

UNIVERSIDAD DE LAS AMÉRICAS

FACULTAD DE INGENIERÍA

DISEÑO DE TECNOLOGÍAS INALÁMBRICAS WAP Y SMS INTEGRADAS E
IMPLEMENTACIÓN PARA LA UNIVERSIDAD DE LAS AMÉRICAS.

MAURICIO FERNANDO PONTÓN CAJILEMA
CARLOS GUILLERMO VARGAS ÁLVAREZ

2004

UNIVERSIDAD DE LAS AMÉRICAS

FACULTAD DE INGENIERÍA

ESCUELA DE SISTEMAS EN INFORMÁTICA Y COMPUTACIÓN

DISEÑO DE TECNOLOGÍAS INALÁMBRICAS WAP Y SMS INTEGRADAS E IMPLEMENTACIÓN PARA LA UNIVERSIDAD DE LAS AMÉRICAS.

Trabajo de titulación presentado en conformidad a los requisitos

Para obtener el título de Ingeniero en Sistemas.

Ing. César Intriago

**MAURICIO FERNANDO PONTÓN CAJILEMA
CARLOS GUILLERMO VARGAS ÁLVAREZ**

2004

AGRADECIMIENTOS

A Dios. Sus enseñanzas, su tenacidad y su fuerza como Cristo hombre, son fuente de inspiración.

A nuestros padres. Los principales gestores de nuestra carrera. Su ejemplo, preocupación, amor y apoyo incondicional, han sido pilar en nuestras vidas. A quienes debemos todo, desde la existencia, hasta lo que hemos llegado a ser.

A nuestras familias, cónyuges, hermanos, quienes nos acompañaron durante todas nuestras circunstancias, tanto benignas como desagradables.

A la Universidad. Sus profesores, sus aulas, y principalmente sus enseñanzas y su ideología, han formado de nosotros unos excelentes profesionales, listos para enfrentar los retos en un mundo tan competitivo como el actual.

Y a nuestros amigos, allegados, a todos quienes han estado junto a nosotros en cada capítulo de nuestras vidas.

DEDICATORIA

Mauricio Pontón: EL presente trabajo está dedicado a mi familia, mis amigos... a mi ser especial y a todo aquel que desee seguir sus estrellas... Gracias por todo su apoyo.

Guillermo Vargas: Cuando llega un hijo a la vida de una persona, este se convierte en la razón misma de vivir y de luchar. Este trabajo lo dedico a mis padres y a mi esposa que siempre me acompañaron y apoyaron, y en especial a mi hijo Guillermo Andrés

RESUMEN

El objetivo del presente trabajo de titulación es el de presentar una arquitectura para integración de tecnologías, un modelo sobre el cual puedan basarse soluciones que requiera el uso de protocolos inalámbricos y desarrollar la plataforma tecnológica que de soporte a esta arquitectura. Adicionalmente, el presente trabajo de titulación abarca el desarrollo aplicación que demuestre como la manera en la que puede desarrollarse un sistema de consulta dentro de la Universidad de las Américas usando esta plataforma tecnológica.

Basados en esta arquitectura propuesta en este trabajo, la aplicación demo, utiliza el mismo entorno que la Universidad posee, su misma estructura informática, y sus recursos existentes. Esta aplicación servirá como base para ejemplificar los requerimientos y pasos necesarios para el desarrollo de soluciones basadas de tecnologías inalámbricas.

La aplicación demostrativa desarrollada, usando programación orientada a objetos, se basa sobre un marco de componentes bien definidos, lo que permite la reutilización de los mismos en diversas aplicaciones que estén dentro del ámbito de la solución. Gracias a esta plataforma, se brinda a los usuarios y desarrolladores más posibilidades de acceso a la información, los cuales, además de poder acceder a los datos por un medio cada vez más común, como es Internet (una terminal con acceso a la Web), el usuario o desarrollador podrá hacerlo sin necesidad de una computadora, ni de un ambiente clásico de Internet. Para este efecto, se utilizará un dispositivo de comunicación muy utilizado por todos: el teléfono celular.

Las tecnologías como WAP Y SMS permiten la conectividad directa con dispositivos de este tipo, convirtiéndolo en una excelente herramienta informativa; sacando el máximo provecho a las herramientas tecnológicas que se utiliza diariamente. se puede lograr un sin número de operaciones, tanto de consultas, como transaccionales.

Mediante el protocolo SMTP para el intercambio de correo electrónico, se realiza el envío de información personalizada para cada estudiante previamente registrado en una base de datos.

De forma complementaria, esta plataforma tecnología incluye la funcionalidad de acceso desde Internet por medio de Servicios Web. Así, la arquitectura integra varias tecnologías, y la aplicación demostrativa implementada en el presente Trabajo de Titulación muestra una herramienta poderosa de consulta.

INDICE

INDICE DE FIGURAS	4
1 CAPÍTULO I: INTRODUCCIÓN	5
1.1 OBJETIVOS	6
1.2 OBJETIVOS ESPECÍFICOS:	6
2 CAPÍTULO II: MARCO TEÓRICO	7
2.1 Protocolo WAP (Protocolo de Aplicación Inalámbrica)	7
2.2 ARQUITECTURA WAP	8
2.2.1 Capa de Aplicación (WAE – Wireless Application Environment)	8
2.2.2 Capa de Sesión (WSP – Wireless Session Protocol)	9
2.2.3 Capa de Transacciones (WTP – Wireless Transaction Protocol)	9
2.2.4 Capa de Seguridad (WTLS – Wireless Transport Layer Security)	9
2.2.5 Capa de Transporte (WDP – Wireless Datagram Protocol)	10
2.3 SERVICIOS WEB	10
2.4 SMS	11
2.4.1 Clasificación	12
2.4.2 MT (Mobile Terminated)	12
2.4.3 MO (Mobile Originated)	12
2.4.4 Enfoque del servicio	13
2.5 ARQUITECTURA MULTICAPA	13
2.5.1 Beneficios	13
2.5.2 Capas y componentes	13
2.5.3 Interfaz lógica	14
2.5.4 Lógica de dominio	14
2.5.5 Persistencia	14
2.5.6 Relaciones entre capas	15
2.6 Plataforma Microsoft .NET Framework	15
2.6.1 Visual C # .NET	16
2.6.1.1 Características de C#	16
2.6.2 Visual Basic.Net	17
3 CAPÍTULO III: DESCRIPCIÓN DEL MODELO PROPUESTO PARA EL DESARROLLO DE APLICACIONES QUE INTEGREN TECNOLOGÍAS INALÁMBRICAS	20
3.1 Resumen de las capas propuestas en el modelo y su relación con el prototipo de ejemplo	23
3.1.1 Esquema gráfico del prototipo de ejemplo	25
3.2 Definición individual de las capas del modelo propuesto y relación entre ellas	26
3.2.1. Capa de persistencia	26

3.2.1.1	Requerimientos de la capa de Persistencia	26
3.2.1.2	Seguridades para la capa de Persistencia	27
3.2.2	Capa de Componentes Lógicos	28
3.2.2.1	Requerimientos de la Capa de Componentes Lógicos	30
3.2.2.2	Seguridades para la Capa de Componentes Lógicos	30
3.2.3	Capa de Servicios Web	31
3.2.3.1	Requerimientos de la Capa de Servicios Web	31
3.2.3.2	Aspectos de seguridad para la Capa de Servicios Web	31
3.2.4	Capa de Presentación	32
3.2.4.1	Requerimientos de la Capa de Presentación	33
3.2.4.2	Seguridades para la Capa de Presentación	34
3.2.5	Descripción Individual del Modelo por Protocolos	34
3.2.5.1	SMS	34
3.2.5.2	WAP	35
4	CAPÍTULO IV: CONCLUSIONES Y RECOMENDACIONES	37
4.1	Conclusiones	37
4.2	Recomendaciones	39
	BIBLIOGRAFIA	40
Documentos Impresos		40
Documentos Electrónicos		40
	ANEXOS	42
ANEXO I		43
METODOLOGIA		44
Metodología OMT		45
Pasos para el proceso de desarrollo OMT		45
ANEXO II		47
DIAGRAMA DE COMPONENTES		48
ANEXO III		54
DIAGRAMA DE CLASES		55
ANEXO IV		57
DIAGRAMA DE DESPLIEGUE		58
ANEXO V		60
DIAGRAMA DE SECUENCIA		61
ANEXO VI		69
DIAGRAMA DE ACTIVIDAD		70
ANEXO VII		74
DIAGRAMA DE CASOS DE USO		75
ANEXO VIII		81
PRUEBAS FUNCIONALES		82
		98

ANEXO IX	107
MANUALES	108
MANUAL DE USUARIO	109
MANUAL TECNICO	156
MANUAL DE INSTALACION	214
ANEXO X	235
ANALISIS TARIFARIO	236
ANEXO XI	239
CD	240
CONTENIDO DEL CD	241

INDICE DE FIGURAS

Figura 2-1.- Esquema de funcionamiento de WAP	6
Figura 2-2.- Arquitectura del modelo WAP	8
Figura 2-3.- Arquitectura WDP con Servicio Portador CDMA	9
Figura 2-4.- Arquitectura Multicapa	12
Figura 3.1.- Esquema del modelo propuesto	18
Figura 3.2.- Interacción de la aplicación con empresa integradora	20
Figura 3.3.- Aplicación WAP	21
Figura 3.4.- Esquema gráfico del prototipo de ejemplo	23
Figura 3.5.- Ejemplo de subdivisión de capas en la capa de componentes lógicos	26

1 CAPÍTULO I: Introducción

Las tendencias globales y los adelantos tecnológicos obligan a las empresas a establecer mecanismos eficientes de gestión de procesos. El manejo adecuado de los recursos permite establecer una ventaja competitiva y por consiguiente mantenerse sólidas y vigentes a lo largo del proceso evolutivo empresarial.

El siguiente trabajo de investigación comprende en proponer un modelo que sirva de guía en el desarrollo de aplicaciones o soluciones que integren varias tecnologías. Principalmente se busca involucrar tecnologías inalámbricas, de esta forma no se requiere tener a mano una computadora con conexión física a Internet para acceder a los sistemas, puesto que otros dispositivos pueden utilizar esta tecnología, tales como teléfonos celulares, que al ser de uso masivo, ofrecen un mayor alcance hacia sus usuarios.

El presente trabajo usa, estudia e integra los protocolos WAP (Protocolo de Aplicación o Acceso Inalámbrica), el protocolo SMS (Servicio de Mensajes Cortos), y el protocolo SMTP (Protocolo de transferencia de correo electrónico simple). A su vez, se ha implementado un prototipo, que abarca todo este trabajo investigativo, destinado a la Universidad de las Américas, una aplicación de consulta, cuyo enfoque va dirigido al cuerpo estudiantil, quienes podrán hacer uso de todas las facilidades que esta solución propone, a fin de contar con su respectiva información académica, de manera eficaz y oportuna.

Este planteamiento establece un nuevo servicio que la universidad ofrece a sus alumnos, una solución completamente innovadora, desarrollada para mejorar la comunicación entre el estudiante y el establecimiento educativo.

1.1 OBJETIVOS

Presentar un modelo que sirva como referencia para la implementación de soluciones que integren protocolos WAP y SMS, y que a su vez estos se relacionen en el desarrollo tradicional de aplicaciones. Este modelo será demostrado mediante un prototipo implementado a través de una arquitectura multicapa, orientada a objetos compuesta de varias librerías de clases agrupadas en componentes y servicios Web, que de soporte a sistemas orientados a Web, inalámbricos y de transmisión de correo electrónico.

1.2 OBJETIVOS ESPECÍFICOS:

- Definir un modelo para el desarrollo de aplicaciones WAP, destinadas a su uso desde mini exploradores propios de los teléfonos celulares.
- Integrar en aplicaciones tradicionales la tecnología SMS, definir su funcionamiento, sus elementos y sus requerimientos.
- Mostrar las características de los teléfonos celulares de tercera generación y plantearlos como herramientas de consulta y transaccionales, y explotar su capacidad de interacción con aplicaciones diversas.
- Exponer la potencia de los servicios Web que permiten la interoperabilidad entre diversas plataformas, basados en el estándar XML (Extensible Markup Language).
- Mostar la potencia de la plataforma Microsoft Visual Studio .NET, en el desarrollo de aplicaciones distribuidas, y en la integración de diversas tecnologías y lenguajes.

2 CAPÍTULO II: Marco Teórico

El presente trabajo de titulación contiene una explicación de los diferentes conceptos y tecnologías que van a ser utilizadas para el desarrollo del presente proyecto. Se mostrará una explicación de cada una, conceptos técnicos y finalmente, su integración.

Los puntos a tratar son:

- Protocolo WAP (Wireless Application Protocol)
- Protocolo SMS (Short Message Service)
- Servicios Web
- Protocolo SMTP (Simple Mail Transfer Protocol)
- Arquitecturas Multicapa
- Microsoft.NET

2.1 Protocolo WAP (Protocolo de Aplicación Inalámbrica)

La tecnología WAP ha sido desarrollada para transmisión de datos a dispositivos que no requieren acceso físico a Internet, tales como teléfonos celulares, computadores de mano, etc.

El siguiente esquema (Figura 1-1) muestra el modelo de funcionamiento de WAP:

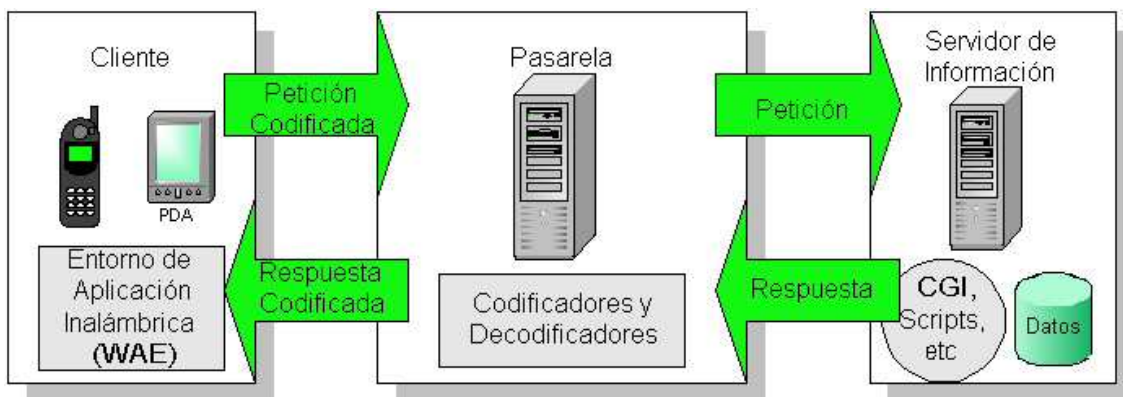


Figura 2-1: Esquema de funcionamiento de WAP

Este esquema muestra un cliente, específicamente un teléfono celular o un dispositivo PDA, de uso tan común en la actualidad. A medida que la tecnología ha avanzado, estos pequeños dispositivos han incorporado un micro navegador, similares en función a los navegadores convencionales de Internet. Otro componente del esquema es el Servidor de Información, que almacena los datos, las páginas de contenido, scripts, etc. Adicionalmente, un servidor

denominado "Pasarela" codifica y decodifica las solicitudes hechas por el micro navegador y las respuestas enviadas por el Servidor de Información.

El teléfono WAP o el dispositivo PDA utilizan las capacidades de información de conexiones inalámbricas convencionales para que el usuario realice peticiones a la pasarela WAP. La pasarela WAP convierte éstas en peticiones HTTP (Hyper Text Transfer Protocol) y las envía a través de Internet. Cuando el servicio requerido responde, la pasarela WAP vuelve a enviar la información al teléfono WAP o al dispositivo PDA.

La pasarela WAP es el núcleo de la plataforma WAP. Su capacidad para actuar en esta clase de dispositivos como un proxy HTTP, permite a los suscriptores acceder a cualquier sitio WWW. Algunos proveedores de información ofrecen igualmente servicios WML (Wireless Markup Language) que usan WML para aprovechar la interfaz del dispositivo WAP. Estos servicios pueden además iniciar la comunicación "impulsando" la información a la pasarela WAP, que como respuesta, transmite la misma a un dispositivo WAP. Este proceso se denomina notificación.

Además de la translación HTML (Hyper Text Markup Language), la oferta de servicios de la pasarela varía. Estos pueden ser un servicio de protección de información por medio del mantenimiento de una base de datos de teléfonos WAP y sus privilegios de acceso, un servicio de fax que permitiese a los usuarios de teléfonos WAP mandar por fax contenido de un sitio Web a una máquina de fax local, o servicios de correo, organizadores o directorios. Todos ellos dependen de la suite de servicios que ofrezca cada pasarela.

2.2 ARQUITECTURA WAP

La arquitectura WAP consiste en capas, las cuales se encuentran accesibles una entre otra, así como a otros servicios y aplicaciones a través de interfaces especificadas. El esquema de la Figura 1-2 muestra la definición de la arquitectura del modelo WAP:

2.2.1 Capa de Aplicación (WAE – Wireless Application Environment)

El entorno inalámbrico de aplicación consiste en un ambiente de aplicación general, formado por la combinación de protocolos Web y tecnologías de Comunicaciones Móviles.

Está formado por un navegador integrado en el dispositivo móvil, el cual soporta lenguajes WML (un tipo de lenguaje similar al HTML, pero optimizado para este tipo de navegadores), una agrupación de formatos de contenido bien definidos y soporte de WMLScript (similar al JavaScript).



Figura 2-2: Arquitectura del modelo WAP

2.2.2 Capa de Sesión (WSP – Wireless Session Protocol)

El protocolo inalámbrico de sesión brinda a la capa de aplicación dos servicios, uno orientado a conexión y otro no orientado a conexión, proporcionando servicio de datagramas seguro y no seguro. Esta capa consiste en servicios y aplicaciones orientadas a la navegación Web.

2.2.3 Capa de Transacciones (WTP – Wireless Transaction Protocol)

El protocolo inalámbrico de transacción funciona sobre los servicios de datagramas, brindando transacciones de peticiones una sola vía (tanto seguras como inseguras), como transacciones de dos vías (es decir, una petición y su respuesta). Adicionalmente, ofrece servicios de transacciones asíncronas.

2.2.4 Capa de Seguridad (WTLS – Wireless Transport Layer Security)

La capa inalámbrica de seguridad de transporte se basa en el estándar de seguridad SSL. Brinda una interfaz de transporte seguro a las capas superiores, con servicios de integridad de datos, privacidad y autenticación. De igual forma, puede ser usado para levantar un vínculo seguro entre terminales.

2.2.5 Capa de Transporte (WDP – Wireless Datagram Protocol)

El protocolo inalámbrico de datagramas brinda un servicio confiable a las capas superiores, con una comunicación transparente. Constituye una interfaz común a las capas de Transacciones, Sesión y Aplicación, lo que permite la independencia de uso de las mismas.

Adicionalmente esta capa es la que establece el enlace de comunicación de una manera transparente con uno de los servicios portadores disponibles, esto es que, mediante los servicios que este protocolo ofrece a los protocolos superiores, tales como direccionamiento por número de puerto, segmentación y re-ensamblado y detección de errores, permite a las aplicaciones de usuario funcionar de forma transparente sobre distintos servicios portadores disponibles, para nuestro caso se tiene tres servicios portadores disponibles, que son el CDMA (Código de División de Múltiple Acceso) que ofrece las compañías de servicios celulares BELLSOUTH y ALEGRO PCS, GSM (Global System for Mobile communications) que ofrece la compañía de servicios celulares PORTA, y TDMA (Time Division Multiple Access) que es el servicio portador antiguo que aún mantienen las compañías BELLSOUTH y PORTA..

La Figura 1-3 ilustra la arquitectura de este protocolo en el caso de utilizar CDMA como servicio portador.



Figura 2-3: Arquitectura WDP con Servicio Portador CDMA

2.3 SERVICIOS WEB

Un servicio Web es un método para la transferencia de datos, cuya interfaz está bien definida y al que se puede acceder a través de Internet. Similar a una página Web que está definida por

un URL (Uniform Resource Locator), un servicio Web está definido por un URI (Uniform Resource Identification) y por su interfaz, a través de la cual se puede acceder a él. A través de un servicio Web se puede ofrecer cotizaciones de la bolsa, tasa de cambio, datos de estados del clima, etc. De esta forma, las aplicaciones se convierten en clientes que integran servicios Web procedentes de diferentes proveedores, y además, se abre la posibilidad de que se cobre por el uso del servicio mas no por cada copia de la aplicación vendida. Para la Microsoft esto es lo más atractivo ya que de esa forma se les presenta la posibilidad de acabar de una vez por todas con la piratería, al alojar partes importantes de las aplicaciones en sus propios servidores, y no en la computadora del cliente o usuario final.

Los servicios Web se dividen en servicios de transporte (los protocolos del nivel más bajo, que codifican la información independientemente de su formato, y que pueden ser comunes a otros servicios), de mensajería, de descripción y de descubrimiento. En la parte más baja se encuentran los servicios de transporte, que establecen la conexión y el puerto usado. Generalmente se usa HTTP, el mismo protocolo que la WWW, pero en sí puede usar también SMTP (el mismo protocolo que el correo electrónico), FTP (File Transfer Protocol), o BEEP (blocks extensible exchange protocol), un protocolo específico para servicios Web, que, a diferencia de los anteriores, no es cliente-servidor, sino "entre pares"; las dos computadoras entre las que se establece la comunicación actúan como clientes y servidores a la vez. Es además extensible, y está especificado en XML; por eso se está haciendo mucho más popular para aplicaciones Web. Los Servicios Web XML permiten la comunicación de datos entre distintas plataformas y tecnologías de forma sencilla, clara y transparente, cualquier sistema informático o dispositivo electrónico que soporte el tratamiento de documentos XML podrá consumir los servicios Web XML.

“Para utilizar un servicio Web, un desarrollador no tiene por qué saber de XML o de otros protocolos como SOAP (protocolo de simple acceso a objetos). Sólo debe preocuparse de llamar al servicio remoto pasando los parámetros correctos y esperar el resultado.”¹

2.4 SMS

Short Message Service (SMS) es un servicio inalámbrico disponible en las redes móviles digitales. Permite la transmisión de mensajes de texto entre teléfonos móviles y otros sistemas como mail electrónico, *paging* y mail de voz. Alrededor de 160 caracteres pueden ser enviados y recibidos a través del sistema de mensajes del operador de red hacia el teléfono móvil.

¹ Súper utilidades para Visual Basic.Net, Kris Jamsa, Editorial McGraw Hill, España, 2003, pág. 632

En los últimos años, debido a la popularización de concursos por televisión, se plantea al servicio SMS como una herramienta poderosa de comunicación masiva. Dado que el teléfono celular es ahora un dispositivo indispensable para gran parte de la población, es correcto pensar en el gran alcance de SMS.

2.4.1 Clasificación

Existen cuatro tipos de intercambio de mensajes:

- Mensajes de celular a celular: A través de las empresas de telefonía celular, o por medio de empresas integradoras (empresas encargadas del envío de mensajes cortos utilizando un canal específico que las empresas de telefonía celular brindan para este efecto), dos usuarios con teléfono que soporten SMS pueden enviarse mensajes cortos de texto.
- Envío desde un móvil a una aplicación: por ejemplo, el envío de un mensaje a una aplicación para registrarse a un concurso (una base de datos) o alguna petición específica.
- Envío desde una aplicación a un móvil: una aplicación puede enviar mensajes publicitarios a los teléfonos celulares registrados previamente en una base de datos.
- Envío desde una aplicación a otra: una aplicación, basándose en la lectura de un dispositivo, puede enviar un mensaje de texto a otra, a manera de desencadenador.

Adicionalmente, el servicio de SMS también puede ser clasificado en función de donde tiene origen el mensaje:

2.4.2 MT (Mobile Terminated)

Denominados así a los mensajes que son enviados desde una aplicación a un dispositivo móvil. Dentro de esta clasificación se incluye al mecanismo de envío masivo de mensajes a usuarios con fines publicitarios.

2.4.3 MO (Mobile Originated)

Denominados así a los mensajes que son originados en un dispositivo móvil y enviados hacia una aplicación. Un claro ejemplo dentro de esta clasificación es la de los usuarios que envían mensajes a un número corto, en espera de un registro en un concurso, programación de una alerta o para obtener información precisa. Este número previamente ha sido contratado a las empresas de telefonía celular por empresas integradoras, las cuales se encargan de brindar servicios y contenidos en SMS.

2.4.4 Enfoque del servicio

Gradualmente, la tecnología asociada a los servicios SMS va adquiriendo importancia, enfocada a ser una herramienta poderosa para cualquier empresa que requiera medios eficaces de comunicación con clientes, proveedores, empleados, alumnos, etc., tanto si se requiere una comunicación masiva a usuarios registrados, como si necesita regularmente enviar información personalizada a cada uno de los abonados al servicio.

2.5 ARQUITECTURA MULTICAPA

La idea principal de la arquitectura multicapa es la de desarrollar subsistemas muy bien definidos, en los que se sitúan clases con sus propios atributos y métodos. Estos subsistemas forman una capa de una solución completa.

La arquitectura multicapa es utilizada en la implementación de sistemas complejos, con la finalidad de fomentar la reutilización de software, aprovechando los conceptos de herencia y encapsulación. Las capas representan una abstracción de las distintas funcionalidades que el sistema ofrece, lo que da como resultado clases independientes, las cuales podrán ser reutilizadas de manera sencilla.

