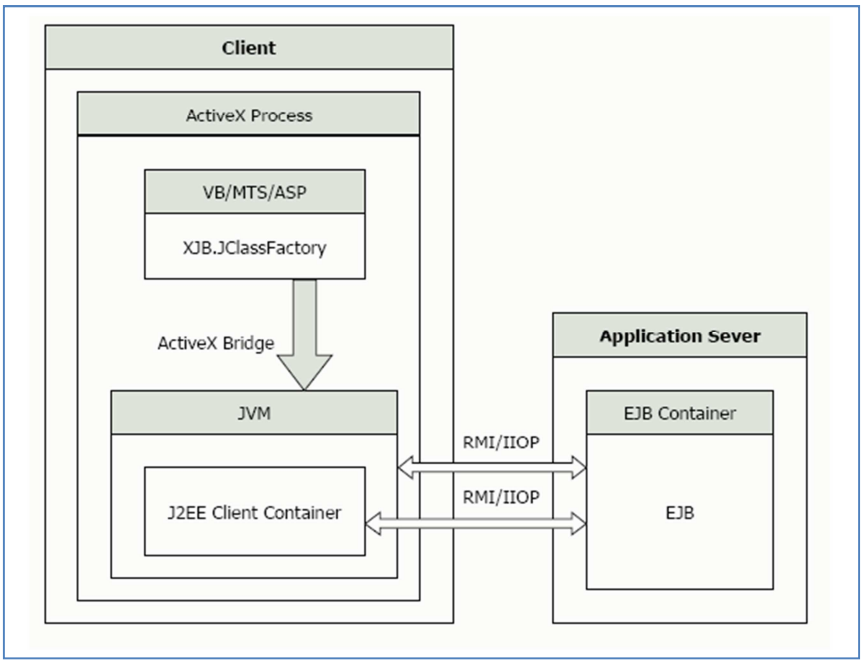

Figura 4:*Arquitectura ActiveX Bridge*
Fuente: *Instituto de Computación. Estudio de Inteoperabilidad .Net y J2EE. Disponible en: http://www.consultec.es/DocInformes/J2EE%20y%20NET.pdf*

El bridge a más del acceso a componentes EJB permite el acceso a las APIs de Java ya sea JNDI (Java Naming and Directory Interface), JDBC (Java DatabaseConnectivity), JMS (Java MessagingService), entre otras, el bridge soporta manejo de control de cargas y seguridad, pero no soporta transacciones distribuidas que involucren ambos ambientes. Otra de las desventajas de esta tecnología es que funciona en un solo sentido es decir solo es posible invocar servicios o componentes Java desde COM y no viceversa.

2.3.4.1. Soporte de ActiveX Bridge: ActiveX Bridge está soportada para ejecutarse sobre:

- Visual Basic
- VBScript
- ASP sobre plataformas NT o Windows 2000 o superior

2.4. ARQUITECTURA J2EE

La arquitectura J2EE es una plataforma de desarrollo que fue creada por SUN en 1997 y en la actualidad pertenece a ORACLE, basa su arquitectura en productos de software libre, con una arquitectura definida en conceptos de capas, containers, componentes, servicios y sus características.

Las arquitecturas J2EE crea aplicaciones empresariales utilizando un modelo de multicapas, dividiendo la aplicación en diferentes niveles: la capa cliente, la capa web, la capa negocio y la capa datos. La figura 5, representa estas capas y los componentes relacionados.

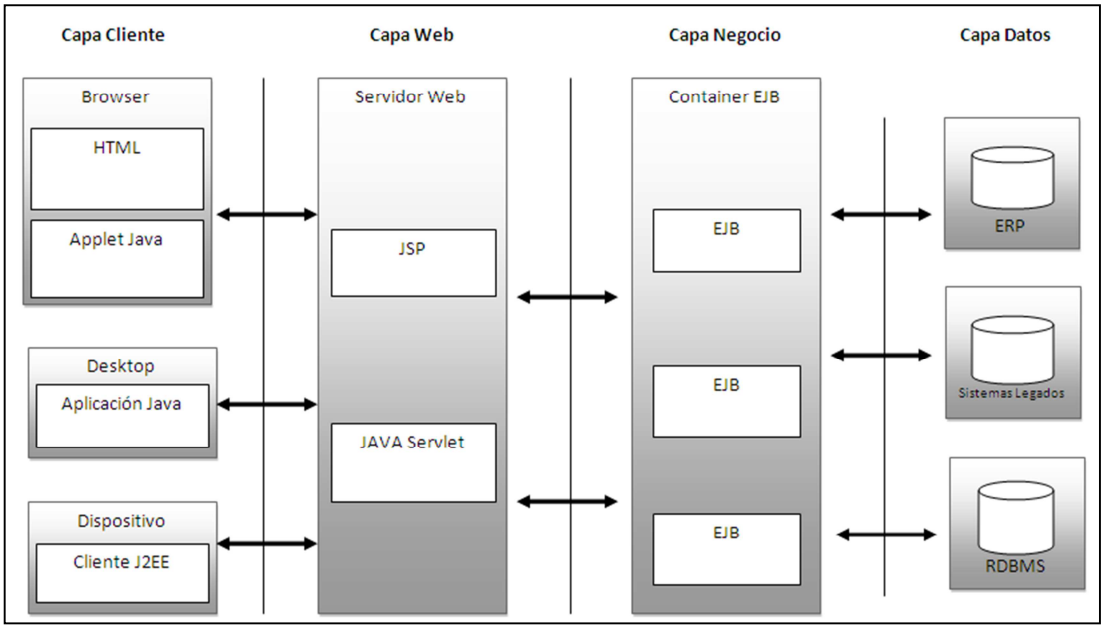

Figura 5: *Arquitectura J2EE*
Fuente: *Uniersidad de Chile. Arquitectura J2EE. Disponible en: http://users.dcc.uchile.cl/~jbarrios/J2EE/node14.html*

2.4.1. Capa Cliente: Esta capa corresponde a la interfaz gráfica del sistema y permite interactuar con el usuario, J2EE soporta diferentes tipos de clientes incluyendo clientes HTML, applets Java y aplicaciones Java.

2.4.2. Capa Web: Se encuentra en el servidor web y contiene la lógica de presentación que se utiliza para generar una respuesta al cliente. Recibe los datos del usuario desde la capa cliente y basado en éstos genera una respuesta apropiada a la solicitud. J2EE utiliza en esta capa las componentes Java Servlets y JavaServerPages para crear los datos que se enviarán al cliente.

2.4.3. Capa Negocio: Se encuentra en el servidor de aplicaciones y contiene el núcleo de la lógica del negocio de la aplicación. Provee las interfaces necesarias para utilizar el servicio de componentes del negocio. Las componentes del negocio interactúan con la capa de datos y son típicamente implementadas como componentes EJB.

2.4.4. Capa Datos: Esta capa es responsable del sistema de información de la empresa o Enterprise InformationSystem (EIS) que incluye bases de datos, sistema de procesamiento datos, sistemas legados y sistemas de planificación de recursos. Esta capa es el punto donde las aplicaciones J2EE se integran con otros sistemas no J2EE o con sistemas legados.

2.4.5. COMPONENTES DE J2EE

Cada componente de J2EE es una unidad de software independiente y funcional que cumple con las condiciones de interfaz definidas por la especificación del componente y sólo tiene dependencias explícitas con su entorno de ejecución. Un componente puede estar compuesto por una o por un conjunto de clases, interfaces y recursos. Estos componentes son:

- **Aplicación cliente:** Representa la interfaz del usuario que maneja el cliente.

- **Applets:** Se ejecutan en un browser y proporcionan una interfaz web mejorada para aplicaciones J2EE.

- **Java Servlets y JavaServerPages:** También llamados componentes web, se ejecutan del lado del servidor web respondiendo a solicitudes HTTP realizadas por el cliente.

- **Enterprise JavaBeans:** Contienen la lógica de negocios, se ejecuta en un ambiente distribuido y que soporta transacciones.

2.4.6. INTEGRACIÓN DE J2EE CON OTROS SISTEMAS

La plataforma J2EE es integrable puesto que está basada en CORBA lo que le permite comunicase sin ningún problema con otras aplicaciones creadas en lenguajes diferentes de Java y viceversa, además permite exponer cualquier EJB como un servicio web facilitando la integración con otras plataformas, otras de las herramientas que posee para la integración son los conectores que poseen una API estándar y sirven como puente entre J2EE y sistemas legacy.

2.5. ARQUITECTURA .NET

Microsoft .Net es una plataforma de desarrollo y ejecución de aplicaciones que además de brindar herramientas y servicios para el desarrollo de aplicaciones empresariales también provee de mecanismos robustos, seguros y eficientes para asegurar que la ejecución de las mismas sean óptimas (Figura 6).

.NET ofrece una infraestructura tecnológica flexible, capaz de adaptarse rápidamente a los cambios del negocio, que es capaz de comunicarse a través de estándares y que en general busca la creación de fáciles de integrar, de modificar, y agilizar la implementación de cualquier tipo de soluciones, para cualquier tipo de dispositivo.

Los principales componentes de la plataforma .Net son:

- **Runtime:** Es el entorno de ejecución de la aplicación .Net.

- **Bibliotecas de funcionalidades y controles reutilizables:** Permite el consumo de componentes ya programados en otras aplicaciones.

- Conjunto de lenguajes de programación de alto nivel, compiladores y linkers que facilitan el desarrollo de aplicaciones.

- Utilitarios y herramientas de desarrollo que facilitan el desarrollo de las aplicaciones.

- Documentación y guías de arquitectura: Describen las mejores prácticas de diseño, organización, desarrollo, prueba e instalación de aplicaciones .NET.

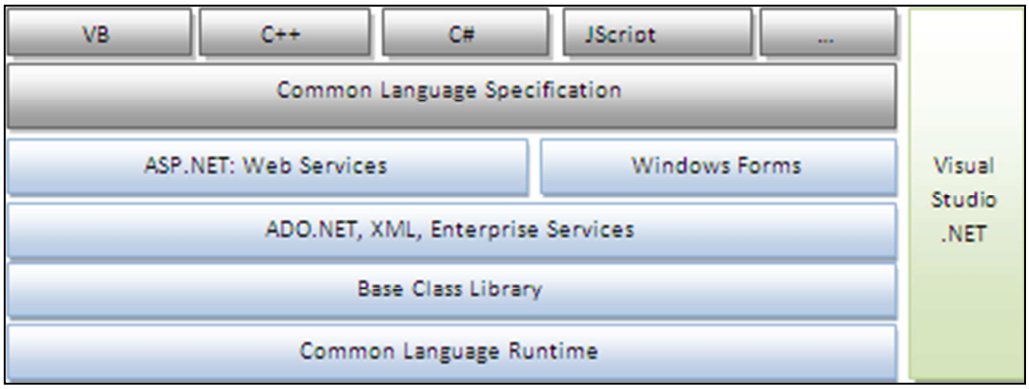

Figura 6: *Arquitectura .Net*
Fuente: *Introducting Microsoft DotNet. Disponible en: httpaspalliance.com264_Introducting_Microsoft_DotNet.3.txt*

2.5.1. VENTAJAS DE .NET

Las principales ventajas de la plataforma .NET son:

- **Interoperatibilidad Multilenguaje:** Soporta aplicaciones con componentes en múltiples lenguajes lo que permite integrar desarrolladores de distintos perfiles.

- **Documentación:** Ofrece documentación de ayuda como herramientas, debuggers, editores incluida en el IDE y de soporte.

- **Rendimiento:** Los códigos que se ejecutan en .NET son compilados.

- **Rápido Aprendizaje:** Es sencillo de aprender por la documentación y el soporte de ayuda.

- **Movilidad:** Las aplicaciones pueden ser desplegadas en una gran variedad de dispositivos.

- **Escalabilidad:** .NET ofrece métodos de escalabilidad como la carga balanceada que permite a un clúster de servidores colaborar y dar un servicio de forma simultánea.

- **Flexibilidad:** El modo de programación que se emplea permite agregar nuevos módulos sin modificar la aplicación en su totalidad.

- **Seguridad:** Da respaldo para ejecutar código no seguro.

- **Estándar abierto:** .NET se basa en estándares abiertos como HTML, XML, SOAP, WSDL, UDDI.

2.5.2. INTEGRACIÓN DE .NET CON OTROS SISTEMAS

La plataforma .Net puede interactuar e integrarse fácilmente con aplicaciones desarrolladas en plataformas anteriores y particularmente con COM; .NET no solo se integra con aplicaciones desarrolladas en plataformas Microsoft sino también en otras plataformas de software, sistemas operativos o lenguajes de programación mediante varios estándares globales tales como: XML, HTTP, SOAP, WSDL y UDDI.

2.6. NET FRAMEWORK 2.0

NET Framework es un componente integral de Windows que admite la creación y la ejecución de la siguiente generación de aplicaciones y servicios Web XML.[9] El diseño de .NET Framework está enfocado a cumplir los objetivos siguientes:

- Proporcionar un entorno coherente de programación orientada a objetos, en el que el código de los objetos se pueda almacenar y ejecutar de forma local, ejecutar de forma local pero distribuida en Internet o ejecutar de forma remota.

- Proporcionar un entorno de ejecución de código que reduzca lo máximo posible la implementación de software y los conflictos de versiones.

- Ofrecer un entorno de ejecución de código que fomente la ejecución segura del mismo, incluso del creado por terceras personas desconocidas o que no son de plena confianza.

- Proporcionar un entorno de ejecución de código que elimine los problemas de rendimiento de los entornos en los que se utilizan secuencias de comandos o intérpretes de comandos.

---

[9] *Información general y conceptual sobre .NET Framework. MSDN Microsoft.com. Pág. 1*

- Ofrecer al programador una experiencia coherente entre tipos de aplicaciones muy diferentes, como las basadas en Windows o en el Web.

- Basar toda la comunicación en estándares del sector para asegurar que el código de .NET Framework se puede integrar con otros tipos de código.

.NET Framework contiene dos componentes principales: Common Language Runtime y la biblioteca de clases de .NET Framework. Common Language Runtime es el fundamento de la tecnología. El motor en tiempo de ejecución se puede considerar como un agente que administra el código en tiempo de ejecución y proporciona servicios centrales, como la administración de memoria, la administración de subprocesos y la interacción remota, al tiempo que aplica una seguridad estricta a los tipos y otras formas de especificación del código que fomentan su seguridad y solidez.

De hecho, el concepto de administración de código es un principio básico del motor en tiempo de ejecución. El código destinado al motor en tiempo de ejecución se denomina código administrado, a diferencia del resto de código, que se conoce como código no administrado. La biblioteca de clases, el otro componente principal de .NET Framework, es una completa colección orientada a objetos de tipos reutilizables que se pueden emplear para desarrollar aplicaciones que abarcan desde las tradicionales herramientas de interfaz gráfica de usuario (GUI) o de línea de comandos hasta las aplicaciones basadas en las innovaciones más recientes proporcionadas por ASP.NET, como los formularios Web Forms y los servicios Web XML.

.NET Framework puede alojarse en componentes no administrados que cargan Common Language Runtime en sus procesos e inician la ejecución de código administrado, con lo que se crea un entorno de software en el que se pueden utilizar características administradas y no administradas. En .NET Framework no sólo se ofrecen varios hosts de motor en tiempo de ejecución, sino que también se admite el desarrollo de estos hosts por parte de terceros.

Por ejemplo, ASP.NET aloja el motor en tiempo de ejecución para proporcionar un entorno de servidor escalable para el código administrado. ASP.NET trabaja directamente con el motor en tiempo de ejecución para habilitar aplicaciones de ASP.NET y servicios Web XML.

Internet Explorer es un ejemplo de aplicación no administrada que aloja el motor en tiempo de ejecución (en forma de una extensión de tipo MIME). Al usar Internet Explorer para alojar el motor en tiempo de ejecución, puede incrustar componentes administrados o controles de Windows Forms en documentos HTML.

En la figura 7, se muestra la relación de Common Language Runtime y la biblioteca de clases con las aplicaciones y el sistema en su conjunto. En la ilustración se representa igualmente cómo funciona el código administrado dentro de una arquitectura mayor.

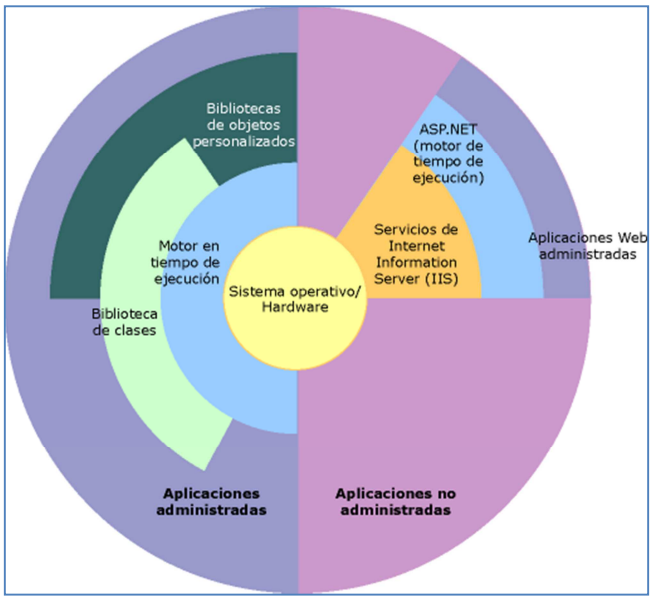

Figura 7:.*NET Framework en contexto*
Fuente: *Información general acerca de .Net Framework. Disponible en: http://msdn.microsoft.com/es-es/library/zw4w595w.aspx*

2.6.1. CARACTERÍSTICAS DE COMMON LANGUAGE RUNTIME

Common Language Runtime administra la memoria, ejecución de subprocesos, ejecución de código, comprobación de la seguridad del código, compilación y demás servicios del sistema. Estas características son intrínsecas del código administrado que se ejecuta en Common Language Runtime.

Con respecto a la seguridad, los componentes administrados reciben grados de confianza diferentes, en función de una serie de factores entre los que se incluye su origen (como Internet, red empresarial o equipo local). Esto significa que un componente administrado puede ser capaz o no de realizar operaciones de acceso a archivos, operaciones de acceso al Registro y otras funciones delicadas, incluso si se está utilizando en la misma aplicación activa.

El motor en tiempo de ejecución impone seguridad en el acceso al código. Por ejemplo, los usuarios pueden confiar en que un archivo ejecutable incrustado en una página Web puede reproducir una animación en la pantalla o entonar una canción, pero no puede tener acceso a sus datos personales, sistema de archivos o red. Por ello, las características de seguridad del motor en tiempo de ejecución permiten que el software legítimo implementado en Internet sea excepcionalmente variado.

Además, el motor en tiempo de ejecución impone la solidez del código mediante la implementación de una infraestructura estricta de comprobación de tipos y código denominado CTS (Common Type System, Sistema de tipos común). CTS garantiza que todo el código administrado es auto descriptivo. Los diferentes compiladores de lenguajes de Microsoft y de terceros generan código administrado que se ajusta a CTS. Esto significa que el código administrado puede usar otros tipos e instancias administrados, al tiempo que se aplica inflexiblemente la fidelidad y seguridad de los tipos.

Además, el entorno administrado del motor en tiempo de ejecución elimina muchos problemas de software comunes. Por ejemplo, el motor en tiempo de

ejecución controla automáticamente la disposición de los objetos, administra las referencias a éstos y los libera cuando ya no se utilizan. Esta administración automática de la memoria soluciona los dos errores más comunes de las aplicaciones: la pérdida de memoria y las referencias no válidas a la memoria.

Además, el motor en tiempo de ejecución aumenta la productividad del programador. Por ejemplo, los desarrolladores pueden crear aplicaciones en el lenguaje que prefieran y seguir sacando todo el provecho del motor en tiempo de ejecución, la biblioteca de clases y los componentes escritos en otros lenguajes por otros colegas. El proveedor de un compilador puede elegir destinarlo al motor en tiempo de ejecución. Los compiladores de lenguajes que se destinan a .NET Framework hacen que las características de .NET Framework estén disponibles para el código existente escrito en dicho lenguaje, lo que facilita enormemente el proceso de migración de las aplicaciones existentes.

Aunque el motor en tiempo de ejecución está diseñado para el software del futuro, también es compatible con el software actual y el software antiguo. La interoperabilidad entre el código administrado y no administrado permite que los desarrolladores continúen utilizando los componentes COM y las DLL que necesiten.
El motor en tiempo de ejecución está diseñado para mejorar el rendimiento. Aunque Common Language Runtime proporciona muchos servicios estándar de motor en tiempo de ejecución, el código administrado nunca se interpreta. Una característica denominada compilación JIT (Just-In-Time) permite ejecutar todo el código administrado en el lenguaje máquina nativo del sistema en el que se ejecuta. Mientras tanto, el administrador de memoria evita que la memoria se pueda fragmentar y aumenta la zona de referencia de la memoria para mejorar aún más el rendimiento.

Por último, el motor en tiempo de ejecución se puede alojar en aplicaciones de servidor de gran rendimiento, como Microsoft® SQL Server™ e IIS (Servicios de Internet Information Server). Esta infraestructura permite utilizar código

administrado para escribir lógica empresarial, al tiempo que se disfruta del superior rendimiento de los mejores servidores empresariales del sector que pueda alojar el motor en tiempo de ejecución.

2.6.2. BIBLIOTECA DE CLASES DE .NET FRAMEWORK

La biblioteca de clases de .NET Framework es una colección de tipos reutilizables que se integran estrechamente con Common Language Runtime. La biblioteca de clases está orientada a objetos, lo que proporciona tipos de los que su propio código administrado puede derivar funciones. Esto ocasiona que los tipos de .NET Framework sean sencillos de utilizar y reduce el tiempo asociado con el aprendizaje de las nuevas características de .NET Framework. Además, los componentes de terceros se pueden integrar sin dificultades con las clases de .NET Framework.

Por ejemplo, las clases de colección de .NET Framework implementan un conjunto de interfaces que puede usar para desarrollar sus propias clases de colección. Éstas se combinarán fácilmente con las clases de .NET Framework.

Como en cualquier biblioteca de clases orientada a objetos, los tipos de .NET Framework permiten realizar diversas tareas de programación comunes, como son la administración de cadenas, recopilación de datos, conectividad de bases de datos y acceso a archivos. Además de estas tareas habituales, la biblioteca de clases incluye tipos adecuados para diversos escenarios de desarrollo especializados. Por ejemplo, puede utilizar .NET Framework para desarrollar los siguientes tipos de aplicaciones y servicios:

- Aplicaciones de consola
- Aplicaciones GUI de Windows (formularios Windows Forms)
- Aplicaciones de ASP.NET
- Servicios Web XML
- Servicios de Windows

Por ejemplo, las clases de formularios Windows Forms son un conjunto completo de tipos reutilizables que simplifican enormemente el desarrollo de interfaces GUI para Windows. Si escribe una aplicación Web Form de ASP.NET, puede utilizar las clases de formularios Web Forms.

2.6.3. DESARROLLO DE APLICACIONES CLIENTE

Las aplicaciones cliente constituyen lo más parecido a una aplicación de estilo tradicional en la programación basada en Windows. En este tipo de aplicaciones se muestran ventanas o formularios en el escritorio, lo que permite al usuario realizar una tarea. Entre las aplicaciones cliente se incluyen los procesadores de texto y las hojas de cálculo, además de aplicaciones empresariales, como herramientas de entrada de datos, de informes, etcétera. En las aplicaciones cliente se suelen emplear ventanas, menús, botones y otros elementos de la interfaz gráfica de usuario, y suelen tener acceso a recursos locales como el sistema de archivos y a dispositivos periféricos como las impresoras.

Otro tipo de aplicación cliente es el tradicional control ActiveX (reemplazado ahora por el control de Windows Forms) implementado en Internet como una página Web. Esta aplicación es muy parecida a otras aplicaciones cliente: se ejecuta de forma nativa, tiene acceso a los recursos locales e incluye elementos gráficos.

En el pasado, los desarrolladores creaban esas aplicaciones mediante C o C++ en combinación con MFC (Microsoft Foundation Classes) o con un entorno RAD (Rapid Application Development, desarrollo rápido de aplicaciones) como Microsoft® Visual Basic®. En .NET Framework se incorporan aspectos de estos productos, que siguen existiendo, en un único entorno de desarrollo coherente que simplifica de forma espectacular el desarrollo de las aplicaciones cliente.

Las clases de formularios Windows Forms contenidas en .NET Framework están diseñadas para utilizarse en el desarrollo de GUI. Puede crear ventanas, botones,

menús, barras de herramientas y demás elementos de pantalla fácilmente con la flexibilidad requerida para adaptarse a la evolución de las necesidades de su empresa.

Por ejemplo, .NET Framework proporciona propiedades simples para ajustar los atributos visuales asociados con los formularios. En determinadas circunstancias, el sistema operativo subyacente no permite cambiar estos atributos directamente y, entonces, .NET Framework vuelve a crear los formularios de forma automática. Ésta es una de las múltiples maneras en que .NET Framework integra la interfaz del programador, con lo que la creación de código resulta más sencilla y más coherente.

A diferencia de los controles ActiveX, los controles de Windows Forms tienen acceso con una confianza parcial al equipo de un usuario. Esto significa que el código binario o que se ejecuta de forma nativa puede tener acceso a algunos de los recursos del sistema del usuario (como elementos de la GUI y acceso limitado a los archivos) sin tener acceso ni comprometer los demás recursos. Debido a la seguridad de acceso a código, muchas aplicaciones que antes era necesario instalar en el sistema de un usuario, ahora se pueden implementar a través del Web. Las aplicaciones pueden implementar las características de una aplicación local a la vez que se implementan como una página Web.

2.6.4. DESARROLLO DE APLICACIONES DE SERVIDOR

Las aplicaciones de servidor en entornos administrados se implementan mediante hosts de motor en tiempo de ejecución. Las aplicaciones no administradas alojan Common Language Runtime, que permite al código administrado personalizado controlar el comportamiento del servidor. Este modelo proporciona todas las características de Common Language Runtime y la biblioteca de clases, además de obtener el rendimiento y la escalabilidad del servidor host.

En la figura 8 se muestra un esquema de red básico donde se ejecuta código administrado en diferentes entornos de servidor. Los servidores como IIS y SQL Server pueden realizar operaciones estándar mientras la lógica de la aplicación se ejecuta en el código administrado.

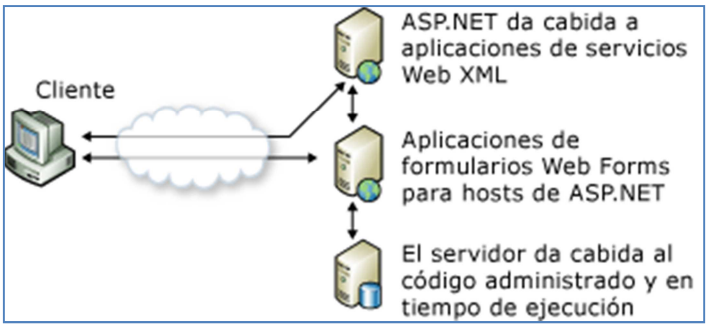

Figura 8: *Código administrado en el servidor*
Fuente: *Información general y concepual sobre .Net Framework. Disponible en: http://msdn.microsoft.com/es-es/library/zw4w595w(v=vs.80).aspx*

ASP.NET es el entorno host que permite a los desarrolladores utilizar .NET Framework con las aplicaciones para el Web. Sin embargo, ASP.NET no es sólo un host de motor en tiempo de ejecución: se trata de una completa arquitectura para el desarrollo de sitios Web y objetos distribuidos en Internet mediante código administrado. Los formularios Web Forms y los servicios Web XML utilizan IIS y ASP.NET como mecanismos de publicación de las aplicaciones y ambos disponen de una colección de clases compatibles en .NET Framework.

Los servicios Web XML, que constituyen una evolución importante de la tecnología basada en el Web, son componentes distribuidos de aplicaciones de servidor similares a los sitios Web comunes. Sin embargo, a diferencia de las aplicaciones basadas en el Web, los componentes de servicios Web XML no tienen interfaz de usuario y no están orientados a exploradores como Internet Explorer y Netscape Navigator. En su lugar, los servicios Web XML consta de componentes de software reutilizables diseñados para que los utilicen otras aplicaciones, como aplicaciones cliente tradicional, aplicaciones basadas en el Web o, incluso, otros servicios Web XML. Como resultado, la tecnología de

servicios Web XML está desplazando rápidamente el desarrollo y la implementación de aplicaciones hacia el entorno altamente distribuido de Internet. El uso de versiones anteriores de la tecnología ASP, permite apreciar de inmediato las mejoras que ofrecen ASP.NET y formularios Web Forms. Por ejemplo, puede desarrollar páginas de formularios Web Forms en cualquier lenguaje compatible con .NET Framework. Además, ya no es necesario que el código comparta el mismo archivo con el texto HTTP (aunque puede seguir haciéndolo, si lo prefiere). Las páginas de formularios Web Forms se ejecutan en lenguaje máquina nativo porque, al igual que todas las aplicaciones administradas, sacan todo el provecho del motor en tiempo de ejecución. En cambio, las páginas ASP no administradas siempre utilizan secuencias de comandos e intérpretes de comandos. El desarrollo de páginas de ASP.NET es más rápido, más funcional y más sencillo que el desarrollo de páginas ASP no administradas, porque interactúan con el motor en tiempo de ejecución como una aplicación administrada.

.NET Framework proporciona también una colección de clases y herramientas para ayudar al desarrollo y uso de las aplicaciones de servicios Web XML. Los servicios Web XML se basan en estándares como SOAP (un protocolo de llamadas a procedimientos remotos), XML (un formato de datos extensible) y WSDL (el Lenguaje de descripción de servicios Web). En .NET Framework se utilizan estos estándares para fomentar la interoperabilidad con soluciones que no son de Microsoft.

Por ejemplo, la herramienta Lenguaje de descripción de servicios Web incluida en .NET Framework SDK puede consultar un servicio Web XML publicado en el Web, analizar su descripción de WSDL y producir código fuente de C# o Visual Basic que la aplicación puede utilizar para convertirse en cliente del servicio Web XML en cuestión. El código fuente puede crear clases derivadas de las clases de la biblioteca de clases que controlan completamente la comunicación subyacente mediante SOAP y análisis de XML. Aunque puede utilizar la biblioteca de clases para usar los servicios Web XML directamente, la herramienta Lenguaje de

descripción de servicios Web y las demás herramientas incluidas en el SDK facilitan el trabajo de desarrollo con .NET Framework.

Si desarrolla y publica su propio servicio Web XML, .NET Framework proporciona un conjunto de clases que cumplen todos los estándares de comunicación subyacentes, como SOAP, WSDL y XML. El uso de esas clases le permite centrarse en la lógica del servicio, sin preocuparse de la infraestructura de comunicaciones que se requiere en el desarrollo de software distribuido. Por último, al igual que las páginas de formularios Web Forms en un entorno administrado, el servicio Web XML se ejecutará con la velocidad del lenguaje máquina nativo mediante la comunicación escalable de IIS.

2.7. NETBEANS 7.2

NetBeans es un entorno de desarrollo integrado libre, hecho principalmente para el lenguaje de programación Java. Existe además un número importante de módulos para extenderlo. NetBeans IDEes un producto libre y gratuito sin restricciones de uso.

NetBeans es un proyecto de código abierto de gran éxito con una gran base de usuarios, una comunidad en constante crecimiento, y con cerca de 100 socios en todo el mundo. Sun MicroSystems fundó el proyecto de código abierto NetBeans en junio de 2000 y continúa siendo el patrocinador principal de los proyectos. La plataforma NetBeans permite que las aplicaciones sean desarrolladas a partir de un conjunto de componentes de software llamados módulos. Un módulo es un archivo Java que contiene clases de java escritas para interactuar con las APIs de NetBeans y un archivo especial (manifest file) que lo identifica como módulo. Las aplicaciones construidas a partir de módulos pueden ser extendidas agregándole nuevos módulos. Debido a que los módulos pueden ser desarrollados independientemente, las aplicaciones basadas en la plataforma NetBeans pueden ser extendidas fácilmente por otros desarrolladores de software.

2.7.1. LA PLATAFORMA NETBEANS

La Plataforma NetBeans es una base modular y extensible usada como una estructura de integración para crear aplicaciones de escritorio grandes. Empresas independientes asociadas, especializadas en desarrollo de software, proporcionan extensiones adicionales que se integran fácilmente en la plataforma y que pueden también utilizarse para desarrollar sus propias herramientas y soluciones.

La plataforma ofrece servicios comunes a las aplicaciones de escritorio, permitiéndole al desarrollador enfocarse en la lógica específica de su aplicación. Entre las características de la plataforma están:

- Administración de las interfaces de usuario (menús y barras de herramientas)
- Administración de las configuraciones del usuario
- Administración del almacenamiento (guardando y cargando cualquier tipo de dato)
- Administración de ventanas
- Framework basado en asistentes (diálogos pasos a paso)

2.7.2. NETBEANS IDE

El IDE NetBeans es un entorno de desarrollo integrado, una herramienta para programadores pensada para escribir, compilar, depurar y ejecutar programas. Está escrito en Java, pero puede servir para cualquier otro lenguaje de programación. Existe además un número importante de módulos para extender el IDE NetBeans. El IDE NetBeans es un producto libre y gratuito sin restricciones de uso.

El NetBeans IDE es un IDE de código abierto escrito completamente en Java usando la plataforma NetBeans. El NetBeans IDE soporta el desarrollo de todos los tipos de aplicación Java (J2SE, web, EJB y aplicaciones móviles). Entre sus

características se encuentra un sistema de proyectos basado en Ant, control de versiones y refactoring.