2.5.1 Beneficios

Dentro de los beneficios de la arquitectura multicapa, se puede destacar:

- Aislamiento de la lógica de la aplicación.- Los componentes del sistema pueden ser abstraídos, separados para su reutilización en otras aplicaciones, como elementos independientes.
- Distribución.- Las capas pueden ser distribuidas en diferentes procesos, e incluso en diferentes máquinas, lo que resulta en mejora del rendimiento y aumenta la coordinación y disponibilidad de información.
- Recursos.- Se dedican recursos para cada una de las capas de manera independiente. Esto abre la posibilidad de desarrollarlas en paralelo.

2.5.2 Capas y componentes

El siguiente (Figura 2-1) es un diagrama de la arquitectura multicapa:

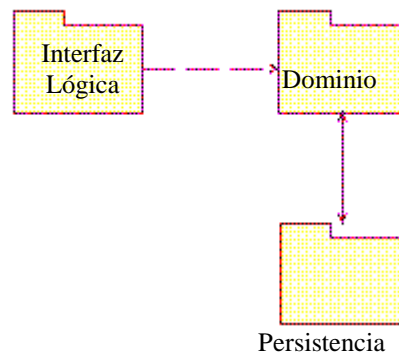


Figura 2-4. Arquitectura Multicapa

2.5.3 Interfaz lógica

La capa de interfaz lógica es la encargada de interactuar con el usuario. Dentro de esta se ubican componentes tales como marcos, pantallas, controles de ingreso de datos por teclado, cuadros de diálogo, etc.

“Todo diseño deberá comenzar por conocer los usuarios destino, así como los perfiles de edad, sexo, habilidades físicas, educación, antecedentes, culturales o étnicos, motivación, objetivos y personalidad”²

2.5.4 Lógica de dominio

Se le denomina lógica de dominio a todos los paquetes y clases que se relacionan con las reglas del negocio y entre sus usos principales están: encapsular la complejidad de acceso y tratamiento a datos de capas inferiores, y contener la lógica para que el resto de capas o componentes pueden interactuar entre sí.

2.5.5 Persistencia

Se le denomina de persistencia a la capa que agrupa componentes y clases relacionadas con el modo de almacenamiento de datos. (Sistemas gestores de bases de datos)

² Ingeniería del Software, Roger Pressman, Editorial McGraw Hill, Quinta Edición, España 2002, pág. 262

2.5.6 Relaciones entre capas

El agrupamiento de capas consiste en componentes con una relación de dependencia unos con otros. Esta dependencia es a causa de elementos dentro de un paquete que requieren elementos de otro paquete.

Los elementos de la capa de interfaz lógica requieren de los elementos de la capa de dominio para funcionar. Estos se encargan de mostrar o recoger información de los usuarios. Muchos de los objetos de la capa de presentación requieren conocer a los elementos de la capa lógica del dominio, pero no al revés, debido a que bajo ningún concepto los objetos de dominio deben depender de los objetos de presentación.

Es diferente en el caso de la relación existente entre la capa de lógica del dominio y la de persistencia. La relación en este caso es de dos vías, esto es, ambos dependen uno del otro en la arquitectura del sistema. La capa de persistencia requiere de la capa de lógica de dominio y viceversa. Esto es debido a que los componentes de la capa lógica querrán almacenar datos, y para ello necesitarán conocer elementos de la capa de persistencia, pero para ello, los elementos de persistencia necesitarán saber el estado de los de dominio.

Cabe resaltar que dentro de cada capa es posible establecer diferentes subcapas, cada una de las cuales con funcionalidades diferentes.

2.6 Plataforma Microsoft .NET Framework

Microsoft.NET comprende un conjunto de tecnologías destinadas a la integración entre personas, sistemas y dispositivos. El objetivo principal de esta nueva plataforma es el mejorar tanto el sistema operativo como el modelo de componentes, y lograr el desarrollo simple de software.

Parte de la estrategia Microsoft .NET es la plataforma .NET Framework, que permite la ejecución de aplicaciones y servicios .NET como: Servicios Web XML, aplicaciones Web, aplicaciones para dispositivos inteligentes, Windows y/o consola.

Los Servicios Web XML son de hecho el pilar de Microsoft .NET. Su principal característica es su completa independencia respecto a su ubicación física real, la plataforma en la que trabaje, el modelo de componentes utilizado para su desarrollo o el lenguaje de programación en el que se base.

Estos servicios aprovechan los estándares de Internet para su acceso. Protocolos como HTTP y el nuevo SOAP (Simple Access Object Protocol), protocolo de acceso simple a objetos, que integra XML y HTTP.

La plataforma .Net también da soporte a la integración completa entre lenguajes, que permite que librerías o componentes hechos en un lenguaje puedan interactuar con otros desarrollados en otro lenguaje.

2.6.1 Visual C # .NET

Visual C# es el nuevo lenguaje de programación que Microsoft presenta dentro de la plataforma de desarrollo .Net. Este lenguaje de programación combina la potencia de lenguajes de bajo nivel como C++, con las utilidades de lenguajes como Java o Visual Basic, tomando los principales elementos de los lenguajes más difundidos.

Acorde con la estrategia de Microsoft, C# pretende desplazar a Java y competir con éxito en el mercado de lenguajes de programación para Internet.

Al igual que los lenguajes C y C++, C# permite la programación a bajo nivel, lo que hace posible la creación de código muy eficiente en donde sea requerido.

2.6.1.1 Características de C#

A continuación se listan algunas de las características más importantes del lenguaje C#. En algunos casos, estas se refieren más a la plataforma .Net, y su incidencia directa en C#:

Sencillez: C# elimina muchos elementos que otros lenguajes incluyen y que son innecesarios en .NET. C# es autocontenido, es decir no necesita adicionar ficheros al código. Además, los tipos de datos básicos poseen tamaño fijo, con lo que se gana portabilidad. Adicionalmente, se ha excluido elementos como operadores diferentes del (.) para acceder a miembros de espacios de nombres.

Modernidad: C# incorpora nuevos tipos de datos e instrucciones. Tipos básicos como *string* o *decimal* apoyan directamente a operaciones de alta precisión y manejo de cadenas de caracteres, e instrucciones como *foreach* permite recorrer matrices o agrupaciones con facilidad.

Orientación a objetos: C# es un lenguaje orientado a objetos, pero basado en un nuevo enfoque que al no permitir variables ni funciones globales, y obligar que los tipos de datos sean creados dentro de definiciones de datos, apoya a la legibilidad y evita conflictos con nombres.

Adicionalmente, C# está diseñado para soportar las características de la programación orientada a objetos: **encapsulación, herencia y polimorfismo**.

Orientación a componentes: C# presenta elementos diseñados para la creación de componentes, como parte integral del lenguaje. Es sencilla la definición de propiedades, eventos o atributos.

Eficiente: La instrucción *unsafe* permite al programador evitar las restricciones de código inseguro. Utilizando punteros en las regiones marcadas bajo esta instrucción, aumenta el rendimiento en escenarios en los que esta característica sea vital.

Compatible: Característica propia de todos los lenguajes .Net, C# permite el uso de librerías de clases creadas en otros lenguajes, tales como Visual Basic.Net o Java.Net. Esta compatibilidad resulta muy útil al momento de desarrollar aplicaciones basadas en componentes, donde las capas pueden ser integradas independientemente del lenguaje de programación que haya sido utilizado.

2.6.2 Visual Basic.Net

Visual Basic.Net constituye la nueva generación de la plataforma de desarrollo Microsoft Visual Basic. Con millones de programadores de este lenguaje en el mundo, Visual Basic.Net es la herramienta más productiva para crear aplicaciones para ser ejecutadas bajo Microsoft Windows.

El lenguaje de Visual Basic .NET se ha actualizado, simplificado y modernizado. Con Visual Basic .NET, ahora tiene acceso a un conjunto de herramientas mucho más completo y eficaz que en versiones anteriores de Visual Basic. Para satisfacer la fuerte demanda de los clientes, Visual Basic .NET ofrece un amplio conjunto de nuevas características, como capacidades de diseño completamente orientado a objetos, subprocesamiento libre y acceso directo a Microsoft .NET Framework. Asimismo, el lenguaje de Visual Basic se ha optimizado, eliminando palabras clave obsoletas que se habían heredado, mejorando la seguridad de tipos y revelando las construcciones de bajo nivel que los programadores avanzados necesitan.

De igual manera, Visual Basic.Net se integra con otros lenguajes de programación de la familia .Net.

“La gran cantidad de programadores que utilizan Visual Basic migrarán a Visual Basic.Net, haciendo de este el lenguaje de programación más exitoso del entorno .Net”³

Crear aplicaciones y servicios Web XML

Visual Basic .NET es la herramienta principal de Microsoft para desarrollar cualquier tipo de aplicación para la plataforma .NET. Con el Diseñador de Web Forms y el Diseñador XML de Visual Studio, los programadores pueden utilizar las características de Microsoft IntelliSense® y la capacidad para completar etiquetas; o bien, pueden elegir el editor WYSIWYG (lo que ve es lo que se obtiene) para crear aplicaciones Web interactivas arrastrando y colocando elementos. Con sólo algunos pasos sencillos, los programadores pueden diseñar, programar, depurar y distribuir fácilmente servicios Web XML que reduzcan el tiempo de programación al poder encapsular los procesos empresariales y permitir ser reutilizados desde cualquier plataforma.

Crear aplicaciones Web móviles

Visual Basic .NET y Visual C# ofrecen características de Internet móvil que permiten a los programadores crear una interfaz Web móvil única, compatible con una amplia gama de dispositivos—como WML 1.1 para teléfonos móviles WAP, HTML compacto (cHTML) para teléfonos i-mode y HTML para Pocket PC, dispositivos de mano y localizadores (pagers). Los controles móviles de servidor generan de manera inteligente la reproducción y paginación apropiada para cada dispositivo Web, proporcionando una experiencia completa y coherente al usuario a la vez que mantiene la flexibilidad del programador.

Visual Studio .NET

La herramienta de desarrollo integrada Visual Studio .NET proporciona el centro de control más avanzado para los programadores. La página de inicio ofrece a los programadores un portal donde obtener información con un solo clic sobre proyectos utilizados recientemente, preferencias, actualizaciones y la comunidad MSDN Online. Las características de Microsoft IntelliSense, el cuadro de herramientas y la lista de tareas proporcionan mejoras de productividad importantes, mientras que las ventanas de ocultación automática y la compatibilidad con varios monitores ayudan a los programadores a maximizar el estado real de la pantalla y personalizar su entorno de desarrollo.

³ Súper utilidades para Visual Basic.Net, Kris Jamsa, Editorial McGraw Hill, España. 2003, pág. 3.

Con los asistentes para aplicaciones, las plantillas de proyectos y el código fuente de ejemplo, los programadores pueden crear rápidamente aplicaciones basadas en .NET para Windows, Internet y dispositivos con una inversión inicial mínima. La ayuda dinámica y Microsoft Developer Network (MSDN®) proporcionan soporte para la tarea que se ejecuta en ese momento, para que los programadores nunca carezcan de la información necesaria acerca de la plataforma Microsoft .NET. Las macros de Visual Studio, al igual que las de Visual Basic para Aplicaciones (VBA) en Office, permiten la automatización de las tareas rutinarias en el IDE, aumentando aún más la productividad general de los programadores de Visual Basic.

3 CAPÍTULO III: Descripción del modelo propuesto para el desarrollo de aplicaciones que integren tecnologías inalámbricas

Para demostrar la aplicación del modelo propuesto en el presente trabajo de titulación, se ha implementado un prototipo que integra las tecnologías WEB, WAP, SMS Y SMTP. Basándose en ellas, el sistema ofrece varias opciones para la consulta de notas y horarios a los estudiantes de la Universidad de las Américas.

Cabe resaltar que este modelo propuesto es genérico, esto quiere decir que dependiendo del escenario bajo el que se vaya a desarrollar una solución informática, muchos de los componentes descritos a continuación podrán ser utilizados, otros omitidos e incluso algunos otros deberán ser adicionados. El prototipo de ejemplo ha sido desarrollado para una situación específica, con un almacén de datos previamente desarrollado, y la solución comprende una aplicación integral de consulta de información de notas y horarios, a través de varios medios, haciendo uso de los protocolos ya mencionados.

El modelo propuesto sugiere una arquitectura de aplicación multicapa, basada en componentes, de modo de tener librerías de clases agrupadas de forma lógica. Esto supone una ventaja de la aplicación, puesto que brinda completa escalabilidad, reutilización, y la posibilidad de distribuir los componentes entre varios servidores, para aumentar su rendimiento.

Así, las capas propuestas son similares a las capas de cualquier aplicación desarrollada bajo la arquitectura multicapa (Figura 3.1):

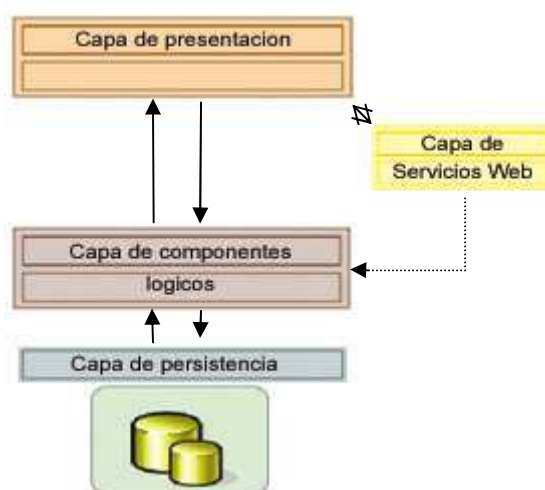


Figura 3.1 Esquema del modelo propuesto

La capa de persistencia contiene los orígenes de datos. Por lo general, los datos son almacenados en bases de datos relacionales. Independientemente del gestor de base de datos (Oracle, SQL Server, MySQL, etc.), su contenido es manejado desde las capas superiores. Específicamente, mediante componentes de conectividad y de seguridad, la capa de componentes lógicos permite o impide el acceso a las bases de datos, manteniendo la información consistente y segura. De igual forma, las funciones implementadas en las librerías de clases permiten la extracción de datos o la modificación de los mismos, dependiendo de las reglas del negocio.

Dentro de la capa de componentes lógicos se almacenan las librerías de clases, que contienen la lógica del negocio. Las librerías de clases implementan funcionalidades diferentes dentro de la aplicación, y pueden ser tan variadas como la de realizar cálculos internos, lograr la conexión a la base de datos, validar usuarios, enviar correo electrónico, modificar datos, etc. La capa de componentes lógicos puede dividirse en subcapas, de modo que los componentes de niveles superiores se apoyen continuamente de los componentes de niveles inferiores (Por ejemplo, dentro de la capa puede existir un componente que modifica un conjunto determinado de los datos de origen, pero este componente debe primero apoyarse en el componente de conectividad, que es el que permite el acceso a la base de datos).

La capa de servicios Web contiene el vínculo entre la aplicación y entidades externas. Concretamente, para lograr la integración de la aplicación con el protocolo SMS, un servicio Web debe ser implementado para encapsular funciones específicas de las clases dentro de la capa de componentes lógicos que permitan la comunicación entre aplicaciones diversas, basándose en el estándar XML. La otra entidad con la que se debe establecer la comunicación es con la Empresa Integradora, como se muestra en la Figura 3.2.

La empresa integradora es la entidad que controla el envío de mensajes de texto a los celulares registrados en cualquier empresa operadora de telefonía celular, por lo que su alcance a los usuarios de teléfonos móviles es del 100%. Es a través de ella que se puede gestionar el envío de mensajes con la información que una aplicación que implementa el protocolo SMS desea enviar a sus usuarios.

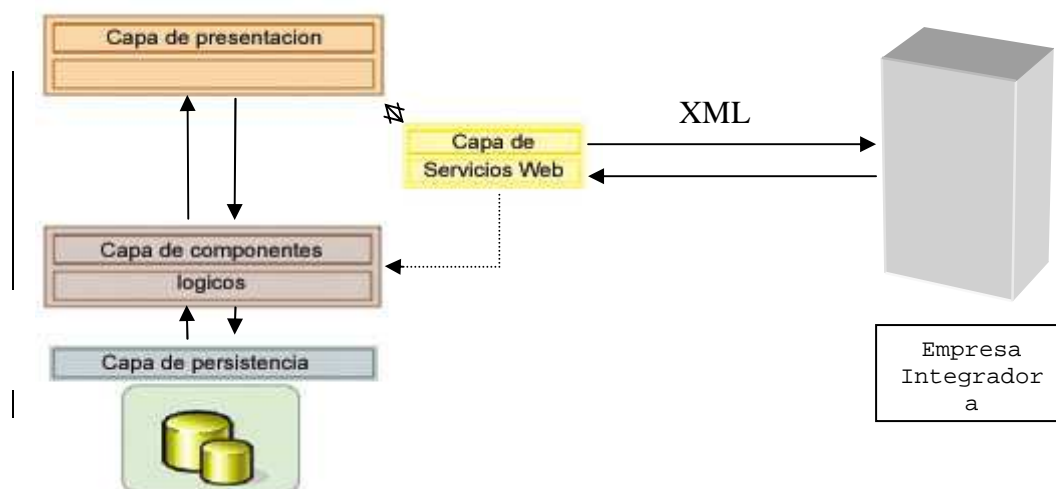


Figura 3.2 Interacción de la aplicación con empresa integradora

Dado que los datos enviados por lo general tendrán el carácter de privado, se debe desarrollar un Servicio Web y publicarlo en un servidor Web, asignando una URL a dicho servicio, tomando en cuenta los aspectos de seguridad que serán especificados más adelante. La empresa integradora tendrá acceso a la URL asignada, y se establecerá el canal de comunicación entre aplicaciones. Mediante mensajes XML se realizarán las peticiones y se retornarán los datos solicitados. La empresa integradora responderá las solicitudes de los usuarios de SMS o bien enviará mensajes de texto masivamente. Todo dependerá de la lógica del negocio, implementada en la capa de componentes lógicos.

Finalmente, la capa de presentación es la encargada de interactuar directamente con los usuarios finales. Dependiendo de la aplicación a desarrollar, se puede optar por una interfaz Windows, o por interfaces basadas en Web. Pero en aplicaciones que además desean implementar interfaces de usuario a través de teléfonos celulares, se debe integrar el protocolo WAP, el cual funciona de manera similar a las aplicaciones Web, y es soportado por teléfonos de última generación. El servicio de Internet móvil es ofertado por las operadoras de telefonía celular, a un precio determinado. Así, en caso de cumplir con estos dos requerimientos mencionados, es posible acceder a las aplicaciones WAP que se encuentren publicadas. El usuario puede acceder a interfaces sencillas, propias de las pantallas de los teléfonos celulares que por su tamaño reducido no dan cabida sino a los mínimos controles para ingreso y despliegue de datos. Botones, etiquetas de enlaces, cuadros de ingreso de texto, etc., son controles que las interfaces WAP implementan para interactuar con el usuario (Figura 3.3).

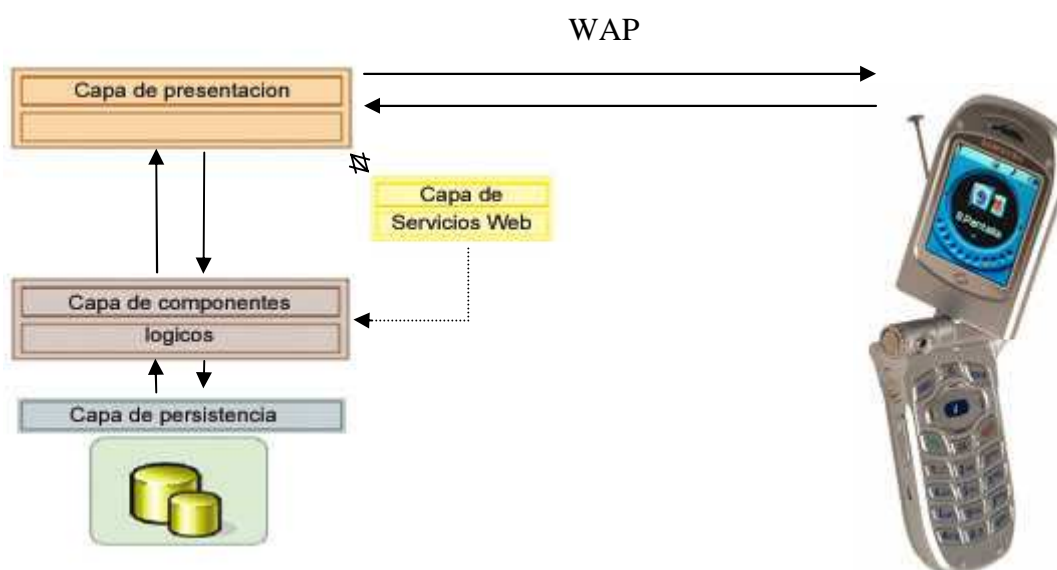


Figura 3.3 Aplicación WAP

Como se puede apreciar en la figura 3.3, la capa de presentación se apoya tanto en la capa de componentes lógicos como en la capa de servicios Web. Esto se debe a que ciertas aplicaciones contenidas en la capa de presentación utilizan directamente las librerías de la capa de componentes lógicos, como el caso de aplicaciones cliente, lo que aumenta su rendimiento. En otro caso, algunas aplicaciones Web se apoyan en los servicios Web desarrollados, debido a que permiten su reutilización y su acoplamiento entre otras plataformas. Siempre dependerá de la funcionalidad que se desea dar a la aplicación, que puede requerir ambos escenarios. Para aplicaciones compatibles con teléfonos celulares, se utiliza los servicios Web directamente.

3.1 Resumen de las capas propuestas en el modelo y su relación con el prototipo de ejemplo

- Dentro de la capa de persistencia se encuentran las bases de datos. Para el ejemplo presentado se han creado bases de datos denominadas SAES y MOVIL respectivamente, construidas bajo el gestor de base de datos SQL Server 2000. La base de datos SAES almacena la información académica sobre la cual se ejecutarán las consultas realizadas por el usuario final (estudiante). La base de datos MOVIL por su parte, almacena la información de los usuarios registrados al sistema, así también como los usuarios para el Configurador y las noticias. Un único componente se encarga de gestionar la conexión entre las capas superiores, y la capa de repositorio de datos, y forma parte de la capa de componentes lógicos.

La capa de persistencia además contiene las consultas implementadas dentro de cada una de las bases de datos, y los procedimientos almacenados desarrollados, siendo estos últimos los destinados a realizar todas las operaciones de manejo de base de datos, mediante peticiones de las capas superiores.

- La capa de componentes lógicos está formada por librerías de clases, cada una de las cuales implementa una funcionalidad distinta en la aplicación. Corresponde a la lógica de la solución planteada. En el prototipo desarrollado se usaron Visual Basic.Net y Visual C#.Net como lenguajes de programación para construir las distintas clases que luego se agruparían para formar los componentes de la capa lógica. De igual forma, estos lenguajes fueron utilizados en el desarrollo de las aplicaciones para el usuario final.
- Los servicios Web XML presentan un nivel adicional de encapsulamiento a las clases que conforman la capa de componentes lógicos. Su uso se justifica en su capacidad de exponer componentes lógicos de las organizaciones sobre Internet y permitir su interacción entre distintas plataformas y sistemas informáticos. Para el prototipo de ejemplo, han sido desarrollados dos Servicios Web XML denominados como *UdlaServicio* y *UdlaServicioSMS*. El primero se lo ha desarrollado con el objetivo de ser un servicio privado para las aplicaciones de la universidad, y el segundo como un servicio público, para interactuar con entidades externas.
- Finalmente, la capa de presentación contiene las aplicaciones para interactuar con el usuario final. Ya sea desde Internet a través de una computadora o desde un dispositivo móvil o usando un teléfono celular que soporte SMS, el usuario final utiliza las aplicaciones para hacer uso del sistema. En el prototipo de ejemplo, el estudiante puede realizar consultas de sus calificaciones y sus horarios a través de varios medios. Como servicios adicionales, puede registrarse en un sitio de Internet, y así recibir por mensajes de texto, calificaciones y noticias enviadas por la Universidad.

3.1.1 Esquema gráfico del prototipo de ejemplo

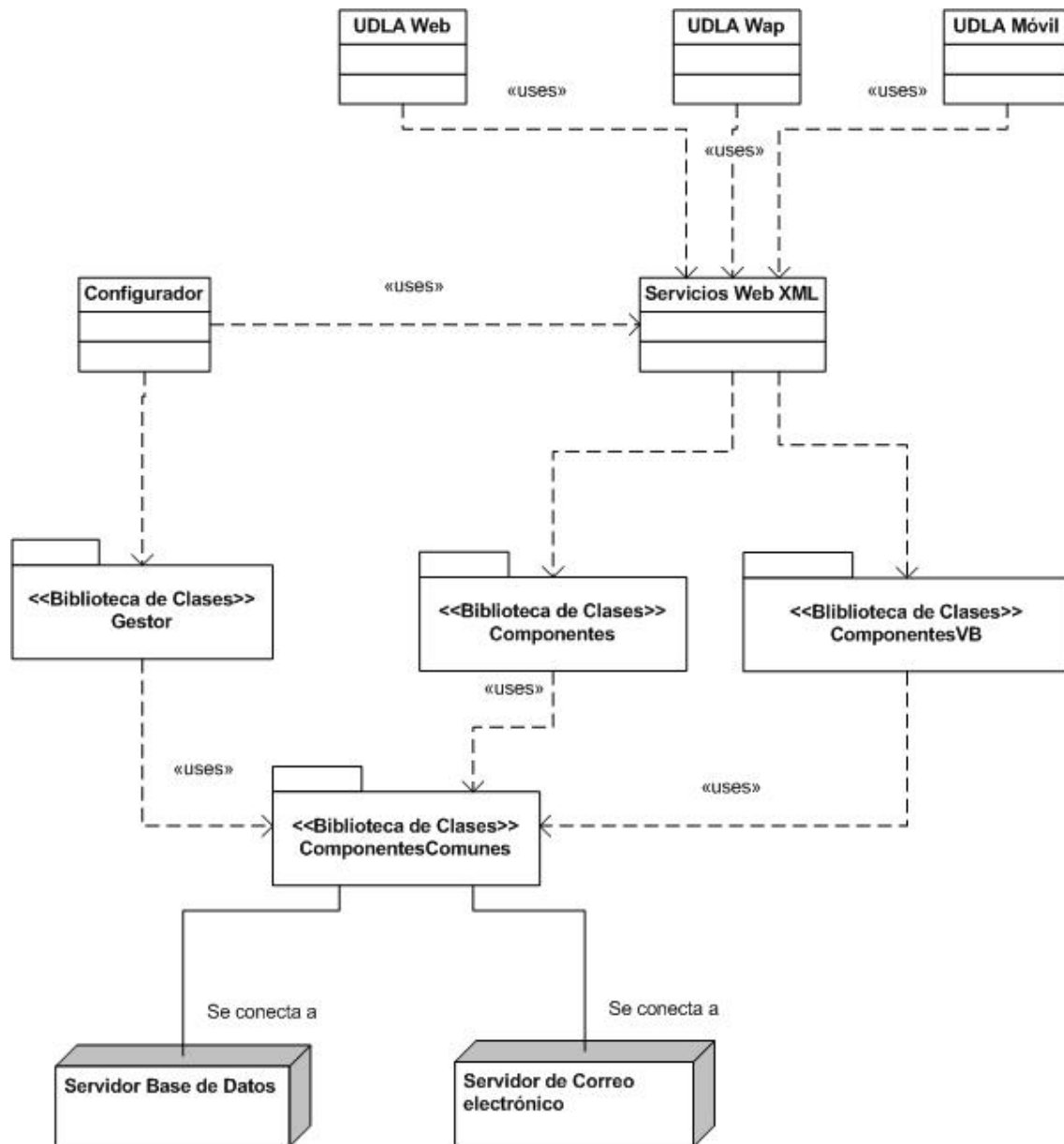


Figura 3.4 Esquema gráfico del prototipo de ejemplo

3.2 Definición individual de las capas del modelo propuesto y relación entre ellas

Una vez presentado el resumen del modelo, y mostrada la relación con el prototipo desarrollado, se describirán con detalle cada una de las capas que el modelo sugiere.

3.2.1. Capa de persistencia

La capa de persistencia o de repositorio de datos es usada generalmente en las aplicaciones para el almacenamiento de los datos que la solución utiliza. El modelo agrupa en esta capa las bases de datos, las consultas y sus procedimientos almacenados.

Para el escenario de ejemplo, se tiene dos bases de datos diferentes, las cuales son:

- Base de Datos SAES⁴
- Base de Datos MOBIL⁵

La capa de persistencia y la capa de componentes lógicos se relacionan mediante una librería de clases destinadas a la conectividad, que forma parte de la capa superior.

3.2.1.1 Requerimientos de la capa de Persistencia

Para esta capa, los requerimientos planteados por el modelo son:

- **Servidor de base de datos:** Es necesario que para el manejo de los datos estos sean almacenados en un servidor de base de datos, específicamente por razones de organización, acceso y seguridad de la información. Dependiendo del escenario bajo el cual se maneje la solución a desarrollarse, la plataforma del servidor podrá ser variada, y se deberá tomar una decisión en cuanto a los componentes que se encarguen de conectar el almacén de datos con las capas superiores. Así mismo, la aplicación puede hacer uso de otras fuentes de datos, servidores Exchange, sistemas de archivos, etc.
- **Gestor de base de datos:** Usualmente, las aplicaciones usan bases de datos relacionales para almacenar los datos. Por ello, dentro del servidor de bases de datos debe instalarse un gestor que maneje la base de datos relacional. De igual manera, dependerá del escenario el tipo de gestor que se vaya a utilizar. Las consideraciones para la selección de un sistema gestor de bases de datos tienen relación con el volumen de datos que se vaya a manejar, la cantidad de usuarios que soportará,

⁴ Base de Datos propia de la Universidad de las Américas.

⁵ Base de Datos sugerida para este Prototipo.

aspectos de seguridad, etc. En el prototipo de ejemplo, se usa Microsoft SQL Server 2000, sobre el que están desarrolladas e implementadas dos diferentes bases de datos.

- **Consultas y procedimientos almacenados:** Las consultas o vistas y los procedimientos almacenados forman parte de las bases de datos, y sirven para aumentar el rendimiento y proteger los datos almacenados. Se deben tomar estrategias de bases de datos para generar consultas de alto rendimiento para luego a través de ellas obtener la información mediante procedimientos.

3.2.1.2 Seguridades para la capa de Persistencia

Una parte importante dentro de la aplicación es la seguridad a nivel de la capa de persistencia. Tanto servidores como las bases de datos deben tener protección y controles de acceso, a fin de mantener la integridad de los datos, y evitar que estos puedan ser eliminados, modificados o suplantados.

- El servidor de base de datos debe contar con seguridades tanto físicas como lógicas. Es fuertemente recomendado la instalación de cortafuegos, una configuración apropiada de la red (se sugiere una configuración DMZ, Zona o red desmilitarizada). Al trabajar bajo plataformas Windows, el servidor debe ser constantemente actualizado con parches y "Service Packs", para evitar ataques directos al servidor por fallas de sistema operativo.
- La seguridad de la base de datos debe enfatizar en el control el acceso de los usuarios a los datos. En el escenario de ejemplo, la base de datos SAES, que almacena toda la información académica, se ha creado inicios de sesión y se ha establecido permisos para los mismos. De esta manera, sólo usuarios autorizados podrían tener acceso a los datos y realizar modificaciones en los mismos.
- Para aumentar la seguridad y el rendimiento en la base de datos, se utiliza vistas para la obtención de información. Estas representan tablas virtuales que pueden contener una o más tablas de la base, y no permiten modificación alguna. De igual forma, se puede incorporar procedimientos almacenados, cuya ejecución puede dar como resultado desde la extracción de datos hasta su modificación.
Las vistas y los procedimientos almacenados deben implementar controles de acceso, de modo de controlar sus permisos de ejecución. Para completa seguridad, los procedimientos almacenados pueden ser cifrados.

3.2.2 Capa de Componentes Lógicos

Los componentes lógicos serán los encargados de ejecutar las operaciones que sean solicitadas por parte de la capa de presentación y la capa de servicios, interactuando y obteniendo la información solicitada de la capa de persistencia.

Dichos componentes deberán ser implementados como librerías de clases que encapsularán todas las tareas que podrían ser invocadas por la capa de presentación. Cada clase implementa una funcionalidad diferente dependiendo de los requerimientos de la solución. Estas librerías de clases pueden ser desarrolladas en cualquier lenguaje de programación. Para el prototipo de ejemplo, se utilizó la plataforma Microsoft .NET .

Los componentes lógicos son el medio de comunicación e interacción entre las capas de: persistencia de datos, servicios Web XML privados y públicos, y de presentación. Implementan clases para realizar la conexión con los orígenes de datos, logrando la interacción con la capa de persistencia. Por otro lado, los servicios Web encapsulan los métodos implementados dentro de los componentes. Finalmente, las librerías resultantes son referenciadas por la capa de aplicación, permitiendo al usuario invocar todas las funcionalidades implementadas.

Dentro de la capa de componentes lógicos puede existir subcapas, organizadas de tal modo que una capa inferior pueda ser reutilizada por otra capa superior.

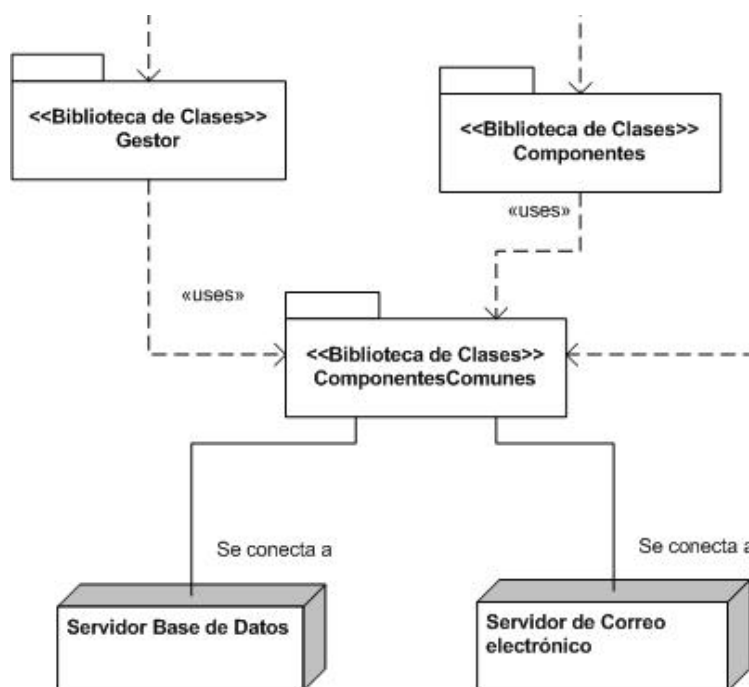


Figura 3.5 Ejemplo de subdivisión de capas en la capa de componentes lógicos

Como se aprecia en la figura 3.2.2, dentro de la biblioteca de clases se tiene un componente denominado “Componentes Comunes”, el cual abarca una clase encargada de la conexión a la base de datos. Cuando los componentes de las capas superiores, en este caso el componente “Gestor” y el componente “Componentes”, se apoyan en la capa inferior para lograr conexión a los orígenes de datos y validar las transacciones.

Cada clase deben incluir código de control de excepciones, para de esta manera poder prever y manejar las excepciones que pudieran producirse en su funcionamiento, que pueden ser ingresos erróneos de datos, lectura de registros vacíos, y obtención de valores nulos, entre otros.

Para el escenario de ejemplo se ha diseñado un modelo que trabaja con tres librerías de clases, dos librerías de clases desarrolladas en C# (Componentes y Gestor, ver Anexo IX-Manual Técnico), y otra desarrollada de Visual Basic .NET (ComponentesVB, ver Anexo IX-Manual Técnico).

Las librerías de clases desarrolladas en C#, permite realizar lo siguiente:

- Obtención de Datos de Alumnos
- Obtención de Datos de Materias
- Obtención de Datos de Notas
- Obtención y Manejo de Datos de Noticias
- Obtención y Manejo de Datos de Usuarios del Sistema Móvil
- Procesos de Envíos de Notas y Horarios por medio de correo electrónico.
- Procesos de Envíos de Noticias por medio de correo electrónico.
- Procesos de Envíos de Noticias por medio de SMS.
- Ejecución y Control de Procesos Automáticos de Envíos.

Mientras la librería de clases desarrollada en Visual Basic, permite realizar lo siguiente:

- Estructuración de Formatos para Notas y Horarios
- Obtención de Datos de Horarios

Los componentes lógicos son utilizados por la capa de servicios de dos formas, la primera mediante el servicio interno UDLAServicio, el cual encapsula los métodos definidos en las clases de la capa de componentes lógicos destinados a ser consumidos por las aplicaciones del prototipo, y la segunda, UDLAServicioSMS, encapsula los métodos destinados a la

obtención de información necesaria para una empresa integradora encargada de hacer llegar estos datos hacia los celulares de los usuarios por medio de SMS.

El prototipo de ejemplo requiere la creación de un nuevo componente, denominado GESTOR, el cual se apoya en los componentes descritos, y a su vez implementa nuevos métodos destinados al control y manejo de procesos automáticos de envíos de Notas, Noticias y Horarios a los estudiantes de la Universidad de las Américas, ya sea por medio de correo electrónico o por medio de mensajes de texto cortos a los celulares de los mismos (SMS).

3.2.2.1 Requerimientos de la Capa de Componentes Lógicos

- El modelo recomienda que esta capa sea instalada en un servidor de aplicaciones, o en el caso de no tener uno disponible se lo debería localizar en un servidor Web, o en el peor de los casos en un servidor diferente al servidor de base de datos. Al tratarse de la capa encargada de realizar las transacciones, debe evitarse sobrecargar al servidor con las peticiones de los usuarios.
- Cada librería de clases debe agruparse en componentes de forma lógica, y deben ser independientes unas de otras, para aumentar el rendimiento en procesos paralelos no relacionados.

3.2.2.2 Seguridades para la Capa de Componentes Lógicos

- Dependiendo del escenario en el que se vaya a desarrollar la aplicación, se debe o bien utilizar el modelo de autenticación de Windows, o bien desarrollar componentes personalizados de autenticación. El prototipo que ejemplifica el modelo propuesto, utiliza componentes de autenticación propios debido a que en este escenario en particular se trabaja con datos ya existentes en una base, que no se encuentra registrada dentro de un servicio Windows como Active Directory, por lo que no utiliza credenciales Windows.
- La llamada a los procedimientos almacenados de las bases de datos tiene que utilizar exclusivamente los componentes desarrollados para la conectividad. De esta manera, cualquier transacción o petición por parte del usuario deberá pasar por el proceso establecido de conexión y validación, evitando violaciones de acceso.

3.2.3 Capa de Servicios Web

La capa de servicios Web contiene los servicios que la aplicación contemple, especialmente si la aplicación va a ser ejecutada desde Internet y se requiere interacción con otras aplicaciones u otras plataformas. Como se ha hablado en el capítulo II, la potencia de los servicios Web radica en XML, estándar de comunicación de datos entre distintas plataformas y tecnologías de manera transparente y simple. Así, desde cualquier sistema informático o dispositivo electrónico que soporte XML, los servicios Web podrán ser consumidos.

Los servicios Web implementados en el prototipo de ejemplo encapsulan los métodos creados en las clases de la capa de componentes lógicos, e interactuarán con las aplicaciones desarrolladas. A su vez, permiten el intercambio de datos con entidades externas, en este caso, con la empresa integradora, encargada del envío de mensajes a los teléfonos celulares. (Anexo II)

3.2.3.1 Requerimientos de la Capa de Servicios Web

- Los servicios Web deben estar publicados en el servidor Web para que puedan ser consumidos. En el escenario de ejemplo, se ha creado el servicio denominado UDLAServicio, el cual está destinado para el uso interno de las aplicaciones del prototipo. (Anexo II)
- Para el caso de comunicación con aplicaciones externas, el servicio debe estar asignado a una URL (Uniform Resource Locator) disponible externamente. El prototipo incluye el servicio UDLAServicioSMS, al que tiene alcance el servicio Web de la empresa integradora.

3.2.3.2 Aspectos de seguridad para la Capa de Servicios Web

El tema de seguridades dentro de los servicios Web es muy amplio, y no se encuentra dentro del alcance del presente Trabajo de Titulación. Sin embargo, es preciso analizarlo y dar una idea de algunos de los más importantes aspectos:

- Si se utiliza el IIS (Internet Information Server) para publicar el servicio, se debe deshabilitar la autenticación de usuarios anónimos, de modo que sólo usuarios autorizados puedan acceder y consumir el servicio.
- También es posible configurar el servidor Web de modo que pueda validar direcciones IP específicas, y autorizar o denegar el acceso al servicio mediante direcciones IP registradas.

- Mediante ASP.NET se puede restringir los usuarios autorizados a una URL específica, de la siguiente manera:

```
<authorization>  
  <allow user = Dominio/usuario">  
</authorization>
```

- Dependiendo del escenario, pueden utilizarse combinaciones con SSL (Secure Sockets Layer).
- La configuración del servidor Web en el que vaya a ser publicado el servicio Web debe ser la adecuada. Como consideraciones especiales, cabe señalar que la arquitectura de seguridad juega un papel muy importante, así como los respectivos parches, cortafuegos, routers y en suma, toda la seguridad que pueda ser implementada para proteger el acceso y mal uso del servicio.

3.2.4 Capa de Presentación

La capa de presentación es la encargada de habilitar la interacción del usuario con la aplicación o con las aplicaciones, según sea el caso. Estos componentes debe ajustarse al escenario bajo el cual se desarrolla la aplicación, esto es, utilizando los controles y manejadores de eventos apropiados con sus correspondientes validaciones y restricciones, y organizados de forma intuitiva y ordenada, a fin de que el usuario pueda interactuar con ellas sin dificultad.

Los componentes de interfaz pueden ser sencillos, tales como los formularios basados en Windows o en Web que presenta la plataforma de desarrollo Microsoft Visual Studio .NET o si se requiere, componentes más complejos personalizados para suplir necesidades de actividades o procesos especializados, como por ejemplo, una actividad secuencial que requiere de un ayudante automático o "Wizard".

Para el modelo propuesto se tiene la necesidad de implementar tres tipos de interfaces de usuario, la primera es una interfaz para los sitios Web (UDLAWeb y UDLAMobil), la segunda es una interfaz para el sitio WAP (UDLAWap), para el uso de los alumnos de la Universidad de las Américas y la tercera es una interfaz basada en formularios de Windows Forms, utilizado por el administrador del sistema interno de la Universidad de las Américas o por el encargado del mismo.

Los sitios Web, UDLAMobil, UDLAWeb y UDLAWap, están orientados a realizar dos actividades primordiales, la una es el registro de usuarios para el acceso a los servicios de

entrega de notas y horarios por medio de correo electrónico y por medio de SMS, y la otra la obtención de información de las mismas notas y horarios en una página Web (ver Anexo IX-Manual Técnico).

El sitio WAP está orientado únicamente a la consulta de notas y horarios por medio de un teléfono celular que contenga un micro-navegador y tenga activado el servicio correspondiente (ver Anexo IX-Manual Técnico).

Por último, la aplicación Windows está orientada al control de procesos de envío de notas, noticias de la universidad y horarios, ya sea por medio de correo electrónico o por medio de SMS. Esta también es la encargada del control y ejecución de los procesos automáticos de envío de información a los usuarios finales (ver Anexo IX-Manual Técnico).

Las aplicaciones Web y WAP interactúan directamente con la capa de servicios, de manera que sean ellos los encargados de interpretar y ejecutar las peticiones del usuario, obtener la información requerida de la capa de datos y presentarla de forma clara y entendible.

Por otro lado, la aplicación Windows interactúa directamente con la capa de componentes lógicos, y es por medio de ella que se obtiene y se maneja la información correspondiente de la capa de datos. Este trabajo lo realiza por medio de la librería de clases Gestor.

3.2.4.1 Requerimientos de la Capa de Presentación

Para el correcto funcionamiento de las soluciones Web será necesario que los clientes tengan acceso a las mismas desde un navegador Web que soporte el entorno de ASP.Net y una conexión a Internet habilitada.

Para la aplicación Windows se tiene como requerimientos mínimos:

- Procesador PIII 550 Mhz
- 256 Mb de memoria RAM
- Sistema Operativo Windows 2000 o Posterior

Nota: Estos requerimientos están basados en el hecho de que la aplicación Windows es una aplicación residente en memoria, por lo que consume recursos permanentemente. Dado que está desarrollada en .Net requiere una plataforma basada en NT para su correcta instalación.

3.2.4.2 Seguridades para la Capa de Presentación

- Las aplicaciones basadas en Web deben ser implementados siempre con una página de inicio, que solicitará al usuario realizar el ingreso de parámetros necesarios para la confirmación o verificación del mismo, esto es, el ingreso de sus credenciales que lo validan como usuario autorizado de la aplicación. Sin estos datos, la solución Web no debe permitir el acceso a páginas posteriores. El archivo web.config permite la posibilidad de configurar el acceso de usuarios, negándolo en caso de ser un usuario no registrado.
- Para las soluciones basadas en Web, se recomienda la creación de páginas personalizadas de error, de modo de ocultar la información demasiado específica de las líneas de código que lo ocasionaron.
- El modelo planteado sugiere que el código desarrollado en la capa de presentación debe ser independiente al de la capa de componentes lógicos, siendo esta última la encargada de ejecutar las peticiones de los usuarios y en general, de la funcionalidad de la aplicación. En caso de escribir lógica del negocio dentro de los componentes de interfaz, su mantenimiento constituiría en una labor más complicada, puesto que se impediría su capacidad de reutilización y escalabilidad.
- Las credenciales de usuario deberán también ser ingresadas en el resto de aplicaciones.
- El modelo también sugiere la implementación de código que controle el inicio de sesión de cada usuario individualmente, y su correspondiente cierre de sesión. En el caso de las aplicaciones basadas en Web, se recomienda un tiempo estándar de 30 segundos de inactividad para que la solución considere que el usuario ha abandonado la sesión, y proceda a cerrarla.

3.2.5 Descripción Individual del Modelo por Protocolos

3.2.5.1 SMS

El modelo plantea los siguientes pasos para el uso del protocolo SMS dentro de la aplicación:

1. El usuario envía al número asignado (Anexo) el mensaje para realizar la consulta desde su teléfono celular.

2. El mensaje es recibido por la empresa integradora, la cual debe ser la encargada de comunicarse con la aplicación a desarrollarse. Es aquí donde se requiere un servicio Web que interactúe con la aplicación externa, independientemente a la solución a desarrollarse. En el escenario de ejemplo, el servicio UDLAServicioSMS recibe el mensaje como parámetro del método específico.
3. El servicio Web, mediante sus métodos, y haciendo uso de los métodos encapsulados de la capa de componentes lógicos, procesa los datos recibidos por la empresa integradora y los devuelve, como un archivo XML. Para el prototipo, el servicio se encarga de validar el usuario, valida el requerimiento del mensaje, selecciona los datos a ser enviados de retorno, los recupera, y finalmente los devuelve al servicio de la empresa integradora.
4. La empresa integradora da formato al XML recibido del servicio Web desarrollado, en este caso UDLAServicioSMS y lo envía al número de celular de origen, presentando los datos de forma legible.

3.2.5.2 WAP

El modelo plantea los siguientes pasos para el uso del protocolo WAP dentro de la aplicación:

1. El usuario ingresa al sitio desarrollado para el protocolo WAP, e introduce como parámetros de validación sus credenciales de ingreso. En el prototipo, los datos de ingreso son el número de matrícula y el número de cédula.
2. Con estos datos, la aplicación valida al usuario, permitiendo o negando el acceso a la misma. Para este caso específico, los componentes de conectividad y de validación se ejecutan simultáneamente, presentando al usuario la pantalla de error en caso de no ser un usuario válido, o la posibilidad de continuar la consulta, solicitando los parámetros Año y Período.
3. El usuario WAP tiene la posibilidad de navegar en el sitio, dependiendo del escenario en el cual la aplicación ha sido desarrollada. El prototipo presenta la opción de consultar notas y horarios, basados en los parámetros de ingreso, tanto de las credenciales del usuario como en los parámetros ingresados posteriormente. Para este fin, el servicio Web desarrollado utiliza los métodos correspondientes, de conexión, de consulta, y de presentación de datos.

4. Finalmente, el sitio WAP permite la opción de cierre de sesión, de modo que la información queda protegida de otros usuarios no autorizados a la misma.

4 CAPÍTULO IV: Conclusiones y Recomendaciones

4.1 Conclusiones

- Se concluye que el modelo de desarrollo de las aplicaciones WAP es similar al de las aplicaciones Web. Aún cuando el lenguaje nativo de WAP es el denominado WML, la tecnología del paquete de desarrollo .Net permite desarrollar aplicaciones WAP multicapa, posibilitando la reutilización de componentes y la publicación en servidores de modo que estén al alcance de todos los usuarios que posean Internet Móvil en sus celulares.
- Se concluye que la integración de aplicaciones tradicionales con el protocolo SMS se da a través de las denominadas Empresas Integradoras, encargada de gestionar el envío de mensajes a todos los usuarios registrados en todas las operadoras de telefonía celular del país. Mediante servicios Web XML, la aplicación y la empresa integradora se comunican enviando peticiones, realizando consultas o transacciones, y devolviendo la información a donde se requiera. El nivel de interacción de la aplicación será definido por la lógica del negocio y por los métodos encapsulados en los servicios Web XML que se publiquen para este fin.
- Se concluye que los teléfonos celulares de tercera generación poseen características que integran tecnologías y protocolos como WAP y SMS, las cuales pueden emplearse para la interacción con diversas aplicaciones. Estos dispositivos incorporan micro navegadores mediante los cuales es posible ejecutar aplicaciones WAP, así como también soporte para SMS, que permiten el envío y recepción de mensajes cortos de texto, mediante los cuales es posible la interacción entre usuarios, o incluso entre un usuario y una aplicación. Todo esto representa ahorro de costos al no necesitarse una computadora con conexión física a Internet, y la facilidad de movilidad, al permitir interactuar con aplicaciones y sistemas informáticos desde cualquier parte.
- Se concluye que la potencia de los Servicios Web se basa en que, al utilizar el estándar XML, es compatible con diferentes plataformas y aplicaciones, y por tanto, puede interactuar con ellas. Es posible encapsular en un servicio Web métodos o funcionalidades específicas de una aplicación, y publicarlo en un servidor, asignándole una dirección determinada, de modo que otras aplicaciones o sistemas informáticos puedan consumir este servicio, con el paso de parámetros por mensajes, a modo petición – ejecución. Una aplicación puede contener uno o más servicios Web, dependiendo de la necesidad de la misma, y mantenerlos privados o públicos, de modo que puedan ser consumidos y reutilizados por una o varias aplicaciones, ya sea

internas o externas, sin mostrar lo que existe detrás de ellos, o dentro de la aplicación en sí, lo que constituye un importante aspecto de seguridad.