NetBeans IDE 6.5, la cual fue publicada el 19 de noviembre de 2008, extiende las características existentes del Java EE (incluyendo Soporte a Persistencia, EJB 3 y JAX-WS). Adicionalmente, el NetBeans Enterprise Pack soporta el desarrollo de Aplicaciones empresariales con Java EE 5, incluyendo herramientas de desarrollo visuales de SOA, herramientas de esquemas XML, orientación a web servicies (for BPEL), y modelado UML. El NetBeans C/C++ Pack soporta proyectos de C/C++, mientras el PHP Pack, soporta PHP 5.

- **Modularidad:** Todas las funciones del IDE son provistas por módulos. Cada módulo provee una función bien definida, tales como el soporte de Java, edición, o soporte para el sistema de control de versiones. NetBeans contiene todos los módulos necesarios para el desarrollo de aplicaciones Java en una sola descarga, permitiéndole al usuario comenzar a trabajar inmediatamente.

Sun Studio, Sun Java Studio Enterprise, y Sun Java Studio Creator de Sun Microsystems han sido todos basados en el IDE NetBeans.
Desde julio de 2006, NetBeans IDE es licenciado bajo la Common Development and Distribution License (CDDL), una licencia basada en la Mozilla Public License (MPL).

En octubre de 2007, Sun anunció que NetBeans desde entonces se ofrecerá bajo licenciamiento dual de Licencia CDDL y la GPL versión 2.

2.7.3. NETBEANS 7.2

La versión es la 7.2, según las notas oficiales de la versión, aumenta bastante el rendimiento y la experiencia de programar. En relación al lenguaje de programación Java, el nuevo NetBeans escanea de forma inteligente el proyecto para corregir cualquier tipo de fallo. También tiene características muy interesantes como la integración de Scene Builder para los gráficos JavaFX.

NetBeans IDE 7.2 es una herramienta de programación integrada. Está enfocado al lenguaje de Programación Java, pero actualmente soporta PHP, C/C++, JavaScript, HTML entre otros. Viene integrado con servidores de aplicaciones GlassFish v3, Apache Tomcat y maneja Bases de Datos MySQL, PostgreSQL y cualquiera que se conecte con JDBC como Oracle, SQL Server, y otros más.

2.7.3.1. CARACTERÍSTICAS

- La nueva versión incluye lanzador para versiones de 64 bits en sistemas operativos Windows.

- Programar será más rápido que nunca, pero lamentablemente, aún no se encuentra disponible en castellano, tan sólo en los idiomas inglesas, portugués, japonés, ruso y chino simplificado.

2.8. GESTORES DE BASES DE DATOS

Un Sistema de Gestión de Bases de Datos (SGBD) es un conjunto de programas que permiten el almacenamiento, modificación y extracción de la información en una base de datos, además de proporcionar herramientas para añadir, borrar modificar y analizar los datos. Los usuarios pueden acceder a la información usando herramientas específicas de interrogación y de generación de informes, o bien mediante aplicaciones al efecto.

Los SGBD también proporcionan métodos para mantener la integridad de los datos, para administrar el acceso de usuarios a los datos y recuperar la información si el sistema se corrompe. Permite presentar la información de la base de datos en variados formatos. La mayoría de los SGBD incluyen un generador de informes. También puede incluir un módulo gráfico que permita presentar la información con gráficos y tartas.

Generalmente se accede a los datos mediante lenguajes de interrogación, lenguajes de alto nivel que simplifican la tarea de construir las aplicaciones. También simplifican la interrogación y la presentación de la información. Un

SGDB permite controlar el acceso a los datos, asegurar su integridad, gestionar el acceso concurrente a ellos, recuperar los datos tras un fallo del sistema y hacer copias de seguridad. Las Bases de Datos y los sistemas para su gestión son esenciales para cualquier área de negocio, y deben ser gestionados con esmero.

2.8.1. SQL SERVER MANAGEMENT STUDIO

SQL Server Management Studio es un entorno integrado para obtener acceso a todos los componentes de SQL Server, configurarlos, administrarlos y desarrollarlos.[10] SQL Server Management Studio combina un amplio grupo de herramientas gráficas con una serie de editores de script enriquecidos para ofrecer acceso a SQL Server a desarrolladores y administradores de todos los niveles de especialización.

SQL Server Management Studio combina las características del Administrador corporativo, el Analizador de consultas y Analysis Manager, herramientas incluidas en versiones anteriores de SQL Server, en un único entorno. Además, SQL Server Management Studio funciona con todos los componentes de SQL Server, como Reporting Services, Integration Services y SQL Server Compact 3.5 SP2. Los programadores obtienen una experiencia familiar y los administradores de bases de datos una única herramienta completa que combina herramientas gráficas fáciles de usar con funcionalidad de scripting enriquecida.

2.8.1.1. CARACTERÍSTICAS DE SQL SERVER MANAGEMENT STUDIO

SQL Server Management Studio incluye las siguientes características generales:

- Compatibilidad con la mayoría de las tareas administrativas de SQL Server.
- Un entorno único integrado para administración y edición de SQL Server Database Engine (Motor de base de datos de SQL Server).

[10] *SQL Server Management Studio y SQL Server Express. MSDN Microsoft.com. Pág. 1*

- Nuevos cuadros de diálogo para la administración de objetos de SQL Server Database Engine (Motor de base de datos de SQL Server), Analysis Services, Reporting Services, Notification Services y SQL Server Compact 3.5 SP2, lo que permite ejecutar las acciones inmediatamente, enviarlas a un editor de código o escribirlas en script para ejecutarlas posteriormente.
- Cuadros de diálogo no modales y de tamaño variable que permiten obtener acceso a varias herramientas mientras un cuadro de diálogo está abierto.
- Un cuadro de diálogo común de programación que permite realizar acciones de los cuadros de diálogo de administración en otro momento.
- Exportación e importación del registro de servidor de SQL Server Management Studio desde un entorno de Management Studio a otro.
- Un nuevo cuadro de mensaje de error e informativo que presenta mucha más información, permite enviar a Microsoft un comentario sobre los mensajes, copiar mensajes en el Portapapeles y enviar fácilmente los mensajes por correo electrónico al equipo de soporte.

2.8.2. MYSQL FRONT

MySQL-Front es una sencilla pero útil aplicación diseñada especialmente para desarrolladores que trabajan con MySQL.[11] (MySQL AB, 2002)

Desde el primer momento en el que se empieza a usar este administrador se descubre la facilidad para obtener información sobre las bases de datos, tanto de sus tablas como de su estructura y contenido.Todo ello desde un interfaz muy intuitivo que recuerda bastante a la estructura del Explorador de Windows.

Con MySQL-Front se pueden realizar acciones básicas como añadir, borrar o modificar tablas, campos, registros, y además:

- Ver variables del servidor
- Ejecutar y matar procesos
- Ejecutar SQL-scripts

[11] *Tutorial Bsico de MySQL. My SQL AB. Pág. 2*

- Exportar tablas a SQL-scripts o a otras bases de datos
- Replicar bases de datos
- Guardar datos en formato HTML o CSV (ideal para Excel)
- Escribir queries en SQL
- Importar datos de ODBC
- Realizar diagnóstico de tablas (optimización, reparación, etc.)
- Ver propiedades avanzadas de tablas (tipo, comentario, key_length, etc.)
- Para funcionar correctamente necesita:
- Microsoft C Runtime Library (msvcrt.dll)
- Protocolo TCP/IP
- ODBC (para la importación de datos)
- La librería Cliente para MySQL Server incluida (libmysql.dll)
- Cambios recientes en MySQL-Front:
- Exportación de tablas ODBC y ficheros Access
- Importación y exportación en formato SQLite
- Diagramas de relaciones entre tablas
- Navegador de datos

CAPÍTULO III

3. ANÁLISIS DE RESULTADOS

3.1. MÉTODOS

Los métodos de investigación que se aplicaron son:

- **Método Inductivo.** Se lleva a cabo una etapa de observación y registro de los hechos que se suscitaron. A continuación se procedió al análisis de lo observado.

- **Método Bibliográfico.** Se determina las fuentes más importantes que proporcionen información y documentación como: código fuente, libros, scripts, etc,

3.2. TIPO DE ESTUDIO

Según el objeto de estudio:

- **Investigación aplicada.-** Es la utilización de los conocimientos en la práctica, para aplicarlos, en la mayoría de los casos, en provecho de la sociedad.

Según la fuente de información:

- **Investigación bibliográfica.-** La investigación bibliográfica es aquella etapa de la investigación científica donde se explora qué se ha escrito en la comunidad científica sobre un determinado tema o problema. ¿Qué hay que consultar, y cómo hacerlo?, permite, entre otras cosas, apoyar la investigación que se desea realizar.

3.2.1. SEGÚN LAS VARIABLES

3.2.1.1. Experimental: El experimento dentro de los métodos empíricos resulta el más complejo y eficaz; este surge como resultado del desarrollo de la técnica y del conocimiento humano, como consecuencia del esfuerzo que realiza el hombre por penetrar en lo desconocido a través de su actividad transformadora. El experimento es el método empírico de estudio de un objeto, en el cual el investigador crea las condiciones necesarias o adecua las existentes, para el esclarecimiento de las propiedades y relaciones del objeto.

3.3. POBLACIÓN MUESTRA

La población objeto de estudio comprenden los desarrolladores de sistemas que aplican metodologías para la realización del mismo, que se ubican en diferentes instituciones y entidades de la ciudad de Riobamba, se seleccionó una muestra al azar de 10 desarrolladores representativos.

3.4. OPERACIONALIZACIÓN DE VARIABLES

3.4.1. IDENTIFICACIÓN DE VARIABLES

3.4.1.1. Hipótesis. La propuesta metodológica, permitirá la integración eficiente de las arquitecutras J2EE y .NET.

3.4.1.2. Variable Independiente: La propuesta metodológica.

3.4.1.3. Variable dependiente: Integración eficiente de las arquitecturas J2EE y .NET.

Tabla 2: *Operacionalización de variables*

VARIABLES	TIPO	DEFINICIÓN CONCEPTUAL	INDICADORES	TECNICAS E INSTRUMENTOS
Propuesta metodológica	Independiente	Una propuesta metodológica se trata de aquella guía o conjunto de pasos que indican que hacer y cómo debemos actuar cuando se requiere integrar tecnologías o arquitecturas.	Eficiente Sistemática Escalable Metódica	Cuestionario estructurado para aplicarse a desarrolladores
Integración eficiente	Dependiente	Creación de uniones más firmes entre los diferentes Sistemas de Información tratando funcionalidades programadas con el mínimo de recursos disponibles y tiempo, logrando la interoperabilidad de las tecnologías	Interoperabilidad Funcionalidad Confiabilidad Mantenibilidad Usabilidad Seguridad	Cuestionario estructurado para aplicarse a desarrolladores

Autor. *Johnny Latta*

3.5. PROCEDIMIENTOS

3.5.1. TÉCNICAS DE INVESTIGACIÓN

3.5.1.1. Observación. Utilizar los sentidos para observar los hechos, características, ventajas y desventajas de lo investigado. Para que dicha observación tenga validez es necesario que sea intencionada e ilustrada (con un objetivo determinado y guiada por un cuerpo de conocimiento).

3.5.1.2. Estadística descriptiva: Para la representación de los datos se basó en la Estadística Descriptiva, que permite decidir cuáles son los aspectos, características o los atributos importantes que se deben observar.

3.6. PROCESAMIENTO Y ANÁLISIS

Para evaluar los indicadores se aplicó una encuesta a desarrolladores, para representar los resultados se aplica estadística descriptiva.

3.6.1. EVALUACIÓN DE LA VARIABLE INDEPENDIENTE

Para este análisis se utilizaron 7 preguntas, de las cuales se escogieron 3 preguntas concretas para la verificación de hipótesis.

Tabla 3: *Preguntas para la evaluación de los indicadores*

VARIABLE INDEPENDIENTE	PREGUNTAS
Propuesta metodológica	¿La metodología es entendible y cumple con el objetivo determinado?
	¿La creación de una metodología de integración de tecnologías Java y .Net se puede ajustar o aplicar a otros sistemas?
	¿La metodología planteada podría llegar a ser poco aceptada reutilizable y poco escalable?
	¿Conoce proyectos reales en los que se apliquen integración de tecnologías JAVA y .NET?
	¿Considera que la heterogeneidad de tecnologías es una ventaja para desarrollar estrategias de integración de alto nivel?
	¿La integración de tecnologías implica aplicar una arquitectura de desarrollo de software común, conoce usted alguno de ellos? ¿Cúal?
	JAVA y .NET son tecnologías que permiten el desarrollo de aplicaciones robustas, por lo tanto ¿Se asegura que son tecnologías para el desarrollo de proyectos de integración?

Autor: *Johnny Latta*

3.6.1.1. ENCUESTA A DESARROLLADORES

1. ¿La metodología es entendible y cumple con el objetivo determinado?

Tabla 4: *Pregunta 1. Encuesta*

Alternativa	Frecuencia	Porcentaje
Si	**8**	**80%**
No	**2**	**20%**
Total	**10**	**100%**

Fuente: *Encuesta estructurada*
Autor: *Johnny Latta*

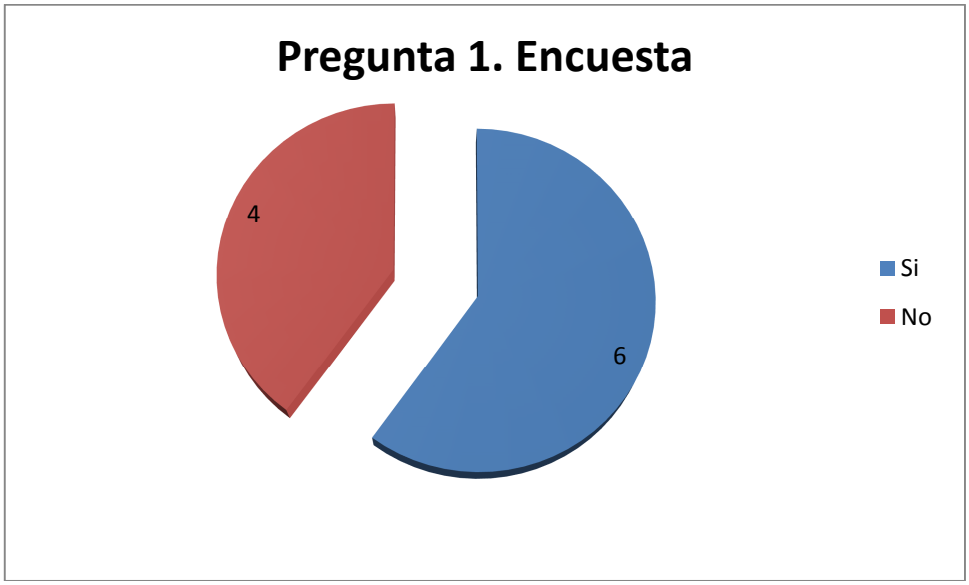

Figura 9 *Pregunta 1. Encuesta*
Autor: *Johnny Latta*

Análisis e interpretación: La metodología propuesta para integrar sistemas, se manifiesta como una estrategia de fácil comprensión para cumplir objetivos de integración tecnológica, en un 80%**,** se determina su aceptación y comprensión. La experiencia en el desarrollo de sistemas con tecnologías heterogéneas facilita un rápido entendimiento entre los desarrolladores.

2. ¿La creación de una metodología de integración de tecnologías Java y .Net se puede ajustar o aplicar a otros sistemas?

Tabla 5: *Pregunta 2. Encuesta*

Alternativa	Frecuencia	Porcentaje
Si	**6**	**60%**
No	**4**	**40%**
Total	**10**	**100%**

Fuente: *Encuesta estructurada*
Autor: *Johnny Latta*

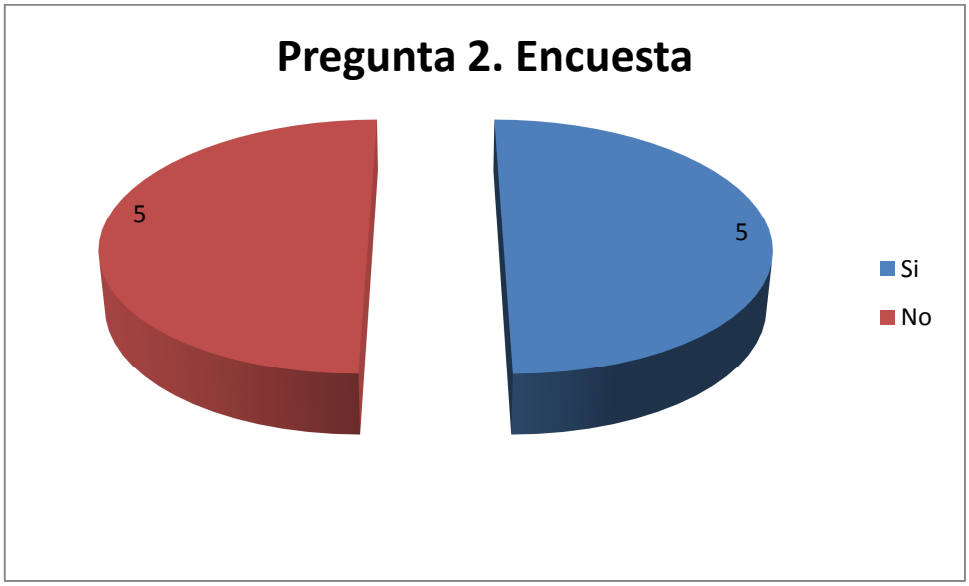

Figura 10: *Pregunta 2. Encuesta*
Autor: *Johnny Latta*

Análisis e interpretación: El nivel de aplicación de la metodología de integración de tecnologías es amplio, ya que los modelos generados son independientes de las tecnologías, llega a un estado que el código se genera cuando ya se define las tecnologías a usar en el proyecto. Por dicha razón el 60% de encuestados afirman que la metodología puede ser aplicable a cualquier tecnología.

3. ¿La metodología planteada podría llegar a ser poco aceptada reutilizable y poco escalable?

Tabla 6: *Pregunta 3. Encuesta*

Alternativa	Frecuencia	Porcentaje
Si	**1**	**10%**
No	**9**	**90%**
Total	**10**	**100%**

Fuente: *Encuesta estructurada*
Autor: *Johnny Latta*

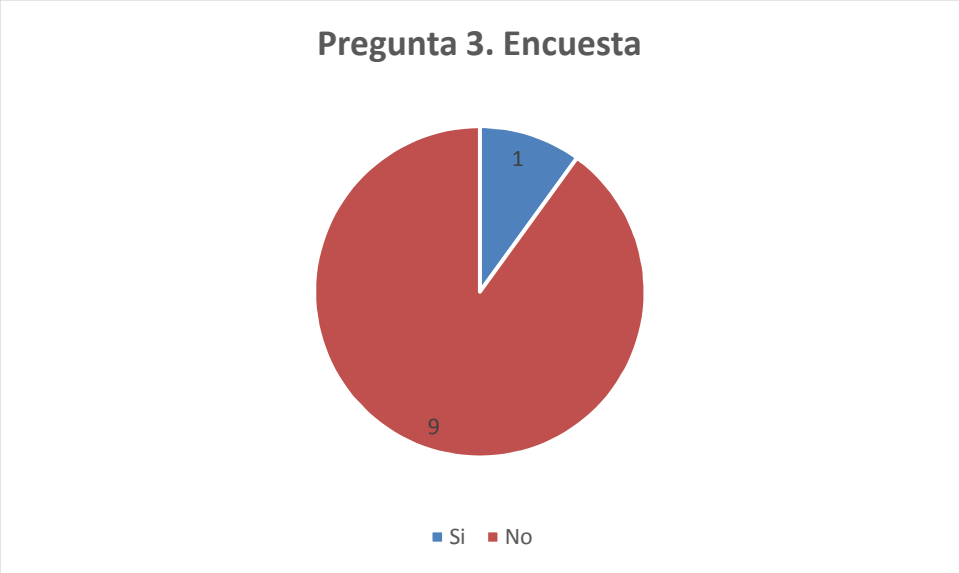

Figura 11: *Pregunta 3. Encuesta*
Autor: *Johnny Latta*

Análisis e interpretación: En un 90% se determina que la metodología puede llegar a ser aceptada por completo, como estrategia de integración tecnológica; se maniefiesta que una nueva innovación de trabajo facilita las tareas del equipo desarrollador.

4. ¿Conoce proyectos reales en los que se apliquen integración de tecnologías JAVA y .NET?

Tabla 7: *Pregunta 4. Encuesta*

Alternativa	Frecuencia	Porcentaje
Si	**6**	**60%**
No	**4**	**40%**
Total	**10**	**100%**

Fuente: *Encuesta estructurada*
Autor: *Johnny Latta*

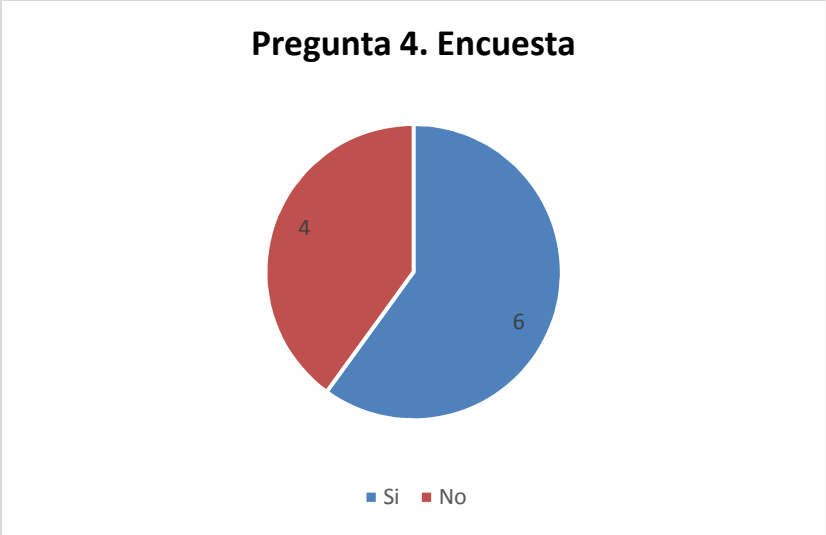

Figura 12: *Pregunta 4. Encuesta*
Autor: *Johnny Latta*

Análisis e interpretación: En un 60% si tienen conocimiento de proyectos reales, donde la integración de tecnologías a sido una estrategia para lograr objetivos del negocio: en Instituciones Educativas de Nivel Superior, Ministerios de Gobiernos, han tenido éxito y siguen en producción.

5. ¿Considera que la heterogeneidad de tecnologías es una ventaja para desarrollar estrategias de integración de alto nivel?

Tabla 8: *Pregunta 5. Encuesta*

Alternativa	Frecuencia	Porcentaje
Si	**5**	**50%**
No	**5**	**50%**
Total	**10**	**100%**

Fuente: *Encuesta estructurada*
Autor: *Johnny Latta*

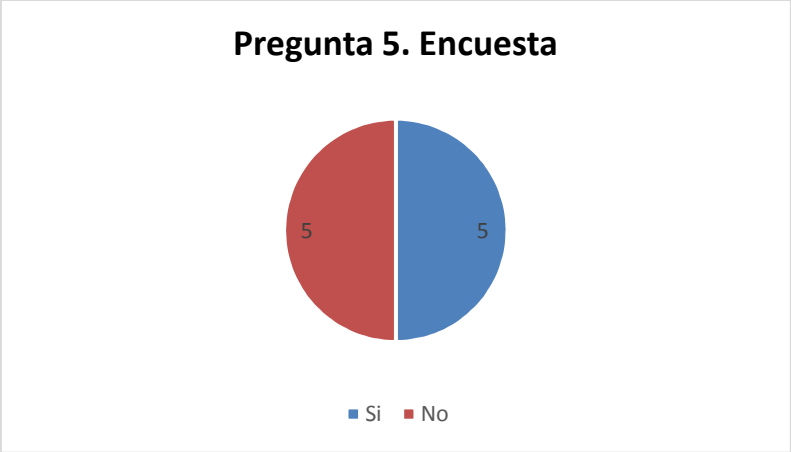

Figura 13: *Pregunta 5. Encuesta*
Autor: *Johnny Latta*

Análisis e interpretación: Las determinaciones son iguales, al mismo tiempo, la heterogeneidad es una ventaja: un reto que retorna soluciones a gran escala en el desarrollo de estrategias de integración que son base o guías para futuras soluciones, pero su implementación implica la realización de pruebas y correcciones que pueden retrasar objetivos trazados.

6. ¿La integración de tecnologías implica aplicar una arquitectura de desarrollo de software común, conoce usted alguno de ellos? ¿Cúal?

Tabla 9: *Pregunta 6. Encuesta*

Alternativa	Frecuencia	Porcentaje
Si	**7**	**70%**
No	**3**	**30%**
Total	**10**	**100%**

Fuente: *Encuesta estructurada*
Autor: *Johnny Latta*

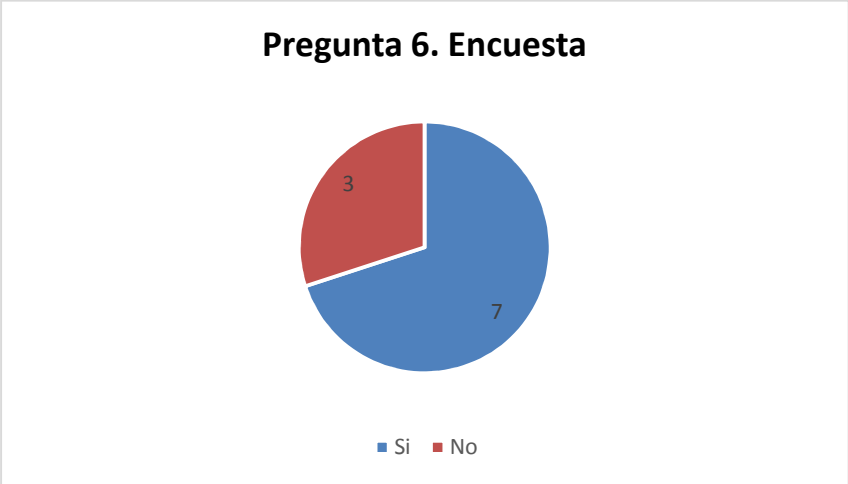

Figura 14: *Pregunta 6. Encuesta*
Autor: *Johnny Latta*

Análisis e interpretación: Un 70% de los desarrolladores conocen la existencia de aquitecturas para el desarrollo de aplicaciones web, y entre las respuestas más comunes fue: MDA, Arquitectura Dirigida por Modelos. Donde diseñar el modelo que tendrá una solución web es importante para su posterior implementación.

7. JAVA y .NET son tecnologías que permiten el desarrollo de aplicaciones robustas, por lo tanto ¿Se asegura que son tecnologías para el desarrollo de proyectos de integración?

Tabla 10: *Pregunta 7. Encuesta*

Alternativa	Frecuencia	Porcentaje
Si	**6**	**60%**
No	**4**	**40%**
Total	**10**	**100%**

Fuente: *Encuesta estructurada*
Autor: *Johnny Latta*

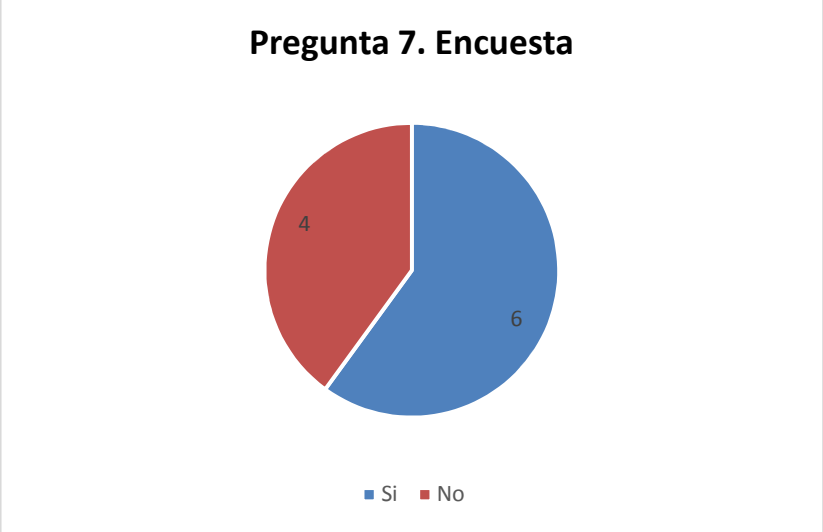

Figura 15: *Pregunta 7. Encuesta*
Autor: *Johnny Latta*

Análisis e interpretación: Con un 60% se afirma por parte de los desarrolladores que Java y .Net son tecnologías que mediante su integración se puede realizar proyectos robustos. Con un 40% se afirma que no son las únicas tecnologías hay otras opciones tecnológicas, pero la experiencia y conocimiento marcan la diferencia al momento de trabajar con efoques de integración tecnológica.

3.6.2. ANÁLISIS DE RESULTADOS

Para verificar la hipótesis se utiliza un estadígrafo, en este caso hablaremos del Chi. La prueba de independencia de Chi – Cuadrado permite determinar si existe una relación entre dos variables categóricas. Es necesario resaltar que ésta prueba indica si existe o no una relación entre las dos variables.

3.6.2.1. PLANTEAMIENTO DE LA HIPÓTESIS

La propuesta metodológica **NO** permitirá la integración eficiente de las arquitecturas J2EE y .NET.

La propuesta metodológica **SI** permitirá la integración eficiente de las arquitecturas J2EE y .NET.

3.6.2.2. DISTRIBUCIÓN DE X^2

Tabla 11: *Distribución de* X^2

Grados de libertad	Probabilidad										
	0,95	0,90	0,80	0,70	0,50	0,30	0,20	0,10	0,05	0,01	0,001
1	0,004	0,02	0,06	0,15	0,46	1,07	1,64	2,71	3,84	6,64	10,83
2	0,10	0,21	0,45	0,71	1,39	2,41	3,22	4,60	5,99	9,21	13,82
3	0,35	0,58	1,01	1,42	2,37	3,66	4,64	6,25	7,82	11,34	16,27
4	0,71	1,06	1,65	2,20	3,36	4,88	5,99	7,78	9,49	13,28	18,47
5	1,14	1,61	2,34	3,00	4,35	6,06	7,29	9,24	11,07	15,09	20,52
6	1,63	2,20	3,07	3,83	5,35	7,23	8,56	10,64	12,59	16,81	22,46
7	2,17	2,83	3,82	4,67	6,35	8,38	9,80	12,02	14,07	18,48	24,32
8	2,73	3,49	4,59	5.53	7,34	9,52	11,03	13,36	15,51	20,09	26,12
9	3,32	4,17	5,38	6,39	8,34	10,66	12,24	14,68	16,92	21,67	27,88
10	3,94	4,86	6,18	7,27	9,34	11,78	13,44	15,99	18,31	23,21	29,59
	No significativo								Significativo		

Fuente: *Estadística y Tecnología de la Información y Comunicación. Disponible en: http://cristina92sm.wordpress.com/2011/05/15/ejercicio-del-seminario-nueve-chi-cuadrado/*

3.6.2.3. SELECCIÓN DEL NIVEL DE SIGNIFICACIÓN

Para la verificación hipotética se utilizará el nivel de significancia α=0.01 obtenido de la tabla 11.

3.6.2.4. DESCRIPCIÓN DE LA POBLACIÓN

Se toma como muestra el total de la población de encuestas realizadas a desarrolladores de aplicaciones web con experiencia en instituciones y entidades de la ciudad de Riobamba.

3.6.2.5. ESPECIFICACIÓN DEL ESTADÍSTICO

Se trata de un cuadro de contingencia de 3 filas x 2 columnas con la aplicación de la siguiente fórmula estadística. Las filas hacen referencia a las preguntas en este caso se han tomado 3 preguntas que son las más relevantes de la encuesta y las columnas hacen referencia a la alternativa de cada pregunta, en este caso las alternativas son: Si y No.

$$X^2 = \sum \left[\frac{(0 - E)^2}{E} \right]$$

X^2= Chi cuadrado

$\sum$ = sumatoria

O= frecuencias observadas

E = frecuencias esperadas

3.6.2.6. ESPECIFICACIÓN DE LAS ZONAS DE ACEPTACIÓN Y RECHAZO

Se procede a determinar los grados de libertad considerando que el cuadro tiene 3 filas (f) y 2 columnas (c), por lo tanto:

gl = (f - 1) (c - 1)

gl = (3 – 1) (2 - 1)

gl = (2)(1)

gl = 2

Con 2 grados de libertad y un nivel de significancia del 0.01 obtenido de la tabla 11, por lo tanto si *c (calculado)* se aceptará caso contrario se la rechazará.

Chi cuadrado para tabular lo podemos graficar de la siguiente manera:

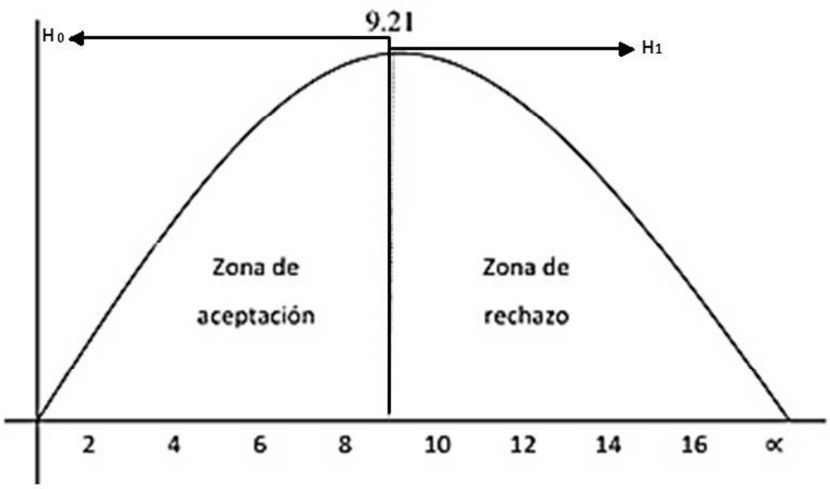

Figura 16: *Chi cuadrado*
Autor: *Jhonny Latta*

3.6.2.7. RECOLECCIÓN DE DATOS Y CÁLCULOS ESTADÍSTICOS.

Se hará uso de las preguntas 1, 2 y 3 de la encuesta dirigida a desarrolladores de instituciones y entidades de la ciudad de Riobamba, siendo éstas las más relevantes para nuestra realización y comprobación del método estadístico chi-cuadrado.

Tabla 12: *Frecuencias observadas desarrolladores*

Nº	Pregunta	Alternativas		Subtotal
		SI	NO	
1	¿La metodología es entendible y cumple con el objetivo determinado?	8	2	10
2	¿La creación de una metodología de integración de tecnologías Java y .Net se puede ajustar o aplicar a otros sistemas?	6	4	10
3	¿La metodología planteada podría llegar a ser poco aceptada reutilizable y poco escalable?	1	9	10
Subtotal		15	15	30

Autor: ***Jhonny Latta***

Tabla 13: *Frecuencias esperadas desarrolladores*

Nº	Pregunta	Alternativas		Subtotal
		SI	NO	
1		4,66	5,33	10
2		4,66	5,33	10
3		4,66	5,33	10
Subtotal		13,98 = 14	15,99 = 16	30

Autor: *Jhonny Latta*

Tabla 14: *Tabla Chi Cuadrado desarrolladores*

O	E	O – E	$(O-E)^2$	$(O-E)^2/E$
8	4,66	3,34	11,1556	2,3939
2	5,33	-3,33	11,0889	2,0805
6	4,66	1,34	1,7965	0,3853
4	5,33	-1,33	1,7689	0,3318
1	4,66	-3,66	13,3965	2,8745
9	5,33	3,67	13,4689	2,5269
TOTAL:				10,5929

Fuente: *Cuestionario*
Autor: *Jhonny Latta*

3.5.2.7.1. Desición: con el 2gl y con un nivel de significancia del 0,01 $x^2t = 9{,}21$

$x^2c = 10{,}5929$ en el caso de los desarrolladores y de acuerdo a las regiones planteadas este último valor es mayor que el primero y se halla por lo tanto en la región de rechazo, se rechaza la hipótesis nula y se acepta la hipótesis alterna que dice:

$\boldsymbol{H_1}$: La propuesta metodológica **SI** permitirá la integración eficiente de las arquitecturas J2EE y .NET.

3.6.3. ANÁLISIS DE RESULTADOS

3.6.3.1. Hipótesis.

La propuesta metodológica, permitirá la integración eficiente de las arquitecutras J2EE y .NET.

De acuerdo el análisis realizado podemos concluir, que la propuesta de una metodología para integrar tecnologías JAVA y .NET permite la interacción entre herramientas de desarrollo, implementar métodos de desarrollo dinámicos, a nivel de una aplicación web que determina el nivel de integración, demostrándose que la hipótesis es verdadera. Se debe tener como consideración que solo es totalmente eficiente en ambientes similares.

3.6.3.2.Justificación.

El manejo de servicios web es una alternativa de soluciones tecnológicas a gran nivel, permite el ahorro de recursos de gestión y representan la realización de tareas rápidas, es más fácil probar y programar componentes que la aplicación completa, además el evaluar la calidad durante todo el desarrollo de la aplicación ayuda a la calidad del producto entregado lo cual repercutirá no sólo en la satisfacción del usuario final sino en la facilidad de mantenimiento, esto se verá traducido en una reducción de costes de mantenimiento y dotará al producto final de un grado de estabilidad que será percibido por el usuario. Por tanto, será una forma de conseguir la satisfacción de los usuarios, y consecuentemente la estabilidad del sistema web. La integración de tecnologías JAVA y .NET desde una perspectiva práctica garantiza la ejecución del sistema porque ayuda al cumplimiento de la eficiencia, funcionalidad y que se requiere en entornos de trabajos demandantes de procesos rápidos, seguros y completos.

CAPÍTULO IV

4. PROPUESTA. APLICACIÓN DE LA METODOLOGÍA PARA INTEGRAR TECNOLOGÍAS JAVA Y .NET

4.1. INTRODUCCIÓN

El desarrollo de software ha tenido una evolución importante en la última década tanto desde el punto de vista conceptual como tecnológico. Se evidencia la incorporación de estándares a los que se adhieren las principales herramientas de desarrollo. De esta manera las nuevas tecnologías marcan una tendencia dentro de la construcción de aplicaciones que podemos decir que definen un paradigma.

Los Servicios Web son una nueva generación de aplicaciones Web. Son componentes de software autocontenidas, autodescriptivas y modulares que pueden ser accedidas, localizadas e invocadas desde cualquier lugar sobre la Internet. Constituyen un modelo de cómputo distribuido que unifica criterios de unificación a otras tecnologías.
El presente trabajo propone una metodología desde el punto de vista del paradigma de la construcción de aplicaciones con tecnología Microsoft y tecnología JAVA.

Dado que la relación se establece desde el punto de vista de la computación distribuida, es importante tener conocimiento de cuál es el modelo de cómputo distribuido que se está comparando. En este sentido, aparece el concepto de Servicio Web como dicho modelo.

Pero más allá de este nuevo modelo de cómputo existen características a relacionar entre las dos tecnologías mencionadas que existen para un modelo de computación distribuida donde no exista el concepto de Servicio Web. Ejemplos de los mismos son: aplicaciones Cliente/Servidor y aplicaciones Web.

Además, podemos decir que el concepto de Servicio Web es una evolución natural de estos dos modelos computacionales mencionados. Ambos tienen en común el hecho de ser un caso particular de software distribuido. El Servicio Web lleva este concepto a la Internet agregando la idea de ser autocontenido y accesible desde cualquier punto.

Una aplicación Cliente/Servidor es software que se construye sobre un modelo computacional distribuido donde existe una comunicación asimétrica entre un proceso servidor que atiende requerimientos y un proceso cliente que los solicita. Ambas componentes son módulos funcionales con interfaces bien definidas y que establecen su comunicación a través de mensajes sincrónicos sin hacer uso de un contexto global. La lógica de la aplicación (lógica del negocio) se encuentra distribuida entre cliente y servidor, mientras que el cliente se hace cargo de la interface de usuario y el servidor de resolver el acceso a los datos.

Una aplicación Web es, al igual que una aplicación cliente/servidor, un tipo especial de aplicación distribuida que se accede vía un navegador Web y cuya lógica de aplicación se encuentra en uno varios servidores accesibles vía un WebServer o un ApplicationServer. Las aplicaciones Web surgieron ante la necesidad de contar con aplicaciones Cliente/Servidor donde los clientes fueran genéricos y livianos y donde su única función sea la de presentar la interface de usuario, llevando la lógica del negocio fuera de esta componente.

En este sentido, hablaremos de plataforma .NET y de plataforma J2EE, que dan soporte al modelo de computación distribuida basada en servicios pero pueden utilizarse de manera tradicional. El aporte de este trabajo es proponer una metodología para integrar plataformas de desarrollo a lo largo de toda la evolución de la computación distribuida, desde el cliente/servidor en dos capas tradicional hasta los nuevos conceptos de servicios Web.

4.2. METODOLOGÍA

La construcción de aplicaciones bajo nuevos estándares y la integración de sistemas heterogéneos es una tarea compleja. En este trabajo se presenta la estrategia seguida para la integración de código plano basado en JSP y J2EE, el sistema transaccional de Java y el gestor de contenidos .Net. El objetivo principal del proyecto ha sido redefinir una metodología de trabajo de los desarrolladores para mejorar su productividad. Siguiendo estrategias fundamentadas en el desarrollo basado en los modelos, y concretamente en MDA, se ha logrado desarrollar un esquema único para los 4 entornos. Se obtuvieron metamodelos independientes de cada uno de ellos para finalmente obtener un metamodelo unificado, flexible y abierto. El desarrollador aumenta su productividad ya que se abstrae de detalles de implementación de cada entorno, centrándose en obtener un modelo de negocio que represente su nuevo sistema, mediante UML, a partir del cual generar el código necesario.

4.3. DEFINICIÓN DE OBJETIVOS DE LA METODOLOGÍA

Proponer una metodología para integrar la plataforma .NET y plataforma J2EE con una perspectiva de la arquitectura cliente/servidor, aplicando criterios de diseño, construcción y mantenimiento que contribuyen al desarrollo de nuevas formas de innovar procesos.

4.4. METODOLOGÍA

Dada la variedad y la complejidad de los entornos con los que se trabajaba se decidió utilizar metodología bottom-up, comenzando por aspectos más específicos para después generalizar y poder abstraer los aspectos comunes. Para ello, se planificaron una serie de tareas repetibles diferenciadas por las tecnologías sujetas a estudio. Al final de estas tareas, se procedió a la integración de las diferentes tecnologías, para acoplarse en un modelo de utilización común e integrada de todas ellas. La figura 9 indica la secuencia de estas tareas así como las actividades que se realizaron en cada una de ellas.

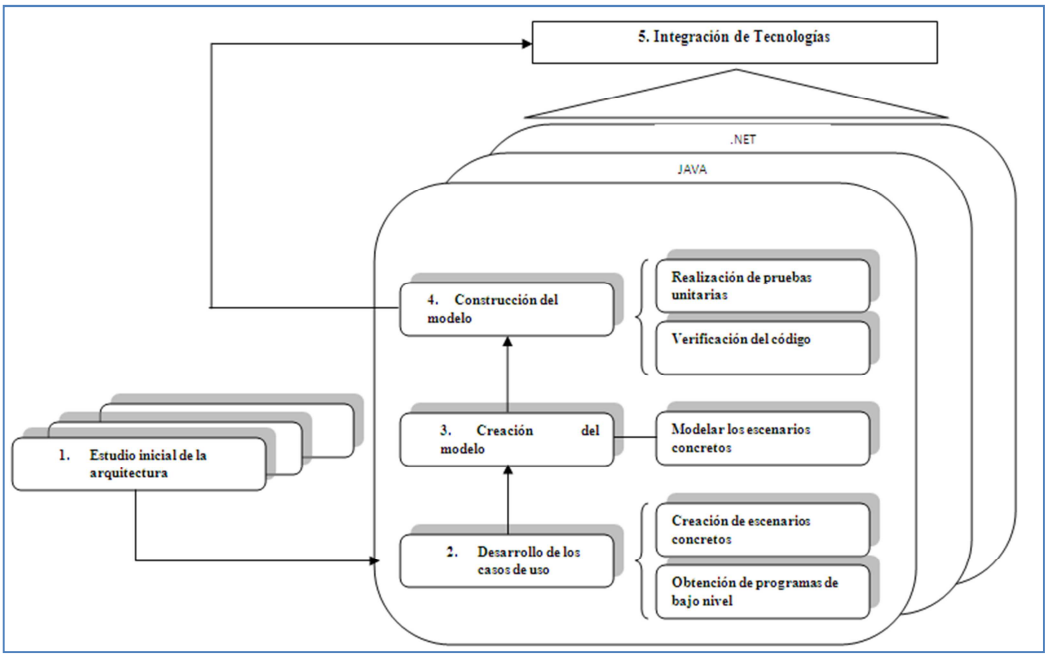

Figura 17: *Esquema de integración*
Autor: *Johnny Latta*

4.5. FASES

Una vez planteada la investigación se precisa la estrategia para llevarla a cabo. En esta etapa se diseña la forma como se obtendrán y analizarán los datos. Para comenzar se deben definen las variables que intervienen en el estudio tanto en el significado de la variable como forma como se cuantificarán precisando los indicadores si fuera necesario. Una vez operacionalizadas las variables se escoge o diseña el instrumento para recolección de datos cuando sea necesario, es decir, la herramienta que va a permitir asignar valores a las variables motivo de estudio, de tal forma que sea precisa y veraz. Se diseña además los cuadros de salida en los cuales podrá resumirse los datos obtenidos para las diferentes variables y las relaciones que se planteen entre ellas. El proceso de medición debe diseñarse con el criterio de eficiencia (ahorro de recursos) y eficacia (permitir medir en forma precisa las variables). Para garantizar la calidad de los datos obtenidos en la medición se definen los criterios de completitud, validez y consistencia de los mismos, previendo que cualquier dato obtenido pueda analizarse según la metodología planteada y acorde con los objetivos del estudio. Los tres pasos que conforman esta fase deben guardar una estrecha relación para

lograr coherencia entre ellos y con lo planteado en la etapa anterior, sin perder de vista lo que se va a realizar en las etapas posteriores.

4.6. DISPOSICIÓN DE LAS TAREAS EN LA METODOLOGÍA

a) **Estudio inicial de las tecnologías**
b) **Desarrollo de los casos de uso**
   - Creación de escenarios concretos
   - Obtención de programas de bajo nivel
c) **Creación del modelo**
   - Modelar los escenarios concretos
d) **Construcción del modelo**
   - Realización de pruebas unitarias
   - Verificación del código
e) **Integración de tecnologías**

4.7. ESTUDIO DE LAS TECNOLOGÍAS CON ENFOQUE DE INTEGRACIÓN

4.7.1. PLATAFORMA DE DESARROLLO J2EE PARA INTEGRACIÓN

Plataforma J2EE. La plataforma J2EE (Java 2 Enterprise Edition) fue diseñada para simplificar el desarrollo, distribución y gerenciamiento de problemas complejos en más de dos capas. Surgió como una evolución a la plataforma J2SE (Java 2 Standard Edition) que permite desarrollar aplicaciones para entornos gráficos, orientadas a objetos y basadas en ventanas.

J2EE es un estándar y no una herramienta. Su arquitectura está basada en Java lo cual permite escribir el código una vez y desplegarlo en cualquier plataforma.

J2EE es una aplicación de Java. Las componentes J2EE son transformadas en "bytecode" e interpretadas por el JRE (Java Runtime Environment) en ejecución.

4.7.2. PLATAFORMA .NET PARA IMPLEMENTACIÓN DE SERVICIOS WEB

Microsoft .NET es una suite de productos que permite a las organizaciones construir aplicaciones empresariales basadas en servicios web o aplicaciones tradicionales en dos o más capas.

Es una reescritura de Windows DNA que fue la plataforma de Microsoft anterior para escribir aplicaciones empresariales. Windows DNA incluía MTS (Microsoft Transaction Server) y COM+, MSMQ (Microsoft Message Queue) y SQL Server como base de datos.

Microsoft .NET ofrece independencia del lenguaje e interoperabilidad, pero no portabilidad. Un componente .NET puede ser escrita en cualquier lenguaje (VB.NET, C #) ya que el código fuente es traducido a un lenguaje intermedio (IL – Intermediate Language) análogo al bytecode de Java, que es interpretado por CLR (Common Language Runtime) análogo al JRE. Esto es lo que se denomina .NET Framework.
Ninguna de estas características ofrece portabilidad, o sea la capacidad de escribir en una plataforma y desplegar en cualquier otra. Microsoft argumenta que la interoperabilidad que ofrecen los servicios web reemplazaría el concepto de portabilidad de los desarrollos Java.

4.7.3. FUNCIONAMIENTO DE LOS WEB SERVICES

Los Web Services son una nueva generación de aplicaciones Web. Son componentes de software autocontenidas, autodescriptivas y modulares que pueden ser accedidas, localizadas e invocadas desde cualquier lugar sobre la Internet. Se construyen sobre estándares como UDDI, WSDL y SOAP.

Los Web Services son:

- Publicados y localizados vía UDDI (Universal Description, Discovery and Integration).
- Descriptos usando WSDL (Web Service Description Language).
- Invocados vía SOAP (Simple Object Access Protocol) sobre HTTP.

4.7.3.1. ESTRUCTURA DE LOS WEB SERVICES

Los Web Services son aplicaciones que utilizan una compleja estructura de protocolos para poder funcionar. Estos se dividen en (Figura 10): Wire Stack (vinculada al mecanismo de comunicaciones), Description Stack (relacionada con la manera de describir el servicio) y Discovery Stack (relacionada con la manera en que el servicio es descubierto por otros servicios).

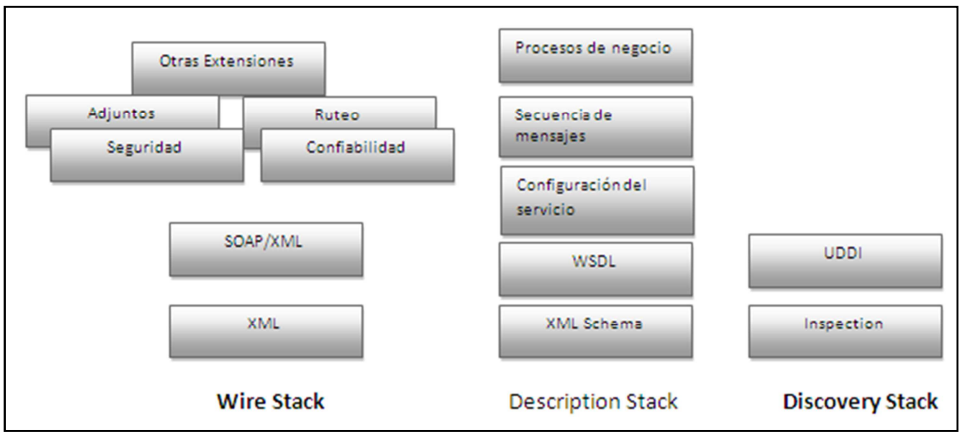

Figura 18: *Estructura de los Web Services*
Fuente: *Rodriguez, M. y Bazán P. (2005) Disponible en: http://www.linti.unlp.edu.ar/uploads/docs/java_y__net_comparacion_de_paradigmas.pdf*

4.7.4. J2EE Y WEB SERVICES

J2EE ha sido históricamente una arquitectura para construir software del lado del servidor en Java. Puede usarse para sitios web tradicionales, componentes de software o aplicaciones empaquetadas. En 2001 extendió su especificación para construir servicios web basados en XML

Las aplicaciones J2EE residen en un contenedor que provee la calidad de servicio necesaria para aplicaciones empresariales como transacciones, seguridad y persistencia. La capa de negocio es implementada por EJB en aplicaciones de mediana a gran envergadura. Este nivel resuelve la lógica de negocios y se conecta vía JDBC a las bases de datos. Los sistemas preexistentes se conectan vía JCA (Java Connector Architecture) Los "business partnerts" conectan con aplicaciones J2EE a través de la tecnología de Web Services (SOAP, UDDI, WSDL y XML). Un servlet (objeto Java orientado al request/reply) puede aceptar requerimientos desde los business partners. Los clientes tradicionales tales como applets y aplicaciones basadas en ventanas, conectan con la capa de EJB a través de IIOP (protocolo Inter-ORB) ya que generalmente estos clientes están escritos en la misma tecnología J2EE con lo cual no necesitan la infraestructura de web services. Los web browsers y dispositivos sin cable, conectan directamente JSP vía http. La figura 11 muestra el modelo de desarrollo de web services en tecnología J2EE.

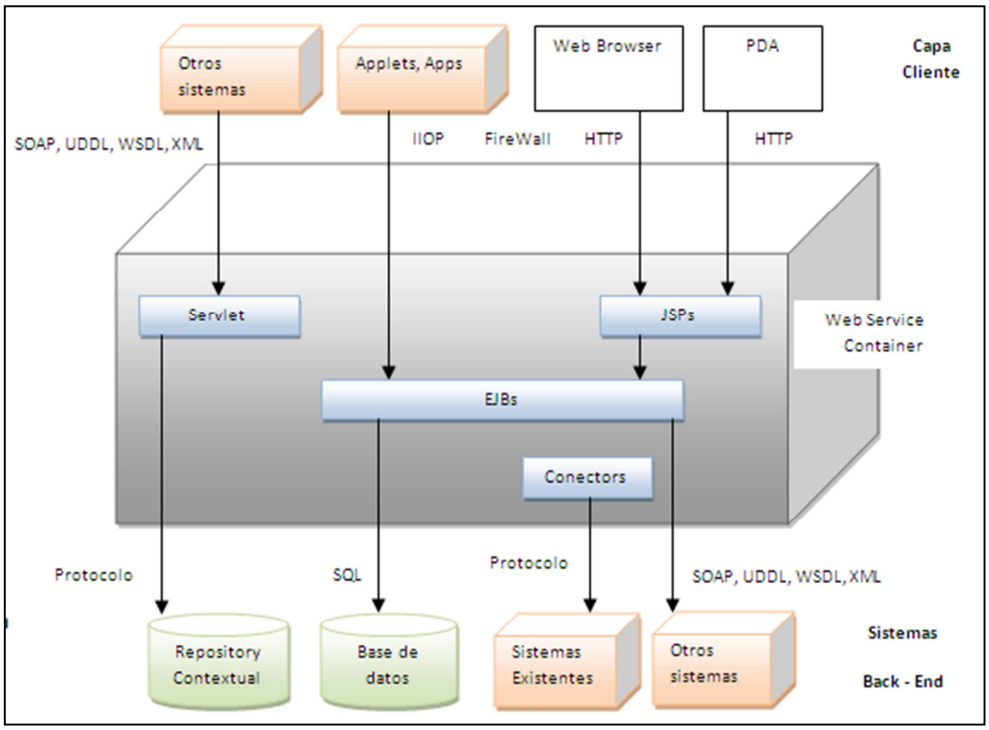

Figura 19: *Modelo de desarrollo de los web service*
Fuente: *Rodriguez, M. y Bazán P. (2005) Disponible en: http://www.linti.unlp.edu.ar/uploads/docs/java_y__net_comparacion_de_paradigmas.pdf*

4.7.5. .NET Y WEB SERVICES

Microsoft .NET es la evolución de Windows DNA para la inclusión de un nivel de Web Services, (Figura 12). Los mismos se encuentran dentro del .NET Framework que también cuenta con un lenguaje de programación mejorado respecto de la tecnología de origen.

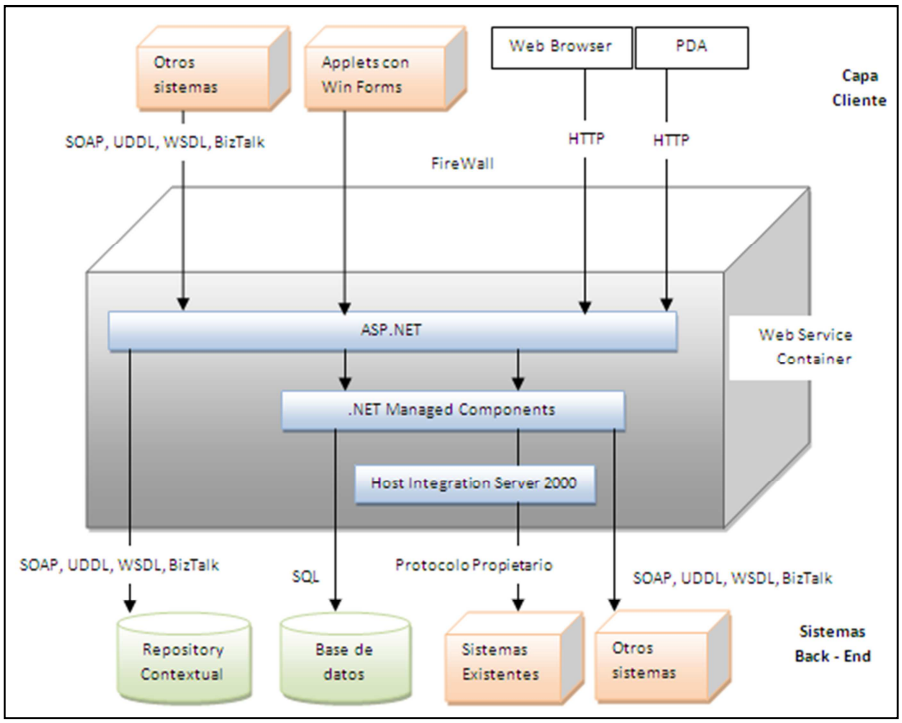

Figura 20:*.Net y web Services*
Fuente: *Rodriguez, M. y Bazán P. (2005) Disponible en: http://www.linti.unlp.edu.ar/uploads/docs/java_y__net_comparacion_de_paradigmas.pdf*

Las aplicaciones .NET residen en un contenedor que provee la calidad de servicio necesaria para aplicaciones empresariales como transacciones, seguridad y persistencia. La capa de negocio de la aplicación .NET se construye con ".NET managed components". Este nivel resuelve la lógica de negocios y se conecta vía ADO.NET a las bases de datos. Los "business partnerts" conectan con aplicaciones .NET a través de la tecnología de Web Services (SOAP, UDDI, WSDL y BizTalk).

Los clientes tradicionales, web browsers y dispositivos sin cable conectan directamente a ASP.NET vía HTTP.

4.7.5.1. Patrones de relación: Se establece en este punto una serie de patrones de relación entre ambas plataformas, algunos de los cuales cobran sentido únicamente considerando la tecnología Web Service, mientras que otros son inherentes exclusivamente a cada plataforma desde el punto de vista conceptual de las mismas, más allá de la inclusión o no del concepto de Web Service.

4.7.5.2. Similitudes: La principal similitud tiene que ver con la normalización y estandarización del intercambio de información a través de la adopción del XML para tal fin.

4.7.5.3. Diferencias: La principal diferencia entre ambas plataformas radica en la tecnología utilizada para realizar la lógica de la aplicación y la manipulación de datos. Java como producto posee un marco muy sólido que puede ser adecuadamente extendido para alcanzar los requerimientos de las aplicaciones sin alterar la estructura del framework. Esto facilita la adopción de esta tecnología por parte de otras compañías. Microsoft, por su parte, si bien aporta .NET framework con CLR para facilitar la corrida en plataformas no Windows, no ofrece un especificación estándar que puedan adoptar otras compañías.

4.7.5.4. Impacto en el mercado futuro: El impacto a futuro de estas tecnologías será más notorio en función del tipo de negocio del que se trate. Contar con la capacidad de centralizar información globalmente impacta sobre el mercado de agencias de viajes, servicios de comidas, teatros. Por otra parte, también constituye un valor agregado para la intranet de una compañía, donde la factorización y dispersión de la información ha sido siempre un problema.

4.7.5.5. Impacto en el mercado actual. El principal impacto de estas dos tecnologías de middleware en el mercado actual es justamente la polarización que produjeron y la necesidad de elegir una u otra. Esta polarización ha sido mayor que la que produjo los sistemas operativos UNÍX y Windows con anterioridad.

4.7.5.6. Lenguaje: Es ampliamente conocido que el 80% del costo de un proyecto de software se encuentra en el mantenimiento de los sistemas. Por lo tanto, contar con un lenguaje simple asegura mantenibilidad. .NET cuenta con varios lenguajes. El sentido común indica que el mercado cuenta con más programadores VB que C#. Por lo tanto sería razonable utilizar VB.NET que es orientado a objetos y posee la característica de CLR. Por otra parte, y justamente por lo enunciado ante, existe un gran "gap" entre VB6 y VB.NET, lo cual hace que los programadores requieran un entrenamiento adicional. De este modo, se debe evaluar el costo del nuevo entrenamiento y allí surge la disyuntiva si se aplica ese entrenamiento sobre VB.NET o Java. En USA, el 78% de las universidades enseñan Java. Por lo tanto, dado que un nuevo entrenamiento es un hecho indiscutible, ya existe una cierta facilidad en los profesionales para aprender Java. Por otra parte, un programador VB no cuenta con experiencia en complejidades tales como programación concurrente y arquitecturas con tolerancia a fallas.

4.7.5.7. Migración de desarrollos preexistentes: Los desarrollos que actualmente se ejecutan en plataforma Windows deberán migrar a .NET. Esto implica varios cambios tanto a nivel de tipos de datos como también de objetos COM. Las aplicaciones VB deberán rescribirse. Por otra parte, los desarrollos Java que incorporen conceptos de Web Services no sufrirán un gran impacto, ya que las nuevas APIs se ubicaron en componentes que facilitan el cambio de las aplicaciones tradicionales a Web Services.

4.7.5.8. Soporte para sistemas existentes: Las organizaciones invierten mucho tiempo y dinero en el desarrollo de sistemas y no siempre se requiere reemplazar los mismos ante los cambios tecnológicos. La integración es uno de los desafíos más importantes a la hora de construir un Web Service. Con tecnología J2EE hay varias formas de realizar la integración: JMS (Java Message Service) para integrar con sistemas existentes, web service para integrar con cualquier sistema, CORBA para interactuar con código escrito en otros lenguajes que existen en máquinas remotas y JNI para cargar librerías nativas y luego invocarlas localmente. .NET ofrece integración a través del Host Integration Server 2000. COM para transacciones colaborativas a través de mainframes. MSMQ para integrar con sistemas IBM MQSeries. Las características de integración ofrecidas por J2EE son superiores a las ofrecidas por .Net.

4.7.5.9. Madurez: Ambas plataformas han madurado al punto de contar con aplicaciones empresariales ya desarrolladas. Los Web Services son nuevos en ambas plataformas y esto hace que las similitudes se separen: en .NET hubo un cambio tecnológico sustancial respecto de sus productos antecesores, incorporando la orientación a objetos y el entorno de ejecución CLR. Para la tecnología J2EE simplemente significó una ampliación de la especificación anterior y una mejora en el manejo de persistencia de los objetos.

4.7.5.10. Portabilidad: Como ya se explicó anteriormente la portabilidad es un concepto diferente de la interoperabilidad. Si bien .NET ha incorporado ventajas para mejorar la manera de interoperar con otras plataformas, sigue dejando cautivos a desarrolladores y usuarios de la plataforma subyacente.

4.7.5.11. Aplicaciones desconectadas: En los próximos años se espera un gran crecimiento de la tecnología alrededor de los dispositivos móviles, lo cual implica la necesidad de contar con aplicaciones "stand-alone" que sincronicen la información cuando establecen una conexión. NET cuenta con tecnología para tal fin como WinForms y MobiliForms pero, nuevamente, se limita a plataforma Windows. Java ha avanzado más en ese sentido contando con una serie de capacidades y herramientas que realizan en forma automática la actualización cuando se establece la conexión.

4.7.5.12. Herramientas: La plataforma J2EE cuenta con una variedad de IDE basados en Java anteriores tanto a J2EE como a .NET. Muchos de ellos comerciales y otros tantos son productos de terceras partes o de software libre. Podemos mencionar en general Visual Age for Java de IBM, Jbuilder de Borland, entre los comerciales. Por su parte Microsoft ya contaba con una ambiente de desarrollo maduro que mejoro ante el lanzamiento de .NET y que es VisualStudio .NET que soporta todos los lenguajes, excepto Java. El problema que presentan los IDE para Java es que no son totalmente interoperables, justamente porque no provienen de un único vendedor. Contexto Compartido:

El concepto de contexto compartido tiene que ver con la naturaleza "sin estado" del protocolo de comunicación de las aplicaciones Web y que generalmente está basado en HTTP. Se requiere un contexto compartido que permita que el usuario ingrese la información una vez y pueda ser luego utilizada. La información debe estar a disposición del usuario y estará protegida por las reglas de seguridad que este defina. Sun J2EE es la de contextos compartidos descentralizados y distribuidos que viven en la Internet. Microsoft .NET lleva a cabo el contexto compartido a través del servicio de Passport.NET.

Es un repositorio alojado por Microsoft que contiene información de la identidad del usuario. El aspecto descentralizado y distribuido de J2EE permite contar con repositorios especializados para diferentes necesidades. No existe el concepto de "hermano mayor" en el sentido que los individuos no necesitan confiar sus datos a cualquiera. No hay un único punto de falla. El aspecto centralizado de Passport .NET no deja duda acerca de cuál es el contexto compartido "oficial" y reduce el riesgo de la proliferación de contextos que implica un esquema descentralizado.

4.7.5.13. Performance: Una plataforma tiene buen rendimiento cuando el tiempo de respuesta ante un requerimiento es "aceptable", donde la definición de este término cambia depende de la naturaleza del problema. Más allá de esto, sin lugar a dudas el principal cuello de botella en estas arquitecturas se encuentra los sistemas de bases de datos back end. J2EE reduce este tráfico y la cantidad de conexiones a la base de datos permitiendo procesos sin estado o "desconectados" y "caching" de memoria por largos periodos de tiempo.

Estas particularidades deben ser utilizadas por programadores experimentados que sepan manejar adecuamente el "tradeoff". Microsoft.NET no ofrece estas características liberando al programador de la toma de decisiones pero impidiendo también manejos de más bajo nivel que podrían mejorar la perfomance.

4.7.5.14. Escalabilidad: Una plataforma es escalable si un incremento en los recursos de hardware resulta en una correspondencia lineal al incremento de carga de usuario que soporta, manteniendo el mismo tiempo de respuesta. En este sentido ambas plataformas son escalables. La única diferencia es que .NET soporta Win32 solamente.

4.7.5.15. Costo: Como ya se mencionó, si una organización no se encuentra ya desarrollando bajo tecnología Java, la adopción de la misma requiere un nuevo entrenamiento de los programadores. La ventaja en cuanto a costos tiene que ver con que existen IDE libres y para todos los sistemas operativos. Por otra parte, si ya se está dentro de la tecnología Microsoft, el re-entrenamiento sigue siendo necesario por los cambios conceptuales

4.8. HERRAMIENTAS SOFTWARE DE INTEGRACIÓN

Tabla 15: *Herramientas de software de integración*

TECNOLOGIA	HERRAMIENTA	VERSION
MySql	MySQLFront	2.5
JAVA	NetBeans	7.2
.Net	VisualStudio 2010	
SQLServer	Sql Management Studio 2008 r2	10.50.16

Autor: *Johnny Latta*

4.9. REQUERIMIENTOS DEL MODELO

El modelo de casos de uso captura todos los requisitos funcionales que cubre la metodología, definiremos un caso de uso de la siguiente manera: "un caso de uso especifica una secuencia de acciones, incluyendo variantes, que el sistemas puede llevar a cabo y que generan un resultado que es observado por un actor".