- Se concluye que la plataforma de desarrollo Visual Studio.Net constituye una herramienta muy poderosa puesto que sus librerías y componentes apoyan al desarrollo de aplicaciones para varias plataformas y varios usos. Dado que incluye varios lenguajes de programación, permite al desarrollador trabajar bajo cualquier sintaxis a la que haya estado acostumbrado, y es posible integrar componentes diversos, sin importar el lenguaje en el que hayan sido creados.

Microsoft Visual Studio.Net facilita la creación de aplicaciones para varias plataformas, manteniendo el esquema visual de desarrollo. Es posible crear aplicaciones Windows, aplicaciones Web, aplicaciones para teléfonos celulares (WAP), Servicios Web, etc. De igual manera, permite la creación de componentes independientes, que pueden ser reutilizados en las diversas aplicaciones.

Por su diversidad de lenguajes, facilidad de integración, posibilidad de desarrollo de aplicaciones para diferentes plataformas, orientación a objetos, y por ofrecer un sin número de herramientas para la creación de componentes y aplicaciones, Visual Studio.Net constituye un paquete de desarrollo completo, estable, seguro y muy confiable.

4.2 Recomendaciones

- Es recomendable desarrollar aplicaciones con arquitectura multicapa para soluciones complejas, debido a que esto brinda mayor grado de escalabilidad en la aplicación. Además, cuando la solución implementa varias aplicaciones en la capa de presentación, los componentes pueden ser reutilizados, brindando mayor rendimiento.
- Se recomienda establecer niveles de acceso dentro de la capa de persistencia, de modo de proteger su integridad y consistencia. En aplicaciones que manejan bases de datos, la información no debe mostrarse a cualquier usuario, y mucho menos estar expuesta a modificaciones, adiciones o eliminaciones. Por ello, se sugiere utilizar procedimientos almacenados para ejecutar operaciones de consulta o de modificación en las bases de datos, y asignar los permisos de ejecución de los mismos a los usuarios según la lógica de la aplicación.
- Se recomienda utilizar los servicios Web XML por su capacidad de reutilización y de compatibilidad con diferentes plataformas. El servicio Web encapsula la funcionalidad de una solución, y puede ser consumido por diversos tipos de aplicaciones, logrando comunicación e interacción entre entidades diferentes. Un ejemplo de ello es la manera en que se integra el protocolo SMS en una aplicación, que para lograr la comunicación con la entidad externa denominada Empresa Integradora, se vale de servicios Web, que se comunican mediante mensajes y ejecutan diversas funcionalidades, todo basado en el estándar XML.
- Para lograr la integración del protocolo SMS en aplicaciones, se recomienda desarrollar y publicar un servicio Web XML que contenga las funcionalidades necesarias, de modo de no comprometer la integridad de la información con la que la aplicación interactúa, debido a que se trata de un servicio Web público, que puede ser consumido por diversas aplicaciones.

BIBLIOGRAFIA

Documentos Impresos

- C# and the .NET Framework. Andrew Troelsen, Apress, 2001
- El Lenguaje de Programación C#. José Antonio Gonzáles Seco. Publicación de Internet. 2002
- Ingeniería de Software, Un Enfoque Práctico. Roger S. Pressman. Mc Graw Hill, Quinta Edición, 2002, Madrid
- Arquitectura de Aplicaciones .NET, Diseño de Aplicaciones y Servicios. Patterns & Practices. Microsoft Corporation. Diciembre 2002

Documentos Electrónicos

- Disponible en World Wide Web:
<http://www.allnetdevices.com/developer/tutorials/2000/08/21/wap_and.html>.
- Disponible en World Wide Web:
<http://alarcos.inf-cr.uclm.es/per/fgarcia/isoftware/doc/LabTr3_Multicapa.pdf>.
- Disponible en World Wide Web:
<<http://www.asp.net/Default.aspx?tabindex=0&tabid=1>>.
- Disponible en World Wide Web:
<<http://www.desarrolloweb.com/manuales/54>>.
- Disponible en World Wide Web:
<<http://www.frost.com/>>.
- Disponible en World Wide Web:
<<http://geneura.ugr.es/CUR/C/>>.
- Disponible en World Wide Web:
<<http://www.josanguapo.com/>>.
- Disponible en World Wide Web:
<<http://www.lawebdelprogramador.com/temas/>>.
- Disponible en World Wide Web:
<<http://www.monografias.com/>>.
- Disponible en World Wide Web:
<<http://www.msdn.microsoft.com/vcsharp/language>>.
- Disponible en World Wide Web:
<http://www.speedware.com/solutions/wireless/mobile_development_tools/MobileDev/>
- Disponible en World Wide Web:
<<http://tdg.lsi.us.es/~csharp>>.

- Disponible en World Wide Web:
<<http://www.wapforum.org/what/copyright.htm>>.
- Disponible en World Wide Web:
<<http://wmlclub.com/docs/>>.
- Disponible en World Wide Web:
<<http://www.wmlclub.com/programas/kits.htm>>.

ANEXOS

ANEXO I

METODOLOGIA

Metodología OMT

La metodología de desarrollo OMT tiene como característica principal el alto nivel de abstracción proveniente de los modelos de datos y de estructuras, cuyo análisis se enfila hacia el mundo real. OMT puede ser enfocado en varios aspectos de implementación, tales como bases de datos relacionales, bases de datos orientadas a objetos, archivos. Esta metodología también se preocupa de aspectos relacionados a los recursos sobre los cuales trabajará la solución a desarrollar.

Dentro de la programación orientada a objetos, y basados en el pensamiento de que los objetos son entidades coherentes con características tales como estado, identidad, conducta, estos pueden ser agrupados y organizados de modo de lograr modelos que OMT utiliza para obtener de forma precisa los requerimientos y especificaciones, lograr un nivel muy alto de abstracción y brindar soporte a múltiples estilos de desarrollo.

OMT se basa en el concepto de entidad, y utiliza sus atributos de modo de formar relación entre ellas, alcanzar un modelo real basado en objetos y abarcar su modo de interacción con bases de datos.

Pasos para el proceso de desarrollo OMT

1. Conceptualización

La conceptualización es el análisis del negocio o empresa, constituyendo una observación crítica de los procesos empresariales. Aquí intervienen preguntas acerca del por qué del desarrollo de la aplicación, de que problema está destinado a resolver, del dónde y cuándo será requerido.

2. Análisis

Durante el análisis, se toma el levantamiento de requerimientos de la conceptualización y con ellos se construye el modelo real. El análisis apunta a la especificación precisa de los requerimientos que deben ser satisfechos. Para ello, se puede utilizar diversas fuentes de información, tales como escenarios, casos de uso, lenguajes formales, etc.

Durante el análisis, se hacen las primeras tentativas de las clases que conformarán el modelo, realizando eliminaciones y optimizaciones según casos de irrelevancia o redundancia. Luego, se hace referencia al modelo con las clases obtenidas y sus respectivos atributos y relaciones, y se busca niveles de abstracción para modelar sub sistemas, y lograr un sistema tangible y sólido. Para culminar, se introduce el concepto de

transacción para modelar procesos o representar cambios de datos, dando como resultado un diccionario de datos de todas las entidades.

3. Diseño

Durante esta etapa, se debe especificar estrategias, arquitecturas y políticas. Aspectos como implementaciones de control externo, paradigmas de administración de bases de datos, estrategias de intercambio de datos, detalles de diseño, deben ser tomados en cuenta.

En este paso, se debe clarificar las clases y métodos individuales, logrando una mejor conceptualización del modelo y brindando la posibilidad de mejorarlo y optimizarlo. Se elabora un modelo de objetos, un modelo funcional que será evaluado y finalmente implementado.

4. Mantenimiento

Al seguir paso a paso la metodología OMT, los documentos de desarrollo y seguimiento constituirán la bitácora mediante la cual el desarrollador se valdrá para su posterior mantenimiento.

ANEXO II

DIAGRAMA DE COMPONENTES

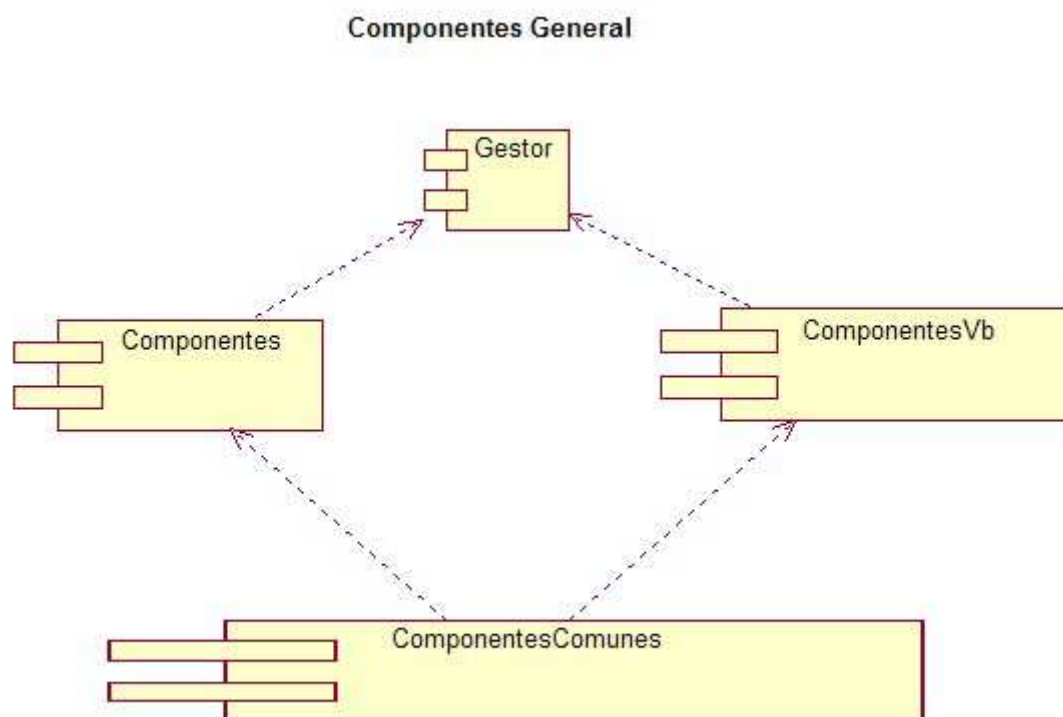


Figura 1. Componentes General

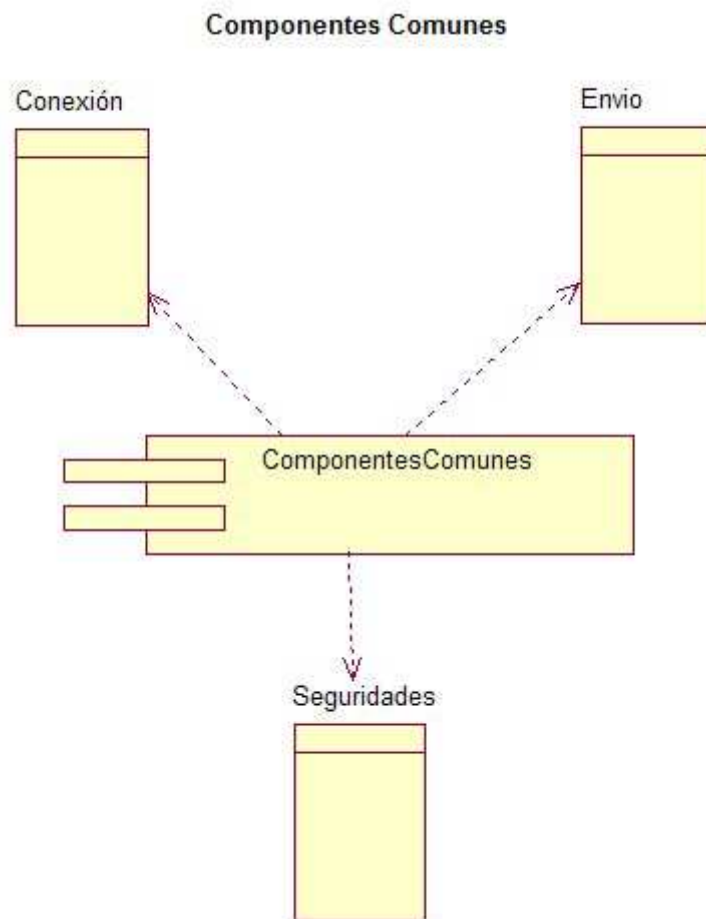


Figura 2. Componente Componentes Comunes

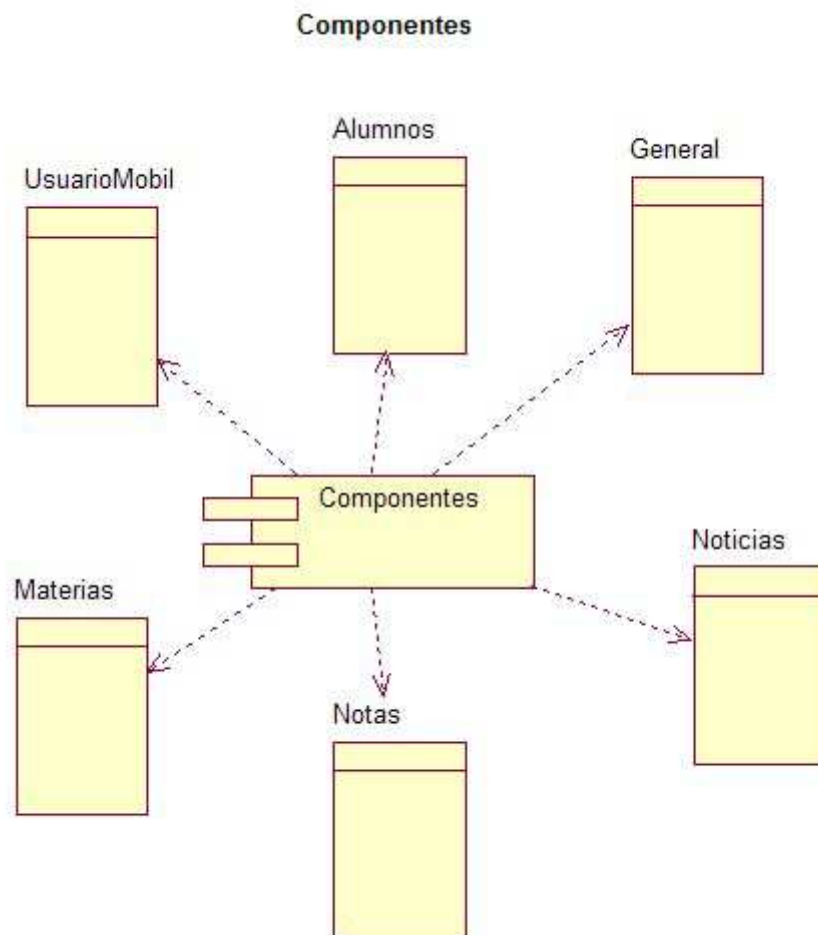


Figura 3. Componente Componentes

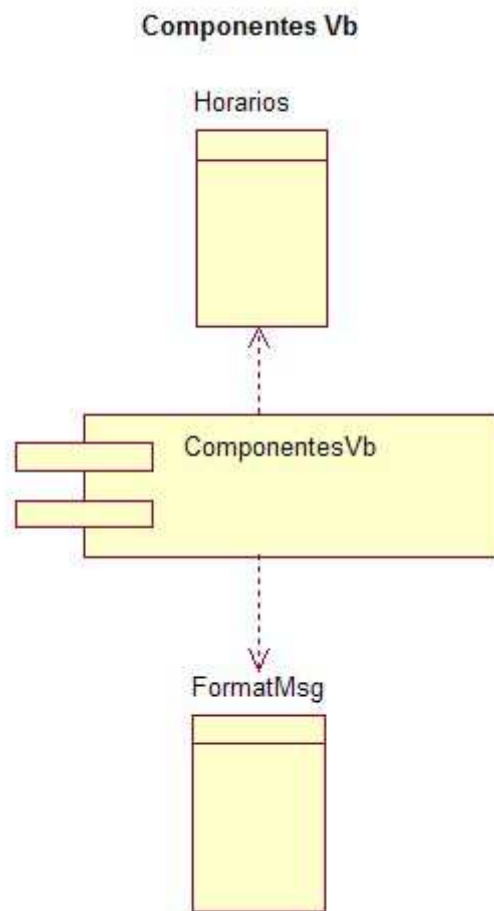


Figura 4. Componente ComponentesVb

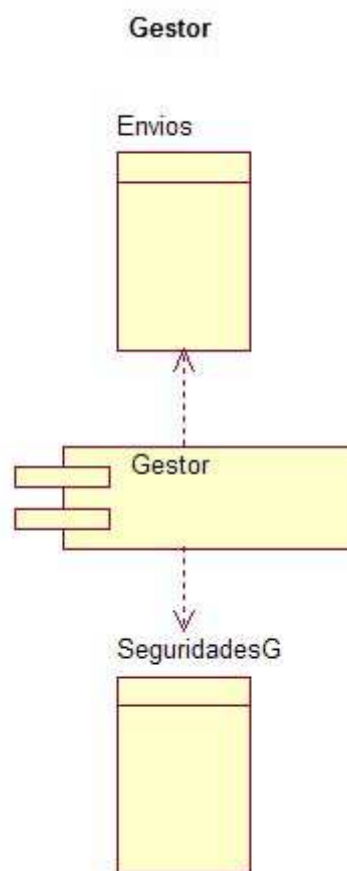


Figura 5. Componente Gestor

ANEXO III

DIAGRAMA DE CLASES

Diagrama de Clases

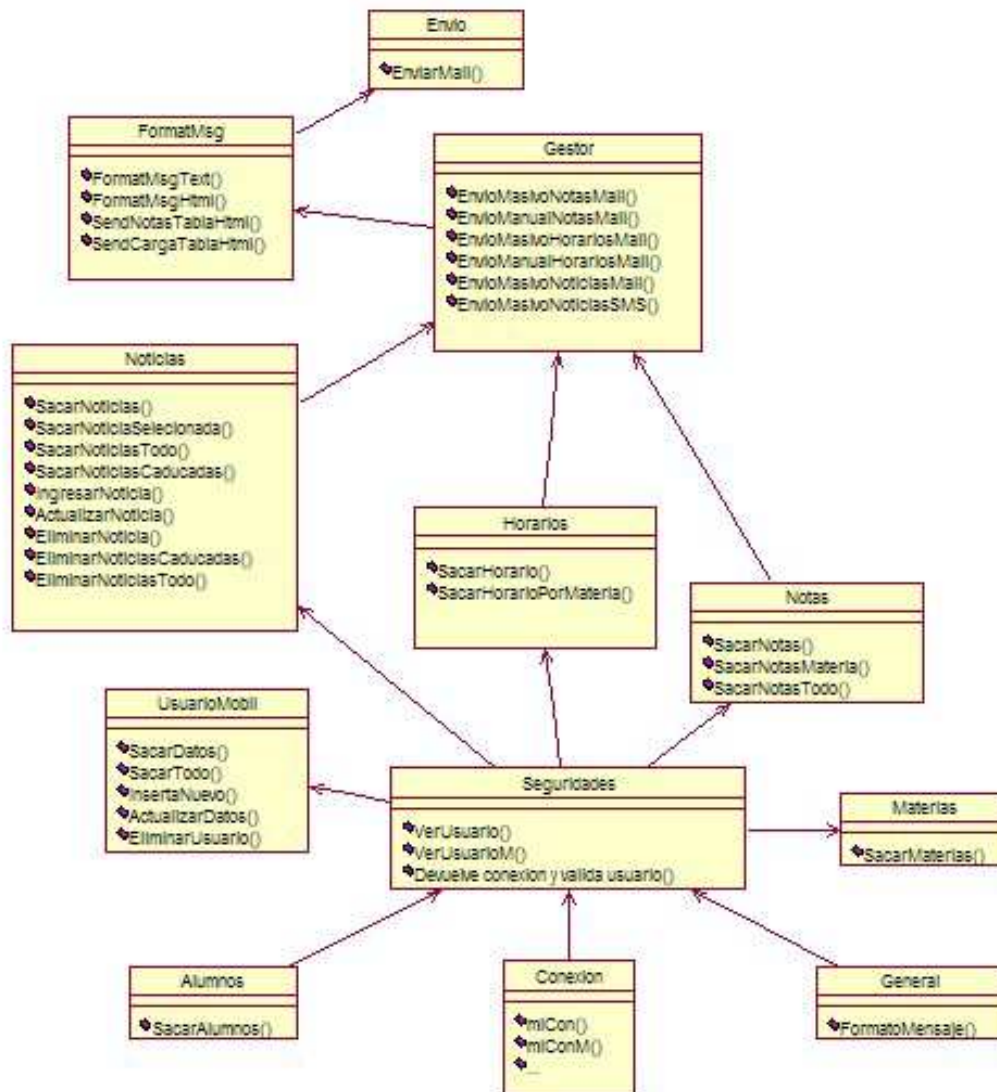


Figura 6. Diagrama de Clases

ANEXO IV

DIAGRAMA DE DESPLIEGUE

Diagrama de despliegue

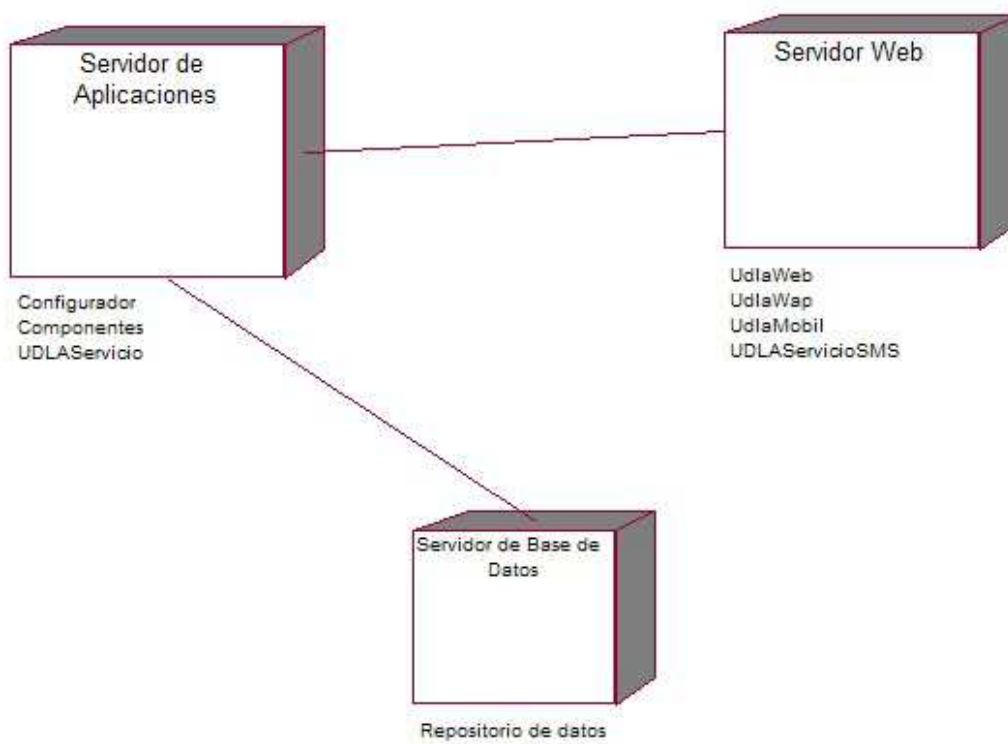


Figura 7. Diagrama de Despliegue

ANEXO V

DIAGRAMA DE SECUENCIA

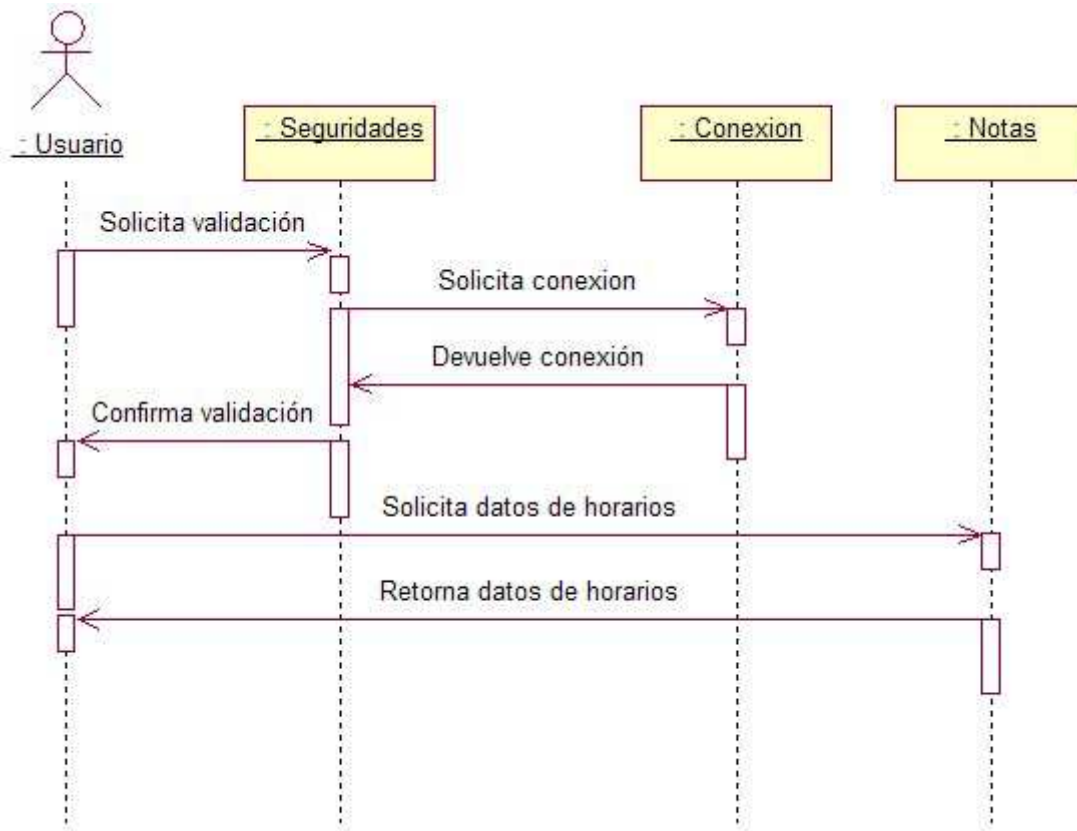


Figura 8. Consulta de Horarios

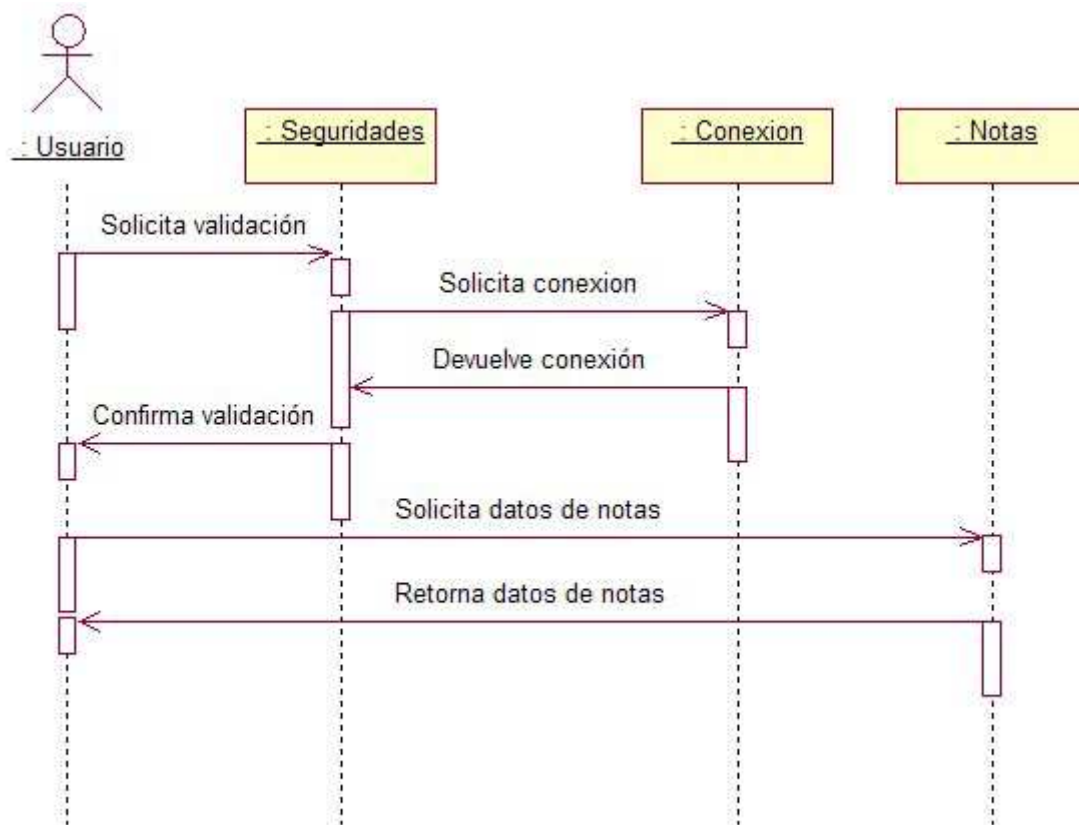


Figura 9. Consulta de Notas

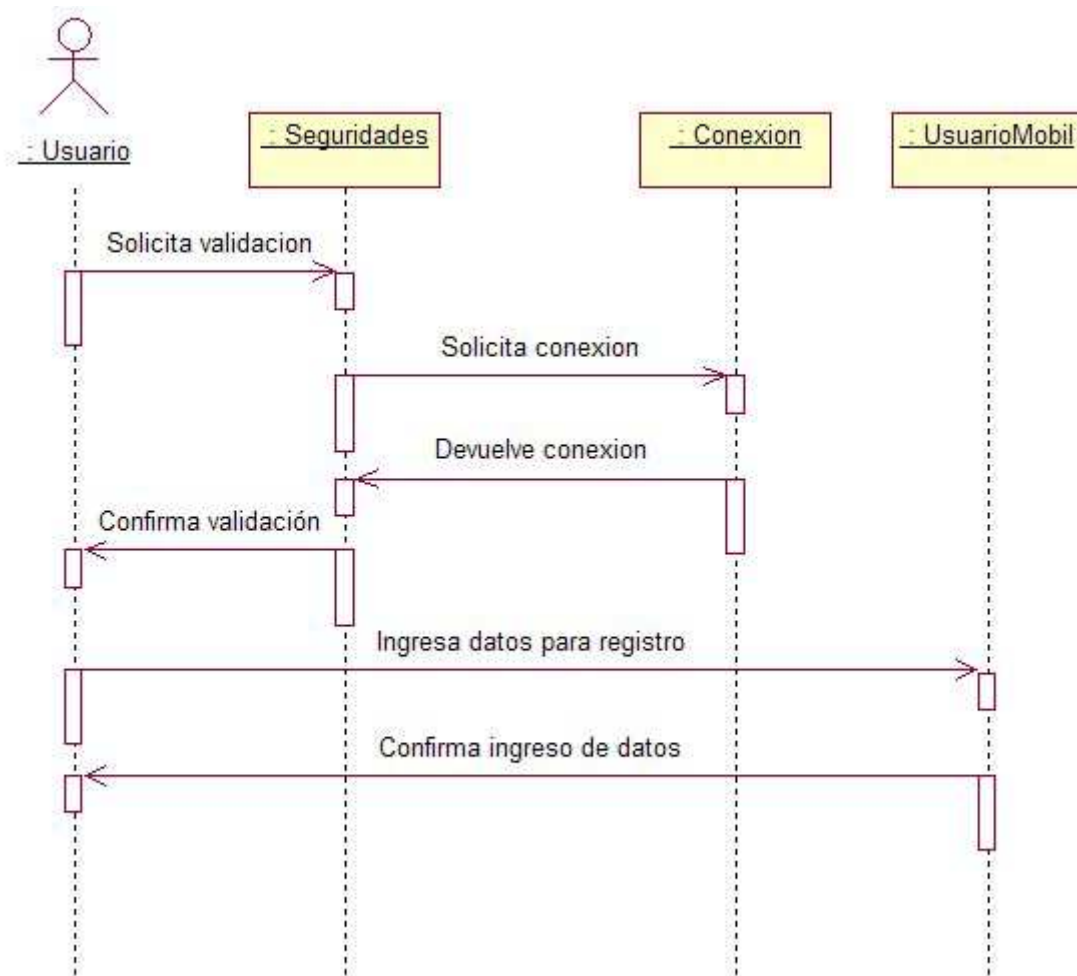


Figura 10. Ingreso de Usuario Nuevo

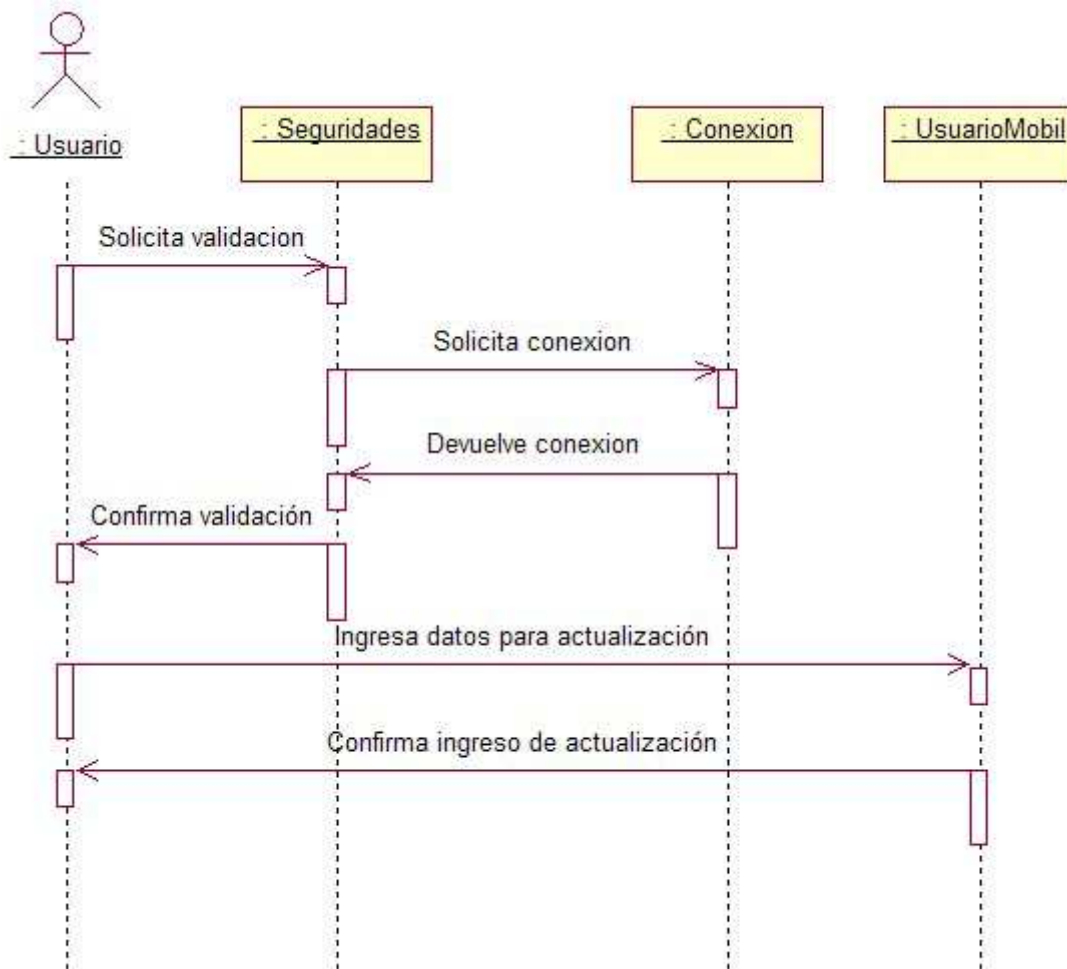


Figura 11. Actualización de Usuario

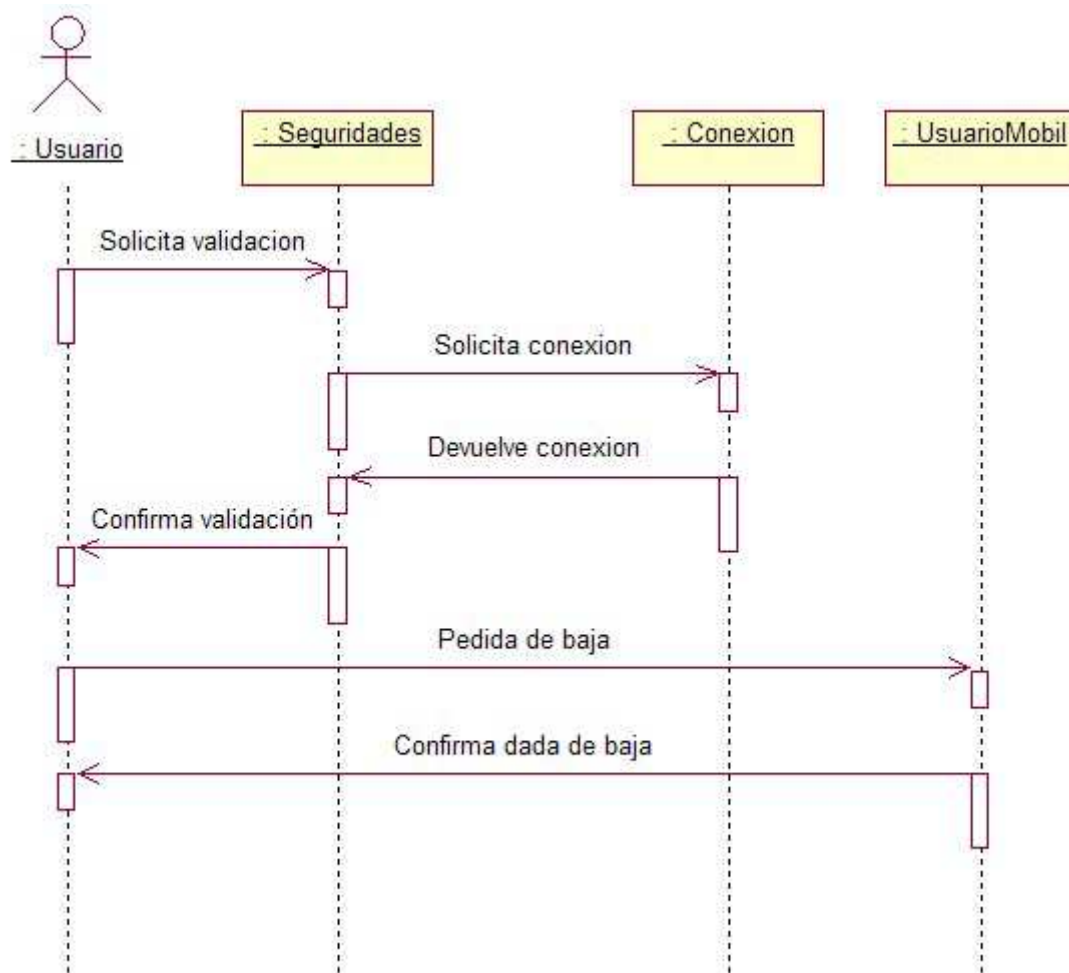


Figura 12. Eliminación de Usuario

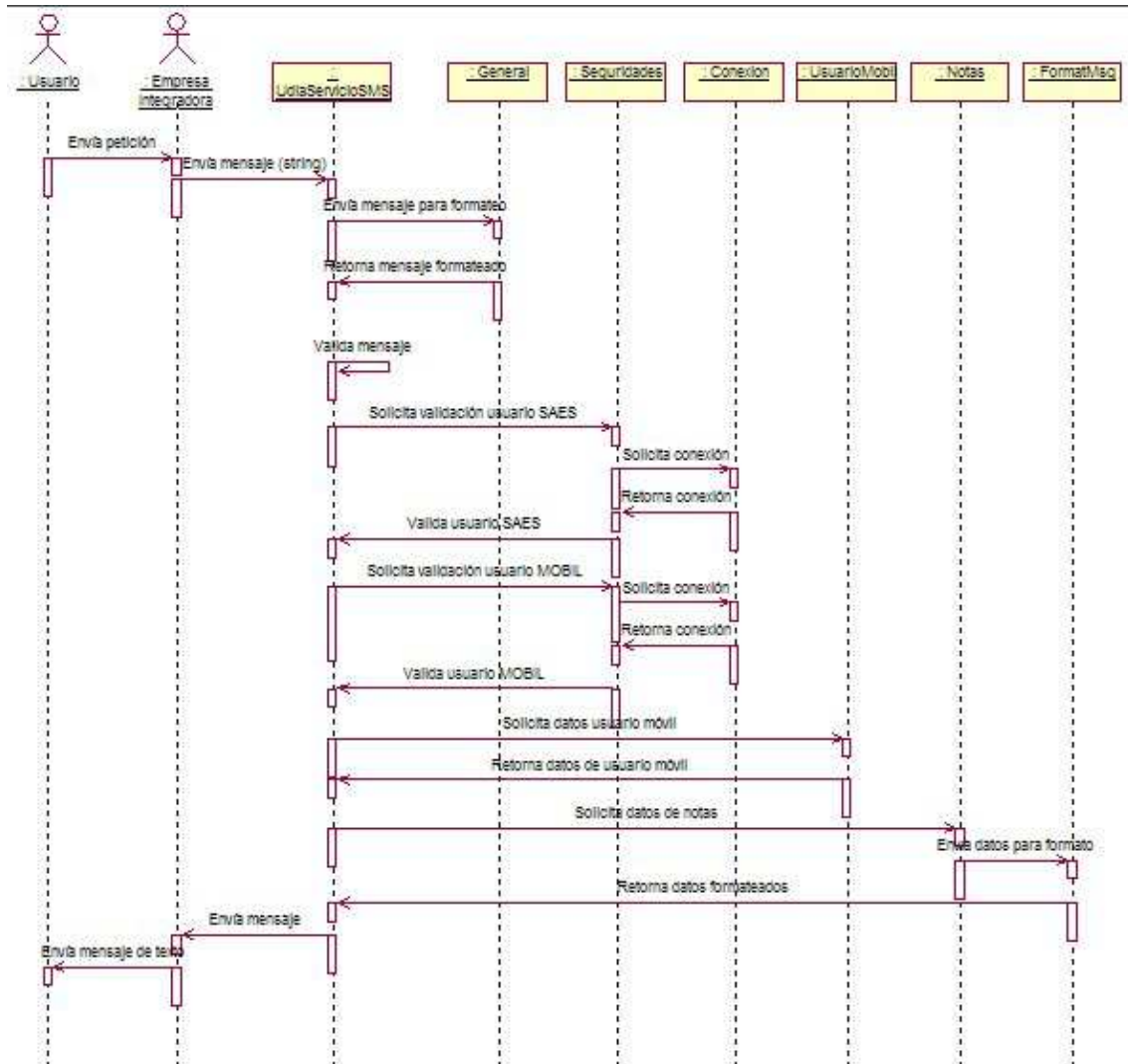


Figura 13. Consulta de Notas SMS

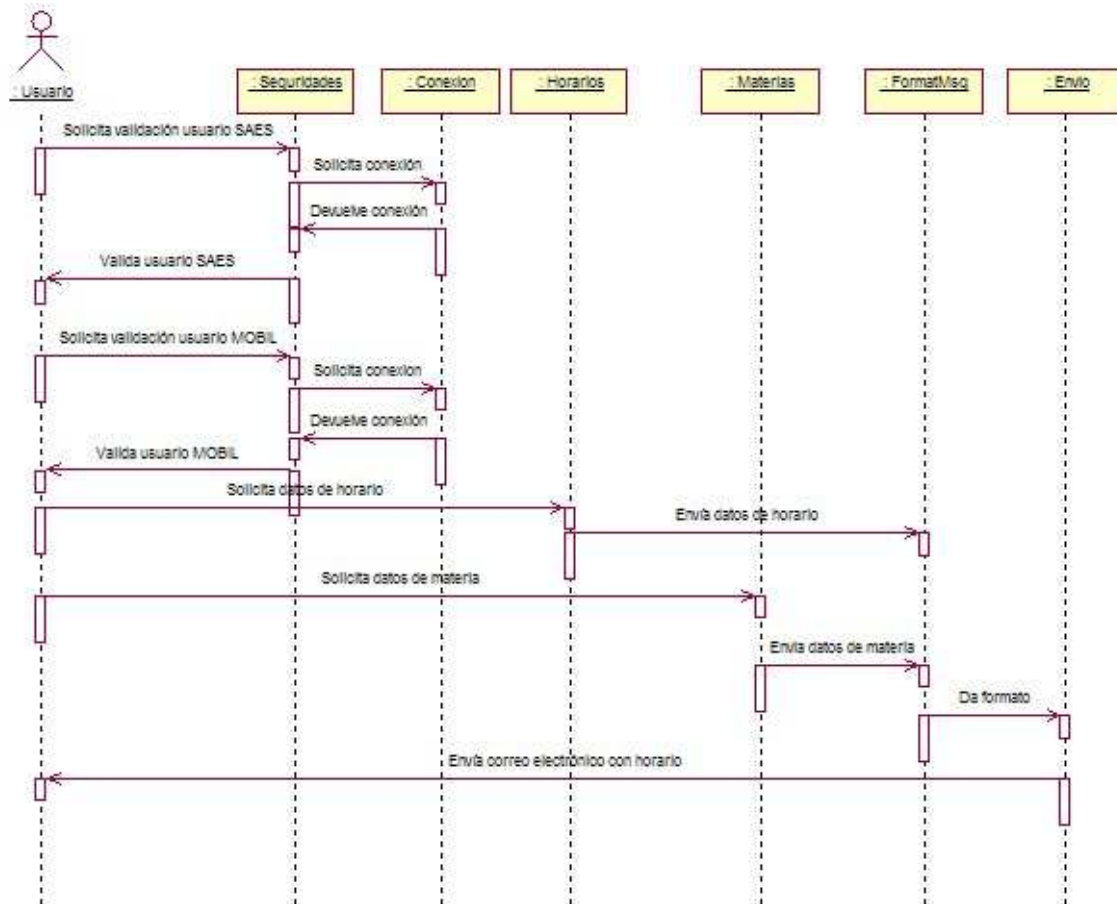


Figura 14. Envío Manual de Horario vía E-mail

ANEXO VI

DIAGRAMA DE ACTIVIDAD

Diagrama de actividad
Consulta de Notas

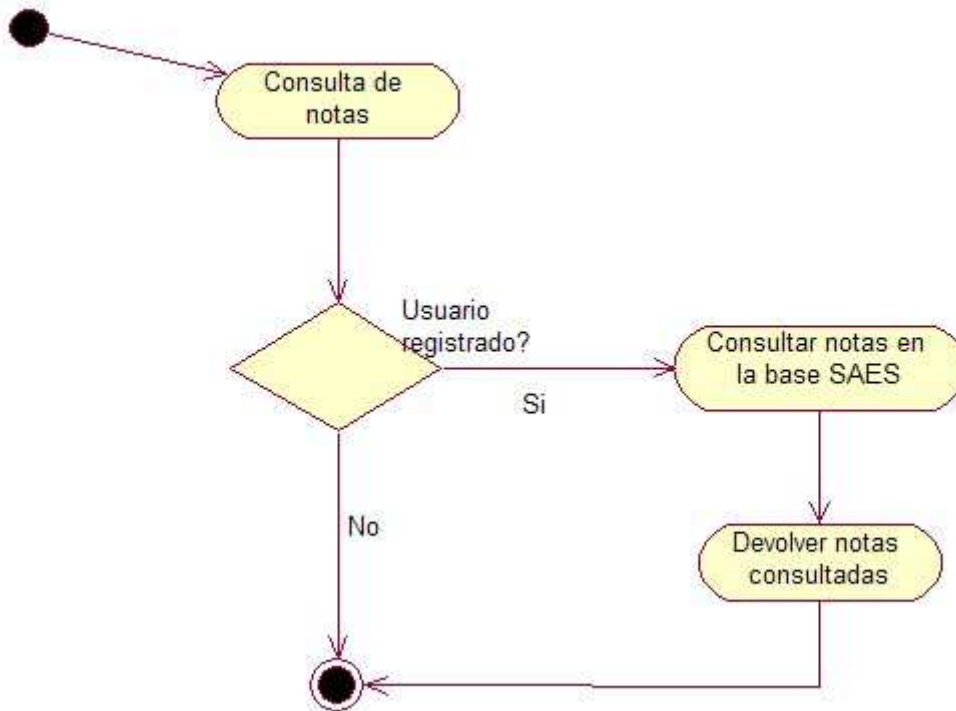


Figura 15. Consulta de Notas

Diagrama de actividad
Consulta de notas SMS

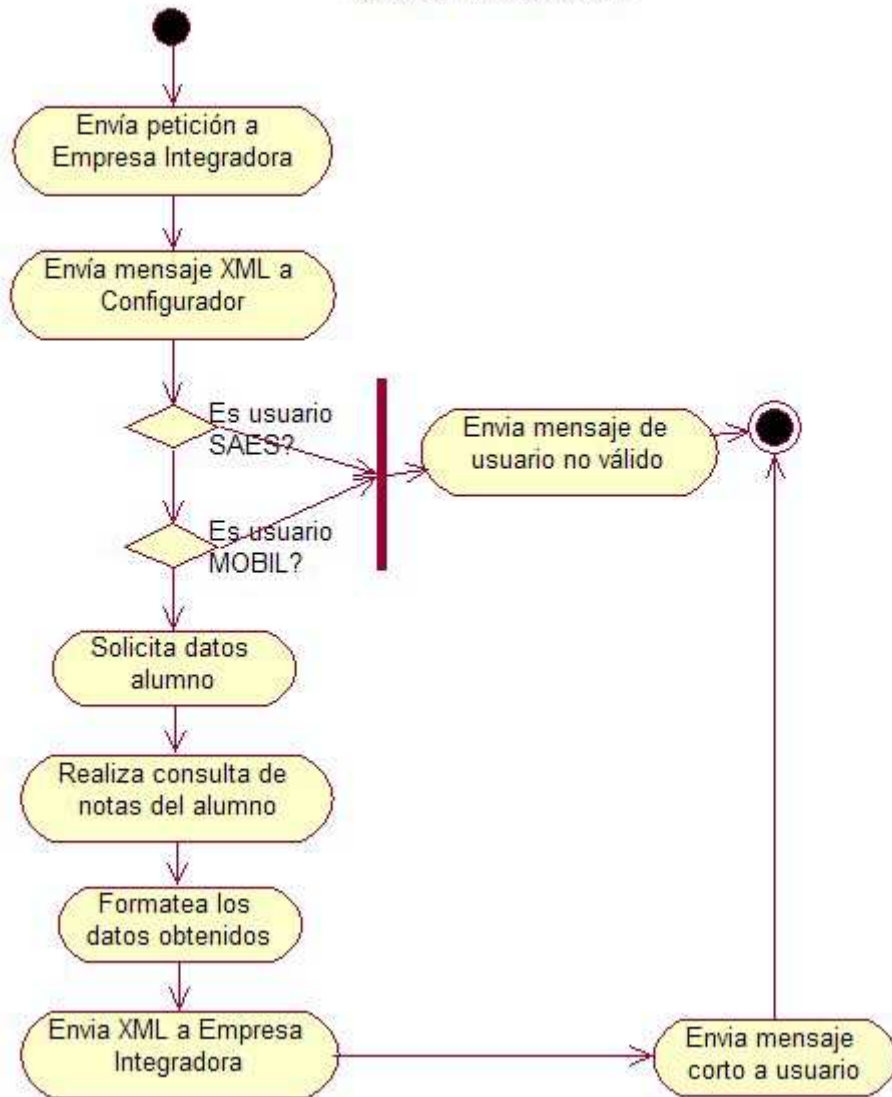


Figura 16. Consulta de Notas vía SMS

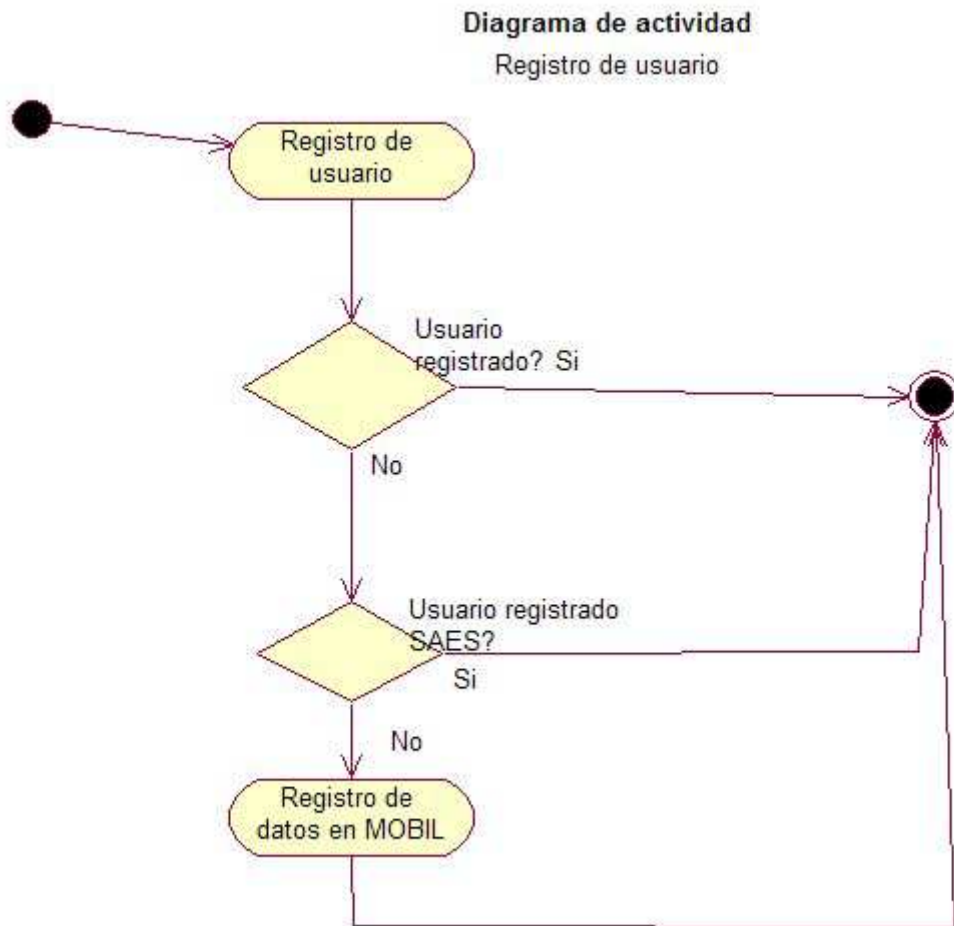


Figura 17. Registro de Usuario Nuevo

ANEXO VII

DIAGRAMA DE CASOS DE USO

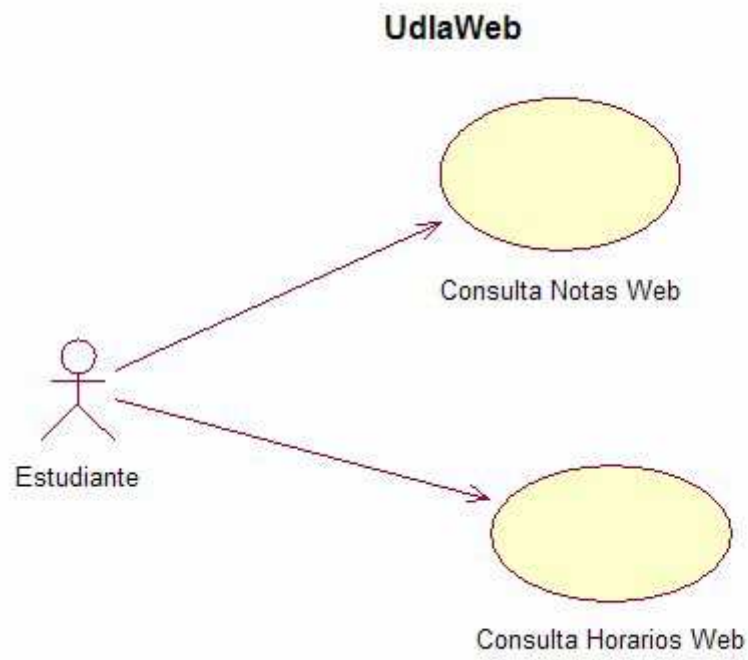


Figura 18. Caso de Uso de UdlaWeb

Configurador

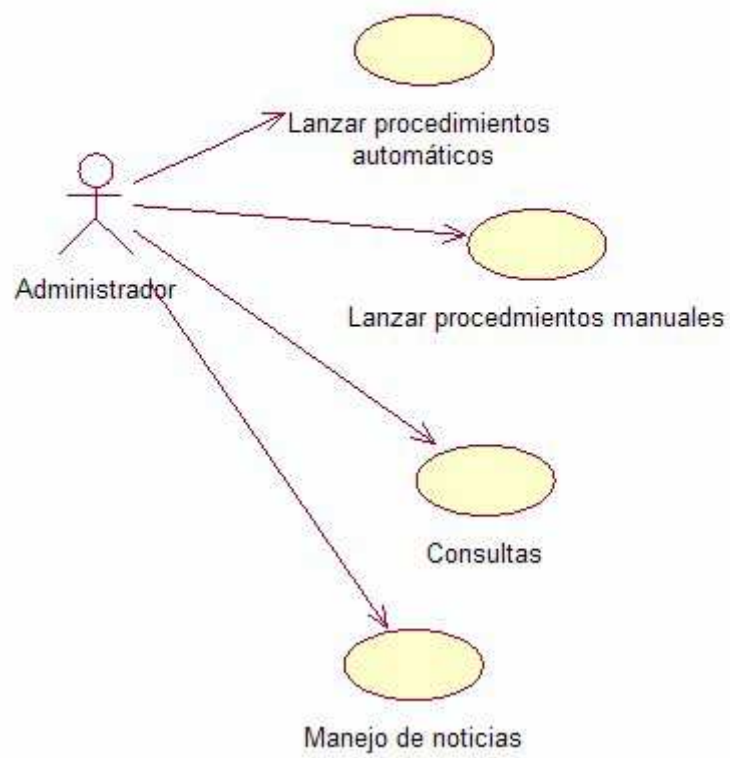


Figura 19. Caso de Uso de Configurador

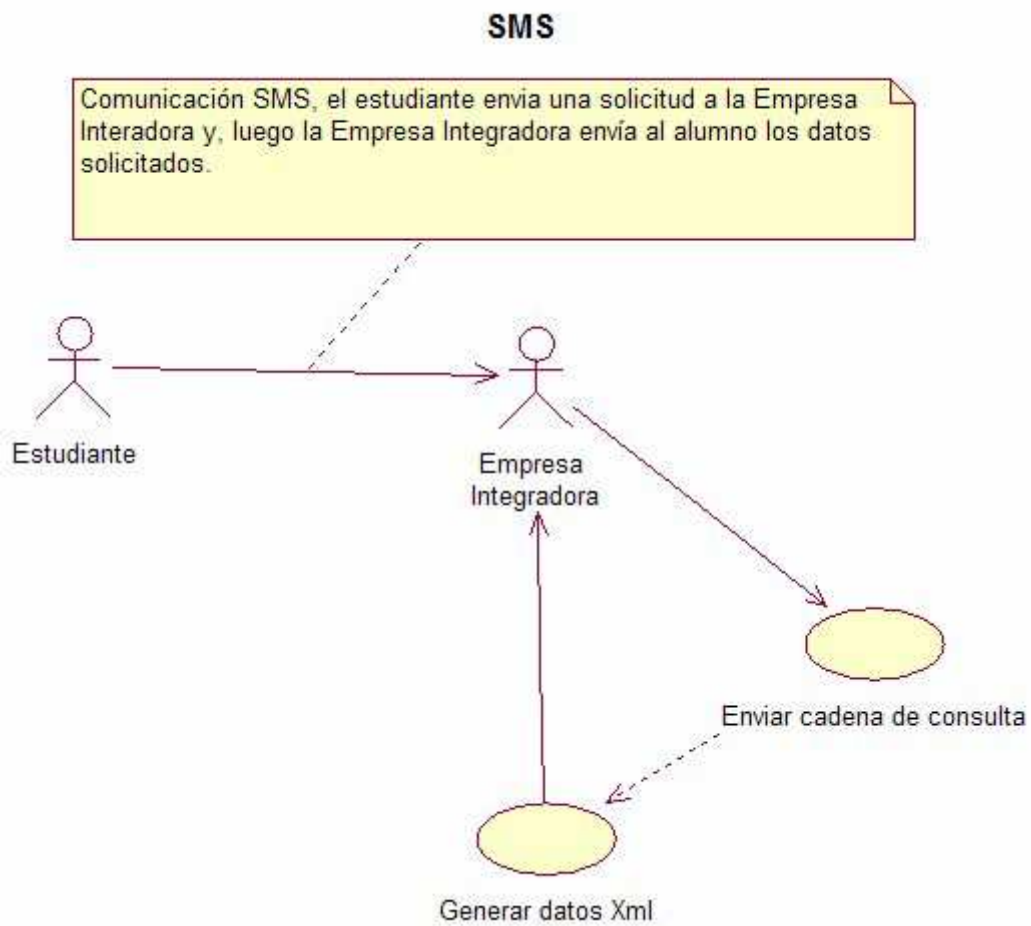


Figura 20. Caso de Uso de SMS

UdlaMovil

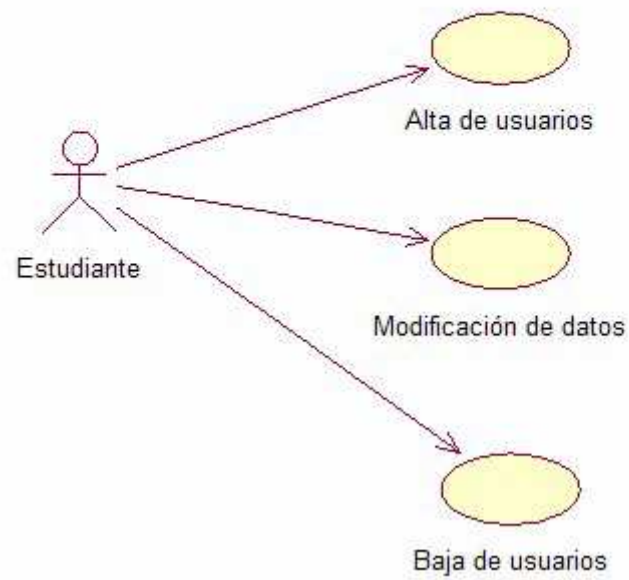


Figura 21. Caso de Uso de UdlaMovil

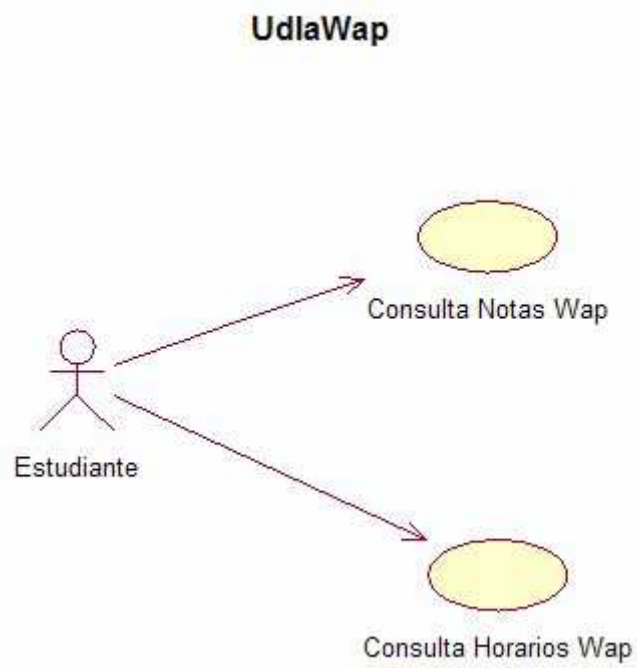


Figura 22. Caso de Uso de UdlaWap

ANEXO VIII

PRUEBAS FUNCIONALES

INDICE

PRUEBAS FUNCIONALES	84
APLICACIÓN UDLAWEB	84
Objetivo	84
Objetivo Específico	84
Conclusión	92
APLICACIÓN UDLAWAP	93
Objetivo	93
Objetivo Específico	93
Conclusión	97
APLICACIÓN WINDOWS	98
Objetivo	98
Objetivo Específico	98
MODULO CONFIGURACIÓN DE PARÁMETROS	98
MODULO PROCEDIMIENTOS MANUALES	104
Conclusión	106

PRUEBAS FUNCIONALES

APLICACIÓN UDLAWEB