A continuación se listan los requerimientos funcionales que la metodología de integración cumplirá:

- Diseñar y codificar el servicio web en .net (Nombre del servicio web: MiEjemplo).
- Realizar pruebas unitarias en el escenario de .net (con datos de estudiantes y docentes de la Unach).
- Integrar el servicio web de .net en Java mediante llamadas (en el sistema web GestionUnach).
- Obtener resultados de peticiones atendidas desde la aplicación Gestión Unach (realizar procesos de prácticas, anteproyectos y tesis).

4.10. DESARROLLO DE LOS CASOS DE USO

En esta etapa se obtienen diferentes aplicaciones (código) para su análisis y definición de la funcionalidad requerida. Como resultado, se deben desarrollar una serie de escenarios concretos que cumpla con la mayor parte de los requisitos funcionales analizados. Se requieren programas de amplio espectro sobre la tecnología implantada. Dichas aplicaciones deben cubrir la mayor parte de las necesidades intrínsecas de NetBeans, cubriendo las tareas más comunes a realizar por los mismos. Se analizan y verifican los programas en los entornos de prueba (en aquellas tecnologías que existan). Estos programas se abstraen y/o se recodifican para facilitar la creación del metamodelo, identificando estructuras y componentes susceptibles a ser abstraídos, conceptos generales que posteriormente puedan ser modelados mediante la creación de plantillas. Unas plantillas que, posteriormente, se utilizarán para la generación automática del código, en fases subsiguientes.

4.10.1. DIAGRAMA DE CASOS DE USO DISEÑAR Y CODIFICAR EL SERVICIO WEB EN .NET

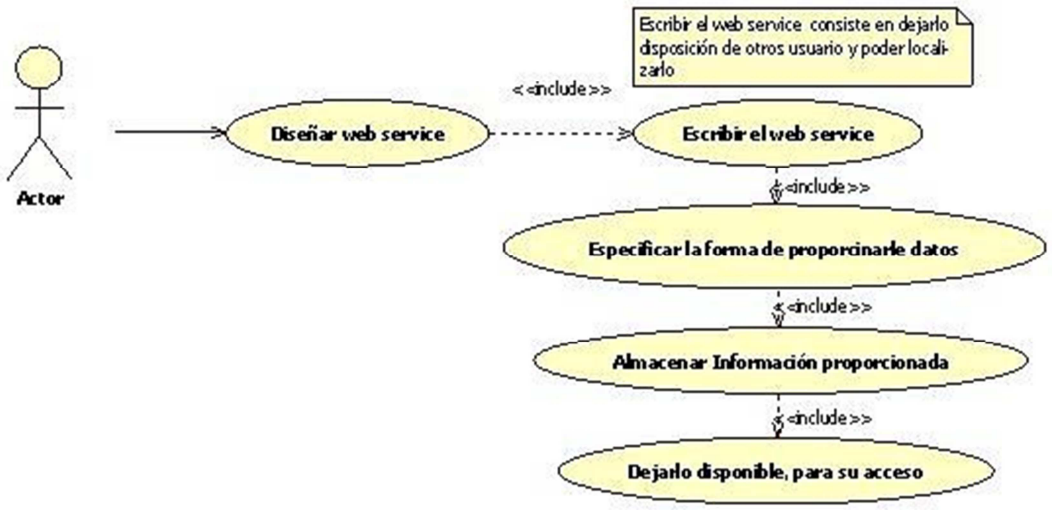

Figura 21: *Diseñar y codificar el servicio web en .net*
Autor. *Johnny Latta*

Actor: El actor representa a la persona encargada de desarrollar, operar y dar mantenimiento al web Service Miejemplo y al sistema software GestionUnach, con el fin de obtener una visión técnica y completa del sistema.

Miejemplo: El Miejemplo representa al web Service que será utilizado por el sistema software GestionUnach para obtener información.

Tabla 16: *Diseñar y codificar el servicio web en .net*

Descripción	El objetivo de este caso de uso es implementar ser servicio web en la plataforma .net	
Precondiciones	La herramienta Visual Studio .Net debe estar instalada correctamente con todos los componentes y librerías.	
Secuencia normal	**Paso**	**Acción**
	1	Ejecutar la herramienta; editor de texto para escribir el archivo
	2	Definir la directiva del servicio web asp.net
	3	Escribir el código, para definir la clase que contendrá al servicio web
	4	Empezaremos con la directiva **using System.Web.Services;** la cual contiene las clases que permite crear servicios Web y clientes de servicios Web.
	5	Incluir el código que contendrá esta clase, en especial la declaración de las funciones (o métodos) que nuestra clase expondrá desde el servicio Web, para ello debemos aplicar a cada uno de los métodos que se desee que el servicio Web exponga, el atributo **WebMethod**. Si no se indica este atributo, el método no será visible (o accesible) desde el servicio Web.
	6	Dejarlo disponible para ser usado.
Poscondiciones	No existen poscondiciones	
Excepciones	**Paso**	**Acción**
	1	Si existe algún error se mostrara el mensaje, proporcionado desde la propia herramienta.
	2	Evaluar y depurar el error.

Autor: *Johnny Latta*

4.10.2. DIAGRAMA DE CASOS DE USO REALIZAR PRUEBAS UNITARIAS EN EL ESCENARIO DE .NET

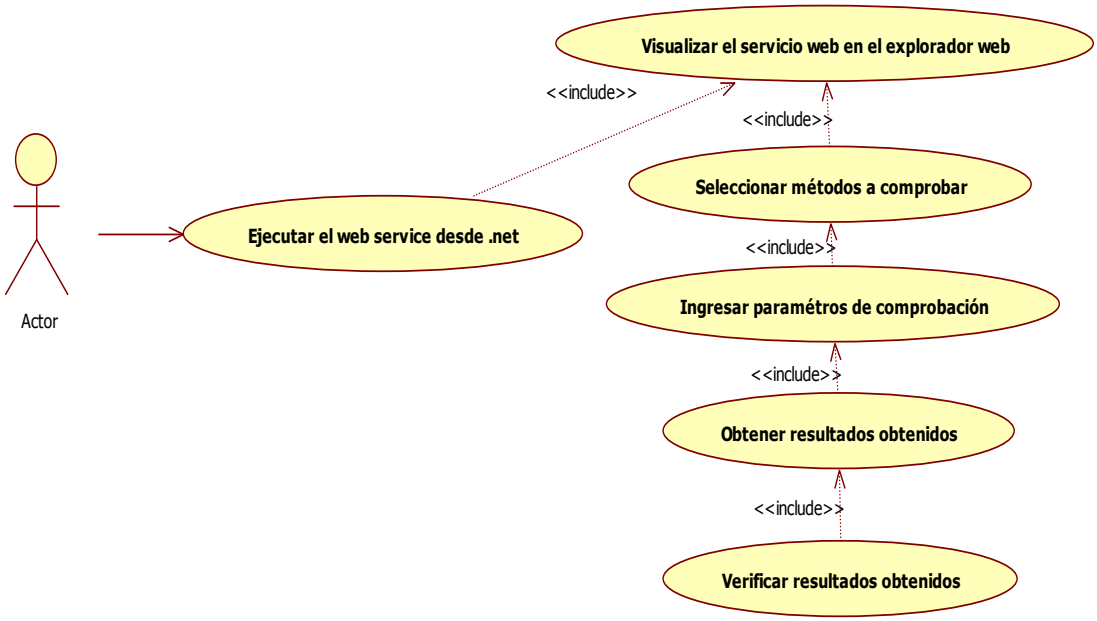

Figura 22: *Realizar pruebas unitarias en el escenario de .net*
Autor: *Johnny Latta*

Actor: El actor representa a la persona encargada de desarrollar, operar y dar mantenimiento al web Service Miejemplo y al sistema software GestionUnach, con el fin de obtener una visión técnica y completa del sistema.

Miejemplo: El Miejemplo representa al web Service que será utilizado por el sistema software GestionUnach para obtener información.

Tabla 17: *Pruebas unitarias en el escenario de .net*

Descripción	El objetivo de este caso de uso es, probar la funcionalidad del web Service implementado; para corregir errores.	
Precondiciones	Los métodos del web Service deben estar ya codificados.	
Secuencia normal	**Paso**	**Acción**
	1	Ejecutar el componente del web Service desde .net
	2	Visualizar en el explorador web
	3	Seleccionar métodos para verificar
	4	Ingresar datos y valorar resultados
Poscondiciones	No existen poscondiciones	
Excepciones	**Paso**	**Acción**
	1	Si existen errores, se mostrara el mensaje de error indicado en el explorador.
	2	Validar errores.

Autor: *Johnny Latta*

4.10.3. DIAGRAMA DE CASOS DE USO INTEGRAR EL SERVICIO WEB DE .NET EN JAVA MEDIANTE LLAMADAS

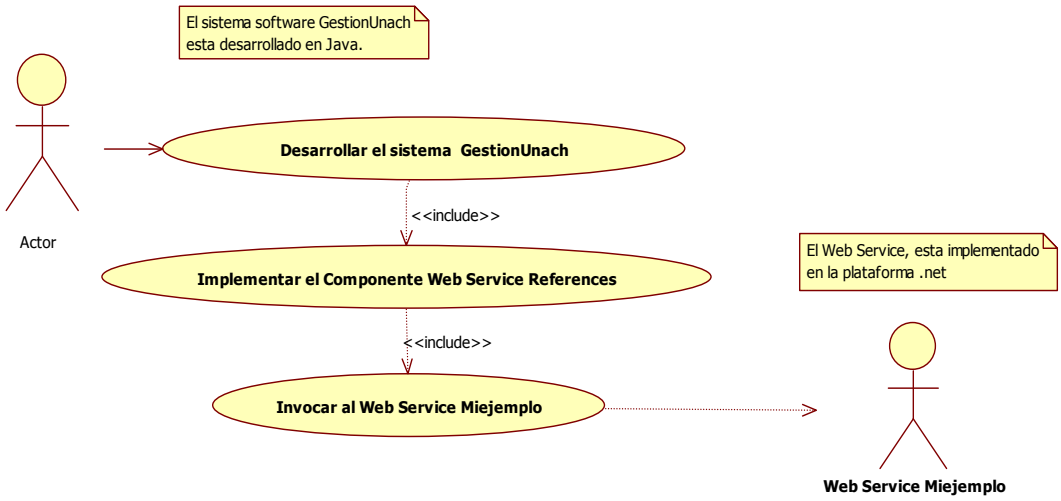

Figura 23: *Integrar el servicio web de .net en Java mediante llamadas*
Autor: *Johnny Latta*

Actor: El actor representa a la persona encargada de desarrollar, operar y dar mantenimiento al web Service Miejemplo y al sistema software GestionUnach, con el fin de obtener una visión técnica y completa del sistema.

Miejemplo: El Miejemplo representa al web Service que será utilizado por el sistema software GestionUnach para obtener información.

Tabla 18: *Integrar el servicio web de .net en Java mediante llamadas*

Descripción	El objetivo de este caso de uso es implementar llamadas a web Service desde Java.	
Precondiciones	Web Service desarrollado y en operación. Sistema software GestionUnach en desarrollo.	
Secuencia normal	**Paso**	**Acción**
	1	Implementar el componente web Service reference desde Java.
	2	Definir el área desde donde se va a invocar al web Service.
	3	Desarrollar la estructura del formulario y sus elementos para trabajar con los métodos del web Service.
	4	Desarrollar los componentes jsp que realiza la llamada a los métodos del servicio web implementados en .net.
Poscondiciones	No existen poscondiciones	
Excepciones	**Paso**	**Acción**
	1	Si existen errores, se mostrara los mensajes de error.
	2	Validar errores.

Autor: *Johnny Latta*

4.10.4. DIAGRAMA DE CASOS DE USO OBTENER RESULTADOS DE PETICIONES ATENDIDAS DESDE LA APLICACIÓN GESTIÓN UNACH

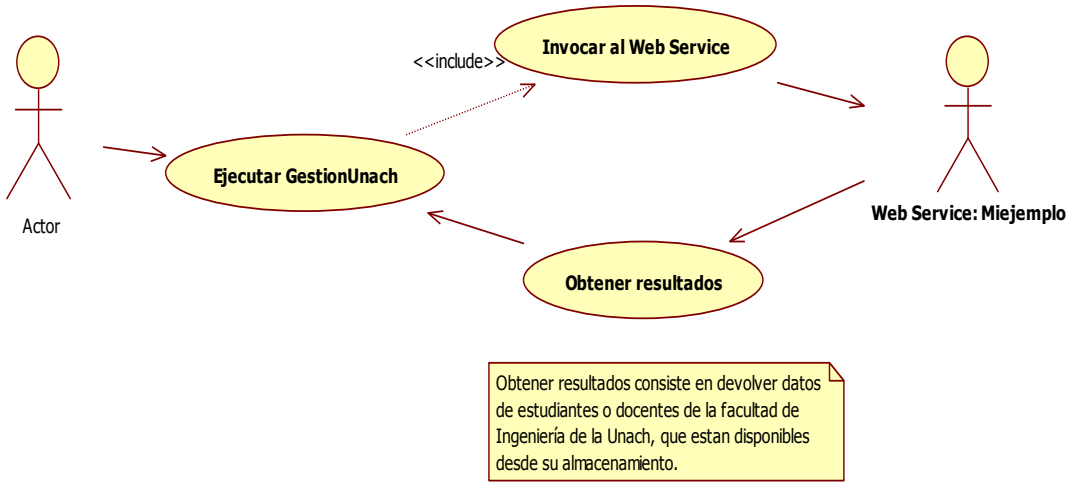

Figura 24: *Obtener resultados de peticiones atendidas desde la aplicación*
Autor: *Johnny Latta*

Actor: El actor representa a la persona encargada de desarrollar, operar y dar mantenimiento al web Service Miejemplo y al sistema software GestionUnach, con el fin de obtener una visión técnica y completa del sistema.

Miejemplo: El Miejemplo representa al web Service que será utilizado por el sistema software GestionUnach para obtener información.

Tabla 19: *Resultados de peticiones atendidas desde la aplicación Gestión Unach*

Descripción	El objetivo de este caso de uso es proveer al usuario los datos obtenidos durante la ejecución del servicio web.	
Precondiciones	Debe ser ejecutado la invocación al servicio web desde el sistema software GestionUnach.	
Secuencia normal	**Paso**	**Acción**
	1	Ejecutar sistema software
	2	Ingresar dato de consulta
	3	Invocar web Service
	4	Obtener resultados
Poscondiciones	No existen poscondiciones	
Excepciones	**Paso**	**Acción**
	1	En caso de existir un error, se presenta el mensaje correspondiente
	2	La ejecución finaliza inmediatamente.

Autor: *Johnny Latta*

4.11. CREACIÓN DEL MODELO

El alineamiento del proyecto con DSDM y en particular con MDA gira en torno al concepto de los Perfiles UML, un mecanismo de especialización definido como parte del propio UML. Los perfiles facilitan el modelado de aspectos específicos de un sistema de software. El principio básico para la obtención de cada uno de los perfiles es indagar generalizaciones entre diferentes lenguajes de programación, plataformas, tecnologías; además de incorporar otros aspectos relevantes relacionados con la integración con sistemas y aplicaciones heredadas.

Mediante al análisis de cada uno de los casos de uso definidos en el Modelo de Casos de Uso, se han identificado las siguientes clases Gestor de Gestión Unach, gestor de ejecución, y web service, que mediante su interacción satisfacen y cumplen con todos los servicios que ofrece el modelo de integración.

- Gestor de ejecución
- Gestor de Sistema software GestionUnach
- Gestor de web Service

En el diagrama de clases siguiente se resume las relaciones existentes entre cada una de las clases identificadas.

4.11.1. DIAGRAMA DE CLASES DEL MODELO

Figura 25: *Clases del modelo*
Autor: *Johnny Latta*

A continuación se presenta el análisis de cada una de las clases identificadas.

4.11.1.1. Gestor de ejecución: clase encargada de iniciar y finalizar la comunicación entre tecnologías.

4.11.1.2. Gestor de Sistema software GestionUnach: interfaz que permite al actor ejecutar la aplicación JAVA.

4.11.1.3. Gestor de web Service: interfaz que permite al sistema software GestionUnach (implementado en JAVA) comunicarse con el web Service (implementado en .Net).

4.11.2. PAQUETES DEL MODELO

Se han definido dos paquetes del modelo. Los mismos que se describen a continuación:

- **GestionUnach:** agrupa las funciones, métodos que permiten ejecutar el servicio web y obtener resultados.

- **Web Service:** Agrupa las funcionalidades para devolver datos en base a requerimientos del actor.

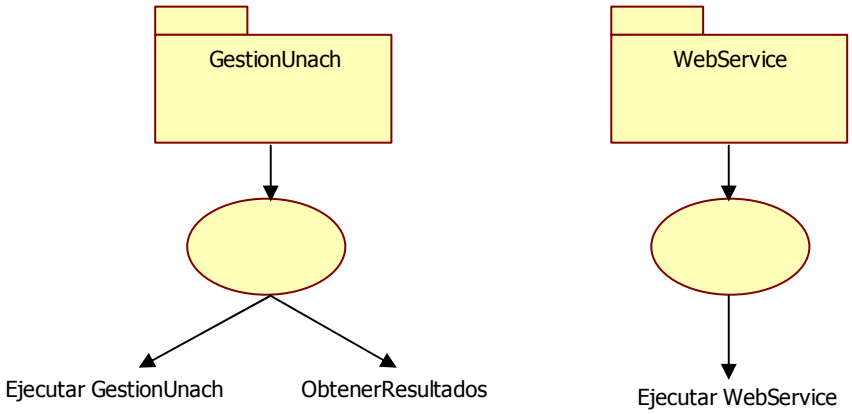

Figura 26: *Paquetes del modelo*
Autor: *Johnny Latta*

4.11.3. RELACIÓN ENTRE COMPONENTES

- **GestionUnach – WebService:** se requiere realizar la integración entre tecnologías JAVA y .NET.

4.12. CONSTRUCCIÓN DEL MODELO

Definidas las funcionalidades y el modelo se comienza con la construcción del mismo, cuya función es la de dirigir la generación, compilación y empaquetado del modelo. En esencia, un modelo reúne la implementación de los Perfiles UML en el contexto de la plataforma y lenguaje de programación sobre el que se generará el código, que en este caso será Java. Dicho modelo contendrá un descriptor donde se definen los perfiles con cada uno de sus estereotipos, asignándoles las plantillas correspondientes. La nueva aplicación se verifica en el entorno tecnológico de NetBeans. Estas 2 tareas se han desarrollado para los entornos de Java y .net que a continuación se detalla, y que en cada caso se han de particularizar al sistema en cuestión.

4.13. APLICACIÓN DE LA METODOLOGÍA

El desarrollo del proyecto se secuenció entre los diferentes entornos tecnológicos. Se comenzó por Java en código plano JSP, ya que se identificó como factor clave para el éxito, así como por su elevada complejidad.

Para el código plano de JSP utiliza la plataforma J2EE que se ejecuta sobre un WebSphere Application Server.

Actualmente se utiliza para el desarrollo de aplicaciones web corporativas. El segundo entorno que se abordó fue el gestor transaccional, identificado en el entorno de .Net. En este entorno existen procesos y lógica de negocio heredadas a nivel corporativo con un alto valor estratégico. En cada entorno se aplicaron las tareas definidas en la metodología previamente explicada. En la última fase del proyecto se dedicó un esfuerzo importante a la integración de todos los entornos en un metamodelo común e integrado.

4.13.1. TAREA 1: ESTUDIO DE LAS TECNOLOGÍAS

La selección de las herramientas de desarrollo constituye un elemento importante, puesto que de estas dependerá el rendimiento, facilidad de uso del producto final y el tiempo que se empleara para la ejecución del mismo.

En base los requisitosde implementación que se han definido, se ha identificado que la herramienta en la que se desarrollara la metodología de integración deberá cumplir con los siguientes criterios:

- Soporte de orientación a objetos
- Soporte para ejecución multithread
- Nivel de conocimiento de la herramienta
- Bajo costo de desarrollo

La herramienta debe facilitar la construcción de aplicaciones orientadas a servicios. Entonces se ha seleccionado las siguientes plataformas de desarrollo:

- JAVA
- .NET

Java y el entorno para su desarrollo NetBeans se utiliza para la creación de aplicaciones J2EE en un entorno corporativo. Básicamente es una unificación de

recursos de desarrollo web, trabajando en la versión 7.2. Dicho entorno está dividido en una serie de módulos de trabajo, entre los que destacan componentes de software para el desarrollo de aplicaciones, el de control, el de presentación, el de negocio y el subsistema de seguridad especializado a un entorno corporativo.

4.13.2. TAREA 2: DESARROLLO DE LOS CASOS DE USO

Se seleccionaron dos herramientas utilizadas para la propia administración del modelo. Estas herramientas se testearon y se pusieron en explotación en los entornos de trabajo. Teniendo dichas herramientas como referencia, se definieron los requisitos de una nueva aplicación y se utilizaron técnicas de reingeniería para su implementación. Durante esta fase se identificaron componentes unitarios del entorno de desarrollo, susceptibles a ser incluidos como componentes parametrizables en el modelo.

4.13.3. TAREA 3: CREACIÓN DEL MODELO

El objetivo del modelo es la creación de un sistema con una arquitectura simplificada, que cumpla los requerimientos del entorno de desarrollo. Para lograr esta simplificación, se mapearon los componentes que ofrecen el mismo a un modelo más orientado al patrón MVC, en el que los dominios estén claramente definidos y enfocados a las funcionalidades de aplicaciones web autocontenidas. Con esta simplificación se consigue una disminución de elementos que el arquitecto MDA debe dominar para la creación de una aplicación más integrada. También permite una mejor distribución del trabajo a desempeñar en áreas especializadas, dividiendo el conocimiento entre las diferentes personas que conforman el equipo.

Se definieron los siguientes dominios y/o capas: dominio de inicialización, presentación, lógica de negocio y el dominio de persistencia.

4.13.4. TAREA 4: CONSTRUCCIÓN DEL MODELO

En esta etapa se propuso como objetivo la generación del 100% del código. Esto aumentó considerablemente la complejidad del problema, sobre todo en la definición de la lógica de negocio. Para lograr esta aproximación se utilizó el Star UML, y con algunas modificaciones. De esta forma se obtuvo un modelo que generaba aplicaciones de forma completamente automatizada. Ahora NetBeans está en capacidad de generar interfaces que utilicen modelos definidos en UML para representar gráficamente necesidades a través de la interacción con .Net y mediante gestores de Bases de datos. El sistema, por tanto, es capaz de generar automáticamente el código a partir de políticas añadidas al negocio.

4.14. UNIFICACIÓN DE MODELOS

La última etapa definida en la metodología 3 se corresponde con la integración de los diferentes entornos. Para ello se plantea una integración a nivel de aplicativo web. Inicialmente se plantea la integración de Java con .Net. En un principio se obtuvo un modelo independiente de un programa concreto al que se le pasaban unos parámetros de entrada y se obtenían unos parámetros de salida. Posteriormente se pasó a incorporar esta funcionalidad como un elemento más, integrado dentro del modelo utilizado en el diseño de una aplicación web. De esta forma, se acometieron dos tipos de integración en el modelo. Una integración en el dominio de presentación, la cual permite la invocación de programa utilizando un formulario web. Y una segunda integración en el dominio de lógica de negocio. Esta última de integración se realizó utilizando diagramas de estado.

4.15. EFECTOS LOGRADOS

Utilizando este modelo, un desarrollador de software puede elaborar una aplicación completa diseñando los patrones adecuados con perfiles ideales. Para ello no hace falta ser un experto en J2EE, .Net, simplemente tener unas nociones básicas de estas tecnologías.

Realizando un correcto modelado, el motor en el que se basa el proyecto es capaz de permitir construir toda la estructura y código necesario para el despliegue y desarrollo de una aplicación completa en un servidor J2EE, en este caso un Websphere Application Server (WAS). Esta aplicación generada podrá haber utilizado, en la capa de negocio, funciones albergadas, o en la capa de persistencia, recursos definidos en la base de datos. Todo ello sin necesidad de ser un experto en las tecnologías utilizadas, simplemente trabajando de una forma correcta y planteando la solución que ellos estimen oportuna.

Para la ejecución de este proyecto se propuso la utilización de código plano JSP en NetBeans 7.2 como solución viable y eficiente para la persistencia de objetos con una base de datos en MySQL. Este componente fue añadido, y correspondientemente modelado e implementado dentro de los formularios adecuados. Para la conexión con los web services, se hizo uso de las librerías aportadas por JCA, de las cuales existen numerosas referencias, documentación y librerías actualizadas. Y para la conversión de datos Cobol a Java y viceversa, una funcionalidad crítica en este ámbito, se utilizaron las librerías y clases proporcionadas por JAVA. El modelo se ha abstraído lo básico como para generalizarlo a otros gestores de contenidos, únicamente bastaría con modificarlo.

Por último, indicar que como resultado final de este proyecto se obtiene la integración de todas las herramientas anteriormente nombradas en un modelo común y unificado. Con ello se logra la integración completa de los entornos especificados, a priori, completamente heterogéneos en un modelo unificado, eficiente y ordenado, que permite el desarrollo de nuevos sistemas. El desarrollador se encuentra con perfiles bien definidos en las 2 plataformas, con el que desarrollará a un mayor nivel de abstracción independizándose de los aspectos tecnológicos.

4.16. CUMPLIMIENTOS

Este nuevo enfoque en la creación de sistemas representa un gran cambio en la forma de trabajo tradicional, que actualmente existe en los equipos de desarrollo. Ello exige el planteamiento de una nueva metodología y forma de trabajo, así como la planificación de una gestión del cambio que permita una adaptación rápida al nuevo escenario y que suponga un mejor aprovechamiento de las ventajas de este nuevo entorno. Los modeladores de los nuevos sistemas deben poseer un mayor conocimiento, principalmente de las tecnologías, a la hora de abordar la abstracción y creación de dichos sistemas. Aparte de las ventajas intrínsecas que se derivan de un sistema basado en una aproximación MDA, se destaca:

La normalización de los sistemas mediante modelos UML.

La integración de sistemas heterogéneos, ocultando las complejidades de cada una de las tecnologías en particular.

Desarrollo de un sistema único basado en la Web.

Aumento considerable en la calidad de los sistemas desarrollados, ya que el código generado ha sido exhaustivamente probado.

Posibilita el desarrollo rápido de prototipos, para que se conviertan fácilmente en sistemas y aplicativos finales.

Mejora la facilidad de la implementación de la persistencia de los modelos, independizándolos de las continuas evoluciones y cambios tecnológicos.

4.17. REALIZACIÓN PRÁCTICA DE LA INTEGRACIÓN DE TECNOLOGÍAS

4.17.1. DIAGRAMA DE FLUJO CONSULTA DATOS ESTUDIANTE: NOMBRE ESTUDIANTE

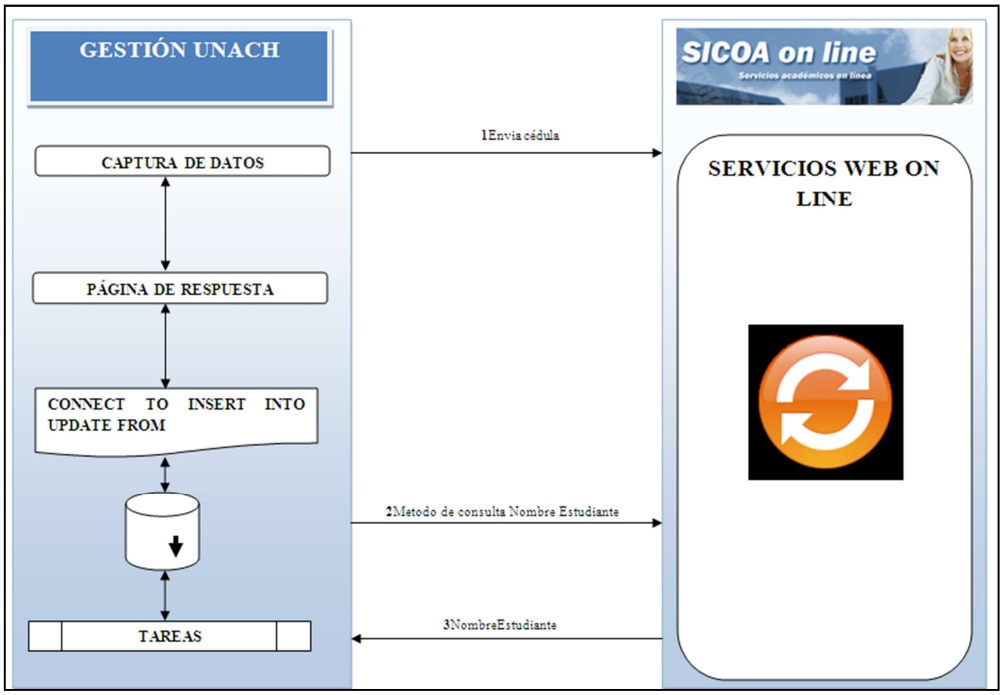

Figura 27:*Nombre estudiante*
Autor: *Johnny Latta*

Nota:

a) Dependiendo del caso, se usa un método de consulta diferente.
b) Dependiendo del método usado para consultar la transacción, traerá un objeto diferente.

Para implementar el registro de prácticas pre profesionales (1) se deben leer la cédula del estudiante, (2) de esta petición recibirá toda la información disponible del objeto estudiante.

Luego se deben enviar los datos para realizar la consulta al WebServices Miejemplo (3). La transacción es procesada y se devuelve una respuesta inmediata el Nombre completo del Estudiante, permitiendo seguir con el proceso de

registro,realizar las actualizaciones de bases de datos que necesite, y mostrándole al usuario una página de respuesta que puede ser personalizada de acuerdo a las necesidades.Pero no en todos los casos, esta respuesta es definitiva, en algunas ocasiones la transacción queda en proceso de validación y requiere de un tiempo mayor para procesarse y tener una respuesta definitiva.

4.17.2. DIAGRAMA DE FLUJO CONSULTA DATOS ESTUDIANTE: ESCUELA ESTUDIANTE

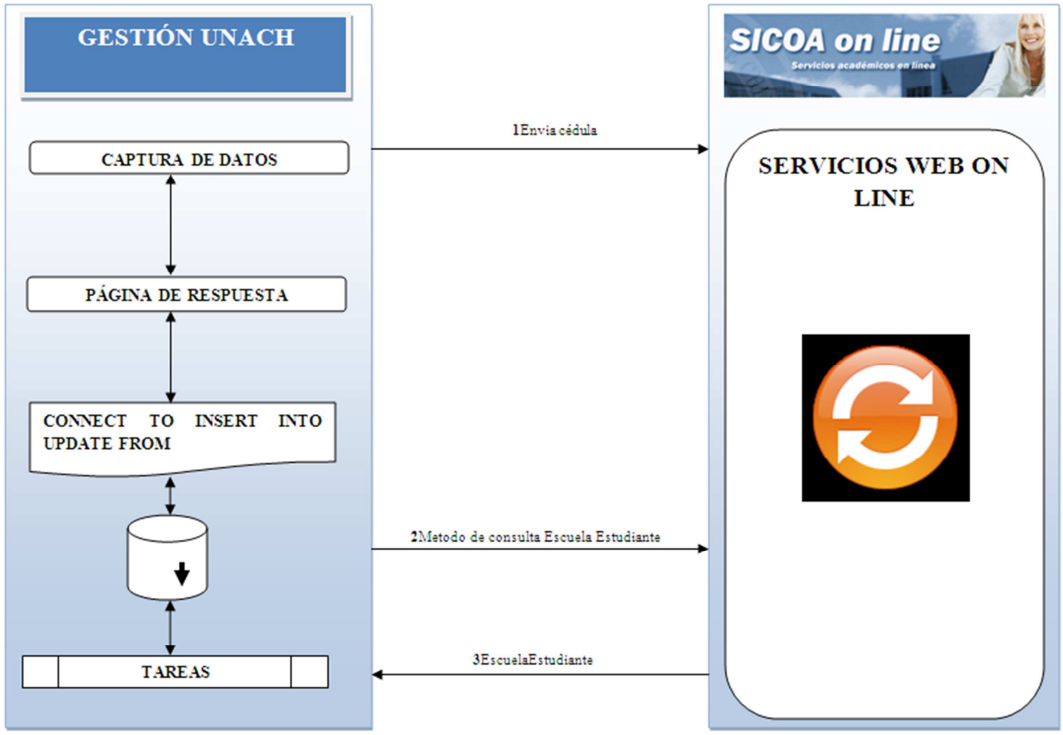

Figura 28:*Escuela estudiante*
Autor: *Johnny Latta*

Nota:

c) Dependiendo del caso, se usa un método de consulta diferente.
d) Dependiendo del método usado para consultar la transacción, traerá un objeto diferente.

Para implementar el registro de prácticas pre profesionales (1) se deben leer la cédula del estudiante, (2) de esta petición recibirá toda la información disponible del objeto estudiante.

Luego se deben enviar los datos para realizar la consulta al WebServices Miejemplo (3). La transacción es procesada y se devuelve una respuesta inmediata el Escuela del Estudiante, permitiendo seguir con el proceso de registro, realizar las actualizaciones de bases de datos que necesite, y mostrándole al usuario una página de respuesta que puede ser personalizada de acuerdo a las necesidades.Pero no en todos los casos, esta respuesta es definitiva, en algunas ocasiones la transacción queda en proceso de validación y requiere de un tiempo mayor para procesarse y tener una respuesta definitiva.

4.17.3. DIAGRAMA DE FLUJO CONSULTA DATOS DOCENTES: NOMBRE DOCENTE

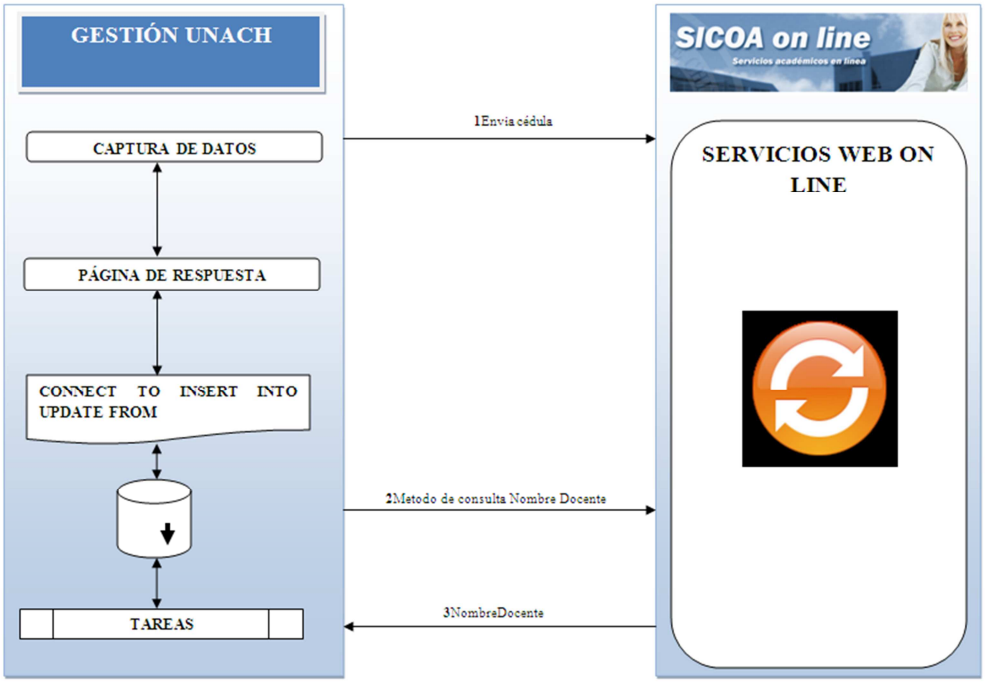

Figura 29:*Nombre docente*
Autor: *Johnny Latta*

Nota:

a) Dependiendo del caso, se usa un método de consulta diferente.
b) Dependiendo del método usado para consultar la transacción, traerá un objeto diferente.

Para implementar la gestión de prácticas pre profesionales y anteproyectos y tesis en la facultad de Ingeniería de la Universidad Nacional de Chimborazo, se

requiere en procesos determinados acceder a la información del Docente que va a estar a cargo de su ejecución. Entonces para acceder a dicha información (1) se deben leer la cédula del docente, (2) de esta petición recibirá toda la información disponible del objeto docente.

Luego se deben enviar los datos para realizar la consulta al WebServices Miejemplo (3). La transacción es procesada y se devuelve una respuesta inmediata el Nombre completo del Docente, permitiendo seguir con el proceso de registro, realizar las actualizaciones de bases de datos que necesite, y mostrándole al usuario una página de respuesta que puede ser personalizada de acuerdo a las necesidades. Pero no en todos los casos, esta respuesta es definitiva, en algunas ocasiones la transacción queda en proceso de validación y requiere de un tiempo mayor para procesarse y tener una respuesta definitiva.

4.17.4. DESCRIPCIÓN DE LOS MÉTODOS: SELECCIONAR NOMBRE ESTUDIANTE

4.17.4.1. Método SeleccionarNombreEstudiante()

Permite consultar el nombre de una solicitud de nombre estudiante en cualquier momento. Recibe como parámetros el ***cedulaestudiante*** retorna un objeto **NombreEstudianteWS**. Este método se puede utilizar para consultar cualquier solicitud de nombre del estudiante, generalmente los que se encuentran en estado: matriculados. El resultado contiene un objeto **NombreEstudiante**que contienen el nombre completo del estudiante que está matriculado en la facultad.

4.17.4.2. Parámetros de entrada:

***Tabla** 20: Método Seleccionar Nombre Estudiante*

Parámetro	Tipo	Descripción
cedulaestudiante	N	Identificador del estudiante.

***Autor:** Johnny Latta*

4.17.4.3. Retorna: El objeto NombreEstudianteWSdatos actualizados del estudiante.

4.17.5. DESCRIPCIÓN DE LOS MÉTODOS: SELECCIONAR NOMBRE DOCENTE

4.17.5.1. MÉTODO SeleccionarNombreDocente()

Permite consultar el nombre de una solicitud de nombre docente en cualquier momento. Recibe como parámetros el ***ceduladocente*** retorna un objeto **NombreDocenteWS**. Este método se puede utilizar para consultar cualquier solicitud de nombre del docente, generalmente los que se encuentran en estado activos. El resultado contiene un objeto **NombreDocente**que contienen el nombre completo del docente que pertenece a la facultad.

4.17.5.2. Parámetros de entrada:

Tabla 21:*Método Seleccionar Nombre Docente*

Parámetro	Tipo	Descripción
Ceduladocente	N	Identificador del docente.

Autor: Johnny Latta

4.17.5.3. Retorna: El objeto NombreEstudianteWSdatos actualizados del estudiante.

4.17.6. DESCRIPCIÓN DE LOS MÉTODOS: SELECCIONAR ESCUELA ESTUDIANTE

4.17.6.1. MÉTODO SeleccionarEscuelaEstudiante()

Permite consultar el nombre de escuela de una solicitud de escuela estudiante en cualquier momento. Recibe como parámetros el ***cedulaestudiante***retorna un objeto **EscuelaEstudianteWS**. Este método se puede

utilizar para consultar cualquier solicitud de escuela del estudiante, generalmente los que se encuentran en estado: matriculados. El resultado contiene un objeto **EscuelaEstudiante**que contienen el nombre de la escuela del estudiante que está matriculado en la facultad.

4.17.6.2. Parámetros de entrada:

Tabla 22:*Método Seleccionar Escuela Estudiante*

Parámetro	Tipo	Descripción
cedulaestudiante	N	Identificador del estudiante.

Autor: *Johnny Latta*

4.17.6.3. Retorna: El objeto EscuelaEstudianteWS datos actualizados del estudiante.

4.18. DESARROLLO DEL SISTEMA: GESTIÓN UNACH DE LA FACULTAD DE INGENIERÍA

4.19. ESPECIFICACIÓN DE REQUISITOS

La ingeniería de software se realiza en base a la metodología de Craig Larman, ya que define una serie de actividades que son adaptables a las condiciones de este proyecto, el mismo que no fija una metodología estricta y posibilita omitir algunas actividades en dicha metodología. Larman propone un ciclo de vida interactivo e incremental haciéndola suficientemente flexible y adaptable a las necesidades de los desarrolladores[12]; esta metodología se va aplicar siguiendo el esquema que se presenta a continuación:

- Planificación y especificación de requisitos.
  - ✓ Definición de requisitos.
  - ✓ Definición de casos de uso en un formato de alto nivel.
- Análisis
  - ✓ Definición de casos de uso en formato extendido.
  - ✓ Definición de los diagramas de secuencia del sistema.
  - ✓ Diagrama de estado.
  - ✓ Modelo conceptual.
  - ✓ Diagrama de actividades.
- Diseño.
  - ✓ Definición de la arquitectura del sistema
  - ✓ Definición de la interfaz de usuario.
  - ✓ Diagramas de interacción
  - ✓ Diagrama de clase de diseño.
  - ✓ Esquema de base de datos.
  - ✓ Diagrama de componentes
  - ✓ Diagrama de Despliegue
- Implementación.

[12] http://www.fdi.ucm.es/profesor/gmendez/docs/publicaciones/jisbd01.pdf

4.20. PLANIFICACIÓN

Si se observa en la metodología de Larman [Larman 99], se parte de la idea de que ya se dispone de un documento de especificación de requisitos, por lo que en ningún momento se describe qué es lo que debe contener este documento o cómo se debe realizar su construcción.

4.20.1. DEFINICIÓN DEL ÁMBITO DE SOFTWARE

La Universidad Nacional de Chimborazo a más proporcionar educación también lleva el control de los procesos académicos seguidos por los estudiantes, prácticas pre profesionales, anteproyectos y tesis y proyectos de vinculación e investigación los mismos que necesitan ser gestionados para verificar el nivel de cumplimiento de requisitos que se exige al nuevo profesional, en la actualidad no existe ningún tipo gestión automatizada con tecnología de punta en la secretaria de la facultad de Ingeniería de la Institución.

Con el objetivo de proveer un elemento software que facilite la gestión de los procesos educativos de los futuros profesionales se desarrollará un sitio Web de Gestión de prácticas, anteproyectos, tesis y gestión de actas el mismo que registrará la información más relevante del estudiante, docentes, instituciones, y calificaciones de las entidades involucradas, los datos almacenados serán procesados para obtener información que demuestre el cumplimiento de las resoluciones emitidas por el consejo directivo de la facultad. Este módulo operacional pretende mostrar resultados que garantice el trabajo de las personas mediante la entrega de reportes que características propias de cada proceso.

El sitio Web será implementado en la secretaría de la Facultad de Ingeniería a la que pertenece la escuela de Ingeniería en Sistemas de la Universidad Nacional de Chimborazo.

La implantación del sistema web permitirá tener una visión de calidad de la gestión del proceso de lo estudiantes, manejar información coherente y completa, estar en la capacidad de emitir reportes de cada una de las resoluciones adoptadas por las autoridades que rigen a dichos procesos.

4.20.2. ANTECEDENTES TECNOLÓGICOS

Actualmente enla secretaría de la Facultad de Ingeniería cuenta con los siguientes recursos:

4.20.2.1. RECURSO HUMANO

Tabla 23:*Recurso Humano*

Nombre	Cargo
Lcda. Yesenia Echeverría	Secretaria de Facultad

Autor: *Johnny Latta*

4.20.2.2. RECURSO HARDWARE

Tabla 24:*Recurso Hardware*

Cantidad	Hardware	Especificación
1	Computador de escritorio	Procesador INTEL Pentium IV 2.0 G.Z Memoria 512 MB DDR2 Disco Duro 320 GB CD-ROM Teclado Mouse Monitor Samsung SyncMaster S58 de 17”

Autor: *Johnny Latta*

4.20.2.3. RECURSO SOFTWARE

Tabla 25:*Recurso Software*

Cantidad	Software	Especificación
1	Windows XP SP3	Sistema Operativo

Autor: *Johnny Latta*

4.20.3. DEFINICIÓN DE LA ALTERNATIVA DE SOLUCIÓN

El sistema Gestión Unach representa una interfaz que permite el ingreso del usuario (Secretaria) con 3 módulos para la gestión de prácticas, anteproyectos y tesis y gestión de actas del Honorable Concejo Directivo, la gestión de cada

módulo consiste en ingresar prácticas, anteproyectos y actas, actualizar prácticas y anteproyectos y tesis, asignar delegados y tribunal, calificar prácticas, matricular anteproyectos, además con la opción de generar reportes exclusivos para cada trámite a seguir.

El sistema está compuesto por los siguientes módulos:

- Módulo de Gestión de Prácticas
- Módulo de Gestión de Anteproyectos y Tesis
- Módulo de Gestión de Actas

El conjunto de módulos en su totalidad forma el sistema de Gestión Unach el mismo que funciona de la siguiente manera: la secretaria se registra ingresando usuario y contraseña a través de una interfaz gráfica el mismo que de acuerdo a sus permisos asignado se le muestra la interfaz correspondiente al usuario, si el usuario pertenece al grupo de usuario normal se muestra una pantalla en el que tiene permisos para realizar tareas administrativas como gestionar prácticas o a su vez anteproyectos y tesis, aquí de acuerdo a la selección realizada pasa a registrar unas nuevas prácticas, definir lo docentes delegados para la revisión correspondiente, además de registrar la calificación de la defensa oral y escrita, etc. Se sigue el mismo proceso para la gestión de anteproyectos y Tesis, cabe indicar que cada actividad en el sistema se respalda de un proceso, que consiste en hacer una búsqueda por cédula del estudiante, para asegurar que el proceso no va a tener contradicciones en la lógica del trabajo. Todo este proceso se encuentra representado en la figura 22.

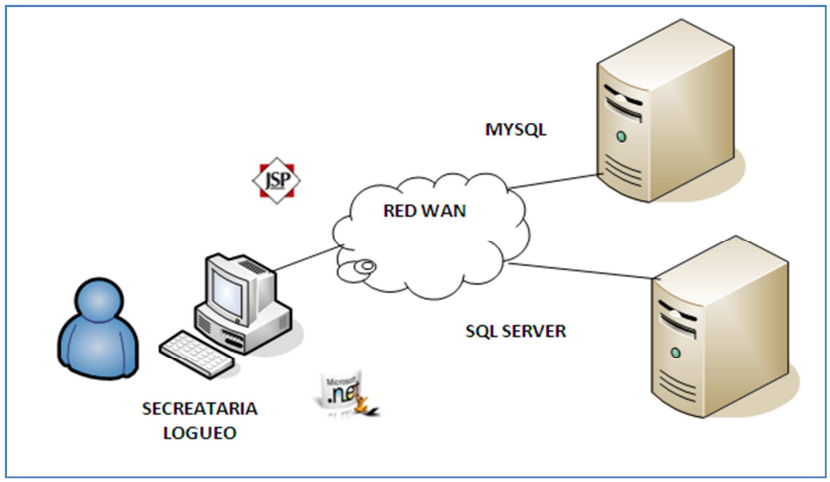

Figura 30: *Definición de la alternativa de solución*
Autor: *Johnny Latta*

4.20.4. CARACTERÍSTICAS DE LOS USUARIOS

Los usuarios o el usuario que utiliza el sistema debe cumplir con ciertos requisitos, tales como: conocer bien el proceso de prácticas, el proceso de anteproyectos y tesis y el proceso generación de actas, conocimientos en computación y manejo del sistema GESTIÓN UNACH.

4.20.5. REQUISITOS FUNCIONALES

Tabla 26: *Requisitos Funcionales*

N º	Requisito	Responsable
RQ1	Definición de usuarios	Administrador
RQ 2	Gestión de prácticas	Secretaria
RQ 3	Definición del sistema	Administrador
RQ 4	Gestión de anteproyectos	Secretaria
RQ 5	Gestión de tesis	Secretaria
RQ 6	Generar Reportes	Secretaria
RQ 7	Cargar resultados	Sistema/Administrador

Autor: *Johnny Latta*

4.20.6. REQUISITOS DE INTERFAZ

4.20.6.1. INTERFAZ CON EL USUARIO

Las salidas del sistema van destinadas al usuario final, y serán ofrecidas a través de la pantalla, con posibilidad de visualizar las opciones debidas para gestionar los módulos y resultados obtenidos después de las tareas.
Además, el usuario podrá elegir entre dos tipos de interfaz para comunicarse con el sistema, bien con las teclas o botones, como se hace habitualmente.

4.20.6.2. INTERFAZ CON OTROS SISTEMAS

El sistema utilizará como repositorio para toda la información registrada de estudiantes y docentes de la Facultad de Ingeniería, una base de datos diseñada en MySql y en SQL Server y para mejorar los procesos se hará consumo de servicios web desde .Net.

4.20.7. REQUISITOS NO FUNCIONALES

4.20.7.1. REQUISITOS HARDWARE

Con el fin de que el sistema Gestión Unach sea lo más accesible y evitar errores de comunicación en su explotación, se propone que todo el equipo hardware del usuario secretaria, sea un ordenador de escritorio con altas capacidades de soporte técnico. Para lo cual se recomienda equipo hardware de tecnología actual, con las siguientes características:

Tabla 27: *Requisitos Hardware*

Cantidad	Hardware	Especificaciones
1	Computador personal de escritorio	Procesador INTEL Core i7 3.0 G.Z mínimo Memoria 1 GB DDR2 mínimo Disco Duro 600 GB, CD-ROM Teclado Mouse Monitor Samsung SyncMaster S58 de 17”

Autor: *Johnny Latta*

4.20.7.2. REQUISITOS SOFTWARE

La aplicación debe funcionar en cualquier ordenador, con cualquier sistema operativo Windows. Lo importante es que debe tener conexión a internet y tener instalado un navegador de páginas web (Mozilla Firefox, Internet Explorer), además de tener levantado el servidor de aplicaciones Internet Information Server (IIS):

- **Disponibilidad:** El producto desarrollado estará disponible para realizar la gestión de prácticas, anteproyectos y tesis, y gestión de actas, siempre y cuando no haya problemas de índole externo como cortes de energía, fallas en las conexiones en la intranet de la Unach, etc.

- **Confiabilidad:** El producto debe ser confiable, almacenará información verídica y sujeta a comprobación con registros manuales de ser necesario.

- **Mantenibilidad:** El sistema será fácil de mantener, además se dejará la documentación para realizar posibles modificaciones de ser necesario.

- **Amigable:** El sistema presentará interfaces amigables, intuitivas y de fácil interacción con el usuario final (administrador y secretaria).

- **Seguridad:** Solo se le permitirá el acceso a los usuarios que posean una cuenta, podrán ingresar su login y password de autenticación.

4.20.8. ESTIMACIÓN DE COSTOS

La estimación de costos para el proyecto se lo realiza con la finalidad de obtener una valoración del esfuerzo y recursos necesarios para el desarrollo del software.

4.20.8.1. COSTOS COMPLEMENTARIOS

Son los costos que intervienen en el desarrollo de la aplicación, tanto en equipo hardware como en herramientas software.

4.20.8.2. COSTOS DE HARDWARE

Durante el análisis, en la definición del ámbito, en las secciones antecedentes tecnológicos se ha definido que la institución cuenta con los recursos hardware necesario para el funcionamiento del sistema Gestión Unach.

4.20.8.3. COSTOS DE SOFTWARE

En cuanto al software de desarrollo no existe costo a pagarse por motivo de licencias, porque se utiliza software libre para el desarrollo de la aplicación.

4.20.9. FACTIBILIDAD

El estudio de la factibilidad es un elemento importante para personalizar una aplicación ya que determina si se puede continuar o no con el desarrollo del sistema, debido a que la factibilidad se relaciona con los recursos que se necesitan y con los que se disponen, el estudio de la factibilidad ayuda a la toma de decisiones. El detalle de los tipos de factibilidad que se considerara para el desarrollo del sistema Gestión Unach se muestra a continuación:

4.20.9.1. FACTIBILIDAD TÉCNICA

Se refiere a la disponibilidad de la tecnología (técnica – humana) que satisfaga las necesidades del usuario, es decir, determina si la solución propuesta puede ser implantada con el hardware, software y recurso técnico (recurso humano) que esté disponible.

Para el desarrollo de la aplicación web Gestión Unach se cuenta con la mayor parte de recursos hardware y software necesarios. A continuación se detalla las características del hardware y características del software requerido, así como el personal técnico y profesional requerido para el desarrollo del sistema:

Tabla 28: *Hardware existente*

Cantidad	Descripción	Observación
2	Computadora	Desarrollo de la aplicación y documentación.
1	Infraestructura de Red	Acceder al internet para consultar las dudas en el desarrollo de la aplicación.

Autor: *Johnny Latta*

El hardware requerido para el desarrollo de la aplicación Gestión Unach es el siguiente:

Tabla 29: *Hardware requerido*

Cantidad	Descripción	Observación
1	Impresora	Impresión de los informes

Autor: *Johnny Latta*

Para el desarrollo de la aplicación Gestión Unach se dispone del siguiente software:

Tabla 30: *Software existente*

Nombre	Descripción	Estado	Observaciones
Windows 7	Sistema Operativo	Legal	Ninguna
Star UML	Herramienta CASE para el análisis y diseño orientado a objetos en el desarrollo de software	Sin licencia	Ninguna
MySQL v2.5	DBMS	Legal	Ninguna
Microsoft Project	Programa para planificación de actividades	Sin licencia	Ninguna

Autor: *Johnny Latta*

Tabla 31: *Software requerido*

Nombre	Descripción	Estado	Observaciones
Ireport Report 3.5.3	Programa para generar reportes de la aplicación web.	Legal	Ninguna

Autor: *Johnny Latta*

Tabla 32: *Recurso Humano requerido*

Función	Formación	Experiencia
Desarrollador	Estudiante de la escuela de Ingeniería en Sistemas	Desarrollo web, html, jsp, java, y java script.
Administrador de base de datos	Estudiante de la escuela de Ingeniería en Sistemas	Transact MySQL

Autor: *Johnny Latta*

4.20.9.2. FACTIBILIDAD OPERATIVA

Se refiere al recurso humano que participan en la operación del sistema cuando se instala:

- **Usuarios directos:** Quienes van a manejar el sistema.
- **Personal a capacitar:** Personal del Decanato de la Facultad de Ingeniería y de la Escuela de Ingeniería en Sistemas de la Universidad Nacional de Chimborazo.

Tabla 33: *Recurso humano*

Nombre	Función
Administrador	Director del Departamento de Sistemas Unach.
Secretaria	Responsable de la gestión de prácticas, anteproyectos y tesis y gestión de actas.

Autor: *Johnny Latta*

4.20.9.3. FACTIBILIDAD LEGAL

El Sistema Gestión Unach para gestionar prácticas pre profesionales, anteproyectos y tesis tiene reservado todos los derechos de autor, cualquier copia parcial o total debe ser autorizada por los autores según la Ley de Propiedad Intelectual. Según lo estipula en la Legislación Ecuatoriana que consta en el código penal con la Ley de Comercio Electrónico, firmas electrónicas y mensajes de datos. Esto asegura que el sistema que se está proponiendo es legalmente factible, ya que no existen impedimentos para que se pueda llevar a cabo.

4.20.9.4. FACTIBILIDAD ECONÓMICA

Tabla 34: *Factibilidad económica*

Costos	Valor
Costos	**$3376,00**
Costos de desarrollo	**$2950,00**
Costo de personal técnico	$ 500,00
Costo de hardware y software	$ 1500,00
Costos de suministros	$ 950,00
Costos de instalación	**$00,00**
Costos de personal para instalación	$00,00
Costos de operación	**$00,00**
Costos de personal de operación	$00,00
Costo de mantenimiento hardware	$00,00
Costos de suministros de operación	$00,00

Autor: *Johnny Latta*

4.20.10. PLANIFICACIÓN Y ANÁLISIS DE RIESGOS

El principal factor crítico en el desarrollo de un sistema software es el análisis de riesgos, esta es una estrategia que se utiliza para gestionar los riesgos de una manera efectiva y de esta forma evitar que dichos riesgos se transformen en problemas.

Nomenclatura utilizada:

RP: Riesgo del proyecto

RT: Riesgo Técnico

RN: Riesgo del Negocio

4.20.10.1. IDENTIFICACIÓN DE RIESGOS

Tabla 35: *Identificación de Riesgos*

Riesgo	Descripción del riesgo	Categoría	Consecuencia
R1	Cambio de asesor técnico del proyecto (director de tesis).	RP	Cambio de proyecto. Pérdida de tiempo
R2	Inexistencia del hardware requerido para la implementación del proyecto en la empresa.	RP,RT	Retraso del proyecto Riesgo de no poder implementar el sistema.
R3	El presupuesto asignado no fue suficiente para culminar el proyecto.	RN	Retraso del proyecto hasta que se obtengan los nuevos recursos.
R4	Falta de formación del equipo de desarrollo en el manejo de las herramientas.	RP	Retraso en la realización del proyecto.
R5	Pérdida de apoyo del nivel estratégico de la empresa.	RN	Desconfianza en el nivel estratégico. No poder continuar con el proyecto.
R6	Cambio del responsable del proyecto.	RN	Implementacióndesordenada de laaplicación.
R7	Daño en los repositorios de almacenamiento de información.	RT	Pérdida significativa del trabajo realizado.

Autor: *Johnny Latta*

4.20.10.2. CATEGORIZAR EL RIESGO

Tabla 36: *Valoración de la probabilidad para los riesgos*

Porcentaje	Descripción	Valor
1% - 33%	Baja	1
34%- 67%	Media	2
68%- 99%	Alta	3

Autor: *Johnny Latta*

Tabla 37: *Valoración del Impacto de los riesgos*

Impacto	Costo	Retraso	Impacto Técnico	Valor
Bajo	< 1 %	1 semana	Ligero efecto en el desarrollo del proyecto.	1
Moderados	< 5%	2 semanas	Moderado efecto en el desarrollo del proyecto.	2
Alto	< 10%	1 mes	Severo efecto en el desarrollo del proyecto	3
Crítico	> 10%	> 1 mes	El proyecto no puede ser culminado.	4

Autor: *Johnny Latta*

Tabla 38: *Valoración de la Exposición de riesgos*

Impacto / Probabilidad	Baja=1	Moderado=2	Alto=3	Crítico=4
Alta=3	3	6	9	12
Media=2	2	4	6	8
Baja=1	1	2	3	4

Autor: *Johnny Latta*

Tabla 39: *Código de colores según la exposición de riesgos*

Exposición del Riesgo	Valor	Color
Baja	1 o 2	Verde
Media	3 o 4	Amarillo
Alta	>= 6	Rojo

Autor: *Johnny Latta*

Tabla 40: *Determinación de la Prioridad de Riesgos*

Riesgo	Probabilidad			Impacto		Exposición al riesgo	
	%	Valor	Proba.	Valor	Impacto	Valor	Exposición.
R2	40%	2	Media	3	Alto	6	Alta
R5	30%	2	Media	3	Alto	6	Alta
R3	40%	2	Media	3	Alto	6	Alta
R1	70%	1	Baja	3	Alto	3	Media
R6	20%	1	Baja	3	Alto	3	Media
R7	10%	1	Baja	3	Alto	4	Media
R4	10%	1	Baja	2	Media	2	Baja

Autor: *Johnny Latta*

A. Hoja de Riesgo

Tabla 41: *Hoja de gestión del riesgo – Cambio de asesor técnico del proyecto*

<table>
<tr><td colspan="4">HOJA DE GESTIÓN DEL RIESGO</td></tr>
<tr><td colspan="2">ID. DEL RIESGO: R1</td><td colspan="2">FECHA: 16-12-2012</td></tr>
<tr><td>Probabilidad: Baja
Valor: 1</td><td>Impacto: Alto
Valor: 3</td><td>Exposición: Media
Valor: 3</td><td>Prioridad: 1</td></tr>
<tr><td colspan="4">DESCRIPCIÓN: Cambio de asesor técnico del proyecto (director de tesis).</td></tr>
<tr><td colspan="4">REFINAMIENTO:
<u>Causas:</u> Solicitud de renuncia, por parte del profesor, enfermedad o disposición de las autoridades de la escuela.
<u>Consecuencias:</u> Cambio de proyecto, pérdida de tiempo.</td></tr>
<tr><td colspan="4">REDUCCIÓN:
Cooperar entre desarrolladores del proyecto de tesis en todos los aspectos.
Informar a las autoridades la situación del proyecto.</td></tr>
<tr><td colspan="4">SUPERVISIÓN:
Al inicio del proyecto se debe quedar de acuerdo en la planificación del proyecto, para no tener problemas a futuro.</td></tr>
<tr><td colspan="4">GESTIÓN:
Agotar todos los medios necesarios para lograr la permanencia del asesor dentro del proyecto.</td></tr>
<tr><td colspan="4">ESTADO ACTUAL:
Fase de reducción iniciada [X]
Fase de Supervisión iniciada [ ]
Gestionando el riesgo [ ]</td></tr>
<tr><td colspan="4">RESPONSABLES: Johnny Danilo Latta Chavarrea</td></tr>
</table>

Autor: *Johnny Latta*

Tabla 42: *Hoja de gestión del riesgo – Inexistencia del hardware requerido*

HOJA DE GESTIÓN DEL RIESGO			
ID. DEL RIESGO: R2		FECHA: 16-12-2012	
Probabilidad: Media Valor: 2	Impacto: Alto Valor: 3	Exposición: Alta Valor: 6	Prioridad: 1
DESCRIPCIÓN: Inexistencia del hardware requerido para la implementación del proyecto en la escuela.			
REFINAMIENTO: Causas: La tecnología que se utilizará requiere equipo de cómputo adecuado para su normal y óptimo rendimiento. Consecuencias: Retraso del proyecto y riesgo de no poder implementar el sistema.			
REDUCCIÓN: Adquirir equipos de última tecnología adecuados para la implementación del sistema.			
SUPERVISIÓN: Solicitar al decano de la facultad la adquisición de los equipos necesarios para la implementación del sistema.			
GESTIÓN: Lograr obtener las características necesarias que deben tener los equipos para el correcto funcionamiento del sistema.			
ESTADO ACTUAL: Fase de reducción iniciada [X] Fase de Supervisión iniciada [ ] Gestionando el riesgo [ ]			
RESPONSABLES: Johnny Danilo Latta Chavarrea			

Autor: *Johnny Latta*

Tabla 43: *Hoja de gestión del riesgo – Presupuesto asignado no es suficiente*

HOJA DE GESTIÓN DEL RIESGO			
ID. DEL RIESGO: R3		FECHA:20-12-2012	
Probabilidad: Media Valor: 2	Impacto: Alta Valor: 3	Exposición: alta Valor: 6	Prioridad: 1
DESCRIPCIÓN: El presupuesto asignado no fue suficiente para culminar el proyecto.			
REFINAMIENTO: Causas: Planificación incorrecta de los recursos disponibles para la ejecución del proyecto o mala inversión de los mismos para llevar a cabo tareas no previstas. Consecuencias: Retraso del proyecto hasta que se obtengan los nuevos recursos.			
REDUCCIÓN: Emplear los recursos económicos solo en lo que necesariamente fue planificado No invertir recursos en tareas no planificadas para evitar que se agoten sobre la marcha del proyecto.			
SUPERVISIÓN: Controlar que los gastos efectuados hasta el momento sean únicamente los ya planificados.			
GESTIÓN: Llevar un gasto completamente eficaz para evitar el desperdicio del recurso económico.			
ESTADO ACTUAL: Fase de reducción iniciada [X] Fase de Supervisión iniciada [ ] Gestionando el riesgo [ ]			
RESPONSABLES: Johnny Danilo Latta Chavarrea			

Autor: *Johnny Latta*

Tabla 44: *Hoja de gestión del riesgo – Falta de formación del equipo de desarrollo*

HOJA DE GESTIÓN DEL RIESGO			
ID. DEL RIESGO: R4		FECHA:22-12-2012	
Probabilidad: Baja Valor: 1	Impacto: Media Valor: 2	Exposición: Media Valor: 2	Prioridad: 1
DESCRIPCIÓN: Falta de formación del equipo de desarrollo en el manejo de las herramientas.			
REFINAMIENTO: Causas: Desconocimiento de la utilización de las herramientas, utilización de nuevas tectologías Consecuencias: Retraso en la realización del proyecto			
REDUCCIÓN: Realizar cursos, recopilar información sobre las herramientas.			
SUPERVISIÓN: Al inicio del proyecto se deben conocer las herramientas que se emplea para así poder seguir investigando sobre las mismas.			
GESTIÓN: Preparar al equipo de desarrollo mediante cursos, bibliografía y consultas con personas que dominen las herramientas.			
ESTADO ACTUAL: Fase de reducción iniciada [X] Fase de Supervisión iniciada [ ] Gestionando el riesgo [ ]			
RESPONSABLES: Johnny Danilo Latta Chavarrea			

Autor: *Johnny Latta*

Tabla 45: *Hoja de gestión del riesgo – Pérdida de apoyo del nivel estratégico*

HOJA DE GESTIÓN DEL RIESGO			
ID. DEL RIESGO: R5		FECHA: 28-12-2012	
Probabilidad: Media Valor: 2	Impacto: Alto Valor: 3	Exposición: Alto Valor: 6	Prioridad: 1
DESCRIPCIÓN: Pérdida de apoyo del nivel estratégico de la escuela.			
REFINAMIENTO: Causas: Falta de comunicación por parte del Jefe del proyecto con el nivel estratégico acerca del avance del proyecto. Consecuencias: Desconfianza en el nivel estratégico con la realización del proyecto, No poder continuar con el proyecto.			
REDUCCIÓN: Mantener una comunicación constante con el nivel estratégico de la empresa.			
SUPERVISIÓN: Presentar los reportes acerca de los beneficios que el proyecto que se encuentra en ejecución traerá a la empresa una vez que este sea terminado.			
GESTIÓN: - Informar periódicamente al nivel estratégico sobre el avance del proyecto en ejecución.			
ESTADO ACTUAL: Fase de reducción iniciada [X] Fase de Supervisión iniciada [ ] Gestionando el riesgo [ ]			
RESPONSABLES: Johnny Danilo Latta Chavarrea			

Autor: *Johnny Latta*

Tabla 46: *Hoja de gestión del riesgo – Cambio del responsable del proyecto*

HOJA DE GESTIÓN DEL RIESGO			
ID. DEL RIESGO: R6		FECHA: 05-01-2012	
Probabilidad: Baja Valor: 1	Impacto: Alto Valor: 3	Exposición: Media Valor: 3	Prioridad: 1
DESCRIPCIÓN: Cambio del responsable del proyecto.			
REFINAMIENTO: Causas: Puede suceder en casos extremos, como fallecimiento, viaje obligado, desacuerdo con otro miembro del proyecto. Consecuencias: Retraso en la entrega, o no se realice el proyecto			
REDUCCIÓN: Mantener una buena relación entre los miembros del equipo de trabajo, en caso de desacuerdo.			
SUPERVISIÓN: Tratar de ser más responsables con las tareas encomendadas. Cumplir con el trabajo.			
GESTIÓN: Tener comunicación permanente con el equipo de trabajo, y comunicar de todos los avances del proyecto que se realice.			
ESTADO ACTUAL: Fase de reducción iniciada [X] Fase de Supervisión iniciada [ ] Gestionando el riesgo [ ]			
RESPONSABLES: Johnny Danilo Latta Chavarrea			

Autor: *Johnny Latta*

Tabla 47: *Hoja de gestión del riesgo – Daños en los repositorios de almacenamiento*

HOJA DE GESTIÓN DEL RIESGO			
ID. DEL RIESGO: R7		FECHA: 15-01-2013	
Probabilidad: Baja Valor: 1	Impacto: Alto Valor: 3	Exposición: Media Valor: 3	Prioridad: 1
DESCRIPCIÓN: Daños en los repositorios de almacenamiento de información.			
REFINAMIENTO: Causas: Puede suceder por mala manipulación del motor de base de datos o borrado de la base de datos. Consecuencias: Pérdida significativa del trabajo realizado.			
REDUCCIÓN: Realizar una o varias copias de seguridad.			
SUPERVISIÓN: Respaldar la información.			
GESTIÓN: Cada avance que se haga de debe guardar un respaldo en otra unidad de disco o en medios extraíbles.			
ESTADO ACTUAL: Fase de reducción iniciada [X] Fase de Supervisión iniciada [ ] Gestionando el riesgo [ ]			
RESPONSABLES: Johnny Danilo Latta Chavarrea			

Autor: *Johnny Latta*

4.20.11. DEFINICIÓN DE LOS CASOS DE USO

Según definió Jacobson un caso de uso es un documento narrativo que describe la secuencia de eventos de un actor (agente externo) que usa un sistema para completar un proceso, es decir, una forma de usar una función que ofrece el sistema.La secretaria de la Facultad de Ingeniería de la Unachconjuntamente con el sistema Gestión Unachtiene la finalidad de seguir el cumplimiento de realización de prácticas, anteproyectos y tesis y gestión de actas, cuya finalidad es aportar con la organización y ejecución de dichos procesos. Para dicha gestión se utiliza información procesada mediante el consumo de servicios web, ingreso de datos, y módulo de gestión del sistema, siguiendo las siguientes fases:

- **Primera Fase:** Configurar usuarios y permisos para el ingreso del usuario.
- **Segunda Fase:** Gestionar prácticas, anteproyectos y tesis y gestinar actas.
- **Tercera Fase:** Generar reportes de cada proceso respectivamente con información disponible en las bases de datos.