Objetivo.- Prueba Funcional de Consulta de Notas

Objetivo Específico.- Verificar que la validación de usuario se lleve a cabo en una de las Aplicaciones Web, y que los datos desplegados sean correctos.

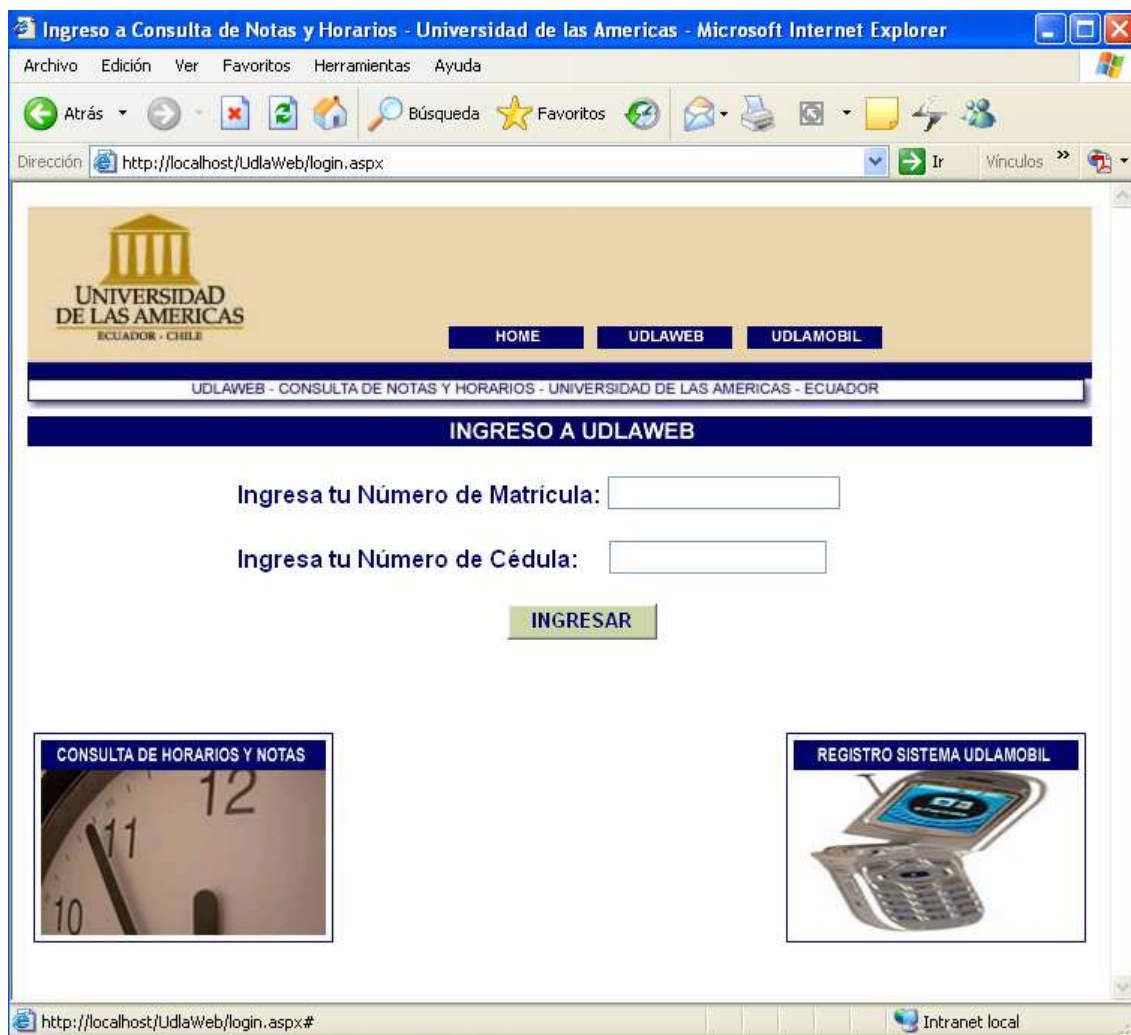


Figura 1. Página de Ingreso a UdlaWeb

En la Aplicación Udlaweb, los estudiantes de la Universidad de las Américas podrán realizar las consultas de Notas y Horarios.

Lo primero que se debe realizar es ingresar el número de matrícula del estudiante en el primer campo, y el número de cédula en el segundo campo, y a continuación presionar el botón INGRESAR.

Si el usuario omite estos campos, ya sea ambos o cualquiera de los dos, y presiona el botón INGRESAR, se presentará un mensaje de que los campos son obligatorios, y no se podrá avanzar a la siguiente página.

Microsoft Internet Explorer
Archivo Edición Ver Favoritos Herramientas Ayuda
Dirección http://localhost/UdlaWeb/login.aspx
Intranet local

UNIVERSIDAD DE LAS AMERICAS
ECUADOR - CHILE

HOME UDLAWEB UDLA MOBIL

UDLAWEB - CONSULTA DE NOTAS Y HORARIOS - UNIVERSIDAD DE LAS AMERICAS - ECUADOR

INGRESO A UDLAWEB

Ingresa tu Número de Matrícula: Obligatorio

Ingresa tu Número de Cédula: Obligatorio

INGRESAR

CONSULTA DE HORARIOS Y NOTAS

REGISTRO SISTEMA UDLA MOBIL

Figura 2. Campos Obligatorios

Una vez ingresados estos dos campos y habiendo presionado el botón INGRESAR, se procede a verificar los mismos, y validar al usuario. En el caso de que no sea un usuario válido, es decir no sea un estudiante de la Universidad de las Américas, se presentará un mensaje de Usuario no válido.

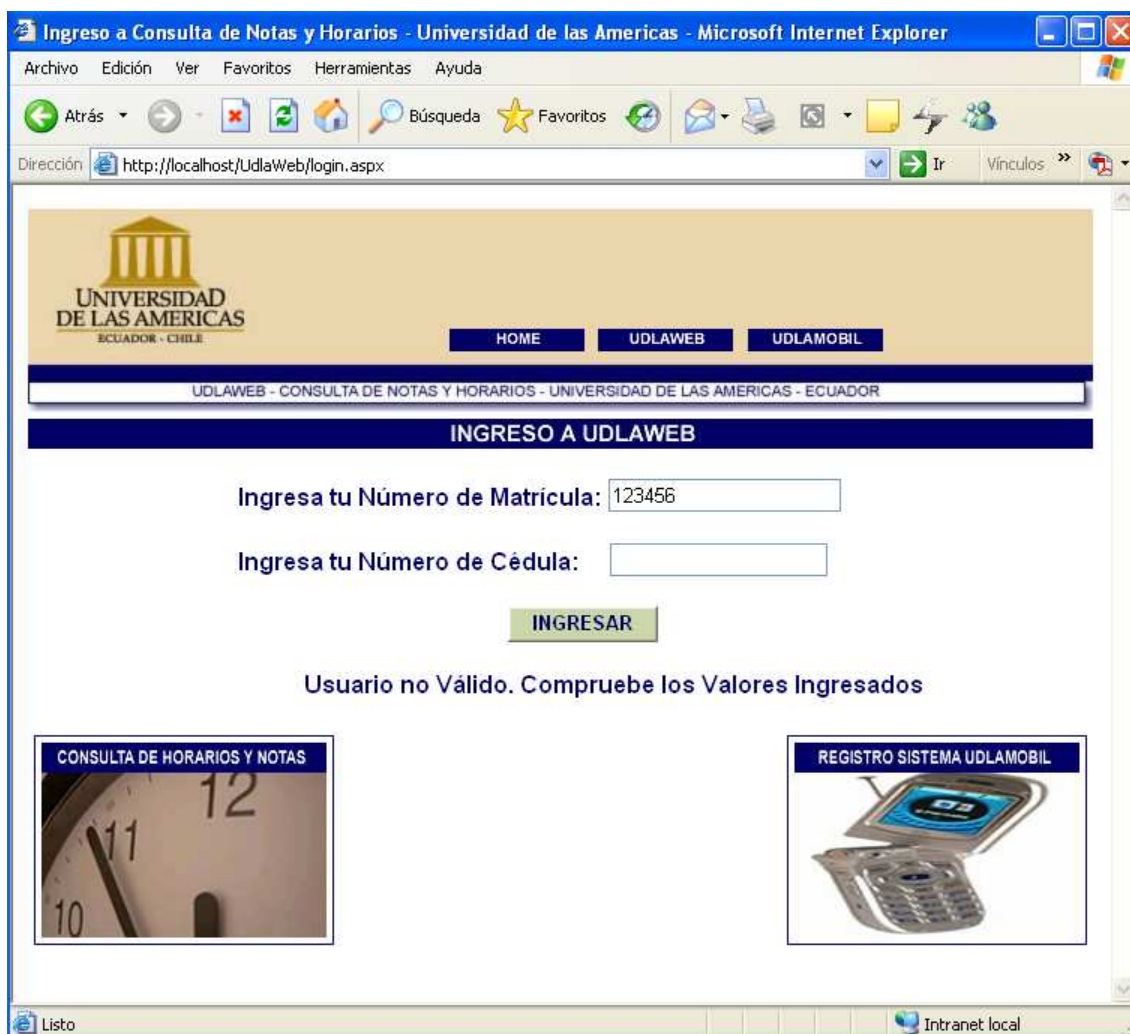


Figura 3. Usuario no válido

Si los valores ingresados corresponden a un usuario válido, se podrá pasar a la página siguiente, en la cual se necesitará que el usuario escoja el año y el periodo lectivo del cual se tomará la información requerida. Los periodos lectivos disponibles son 1er Semestre, que corresponde al semestre comprendido entre el mes de Octubre y el mes de Febrero, 2do Semestre, que corresponde al semestre comprendido entre los meses de Marzo y Julio, y por último el periodo de Verano, que corresponde a los meses de Agosto y Setiembre.

Una vez seleccionados estos valores se debe presionar el hipervínculo Aceptar, para poder pasar a la siguiente página.

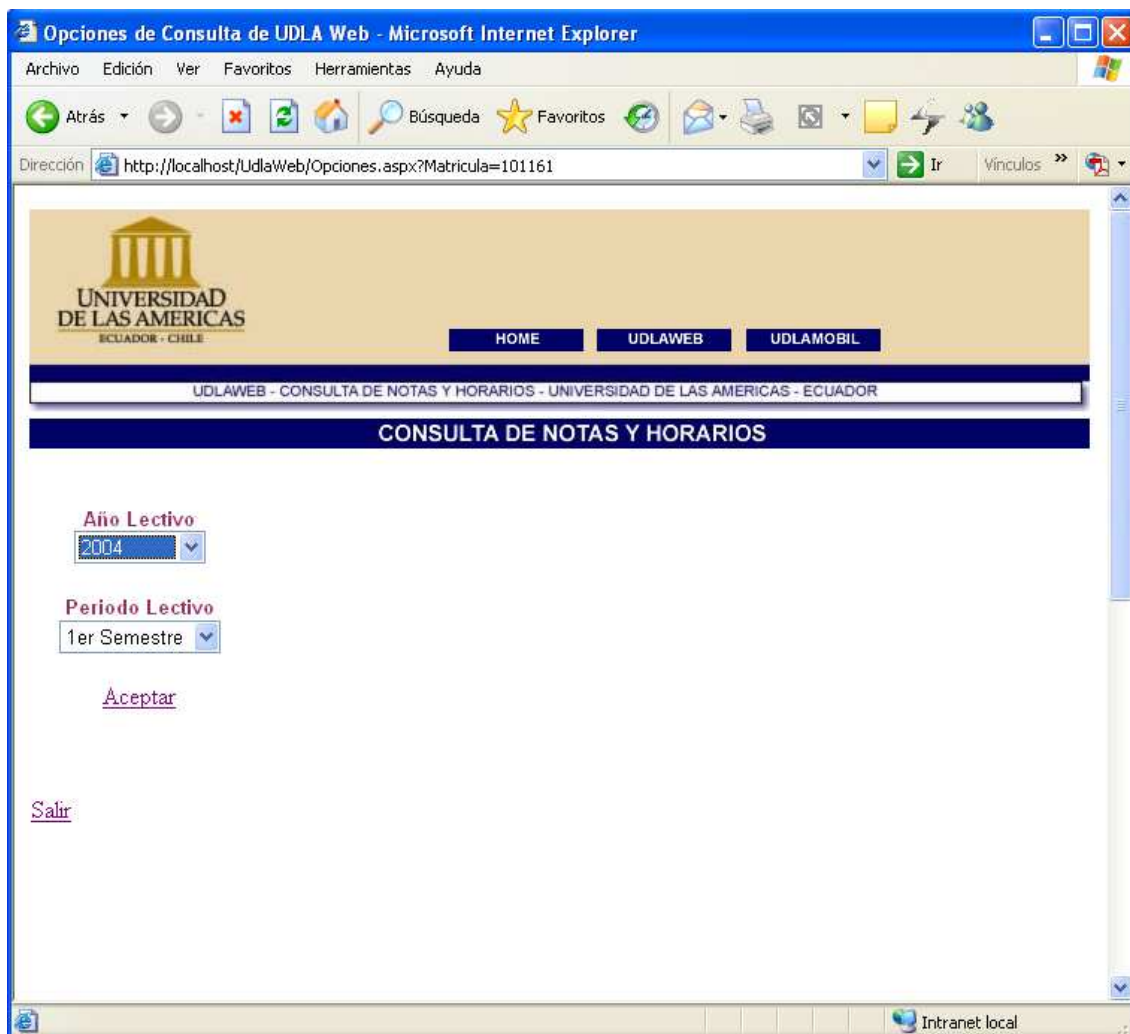


Figura 4. Parámetros de Consulta

A continuación se presentarán las opciones de consulta que se encuentran disponibles para el usuario.