4.20.11.1. Diagramas de Casos de uso

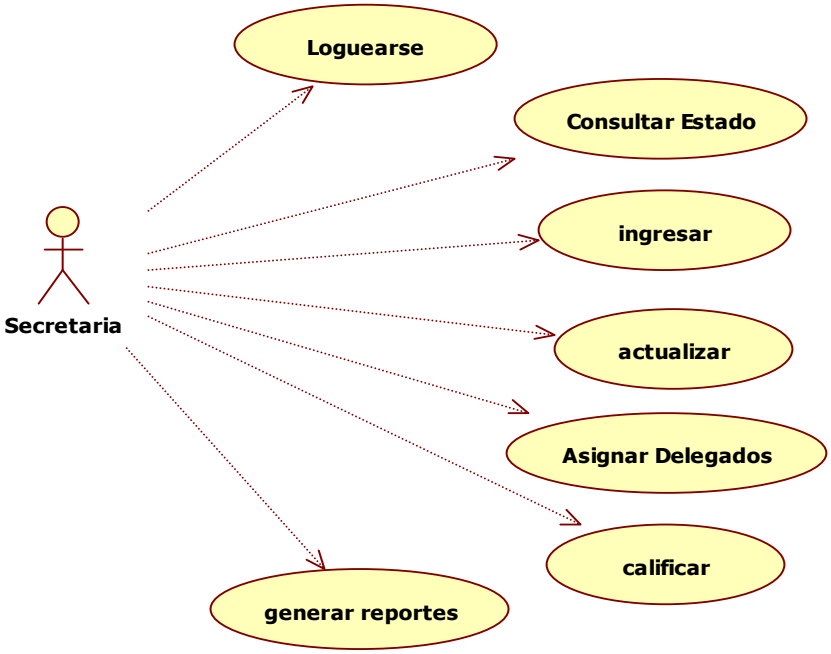

Figura 31*Diagrama casos de uso Gestión Prácticas*
Autor: *Johnny Latta*

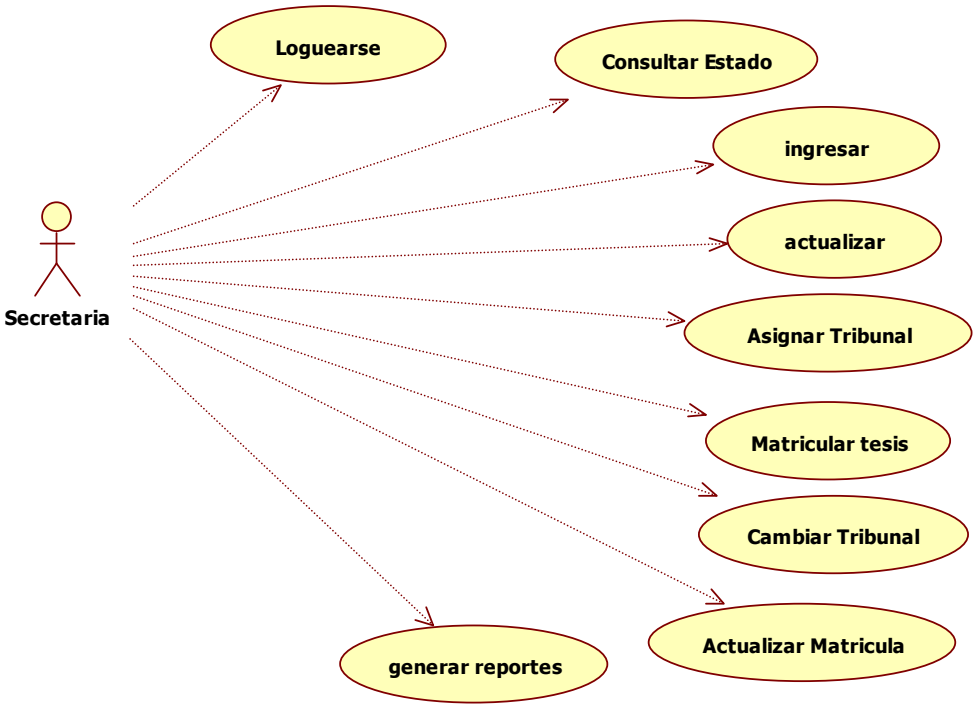

Figura 32: *Diagrama casos de uso Gestión Anteproyectos y Tesis*
Autor: *Johnny Latta*

4.20.11.2. Casos de uso de alto nivel

4.20.11.3. Secretaria – Gestión Prácticas

Tabla 48: *Caso de uso Loguearse*

Caso de Uso	Loguearse
Actores	Secretaria
Tipo	Primario
Descripción	Cuando se ingresa a la página inicial del sistema Gestión Unach, se procede a ingresar a la página de logueo, aquí se tiene q indicar el nombre del usuario y la clave. Entonces si los datos son correctos se ingresa al módulo de gestión.

Autor: *Johnny Latta*

Tabla 49: *Caso de uso Consultar Estado*

Caso de Uso	Consultar Estado
Actores	Secretaria
Tipo	Primario
Descripción	Para el correcto manejo de información, se procede a consultar si existe información por estudiante y su proceso de prácticas.Conocer que falta al proceso, asignar tribunal., calificar o el proceso esta terminado.

Autor: *Johnny Latta*

Tabla 50: *Caso de uso Ingresar nueva práctica*

Caso de Uso	Ingresar nueva práctica
Actores	Secretaria
Tipo	Primario
Descripción	Consiste básicamente en registrar información del estudiante de la facultad, información de las prácticas que realiza: lugar, duración, fecha inicio, fecha fin, tema de las prácticas, etc. Almacenando dicha información para actividades posteriores.

Autor: *Johnny Latta*

Tabla 51: *Caso de uso Actualizar Datos*

Caso de Uso	**Actualizar Datos**
Actores	Secretaria
Tipo	Primario
Descripción	Por razones naturales o particulares se procede a modificar la información de las prácticas que ya están ingresadas, previo a una consulta por número de cédula del estudiante. Cabe indicar que solo se actualiza cierta información, más no todos los campos.

Autor: *Johnny Latta*

Tabla 52: *Caso de uso Asignar Delegados*

Caso de Uso	**Asignar Delegados**
Actores	Secretaria
Tipo	Primario
Descripción	Cuando el proceso de prácticas pre profesionales ha culminado, se procede a valorar el trabajo realizado, para eso se designa a docentes de la facultad como miembros del tribunal para la calificación correspondiente.

Autor: *Johnny Latta*

Tabla 53: *Caso de uso Calificar práctica*

Caso de Uso	**Calificar práctica**
Actores	Secretaria
Tipo	Primario
Descripción	Una actividad que se realiza a un proceso de prácticas, es la calificación tanto a la defensa oral y escrita que realiza el estudiante, en este punto se registra una valoración cuantitativa de acuerdo al puntaje asignado por cada miembro del tribunal.

Autor: *Johnny Latta*

Tabla 54: *Caso de uso Generar reportes*

Caso de Uso	Generar reportes
Actores	Secretaria
Tipo	Primario
Descripción	Cuando ya se cuenta con los requerimientos del estudiante culminados, se procede a generar documentos que permitan registrar formalmente la terminación de las prácticas, constituyéndose en un requisito cumplido para el futuro profesional.

Autor: *Johnny Latta*

4.20.11.4. Secretaria – Gestión Anteproyectos y Tesis

Tabla 55: *Loguearse*

Caso de Uso	Loguearse
Actores	Secretaria
Tipo	Primario
Descripción	Cuando se ingresa a la página inicial del sistema Gestión Unach, se procede ingresar a la página de logueo, aquí se tiene que indicar el nombre del usuario y la clave. Entonces si los datos son correctos se ingresa al módulo de gestión.

Autor: *Johnny Latta*

Tabla 56: *Caso de uso Consultar Estado del Anteproyecto o Tesis*

Caso de Uso	Consultar Estado del Anteproyecto o Tesis
Actores	Secretaria
Tipo	Primario
Descripción	Para el correcto manejo de información, se procede a consultar si existe información por estudiante y su proceso de presentación de anteproyectos y ejecución del mismo. Conocer el estado del proceso, en ejecución, en ejecución con prorroga, etc.

Autor: *Johnny Latta*

Tabla 57: *Caso de uso Consultar Estado del Anteproyecto o Tesis*

Caso de Uso	**Ingresar nuevoanteproyecto**
Actores	Secretaria
Tipo	Primario
Descripción	Consiste básicamente en registrar información del estudiante de la facultad, información del anteproyecto de tesis que se pretende realizar: tema, fecha de registro, estado, número de estudiantes, escuela, facultad, tribunal de análisis.

Autor: *Johnny Latta*

Tabla 58: *Caso de uso Acualizar Datos del Anteproyecto*

Caso de Uso	**Acualizar Datos del Anteproyecto**
Actores	Secretaria
Tipo	Primario
Descripción	Por razones naturales o particulares se procede a modificar la información del anteproyecto que ya está registrado, previo a una consulta por número de cédula del estudiante, si es más de un estudiante, con el número de cédula de cualquiera de los estudiantes. Cabe indicar que solo se actualiza cierta información, más no todos los campos.

Autor: *Johnny Latta*

Tabla 59: *Caso de uso Asignar Tribunal*

Caso de Uso	**Asignar Tribunal**
Actores	Secretaria
Tipo	Primario
Descripción	Cuando se ha presentado el anteproyecto de tesis, se procede a registrar al tribunal que realizará la valoración del mismo, se registra a docentes de la facultad para el proceso.

Autor: *Johnny Latta*

Tabla 60: *Caso de uso Matricular Tesis*

Caso de Uso	Matricular Tesis
Actores	Secretaria
Tipo	Primario
Descripción	Una vez que el anteproyecto es aceptado se constituye en el proyecto de tesis, por tal razón se procede a matricular, dejando constancia de: fecha inicio, fecha fin, fecha de resolución,

Autor: *Johnny Latta*

Tabla 61: *Caso de uso Cambiar Tribunal*

Caso de Uso	Cambiar Tribunal
Actores	Secretaria
Tipo	Primario
Descripción	Cuando un anteproyecto se constituye en un proyecto de investigación o tesis, el tribunal de análisis del anteproyecto se puede constituir en delegados del anteproyecto o a su vez puede ser modificado, respetando la resolución del consejo directivo.

Autor: *Johnny Latta*

Tabla 62: *Caso de uso Actualizar Matricula de Tesis*

Caso de Uso	Actualizar Matricula de Tesis
Actores	Secretaria
Tipo	Primario
Descripción	La información registrada de la tesis, tiene la opción a ser modificada, cabe indicar que solo se actualiza campos específicos, por conservar la consistencia de la información.

Autor: *Johnny Latta*

Tabla 63: *Caso de uso Generar reportes*

Caso de Uso	Generar reportes
Actores	Secretaria
Tipo	Primario
Descripción	Cuando ya se cuenta con requerimientos cumplidos por parte del estudiante, se procede a generar documentos que permitan registrar formalmente la presentación, defensa privada, defensa oral, notas de grado, etc., constituyéndose en un requisito cumplido para el profesional.

Autor: *Johnny Latta*

4.20.11.5. Conceptos de casos de uso

Cabe mencionar que no todas las funciones que realiza el sistema se producen a través de la interacción con el usuario, lo cual imposibilita que se puedan definir a través de los casos de uso. Esto sucede con las acciones que el usuario delega en el sistema o con las que se realizan de manera automática, razón por la cual se utilizará la representación en forma de conceptos.

Tabla 64: *Conectar el sistema de evaluación con la base de datos*

Concepto de Uso	Concepto de la Operación
Conexión del Sistema Gestión Unach con la (s) base de datos	**Propósito:** Conectar el sistema de evaluación con la base de datos. **Modo de funcionamiento:**cuando se realice el consumo de servicios web se procede a la interacción del sistema web con la base de datos diseñada en SQL Server, debe existir una conexión automática, mediante su ODBC, debido a que se controla datos que se solicitan con los almacenados en la base de datos. **Dinámica:** Permanecer conectada durante el funcionamiento del sistema.

Autor: *Johnny Latta*

Tabla 65: *Contar con la información base para el trabajo de sistema Gestión Unach*

Concepto de Uso	Concepto de la Operación
Ingreso de información de prácticas, anteproyectos, tesis y estudiantes en la base de datos	**Propósito:**Contar con la información base para el trabajo de sistema Gestión Unach. **Modo de funcionamiento:** una vez que se define el proceso que se va a gestionar, esta debe ser validada para el ingreso debido las respuestas del proceso deben ser en tiempos. **Dinámica:** Funciona sólo al momento que se requiera.

Autor: *Johnny Latta*

4.21. DIAGRAMAS DE SECUENCIA DEL SISTEMA

- **Casos de uso Prácticas**

Logueo usuario

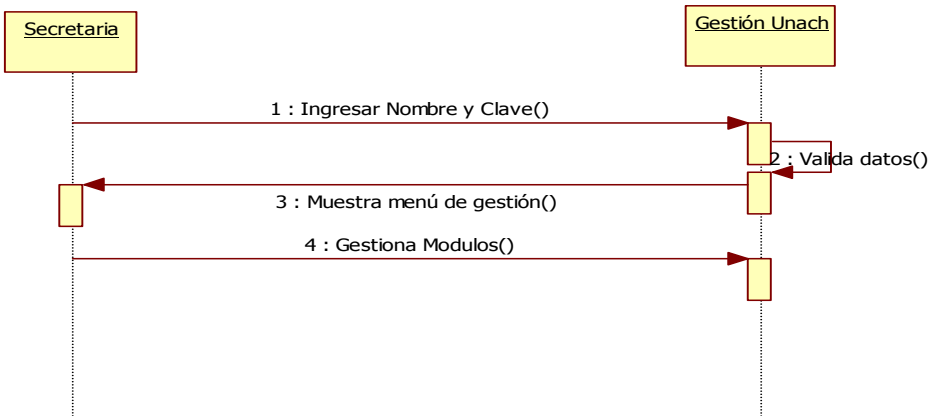

Figura 33: *Diagrama de secuencia Logueo Usuario*
Autor: *Johnny Latta*

➢ Consultar estado

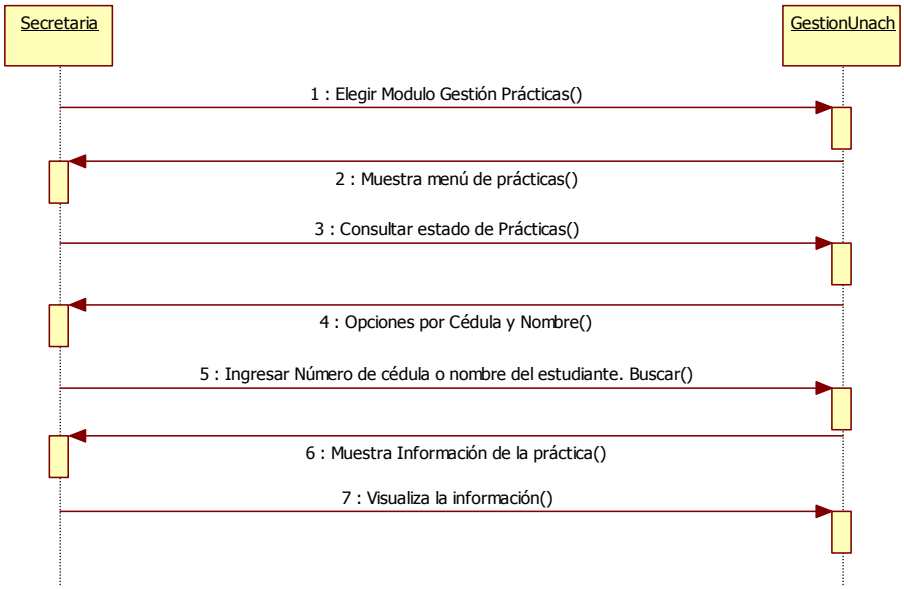

Figura 34: *Diagrama de secuencia Consultar Estado*
Autor: *Johnny Latta*

➢ Ingresar prácticas

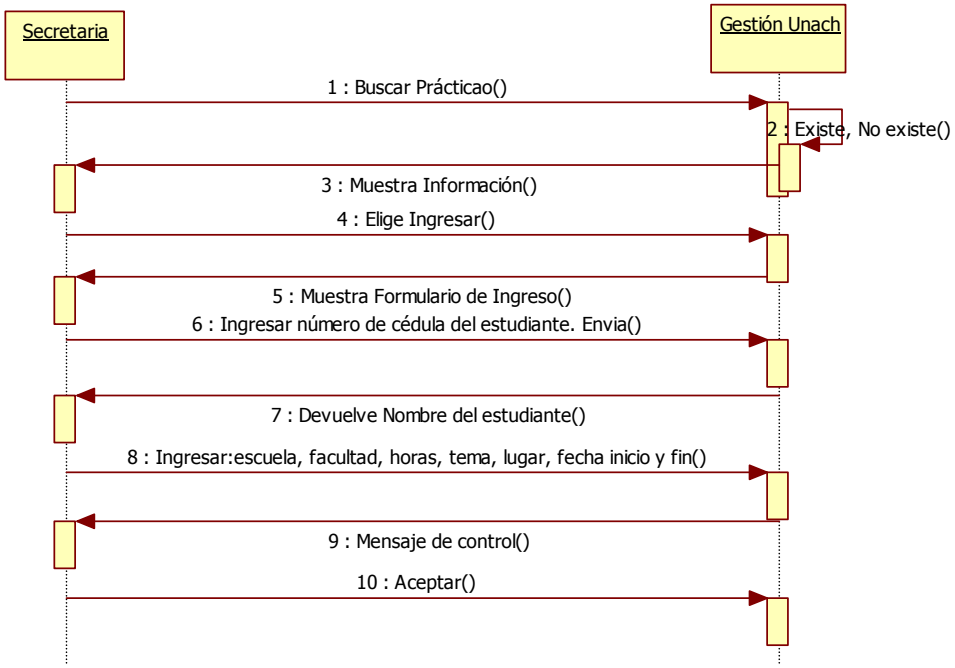

Figura 35: *Diagrama de secuencia Ingresar Práctica*
Autor: *Johnny Latta*

➢ Actualizar datos

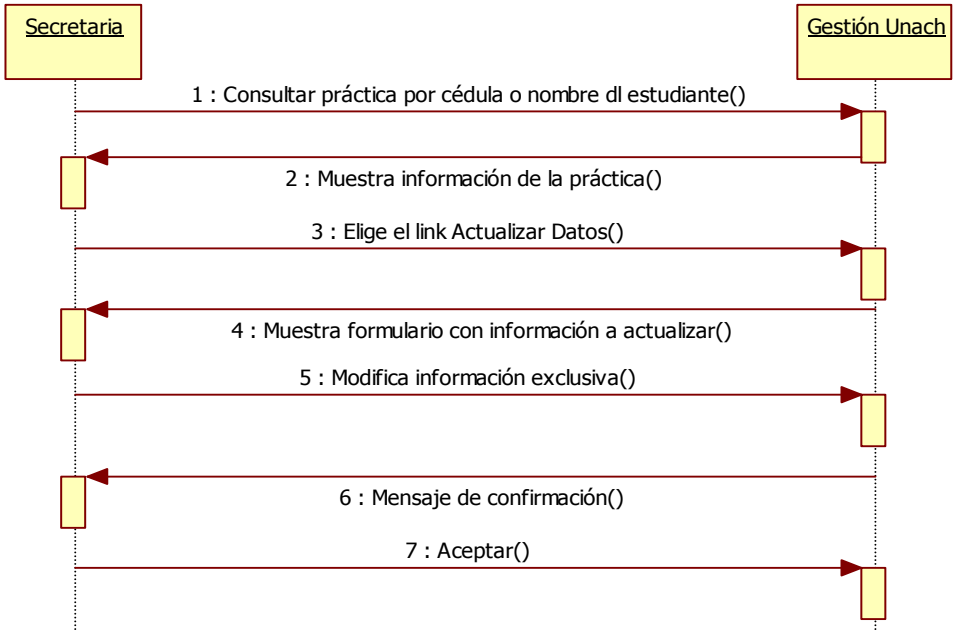

Figura 36: *Diagrama de secuencia Actualizar datos*
Autor: *Johnny Latta*

➢ Asignar delegados

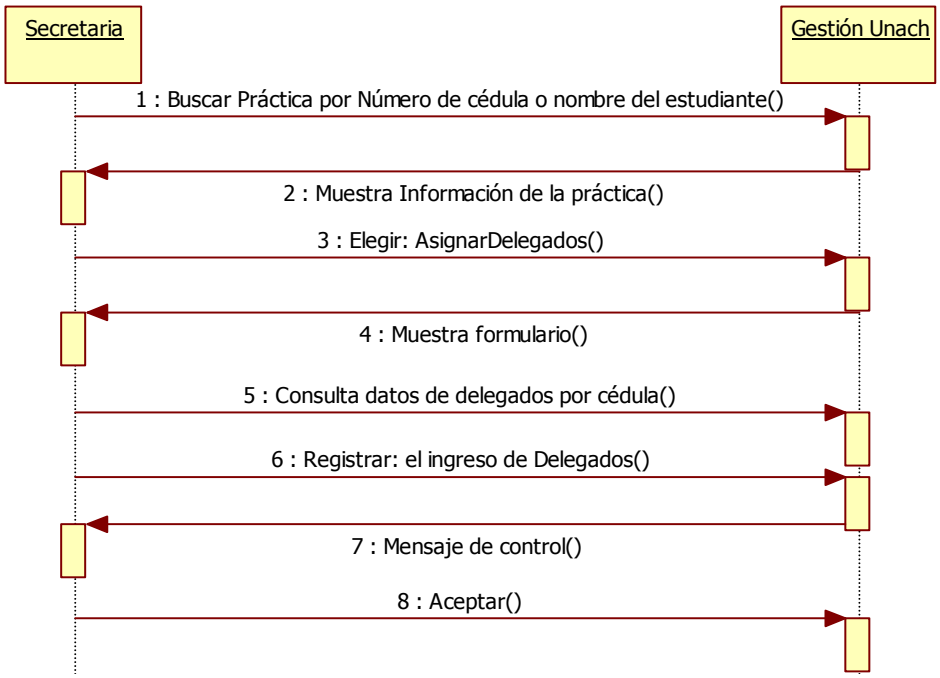

Figura 37: *Diagrama de secuencia Asignar Delegados*
Autor: *Johnny Latta*

➢ Calificar prácticas

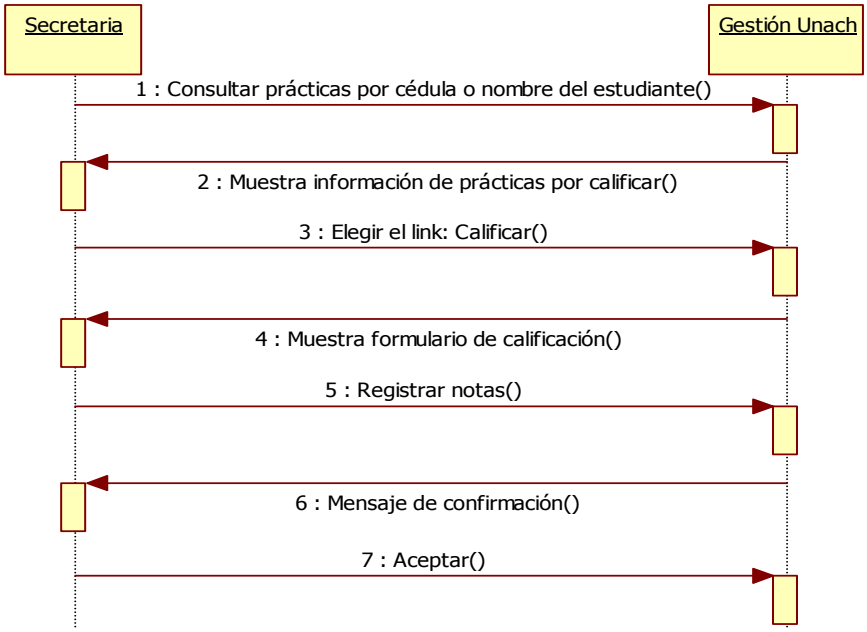

Figura 38: *Diagrama de secuencia Calificar Prácticas*
Autor: *Johnny Latta*

➢ Generar reportes

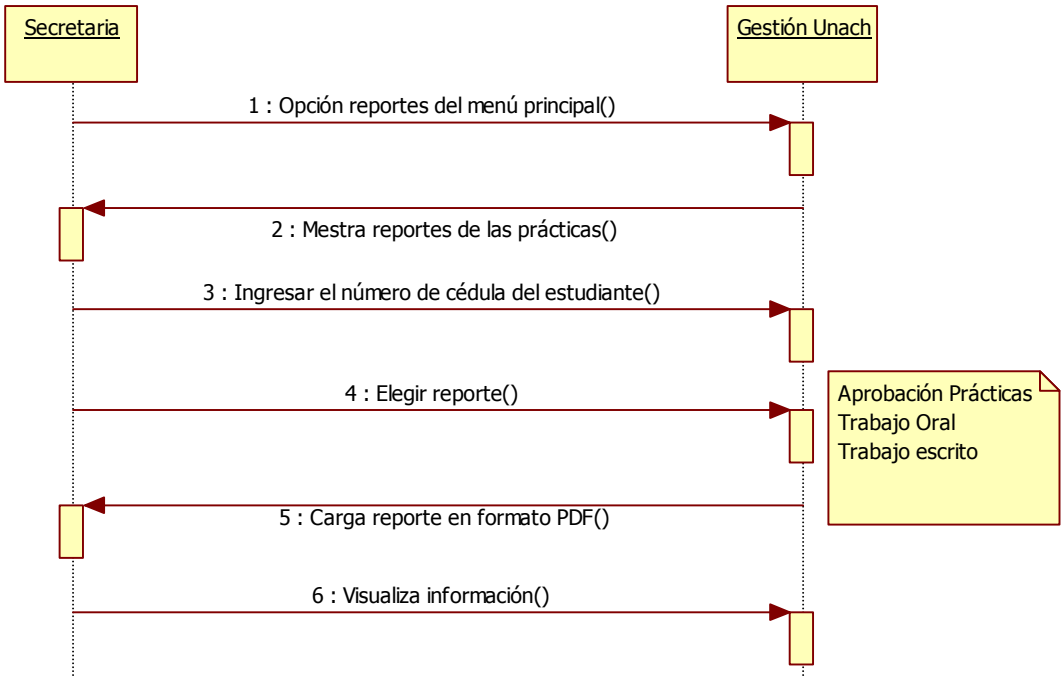

Figura 39: *Diagrama de secuencia Generar Reportes*
Autor: *Johnny Latta*

4.21.1.1. Casos de uso Anteproyectos y tesis

- **Logueo de usuario**

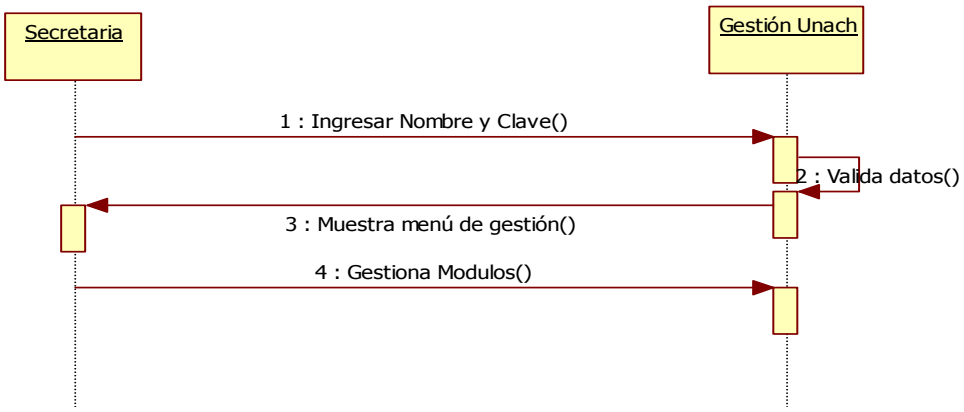

Figura 40: *Diagrama de secuencia Logueo Usuario*
Autor: *Johnny Latta*

- **Consultar estado**

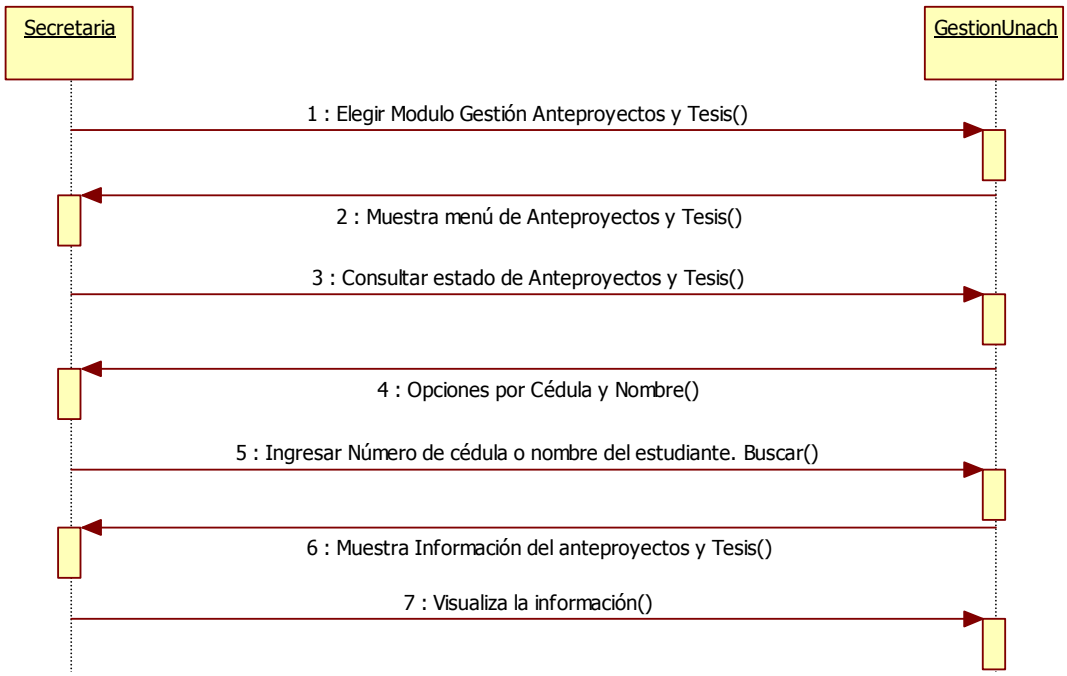

Figura 41: *Diagrama de secuencia Consultar estado*
Autor: *Johnny Latta*

➢ Ingresar anteproyecto

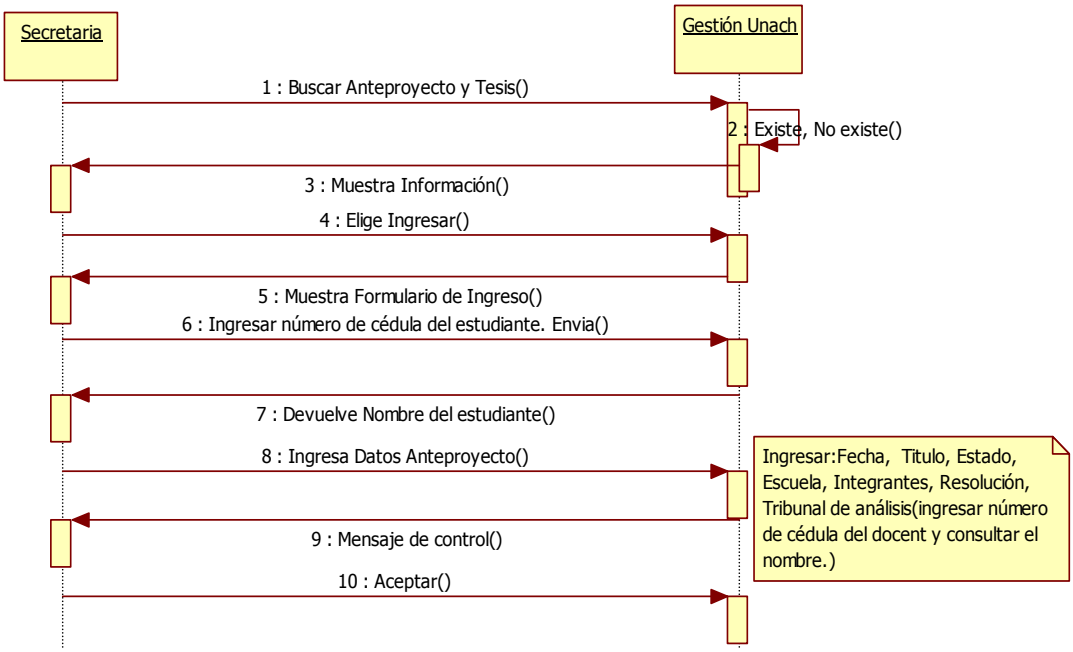

Figura 42: *Diagrama de secuencia Ingresar Anteproyecto*
Autor: *Johnny Latta*

➢ Actualizar datos

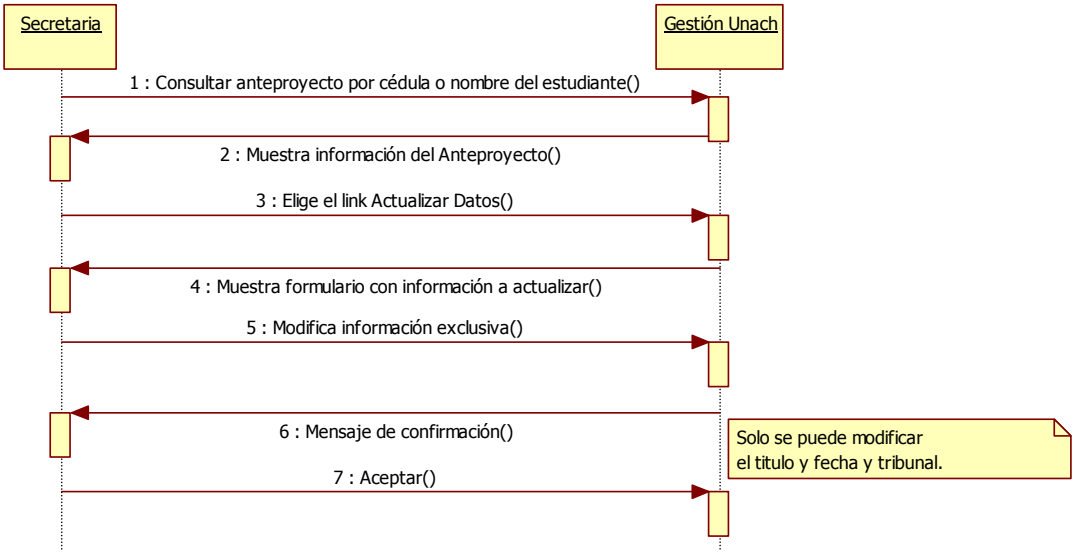

Figura 43: *Diagrama de secuencia Actualizar datos*
Autor: *Johnny Latta*

➢ Asignar delegados

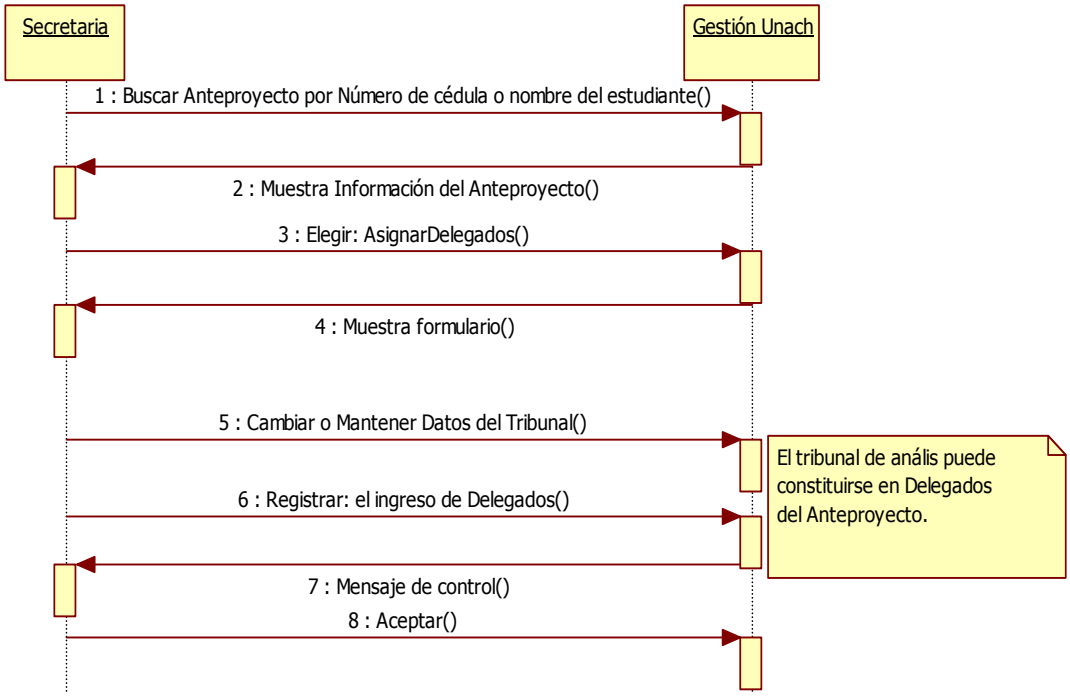

Figura 44: *Diagrama de secuencia Asignar Delegados*
Autor: *Johnny Latta*

➢ Matricular tesis

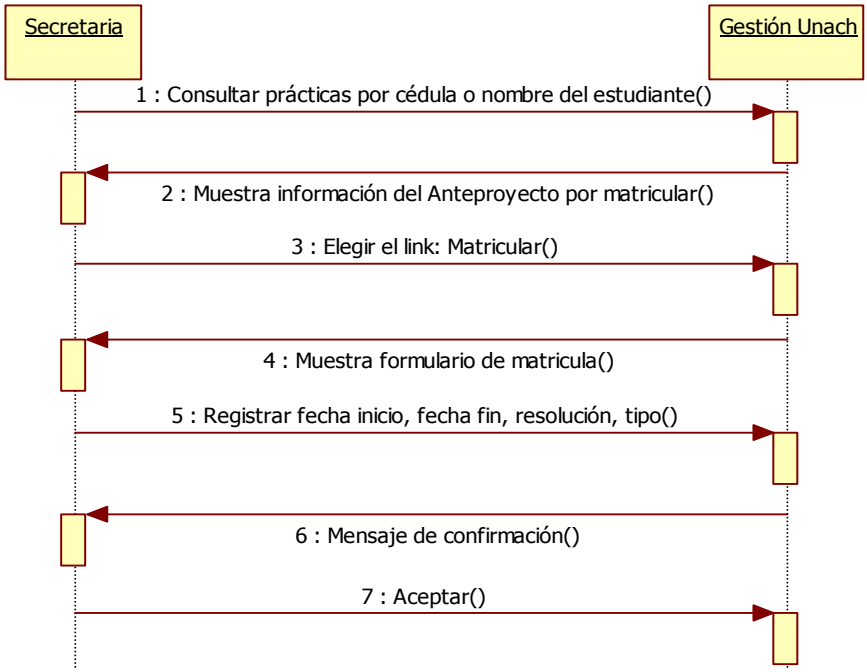

Figura 45: *Diagrama de secuencia Matricular Tesis*
Autor: *Johnny Latta*

➢ Actualizar matrícula

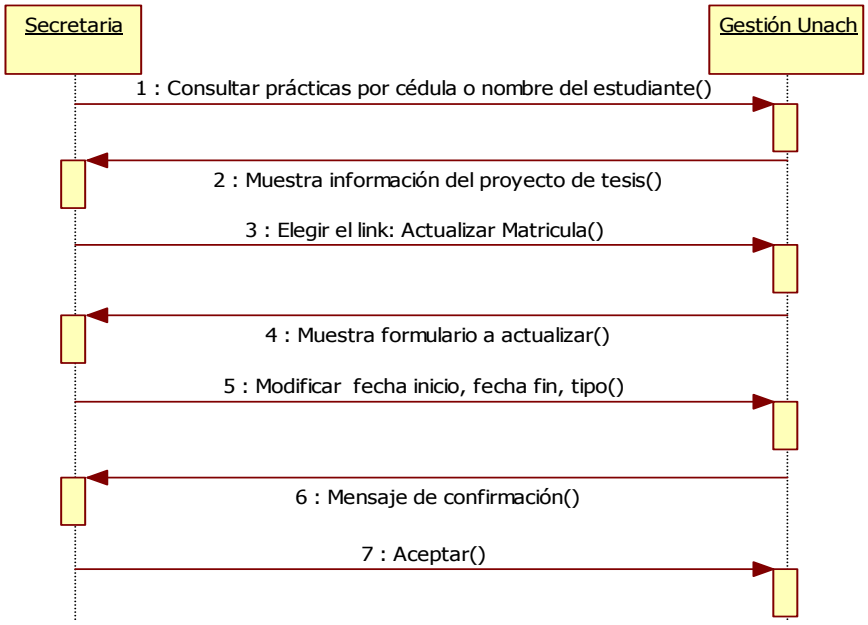

Figura 46: *Diagrama de secuencia Actualizar matrícula*
Autor: *Johnny Latta*

➢ Generar reportes

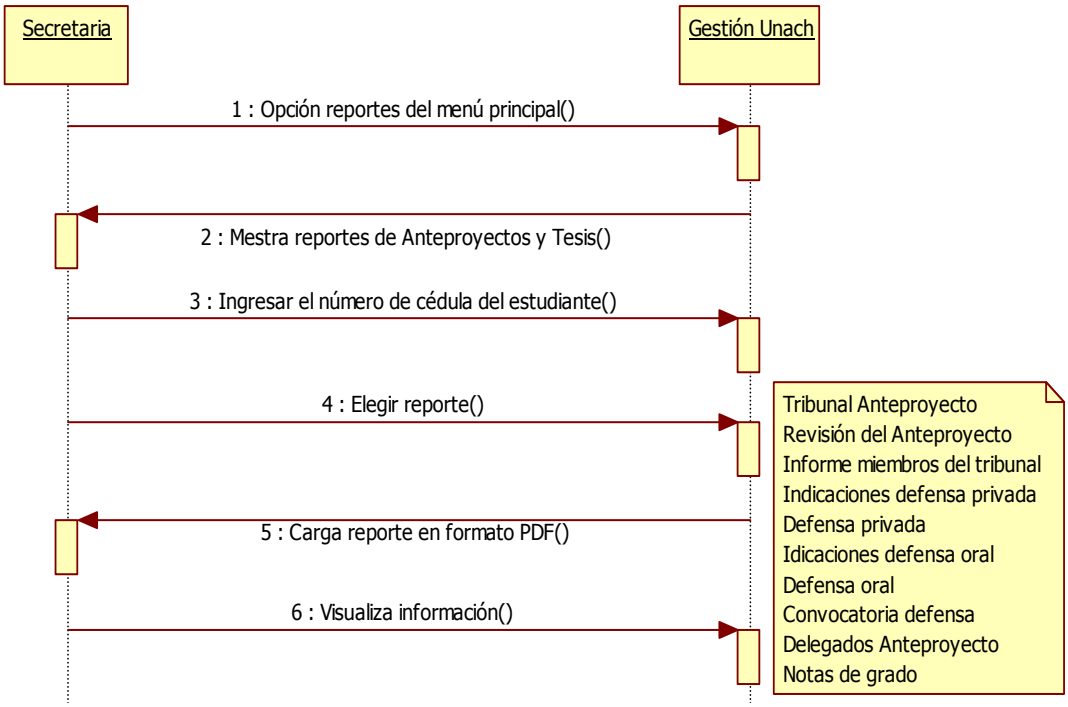

Figura 47: *Diagrama de secuencia Generar reportes*
Autor: *Johnny Latta*

4.21.2. DIAGRAMA DE ESTADOS

- **Gestión Prácticas**

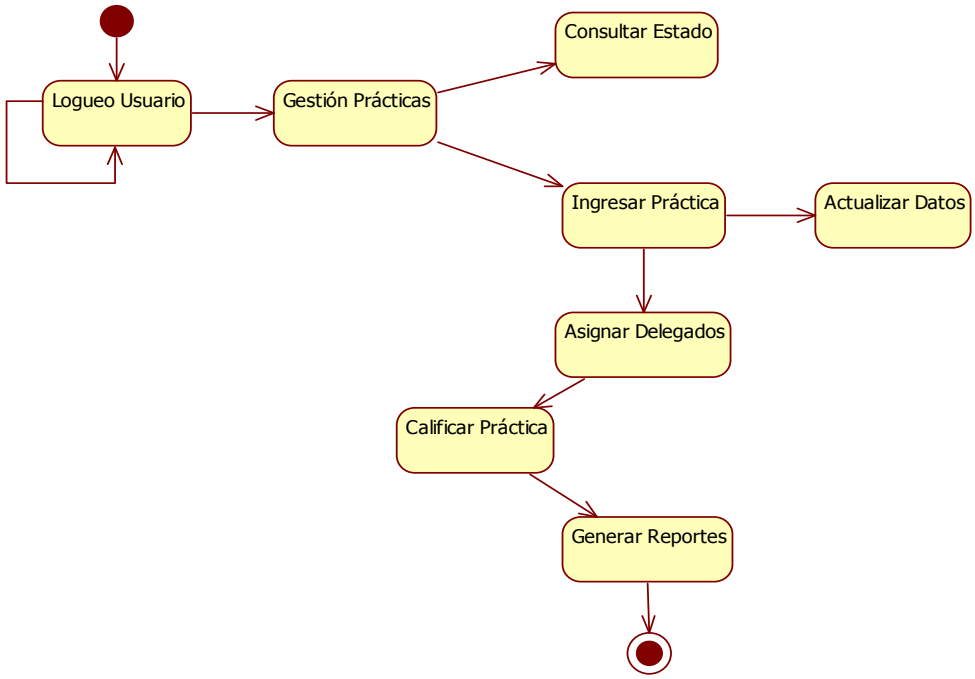

Figura 48: *Diagrama de estado Gestión Prácticas*
Autor: *Johnny Latta*

- **Gestión Anteproyectos y Tesis**

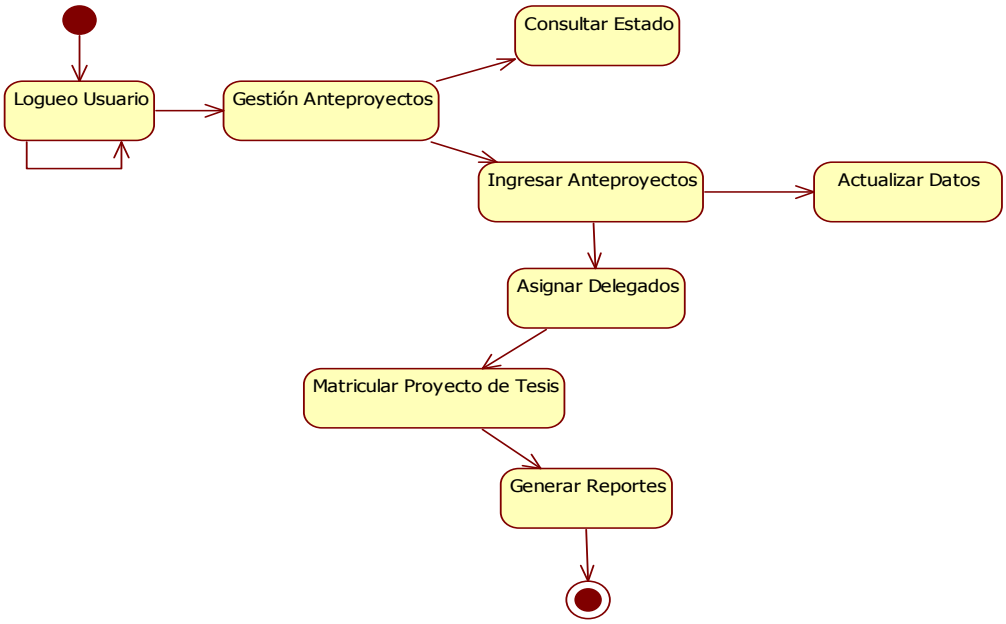

Figura 49: *Diagrama de estado Gestión Anteproyectos*
Autor: *Johnny Latta*

4.21.3. DIAGRAMA DE ACTIVIDADES

- **Gestión Prácticas**

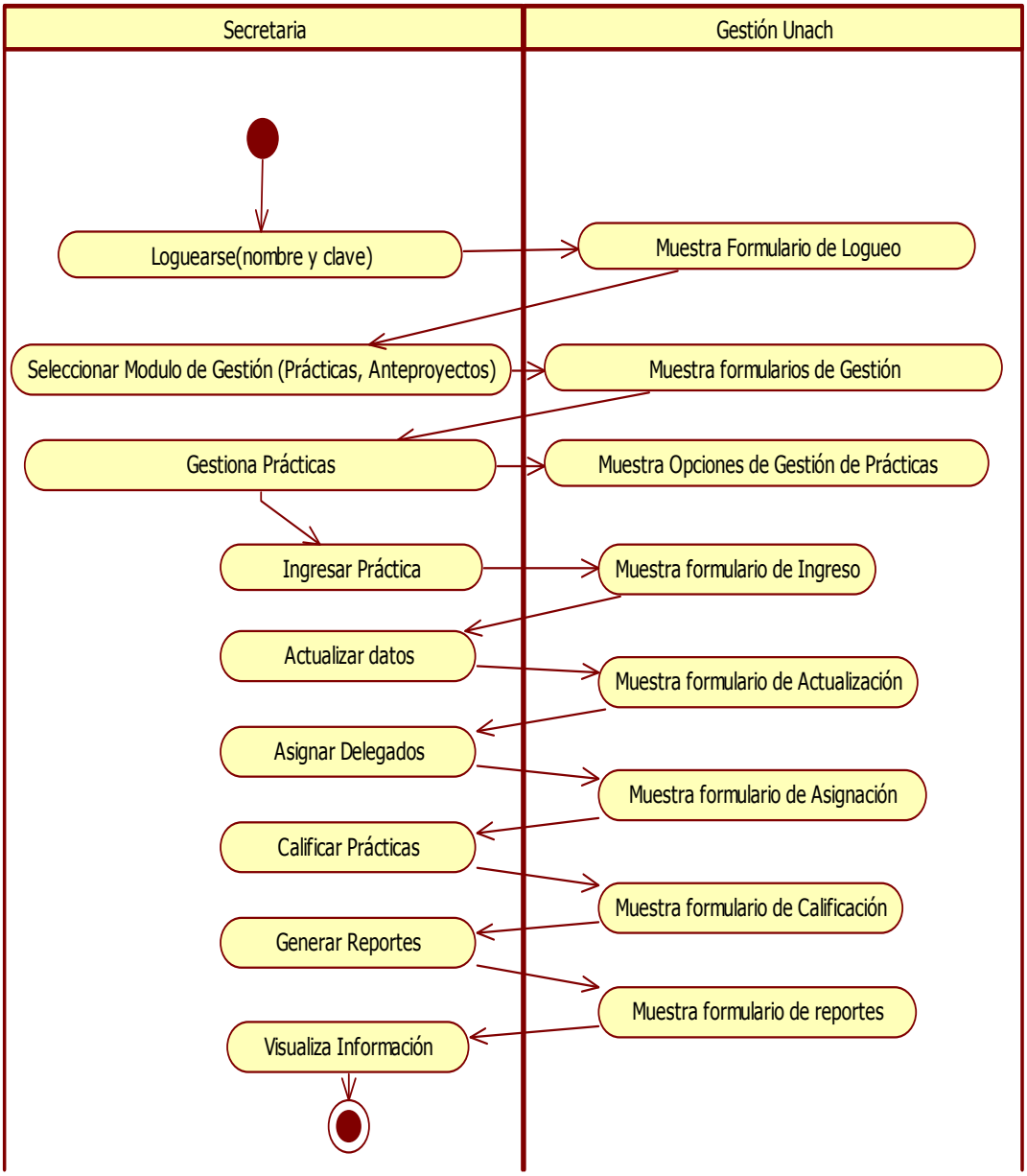

Figura 50: *Diagrama de actividades Gestión Prácticas*
Autor: *Johnny Latta*

➢ Gestión Anteproyectos y Tesis

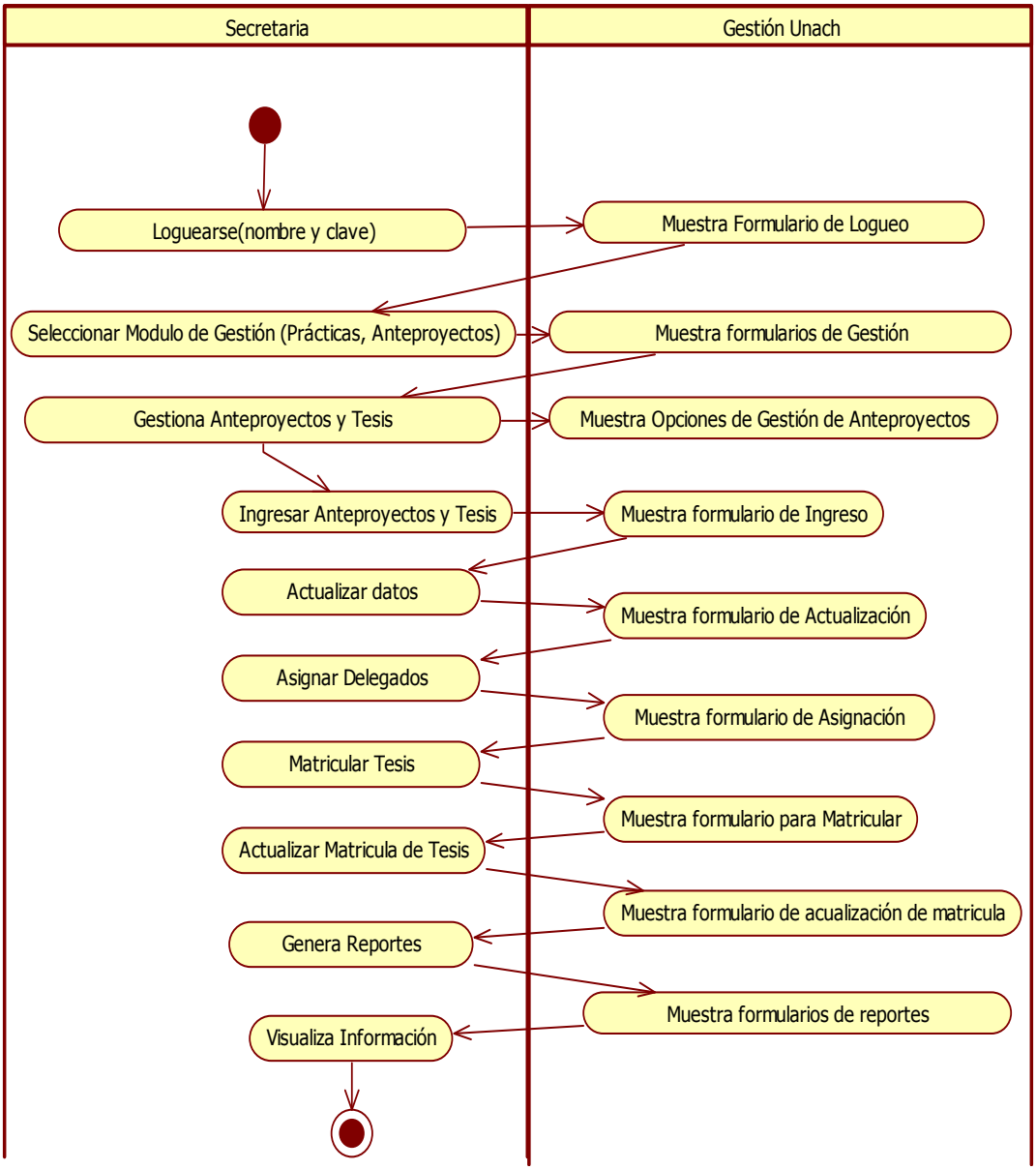

Figura 51: *Diagrama de estado Gestión Anteproyectos y tesis*
Autor: *Johnny Latta*

4.21.4. MODELO CONCEPTUAL

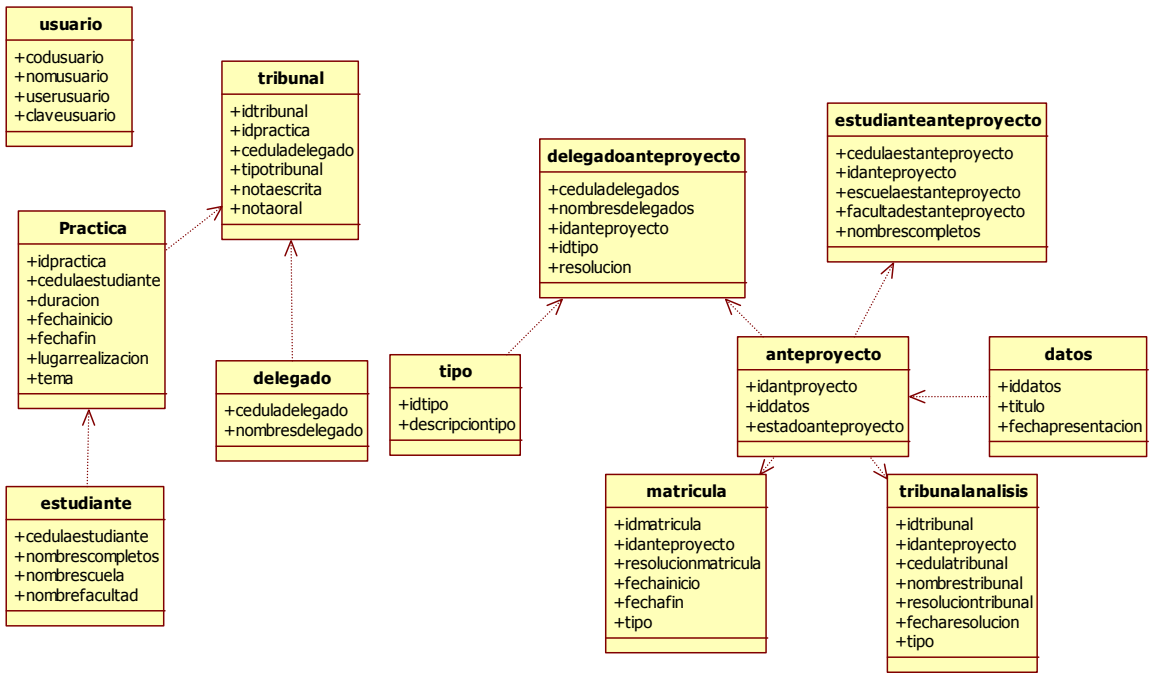

Figura 52:*Modelo conceptual Gestión Unach*
Autor: *Johnny Latta*

4.22. DISEÑO

La decisión de qué software y qué hardware se utilice, es fundamental, en esta fase se define la arquitectura del sistema, lo que permite tener una visión de la organización global del sistema que incluye sus componentes, las relaciones entre sí y el ambiente.

Se define las interfaces de usuario, cuál es el medio con que el usuario puede comunicarse con la computadora, la interacción del usuario con el sistema reflejada con los diagramas de interacción.

Además se elaborará un diagrama de clases que representará las clases que serán utilizadas dentro del sistema y las relaciones que existen entre ellas, a partir de ella tener un esquema de base de datos.

También para representar cómo un sistema está dividido en componentes y el hardware utilizado en las implementaciones de sistemas y las relaciones entre sus componentes se realizará el diagrama de componentes y despliegue.

4.22.1. DEFINICIÓN DE LA ARQUITECTURA DEL SISTEMA

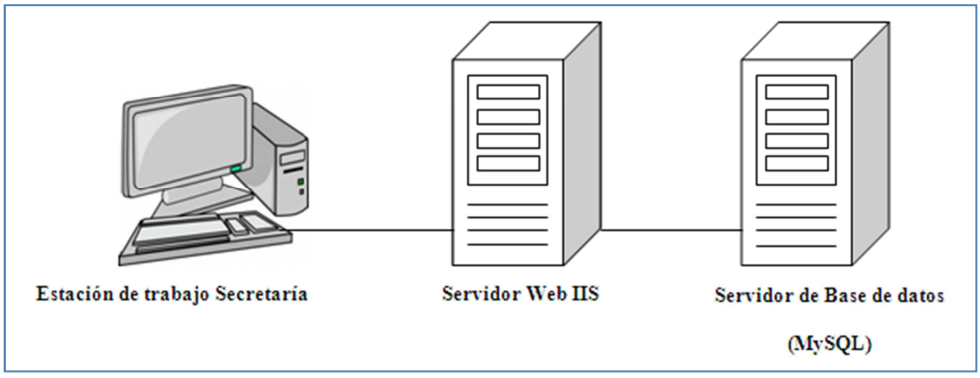

Figura 53: *Arquitectura del sistema*
Autor. *Johnny Latta*

4.22.2. DEFINICIÓN DE LA INTERFAZ DE USUARIO

- **Ingreso al sistema Gestión Unach y logueo de usuario**

Figura 54: *Página de inicio*
Fuente: *Sistema Web GestionUnach*

Figura 55: *Logueo de usuario*
Fuente: *Sistema Web GestionUnach*

Figura 56: *Opciones de Gestión Unach*
Fuente: *Sistema Web GestionUnach*

- **Módulo Gestión Prácticas**

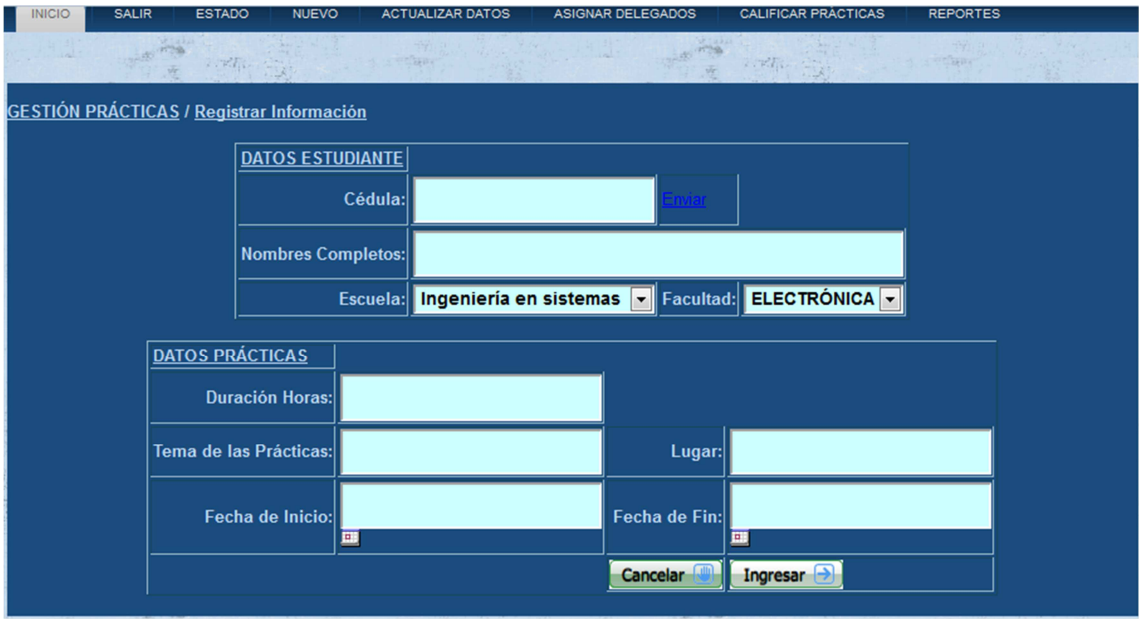

Figura 57: *Ingresar Práctica*
Fuente: *Sistema Web GestionUnach*

Figura 58: *Consultar Práctica*
Fuente: *Sistema Web GestionUnach*

Figura 59: *Actualizar datos*
Fuente: *Sistema Web GestionUnach*

Figura 60: *Asignar Delegados*
Fuente: *Sistema Web GestionUnach*

Figura 61: *Estado de la práctica*
Fuente: *Sistema Web GestionUnach*

Figura 62: *Calificar Práctica*
Fuente: *Sistema Web GestionUnach*

Figura 63: *Reportes Práctica*
Fuente: *Sistema Web GestionUnach*

- **Módulo Gestión Anteproyectos y Tesis**

Figura 64: *Menú Gestión Anteproyectos y tesis*
Fuente: *Sistema Web GestionUnach*