Estas opciones son:

- Notas y
- Horarios

Para nuestro caso seleccionaremos la opción de Notas.

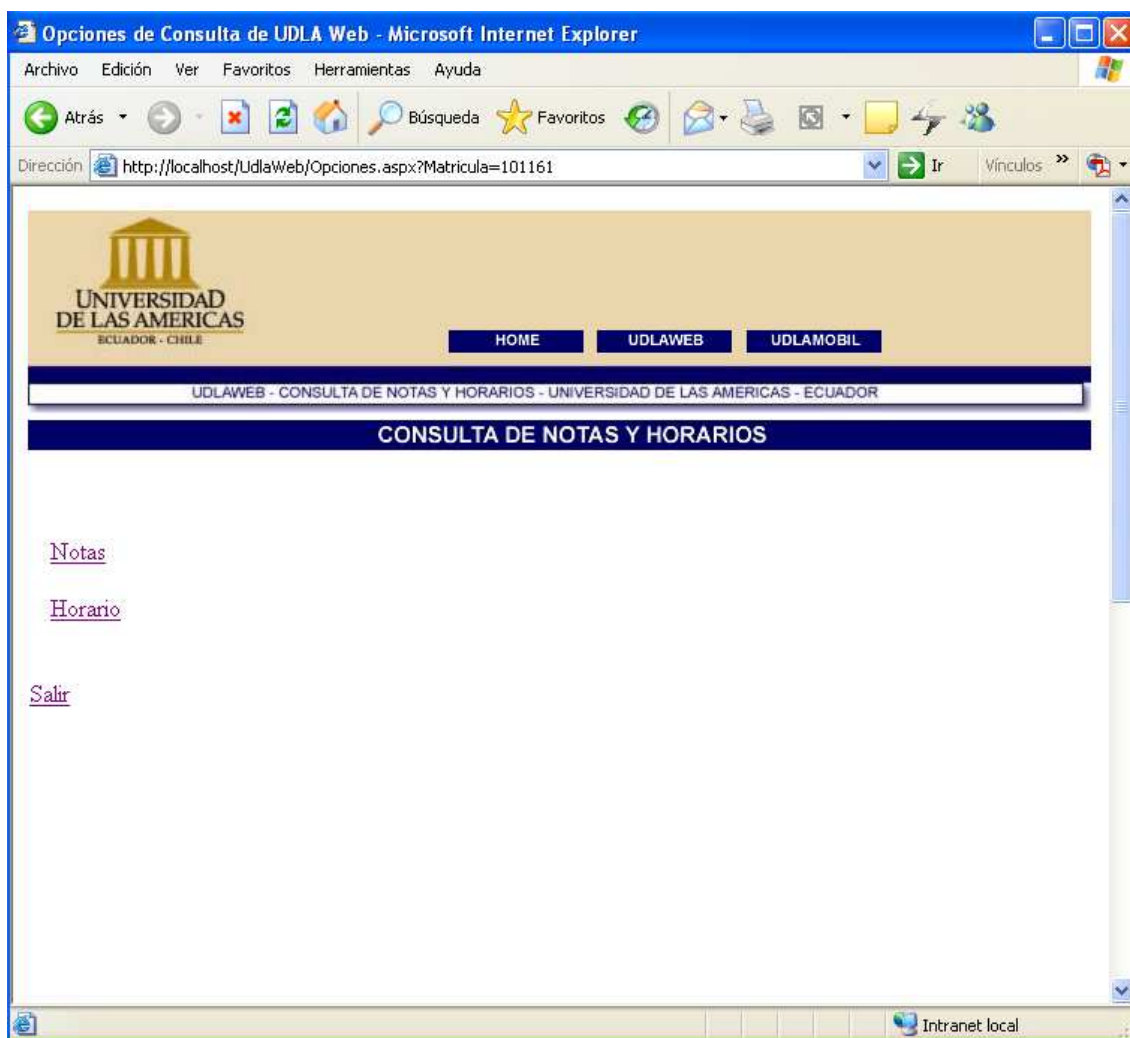


Figura 5. Opciones de Consulta

Habiendo seleccionado la opción de Notas, se presentarán varias opciones con el fin de facilitar la búsqueda al usuario, estas opciones son las siguientes:

- Por Materia
- Cátedra 1
- Cátedra 2
- Controles
- Examen Final
- Recuperación
- Todas las Notas

En primera instancia seleccionaremos la opción de Por Materia.

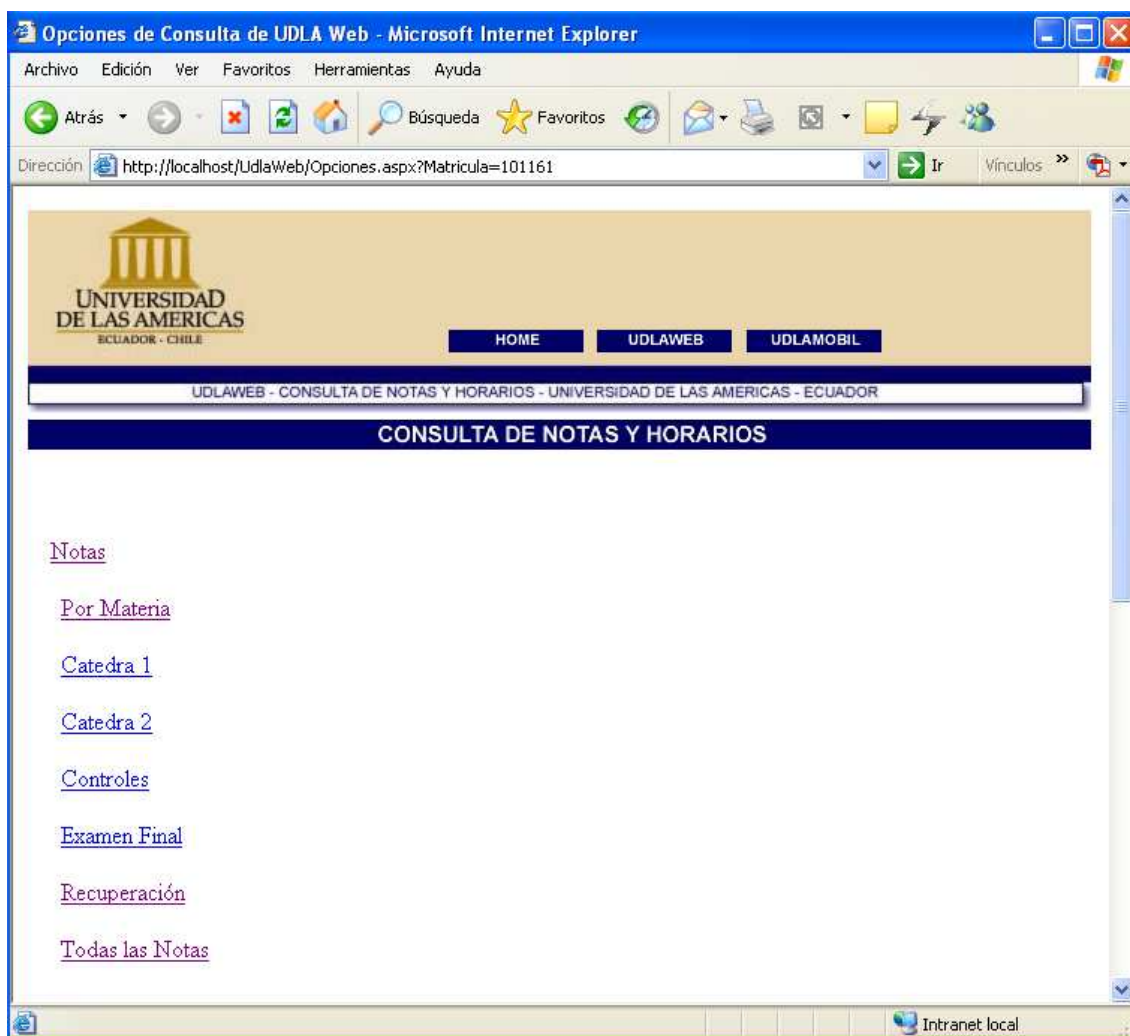


Figura 6. Opciones de Notas

Al seleccionar esta opción, se desplegarán todas las materias que el estudiante que está realizando la consulta tenga registradas como carga académica en la Universidad de las Américas en concordancia con el año y el período seleccionado con anterioridad.

Aquí el usuario deberá seleccionar una de las materias que se le presente, para que se despliegue las notas de la opción seleccionada.

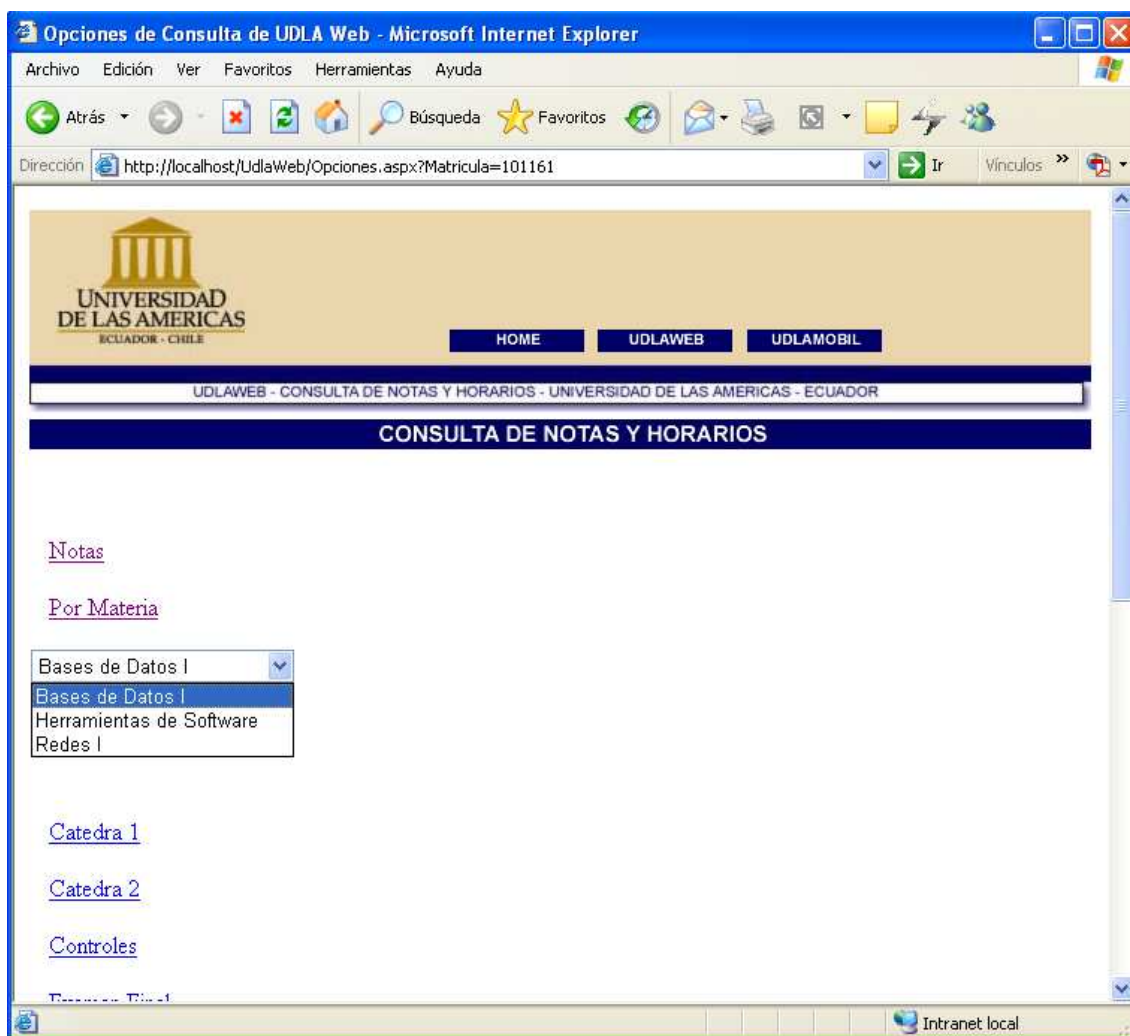


Figura 7. Despliegue de materias disponibles

Una vez seleccionada la materia de la que se desea consultar las notas, se presentará la página en la cual el usuario podrá visualizar las notas de la opción escogida, en forma de una tabla, en la cual se describe a que periodo corresponde la Nota, que puede ser Cátedra 1, Cátedra 2, Controles, Examen Final o Recuperación, así como el valor de la nota en si.

Una vez realizada la consulta el usuario tiene la opción de realizar otra, o salir de UdlaWeb, lo cual lo puede hacer mediante un botón de Salir que se encuentra ubicado en la parte inferior de la tabla en la que se despliegan los datos, o también mediante el hipervínculo Salir, situado en la parte Inferior izquierda de la Pagina.

The screenshot shows a Microsoft Internet Explorer window titled 'Opciones de Consulta de UDLA Web'. The address bar shows 'http://localhost/UdlaWeb/Opciones.aspx?Matricula=101161'. The page header includes the Universidad de las Américas logo and navigation buttons for 'HOME', 'UDLAWEB', and 'UDLAMOBIL'. Below the header, the page title is 'UDLAWEB - CONSULTA DE NOTAS Y HORARIOS - UNIVERSIDAD DE LAS AMERICAS - ECUADOR'. The main content area is titled 'CONSULTA DE NOTAS Y HORARIOS' and displays the following information:

[Notas](#)
[Horario](#)
[Salir](#)

Matricula:101161 **Cedula:**501828792 **Alumno:**Guillermo Vargas Alvarez

Periodo	Nota
Controles	9

Below the table is a 'Salir' button. The status bar at the bottom shows 'Listo' and 'Intranet local'.

Figura 8. Despliegue de Notas por Materia

Por otro lado si se hubiera escogido la opción de Notas – Todas las Notas, se hubiera obtenido todas las notas que se encontraran disponibles y para todas las materias que tuviere registradas, en una tabla en la que se describe, la Materia a la cual pertenece la nota, al periodo al que pertenece la nota y el valor de la nota.

The screenshot shows a Microsoft Internet Explorer window titled 'Opciones de Consulta de UDLA Web - Microsoft Internet Explorer'. The address bar shows the URL: `http://localhost/UdlaWeb/Opciones.aspx?Matricula=101161`. The page content includes the university logo and navigation buttons for 'HOME', 'UDLAWEB', and 'UDLAMOBIL'. Below this, a banner reads 'UDLAWEB - CONSULTA DE NOTAS Y HORARIOS - UNIVERSIDAD DE LAS AMERICAS - ECUADOR'. The main heading is 'CONSULTA DE NOTAS Y HORARIOS'. On the left, there are links for 'Notas', 'Horario', and 'Salir'. The main content area displays student information: 'Matricula:101161 Cedula:501828792 Alumno:Guillermo Vargas Alvarez'. Below this is a table with the following data:

Materia	Periodo	Nota
Herramientas de Software	Recuperación	8,7
Redes I	Controles	9

At the bottom of the table area, there is a 'Salir' button. The status bar at the bottom of the browser shows 'Listo' and 'Intranet local'.

Figura 9. Despliegue de Todas las Notas

Conclusión

Por lo presentado, se puede concluir que la Aplicación Web UdlaWeb, está realizando en forma correcta la validación de usuarios y que está presentando de buena forma la información de notas necesitada y solicitada por el usuario. Además se puede ver que se le presenta diferentes opciones a las que puede acceder de acuerdo a su conveniencia.

APLICACIÓN UDLAWAP

Objetivo.- Prueba Funcional de Consulta de Horarios

Objetivo Específico.- Verificar que la validación de usuario se lleve a cabo en la Aplicación WAP, y que los datos desplegados sean correctos.



Figura 10. Página de Ingreso a Udlawap

En la aplicación Udlawap, los estudiantes de la Universidad de las Américas, podrán realizar consultas de Notas y Horarios, sin necesidad de utilizar un computador, utilizando las tecnologías que nos ofrecen los nuevos equipos y empresas de servicios celulares. Esto es que el estudiante podrá realizar sus consultas utilizando su equipo celular, siempre y cuando tenga contratado el servicio de navegación en Internet, y su equipo se lo permita.

La primera página que se le desplegará al usuario en su micro-navegador, es la correspondiente al ingreso de sus credenciales que lo certifiquen como usuario válido.

El primer campo corresponde al número de matrícula del alumno en la Universidad de las Américas, y el segundo campos es para el ingreso de su número de cédula de ciudadanía.

Si el usuario omite ambos o cualquiera de estos dos valores, no podrá pasar a la siguiente página correctamente.

Una vez ingresados los valores para estos campos, se procede con la verificación de los mismos y con la validación de usuario. Si los valores ingresados no corresponden a un usuario válido se presenta una página de Error de Ingreso.



Figura 11. Error de Ingreso

Si los datos ingresados corresponden a un usuario válido, se pasa a la siguiente página, en la cual se deberá ingresar el año y el periodo del cual se desea obtener la información. Para el campo de periodo se tiene los valores de 1 para el semestre comprendido entre Octubre y Febrero, 2 para semestre comprendido entre Marzo y Julio y 3 para el periodo de verano. A continuación se debe seleccionar la opción de Consultar ubicada en la parte inferior a continuación del campo de periodo.



Figura 12. Parámetros de Consulta

Seguidamente se presenta la página de Menú Principal, en la cual se tendrán las opciones disponibles de consulta, estas son, Notas y Horarios.



Figura 13. Menú Principal

Para este caso vamos a seleccionar la opción de horarios, después de lo cual se presenta una página en la cual se desplegarán todas las materias que hayan sido registradas para el usuario que se encuentra realizando la consulta.

Al seleccionar una de las materias, se procederá a presentarse la información de los días y las horas en las que el estudiante tomará dicha asignatura.



Figura 14. Asignaturas Disponibles para la Consulta



Figura 15. Despliegue de Horarios

Conclusión

Por lo presentado, se puede concluir que la Aplicación WAP UdlaWap, está realizando en forma correcta la validación de usuarios y que está presentando de buena forma la información de horarios necesitada y solicitada por el usuario. También se puede ver que las opciones son presentadas en una forma reducida pero entendible con claridad, con el objetivo de que pueda ser presentada en la pantalla de un equipo celular.

APLICACIÓN WINDOWS

Objetivo.- Prueba Funcional del Envío Manual de Notas

Objetivo Específico.- Verificar que el ingreso de parámetros sea válido, que los parámetros sean almacenados en un archivo y que el envío manual de Notas se realice correctamente.

MODULO CONFIGURACIÓN DE PARÁMETROS

Principal - [Configuración de Parámetros]

Opciones Ventana

Configuración General | Configuración de Horarios | Configuración de Notas Parciales | Configuración de Notas Finales | Opciones

Configuración Parámetros Generales Ingreso de Parámetros Generales

Remitente: UDLA

Servidor: GAVT

Año: 2004

Periodo: 1

Aceptar Cancelar Reiniciar Salir

Figura 16. Configuración de Parámetros Generales

En este módulo, se realiza el ingreso de todos los parámetros necesarios para realizar la configuración de los procedimientos que automáticamente realizara el Sistema Configurator. Esto es el ingreso de los parámetros para que el sistema realice automáticamente el envío de Notas y Horarios a los estudiantes que se hayan registrado previamente en la Página Web de la aplicación UdlaMobil.

La primera pestaña que tenemos es la de Configuración General, en donde se realiza el ingreso de el nombre que tomará el remitente de los correos electrónicos que se envíen a los estudiantes que en este caso será UDLA, el nombre del servidor de Correo Electrónico, que en este caso es GAVT, y por último el año y el periodo del cual se va a tomar la información referente a Notas y Horarios.

Una vez ingresados estos valores, se deberá presionar el botón aceptar, si los datos son correctos, Cancelar si no se está de acuerdo con lo ingresado, con el fin de que se vacíen los campos ingresados y se pueda ingresar otros o salir si se desea abandonar este módulo.

Al presionar el botón aceptar, aparece una pantalla en la cual se pide la confirmación de almacenaje de los parámetros en variables locales.

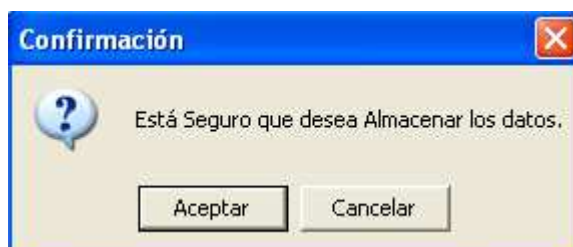


Figura 17. Petición de Confirmación

Si se confirma, se tendrá un mensaje de que los datos han sido almacenados correctamente, caso contrario, si se no se confirma, los valores no se almacenarán en sus respectivas variables.

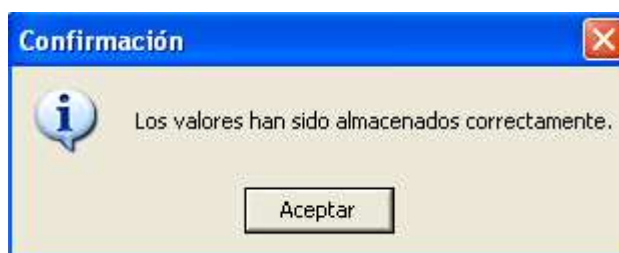


Figura 18. Mensaje de Confirmación de Almacenaje

La segunda pestaña que se tiene, es la de Configuración de Horarios, en la cual se pide el ingreso de el dato que corresponderá al Asunto de los correos electrónicos que se envíen a los estudiantes, de la fecha en la cual deberán ser enviados así como la hora en el que iniciará este proceso de envío.

Las siguientes pestañas de Configuración de Notas Parciales y Configuración de Notas Finales, siguen el mismo esquema de esta pestaña, teniendo como parte de sus valores el Asunto, las fechas y las horas en las que se realizarán sus respectivos envíos. Esto se lo hace ya que el Sistema Configurator, deberá gestionar automáticamente el envío de información a todos los estudiantes que hayan sido registrados, en las fechas y horas ingresadas aquí.

Los campos de fechas y horas serán validados para un ingreso correcto de valores.

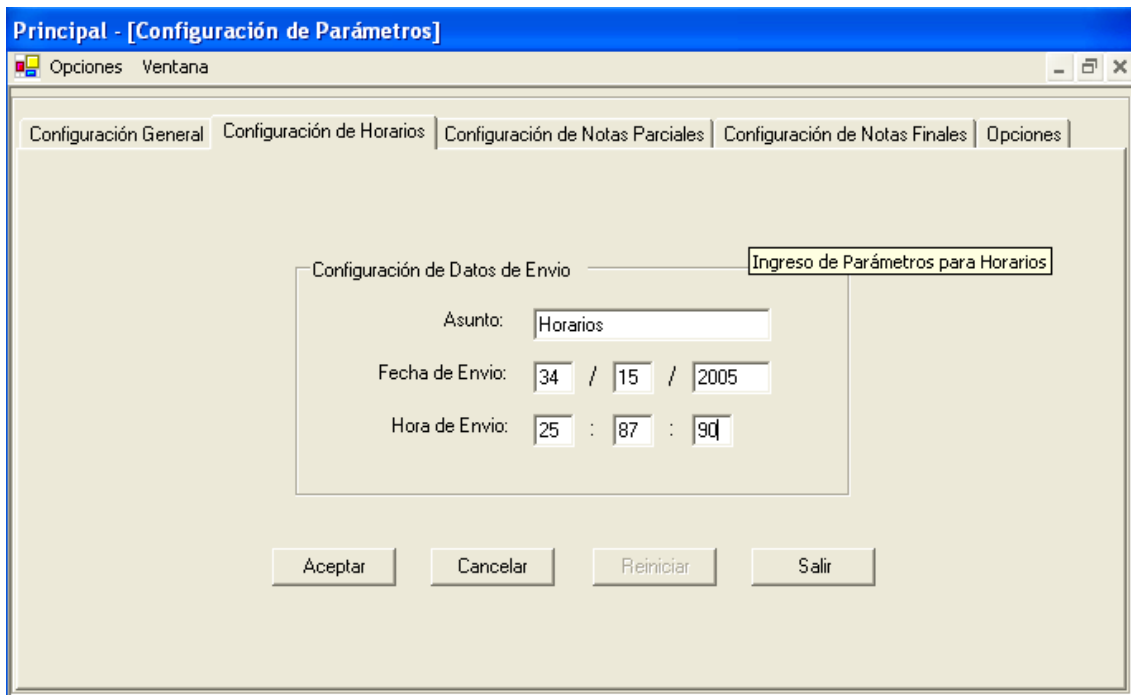


Figura 19. Configuración de Parámetros de Horarios

Si un valor de los campos para fecha y hora es ingresado incorrectamente, se presenta una pantalla de error para el primer campo en el que se lo haya encontrado y sugiere una revisión.