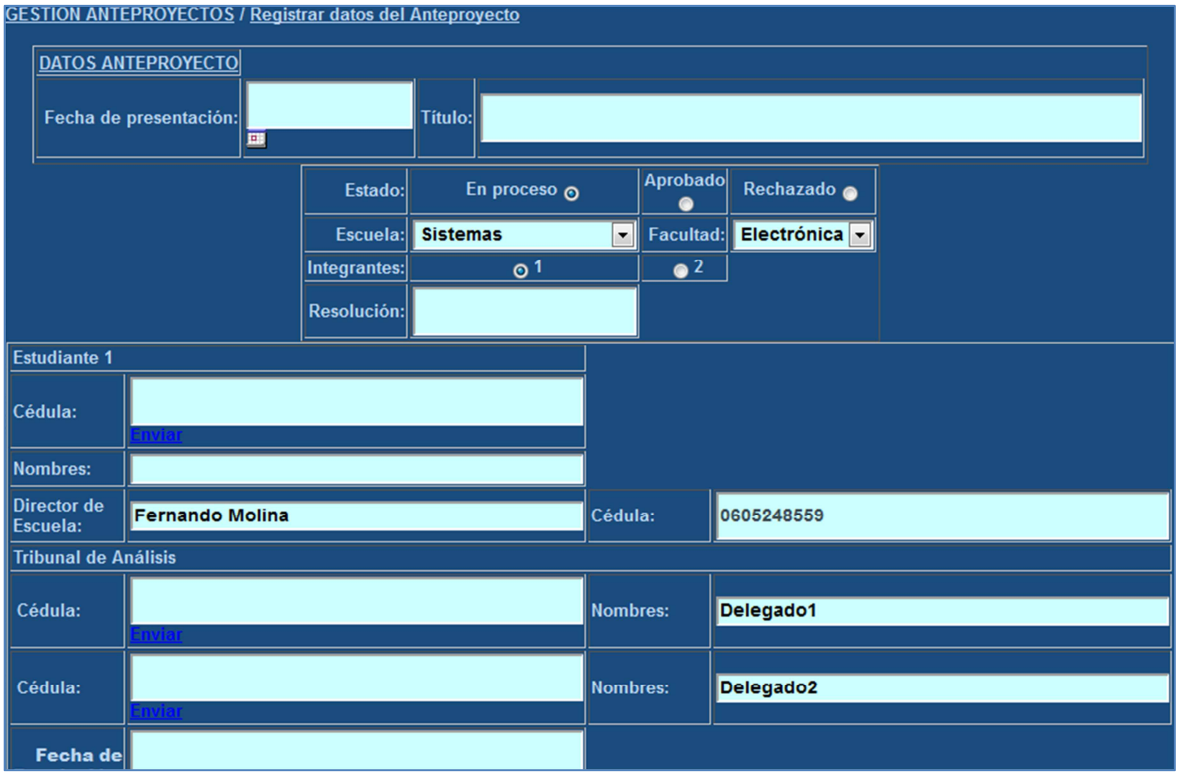

Figura 65: *Registro de Anteproyecto*
Fuente: *Sistema Web GestionUnach*

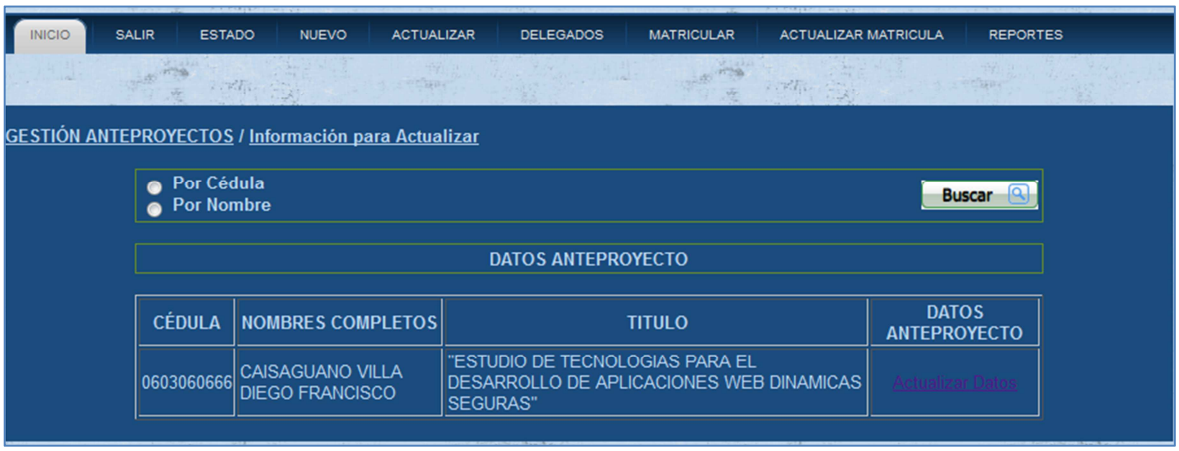

Figura 66: *Consultar estado de Anteproyecto*
Fuente: *Sistema Web GestionUnach*

Figura 67: *Seleccionar Proyecto de Tesis*
Fuente: *Sistema Web GestionUnach*

Figura 68: *Reportes Anteproyectos y tesis*
Fuente: *Sistema Web GestionUnach*

Figura 69: *Consultar Reportes Anteproyectos*
Fuente: *Sistema Web GestionUnach*

Figura 70: *Reportes Anteproyectos*
Fuente: *Sistema Web GestionUnach*

Figura 71: *Reportes Presentación Tesis*
Fuente: *Sistema Web GestionUnach*

- **Modulo Gestión Actas**

Figura 72: *Consultar o ingresar Actas*
Fuente: *Sistema Web GestionUnach*

Figura 73: *Consulta de acta e imprimir reporte*
Fuente: *Sistema Web GestionUnach*

Figura 74: *Ingreso de nueva acta con generación automática*
Fuente: *Sistema Web GestionUnach*

Figura 75: *Ingreso de datos del Acta*
Fuente: *Sistema Web GestionUnach*

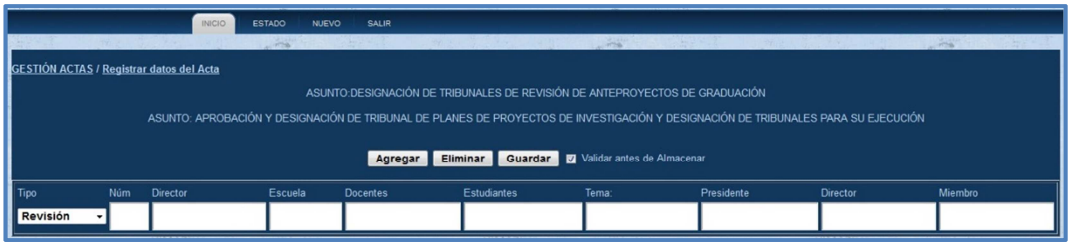

Figura 76: *Ingreso de revisiones , asignaciones y aprobaciones*
Fuente: *Sistema Web GestionUnach*

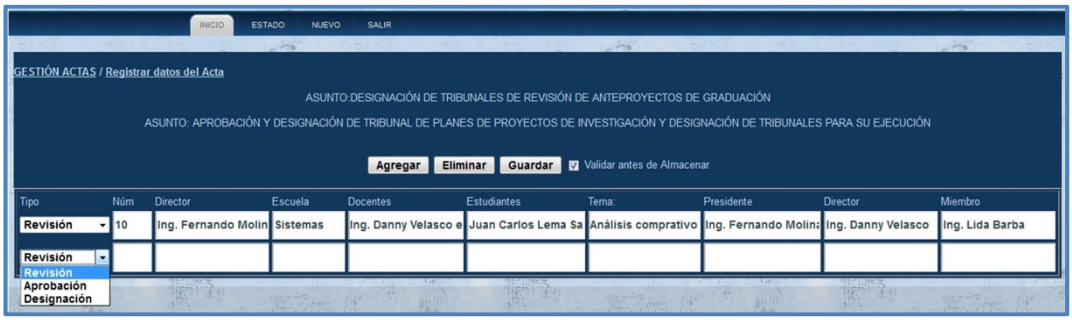

Figura 77: *Ingreso otro tipo de información*
Fuente: *Sistema Web GestionUnach*

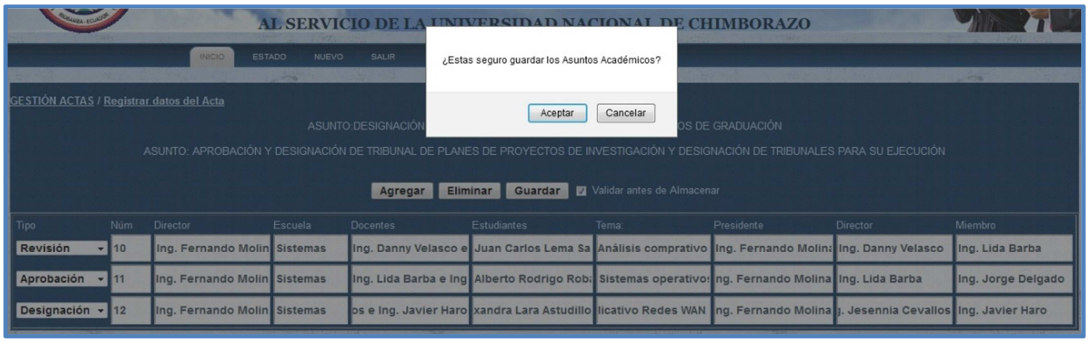

Figura 78: *Ingreso de asuntos relacionados a la revisión, aprobación y designación*
Fuente: *Sistema Web GestionUnach*

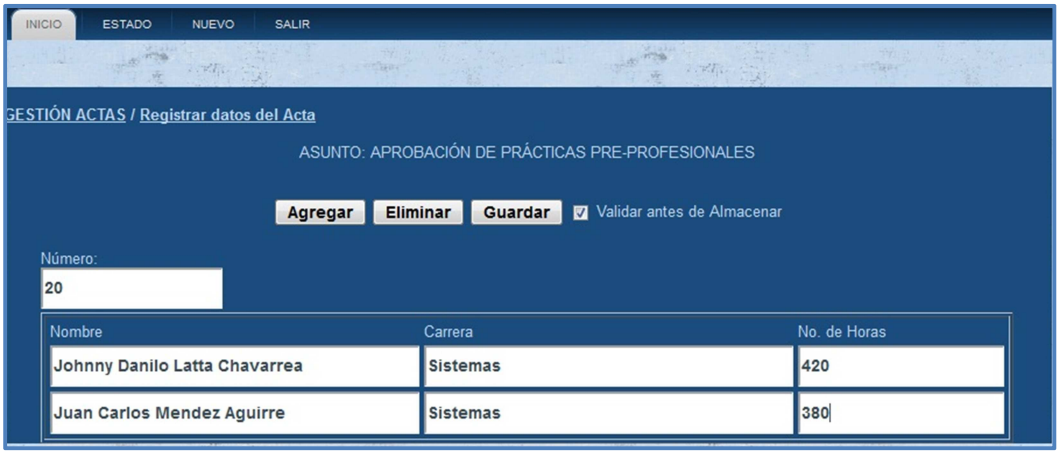

Figura 79: *Ingreso de asuntos de la aprobación de prácticas preprofesionales*
Fuente: *Sistema Web GestionUnach*

Figura 80: *Ingreso asuntos varios de la misma acta*
Fuente: *Sistema Web GestionUnach*

Figura 81: *Finalización del proceso de actas y se puede generar el reporte*
Fuente: *Sistema Web GestionUnach*

4.22.3. DIAGRAMAS DE CLASES DE DISEÑO

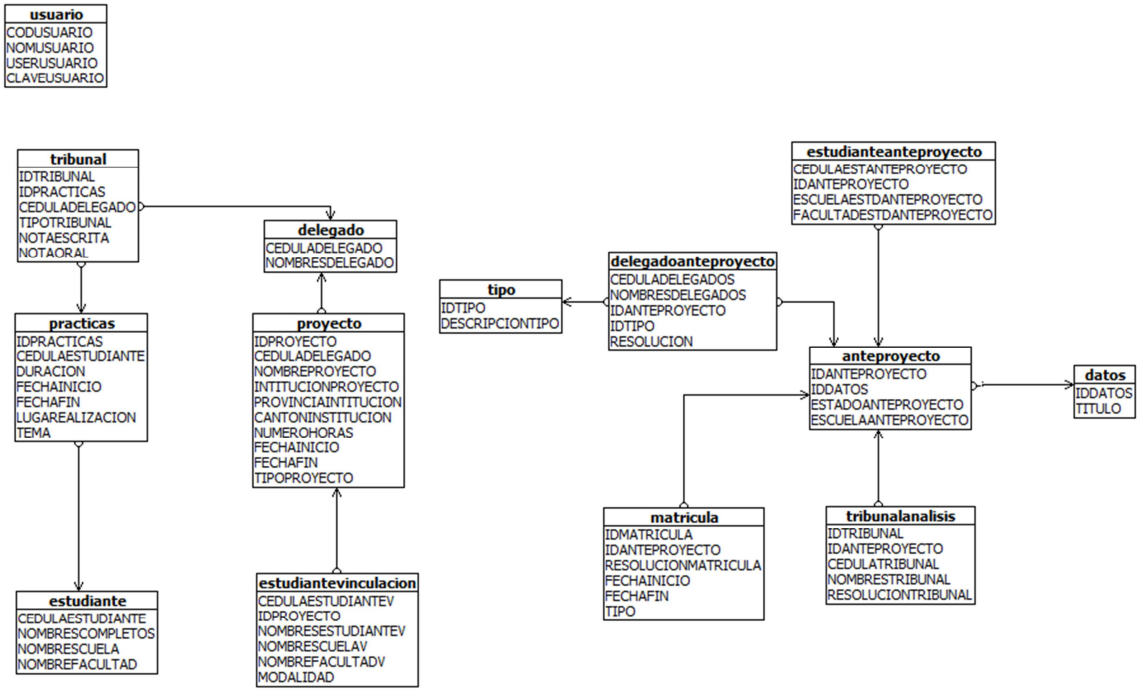

Figura 82: *Diagrama de clases Gestión Unach*
Autor: *Johnny Latta*

4.22.4. ESQUEMA DE LA BASE DE DATOS

El sistema Gestión Unach utiliza una base de datos relacional, elaborado en MySql que enfatiza las relaciones que existe entre los datos almacenados en las tablas para convertirse en información luego de un modelamiento completo,el esquema se detalla en base a los dos módulos:

4.22.4.1. Base de datos en MySQL: gestionunach

- **Tablas del módulo gestión prácticas, gestión anteproyectos y tesis:**

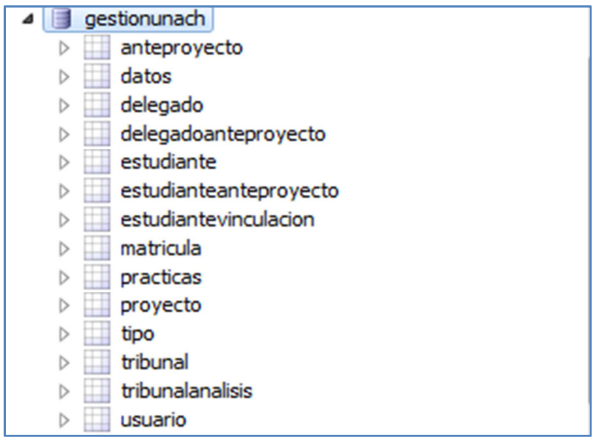

Figura 83: *Tablas del Módulo Gestión Unach*
Autor: *Johnny Latta*

- **Tablas de Gestión Unach en SQLServer: GESTIONDANILO**

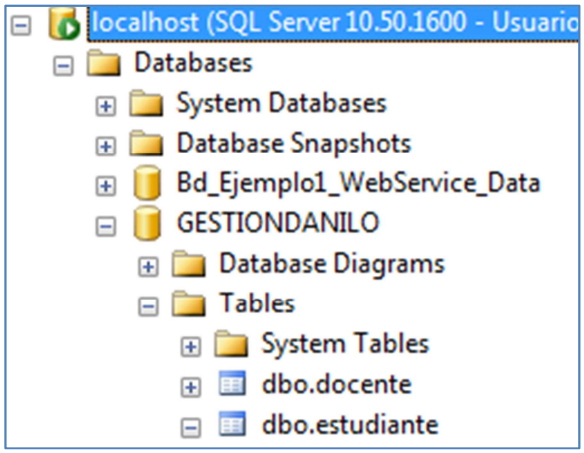

Figura 84: *Tablas del Módulo Gestión*
Autor: *Johnny Latta*

4.23. DIAGRAMA DE COMPONENTES

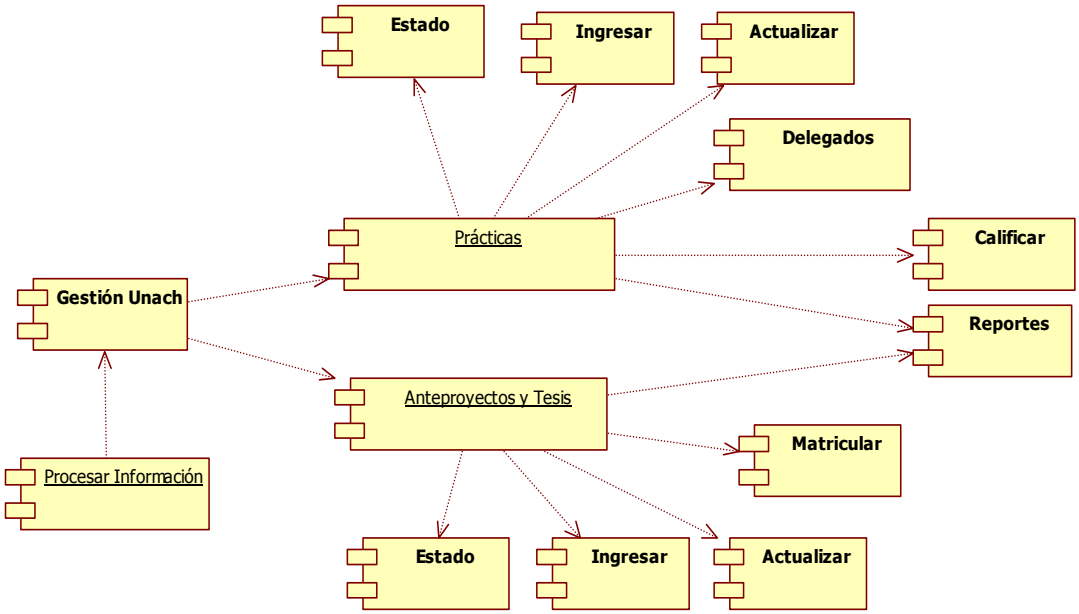

Figura 85: *Diagrama de componentes Gestión Unach*
Autor: *Johnny Latta*

4.24. DIAGRAMA DE DESPLIEGUE

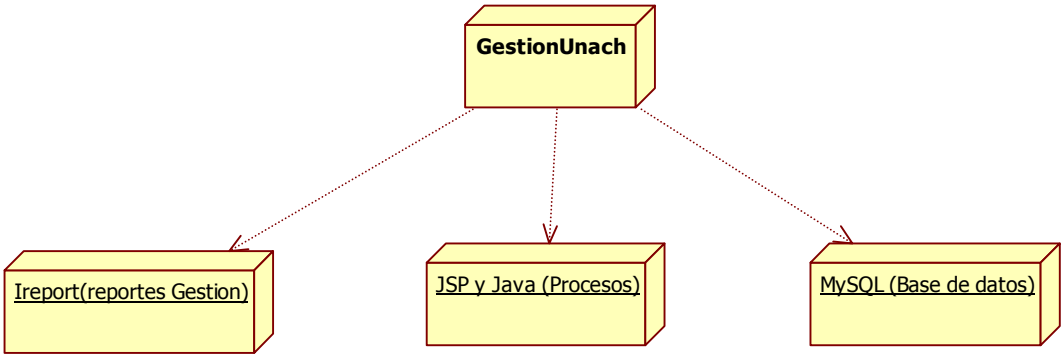

Figura 86: *Diagrama de despliegue Gestión Unach*
Autor: *Johnny Latta*

4.25. IMPLEMENTACIÓN

Una vez modelado el sistema, se procede a la programación e implementación del mismo. Esta fase se centra en crear una aplicación que permita el registro y la generación de reportes de un proceso de prácticas, anteproyectos y tesis en la Unach para lo cual se emplea MySQL Front 2.5 para la gestión de datos e información por curso a evaluar y para la programación de la aplicación se utiliza NetBeans 7.2 que es una herramienta de desarrollo Web y el código fuente es generado en lenguaje JSP y Java.

4.26. IMPLEMENTACIÓN DEL SISTEMA GESTIÓNUNACH

En el desarrollo del GestiónUnach (sistema de gestión), la base de datos gestionunach diseñada es muy importante en registrar datos de estudiantes, prácticas, anteproyectos y tesis, ya que involucra todos los criterios tomados en cuenta para el proceso, además almacena datos de los usuarios del sistema. Cada módulo se encarga de la gestión independiente de los procesos que debe seguir el futuro profesional y se tiene la opción de generar reportes por cada estudiante en cada proceso.

CONCLUSIONES

- El análisis de los métodos de integración de sistemas informáticos, es una técnica de trabajo profesional que permite obtener una visión práctica para facilitar la realización de proyectos con orientación de integración de herramientas Java y .Net.

- Una metodología para la integración de sistemas, implica establecer las estrategias para la integración de código plano basado en JSP y J2EE, el sistema transaccional de Java y el gestor de contenidos .Net, con estrategias fundamentadas en el desarrollo basado en los modelos, y concretamente en MDA, se logra desarrollar un esquema único para los entornos.

- El desarrollo del sistema de gestión académico universitario, generaliza la integración de tecnologías, mediante acciones que incluyen el consumo de los servicios web de la Unach para obtener información de estudiantes y docentes, desde un sistema web independiente, proyectando independencia, estabilidad y eficiencia en los recursos tecnológicos.

- La comprobación de hipótesis mediante el método chi-cuadrado se deduce que la propuesta metodológica es eficiente y cumple con los requisitos necesarios en la integración de las tecnologías J2EE y .NET, lo cual permite obtener un sistema de calidad.

RECOMENDACIONES

- Se recomienda realizar un análisis organizado y profundo de los métodos de integración de sistemas informáticos, para obtener una visión práctica y facilitar la realización de proyectos con orientación para la integración de herramientas tecnológicas del mercado en especial de Java y .Net.

- La metodología de integración de sistemas establece las estrategias para la integración de los componentes de trabajo en los sistemas transaccional y en los gestores de contenidos a utilizar, por tanto redefinir el trabajo de modelar sistemas para mejorar la productividad será una de las estrategias fundamentadas a seguir en el desarrollo para lograr esquemas únicos de integración.

- El desarrollo de un sistema software para la gestión de los procesos de prácticas, anteproyectos, tesis y actas, tiene un enfoque de ser una solución web para la institución, por lo cual se recomienda un mantenimiento del servidor que contiene la aplicación para que no exista futuros problemas en la ejecución del sistema.

- Se recomienda a la persona encargada de llevar a cabo los procesos y reportes que se generan en el sistema, luego de haber recibido indicaciones previas para su uso, leer el manual de usuario para que no exista complicaciones o mal uso del mismo.

BIBLIOGRAFÍA

- [1] Desarrollo de aplicaciones web dinámicas con XML y JAVA, Anaya, Parson David. (2009).

- [2] Estrategias de integración de sistemas. [En línea] http://www.netzima.com/pdf/integracion_de_sistemas.pdf2011/10/20

- [3] Estudio de interoperabilidad .NET/J2EE. [En línea] http://www.consultec.es/DocInformes/J2EE%20y%20NET.pdf

- [4] Ingeniería de Software. (6 Ed).McGraw-Hill, Roger S. Pressman. (2005).

- [5] Integración de sistemas. [En línea] http://www.vertis.cl/soluciones/integracion-de-sistemas.html 2011/10/23

- [6] Libro de Metodología De La Investigación. **Autor:** Hernández Sampieri Roberto. 2011/10/23

- [7] Libro de Implantación De Sistemas. **Autor:** Fuoc. 2011/10/23

- [8] Libro de Programación Java Server J2ee. **Autor:** Wrox. 2011/10/22

- [9] NetBeans. [En línea]
  http://ifile.it/pdo94ra/netbeans-6.9.1-201011281701-ml-visualweb.zip
  http://antiguo.itson.mx/die/mdomitsu/bibliotecaDigital/Tutoriales_NetBeans/
  http://netbeans.org/kb/trails/platform.html
  http://www.planetnetbeans.org/es/

- [10] Objects, components and frameworks with UML. The catalysis approach. Addison-Wesley, Reading Mass. **Autor** D´Souza, D. F. y Cameron W. A (1999),

- [11] Pyme. Integración de sistemas. [En línea] http://www.conectapyme.com/lectura.asp?id_nodo=1731&id_puntoDpto=3932 2011/10/22

- [12] Sistemas de información III. [En línea] http://prof.usb.ve/lmendoza/Documentos/PS-6117%20%28Teor%EDa%29/Teor%EDa%20PS6117%20Reto%20de%20los%20SI.pdf 2011/10/24

- [13] Tectura.Integración de sistemas. [En línea] http://www.es.tectura.com/Page/cm172/Integracin_de_sistemas_172.asp?d=1 2011/10/20

ANEXOS

ANEXO A
CODIFICACIÓN ACCESO A DATOS

Comando.java

```
package AccesoDatos;
//import java.util.*;
import java.util.ArrayList;
/**
 *
 * @author root
 */
public class Comando {
private String setenciaSql;
private ArrayList<Parametro> lstParametros;
public Comando()
    {
setenciaSql="";
lstParametros= new ArrayList<Parametro>();
    }

    /**
     * @return the setenciaSql
     */
public String getSetenciaSql() {
return setenciaSql;
    }

    /**
* @param setenciaSql the setenciaSql to set
     */
public void setSetenciaSql(String setenciaSql) {
this.setenciaSql = setenciaSql;
    }

    /**
     * @return the lstParametros
     */
public ArrayList<Parametro> getLstParametros() {
return lstParametros;
    }

    /**
* @param lstParametros the lstParametros to set
*/
public void setLstParametros(ArrayList<Parametro> lstParametros) {
```

```
        this.lstParametros = lstParametros;
    }
}
```

Conexión.java

```
package AccesoDatos;
import java.sql.*;
import java.util.*;

/**
*
 * @author root
 */
public class Conexion {
public  String driver;
        public  String url;
        public  String user;
        public String pass;

        public Conexion(){
        }

        public String getDriver(){
                return this.driver;
        }

        public String getUrl(){
                return this.url;
        }
        public String getUser(){
                return this.user;
        }

        public void setUser(String user){
                this.user=user;
        }

        public void setPassword(String pass){
                this.pass=pass;
        }

        public   Conexion(String pdriver, String pURL, String pusuario,String
pclave ){
                this.driver=pdriver;
      this.url=pURL;
      this.user=pusuario;
      this.pass= pclave;
        }
```

```
//    public ResultSet ejecutaQuery(String sql) throws Exception
//    {
//        ResultSet rs = null;
//         try {
//         Class.forName(this.driver);
//          //  System.out.println("OK al cargar");
//            try {
//            Connection con =
DriverManager.getConnection(this.url,this.user,this.pass);
//             // System.out.println("Conectado a Oracle");
//              Statement st = con.createStatement();
//              rs = st.executeQuery(sql);
//
//        } catch (SQLException exConec) {
//              throw  new Exception(exConec.getMessage());
//            }
//        }
//        catch (ClassNotFoundException exCarga) {
//            throw  new Exception(exCarga.getMessage());
//        }
//         return rs;
//    }


public ResultSet ejecutaPrepared(PreparedStatement prStm) throws Exception
        {
       ResultSet rts=null;
try {
rts =prStm.executeQuery();

        } catch (SQLException exConec) {
throw  new Exception(exConec.getMessage());
        }
return rts;
        }


public PreparedStatement creaPreparedSmt(String sql) throws Exception
        {
         PreparedStatement prStm=null;
try {
Class.forName(this.driver);
          //  System.out.println("OK al cargar");
try {
           Connection con =
DriverManager.getConnection(this.url,this.user,this.pass);
            // System.out.println("Conectado a Oracle");
prStm = con.prepareStatement(sql);
```

```
        } catch (SQLException exConec) {
throw  new Exception(exConec.getMessage());
          }
       }
catch (ClassNotFoundException exCarga) {
throw  new Exception(exCarga.getMessage());
      }
return prStm;
      }

public boolean  ejecutaPreparedTransaccion(ArrayList<Comando> lstComandos)
throws Exception
      {
boolean respuesta=false;
Class.forName(this.driver);
         Connection con =
DriverManager.getConnection(this.url,this.user,this.pass);
try {

if (con.getAutoCommit()) {
con.setAutoCommit(false);
}
for (Comando comando : lstComandos) {
                PreparedStatement ptrs =
con.prepareStatement(comando.getSetenciaSql());
for (Parametro parametro : comando.getLstParametros()) {
ptrs.setObject(parametro.getPosicion(), parametro.getValor());
}
                ResultSet rst = ptrs.executeQuery();
if (rst.next()) {
                   String bandera = rst.getString(1);
respuesta = bandera.equals("t")?true:false;
                }
             }
con.commit();
         }
catch (SQLException exConec) {
con.rollback();
throw  new Exception(exConec.getMessage());
      }
return respuesta;
      }

public      int             ejecutaPreparedTransaccionFactura(ArrayList<Comando>
lstComandos) throws Exception
      {
int respuesta=-1;
boolean respuestaItems=false;
```

```
int respuestaEncabezado=-1;
Class.forName(this.driver);
        Connection con =
DriverManager.getConnection(this.url,this.user,this.pass);
try {

if (con.getAutoCommit()) {
con.setAutoCommit(false);
}
            Comando encabezado= lstComandos.get(0);
            PreparedStatement ptrsencabezado =
con.prepareStatement(encabezado.getSetenciaSql());
for (Parametro parametro : encabezado.getLstParametros()) {
ptrsencabezado.setObject(parametro.getPosicion(), parametro.getValor());
            }
             ResultSet rstencabezado = ptrsencabezado.executeQuery();
if (rstencabezado.next()) {
respuestaEncabezado = rstencabezado.getInt(1);
            }


             //
int p=0;
p=lstComandos.size();

for (int j = 1; j < p; j++){
            Comando items= lstComandos.get(j);
            PreparedStatement ptrsItems =
con.prepareStatement(items.getSetenciaSql());
for (Parametro parametro : items.getLstParametros()) {
if(parametro.getPosicion()==2)
                {
parametro.setValor((Object)respuestaEncabezado);
                }
else
                {}
ptrsItems.setObject(parametro.getPosicion(), parametro.getValor());
            }
ResultSet rstitems = ptrsItems.executeQuery();
if (rstitems.next()) {

                    String bandera = rstitems.getString(1);
respuestaItems = bandera.equals("t")?true:false;
//respuestaItems = rstitems.getInt(1);
            }

            }
if((respuestaEncabezado>0)&&(respuestaItems==true))
```

```
                //&& (respuestaPagoEfectivo==true)&&
(respuestaPagoTarjeta==true)
            {
respuesta= respuestaEncabezado;
            }
rstencabezado.close();
con.commit();
con.close();
         }
catch (SQLException exConec) {
con.rollback();
throw  new Exception(exConec.getMessage());
}
return respuesta;
      }
}
```

Global.java

```
package AccesoDatos;

/**
*
 * @author root
*/
public class Global {
public static final  String url="jdbc:mysql://usuario5-pc:3306/gestionunach";
public static final  String user="root";
public static final  String pass="123456";
public static final  String driver="com.mysql.jdbc.Driver";
}
```

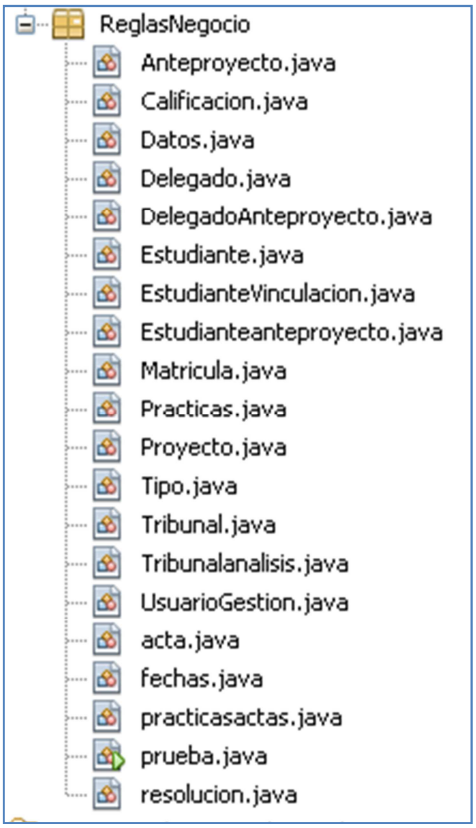

Figura 87: *Codificación lógica del negocio*
Autor: *Johnny Latta*

ANEXO B

MODELO DE ENCUESTA APLICADA A DESARROLLADORES

1. **¿La metodología es entendible y cumple con el objetivo determinado?**

   SI ___ NO ___

2. **¿La aplicación de una metodología de integración de tecnologías Java y .Net se puede ajustar o aplicar a otros sistemas?**

   SI ___ NO ___

3. **¿La metodología planteada podría llegar a ser poco aceptada reutilizable y poco escalable?**

   SI ___ NO ___

4. **¿Conoce proyectos reales en los que se apliquen integración de tecnologías JAVA y .NET?**

   SI ___ NO ___

5. **¿Considera que la heterogeneidad de tecnologías es una ventaja para desarrollar estrategias de integración de alto nivel?**

   SI ___ NO ___

6. **¿La integración de tecnologías implica aplicar una arquitectura de desarrollo de software común, conoce usted alguno de ellos? ¿Cúal?**

   SI ___ NO ___

7. **JAVA y .NET son tecnologías que permiten el desarrollo de aplicaciones robustas, por lo tanto ¿Se asegura que son tecnologías para el desarrollo de proyectos de integración?**

SI ___ **NO ___**

Gracias por su colaboración