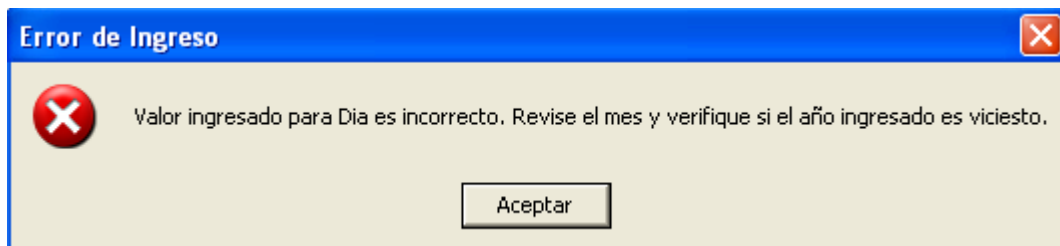


Figura 20. Error de Ingreso de Parámetros

Una vez ingresados todos los valores correctamente, se presenta un mensaje de confirmación de almacenaje de los parámetros en variables locales.

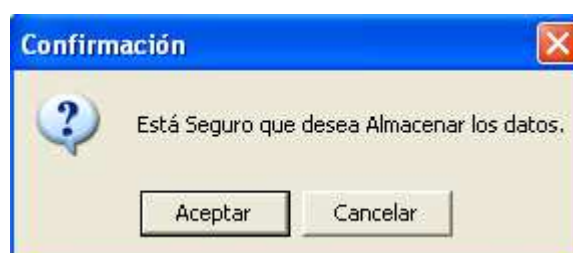


Figura 21. Petición de Confirmación

Si se confirma, se tendrá un mensaje de que los datos han sido almacenados correctamente, caso contrario, si se no se confirma, los valores no se almacenarán en sus respectivas variables.

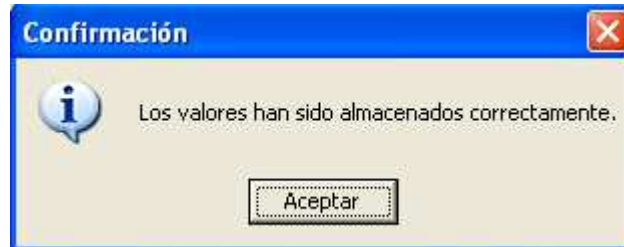


Figura 22. Mensaje de Confirmación de Almacenaje

La última opción que se tiene en el módulo de Configuración de Parámetros es la Opciones, en la cual se podrá realizar el guardar todos los parámetros de configuración en un archivo de texto o a su vez si ya se los tenía almacenados obtenerlos desde un archivo de texto.

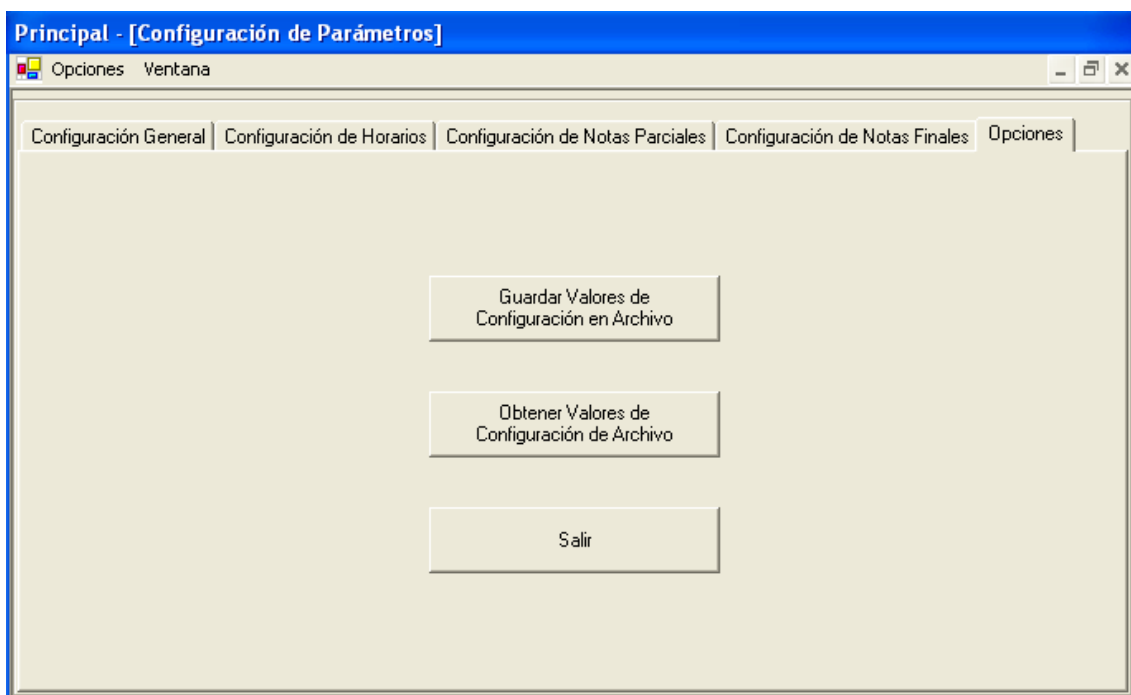


Figura 23. Opciones Adicionales

Al hacer clic en el botón Guardar Valores de Configuración en Archivo, se presenta la correspondiente ventana de Guardar para escoger la ubicación en donde vamos a almacenar el archivo y poner su respectivo nombre.

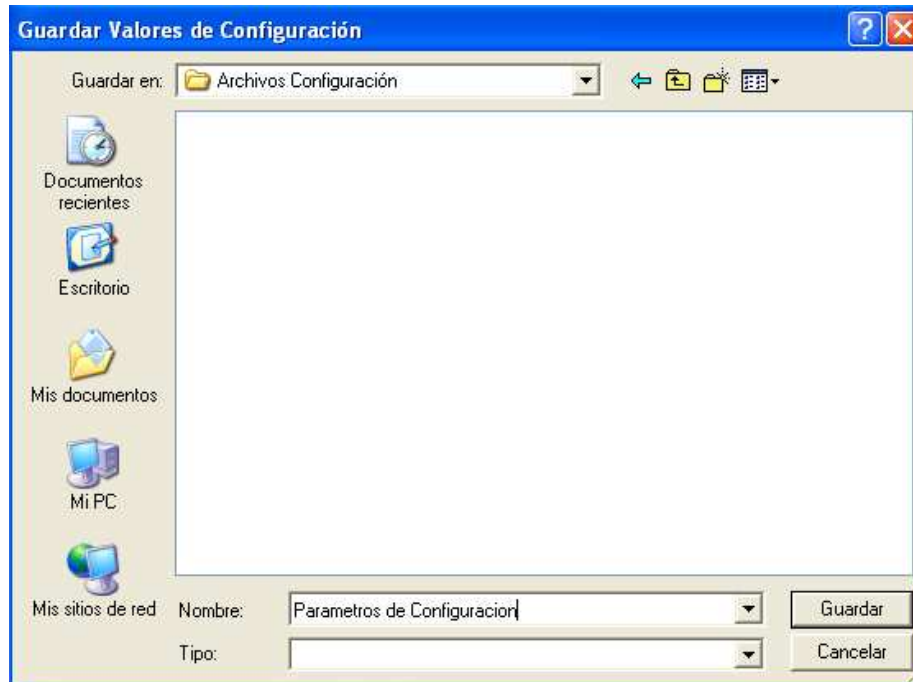


Figura 24. Guardar Valores en Archivo de Texto

Al haber realizado esta acción, se puede ubicar al archivo en la ubicación en la que se haya almacenado y verificar el correcto funcionamiento de lo antes mencionado.

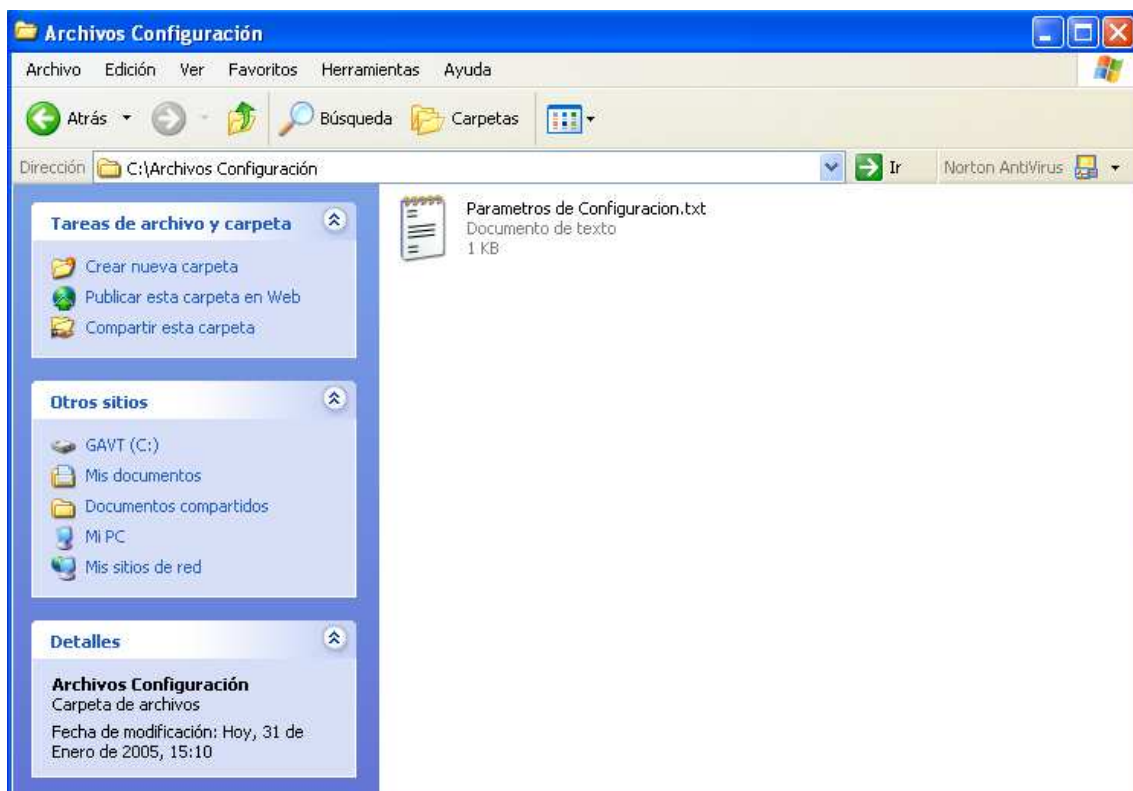


Figura 25. Comprobación de Archivo Almacenado

Una vez ubicado el archivo, se lo puede abrir para verificar que los datos almacenados sean los anteriormente ingresados.

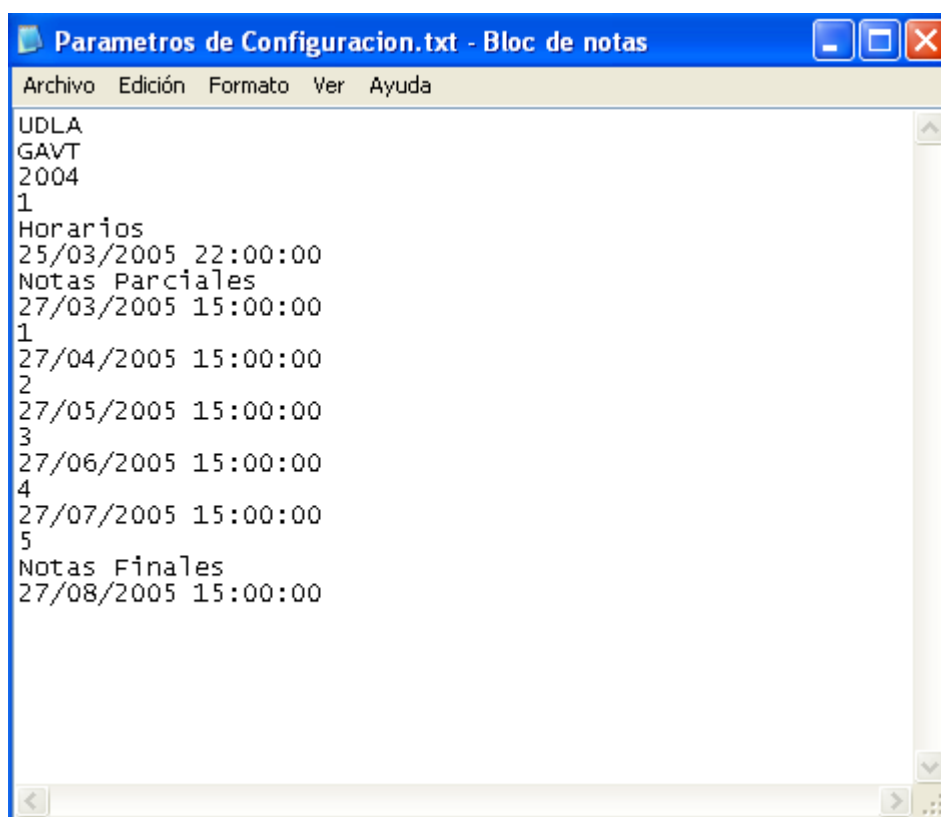


Figura 26. Valores almacenados en Archivo

Realizada la configuración de parámetros, se puede lanzar los procedimientos automáticos, para que el sistema por si solo verifique las fechas y las horas de envío y proceda con su ejecución.

MODULO PROCEDIMIENTOS MANUALES

Principal - [Procedimientos Manuales]

Opciones Ventana

Datos del Alumno

Ingrese el Número de Matrícula: 101161

Ingrese el Número de Identificación: 0501828792

Aceptar

Otro Alumno Salir

Opciones Manuales

Envio Manual de Horarios via mail

Envio Manual de Notas Parciales via mail

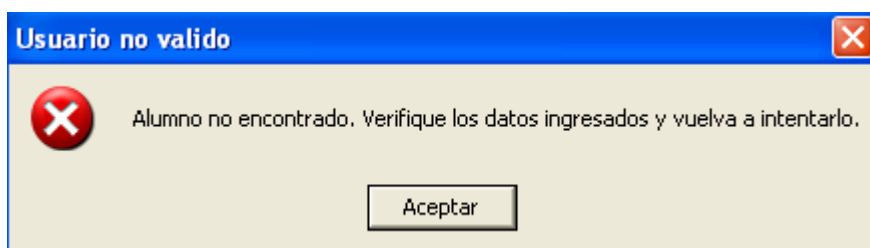
Envio Manual de Notas Totales via mail

Figura 27. Procedimientos Manuales de Notas y Horarios

Este módulo está diseñado para atender peticiones extras de envío de Notas y Horarios por parte de los alumnos de la Universidad de las Américas, cuando por diferentes motivos se requiera un envío adicional de los mismos.

Lo primero que se debe hacer aquí es ingresar el número de matrícula y el número de cédula del estudiante y presionar el botón aceptar, con lo cual el sistema verificará, primero si es un estudiante de la Universidad de las Américas y a continuación si el estudiante es un usuario registrado en Udlamobil.

Si los datos ingresados no corresponden a un estudiante de la Universidad de las Américas, se presenta un mensaje de Usuario no válido.

**Figura 28. Error de datos de Alumno**

Si los valores ingresados corresponden a un estudiante de la Universidad de las Américas, pero este no se encuentra registrado en UdlaMobil, también se presenta un mensaje de error de Usuario no válido.

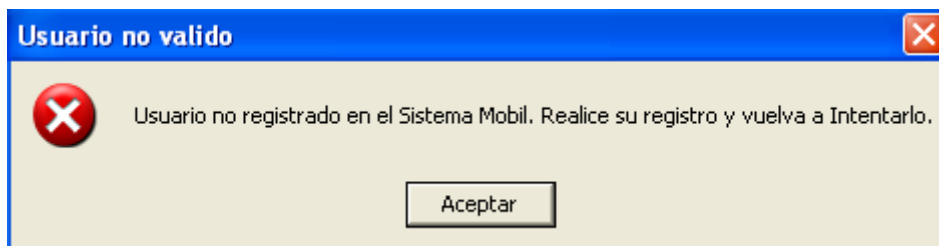


Figura 29. Error de Usuario registrado en UdlaMobil

Una vez ingresado los datos de un usuario válido, se presentan las opciones disponibles para el envío manual, estas son:

- Envío Manual de Horarios vía mail
- Envío Manual de Notas Parciales vía mail
- Envío Manual de Notas Totales vía mail

Para este caso se ha seleccionado la tercera opción, después de lo cual aparece una ventana de confirmación del envío.



Figura 30. Confirmación de Envío

El envío se lo puede verificar en este caso de ejemplo, primero por una ventana emergente producida, de la siguiente forma:

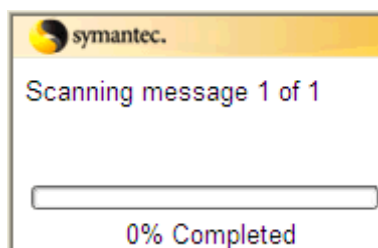


Figura 31. Corroboración de Proceso de Envío

La segunda forma de verificar si se ha realizado el envío de la información solicitada es mediante la apertura del mail correspondiente.



Figura 32. Comprobación de Mensaje Enviado

Conclusión

Por lo presentado, se puede concluir que la Aplicación Windows, está realizando en forma correcta la validación de los parámetros ingresados, que los mismos son almacenados como se ha dicho en un archivo de texto y que está realizando de buena forma el envío de la información solicitada por un usuario, que primeramente es validado.

ANEXO IX

MANUALES

MANUAL DE USUARIO

INDICE

MANUAL DE USUARIO DE UDLA WEB	112
INTRODUCCIÓN	112
REQUERIMIENTOS BÁSICOS	112
DESCRIPCIÓN DE LA APLICACIÓN	113
Login	113
Errores de Ingreso	114
Opciones	115
Notas	117
Horario	117
MANUAL DE USURIO DE UDLA MOBIL	120
INTRODUCCIÓN	120
REQUERIMIENTOS BÁSICOS	120
DESCRIPCIÓN DE LA APLICACIÓN	121
Login	121
Errores de Ingreso	122
Ingreso Datos Usuario Nuevo.	123
Actualización Datos.	127
MANUAL DE USUARIO DE UDLA WAP	129
INTRODUCCIÓN	129
REQUERIMIENTOS BÁSICOS	129
DESCRIPCIÓN DE LA APLICACIÓN	130
Login	130
Error de Ingreso	130
Ingreso al Sistema	131
El Menú Principal	132
Consulta de Notas	132
Consulta de Horarios	133
MANUAL DE USUARIO DE CONFIGURADOR	134
INTRODUCCIÓN	134
REQUERIMIENTOS BÁSICOS	134
DESCRIPCIÓN DE LA APLICACIÓN	135
Presentación	135
Login	135
Errores de Ingreso	136
Principal	136

Configurar Automático	137
Procedimientos Automáticos	143
Lanzar Envío de Horarios vía mail	144
Lanzar Envío de Notas Parciales vía mail	144
Lanzar Envío de Notas Finales vía mail	144
Lanzar Todos los Procedimientos Automáticos	144
Salir de Procedimientos Automáticos	145
Descripción de Errores	145
Procesos Manuales de Horarios y Notas	145
Envío Manual de Horarios vía mail	146
Envío Manual de Notas Parciales vía mail	146
Envío de Notas Finales vía mail	146
Procesos Manuales de Noticias	147
Consulta de Usuarios General	153
Consulta de Usuarios Parametrizada	154
Salir	155

MANUAL DE USUARIO DE UDLA WEB

INTRODUCCIÓN

UDLA Web es la Página Web mediante la cual los estudiantes podrán acceder al sitio Web de la Universidad de la Américas, y tendrán disponible los servicios de consulta de sus Horarios de clases y de sus Notas.

REQUERIMIENTOS BÁSICOS

- Computador PIII 550Mhz o Superior.
- 64Mb de memoria RAM
- Tener instalado un navegador de Internet como Internet Explorer

DESCRIPCIÓN DE LA APLICACIÓN

Login. (Figura 1)

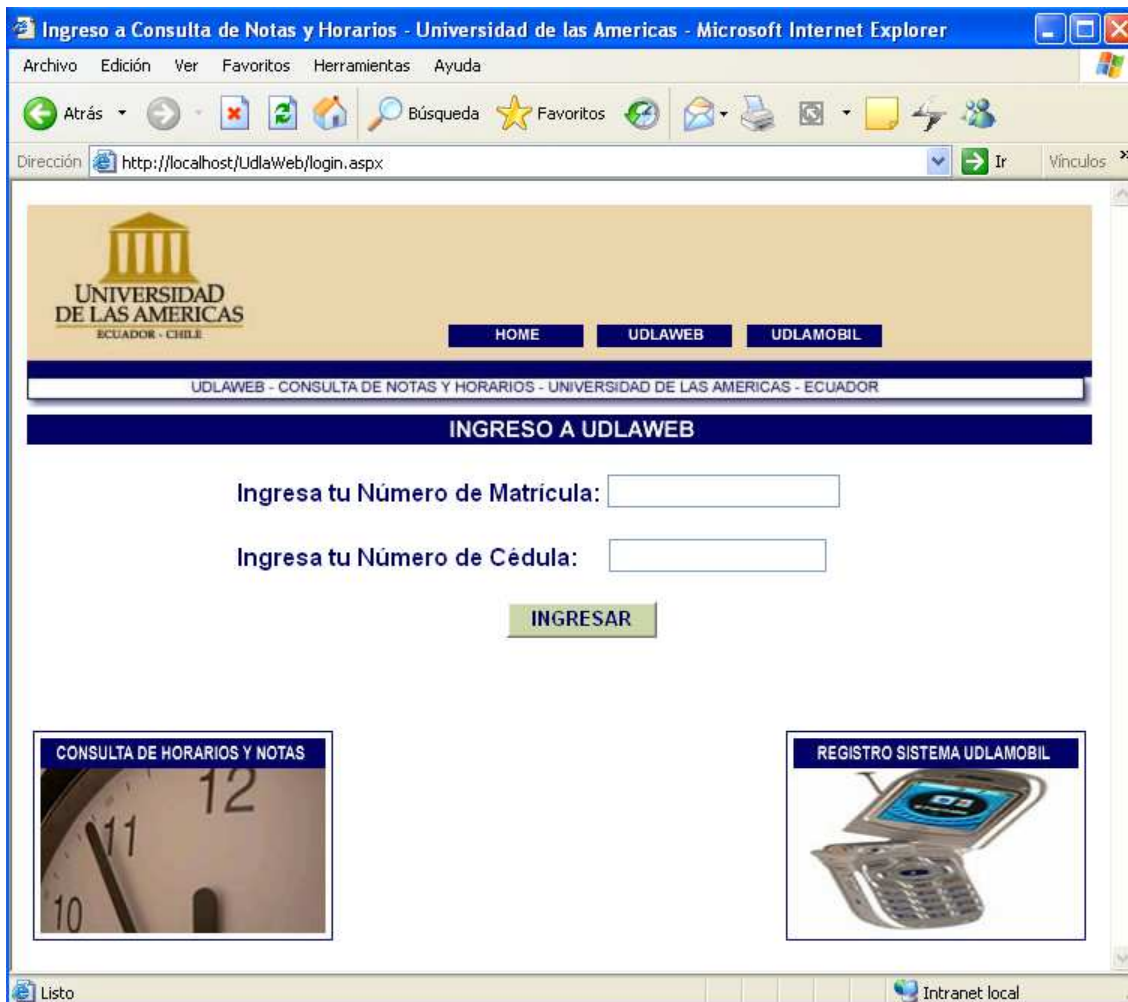


Figura 1. Página de Ingreso a UdlaWeb

Esta página es la que permite al estudiante el ingreso al sitio Web de la universidad de las Américas, consta de dos campos:

Número de Matrícula: Número de Matrícula en la Universidad de las Américas del estudiante que desea Ingresar. Campo tipo texto y de longitud máxima de 6 caracteres.

Número de Cédula: Número de identificación Personal del Estudiante que desea Ingresar. Campo tipo texto y de longitud máxima de 10 caracteres. Se debe ingresar el valor sin espacios ni guión.

Ingresar: Control que valida los datos ingresados y permite la búsqueda del estudiante en la Base de Datos SAES. Realizada la comprobación de la existencia del estudiante, este control presenta la Página Web de opciones.

Errores de Ingreso

En el caso de que no se haya ingresado valor alguno en los campos de Matrícula y Cédula, se presenta un mensaje que indica que los dos valores son obligatorios para realizar la correspondiente identificación del alumno, de la siguiente manera.

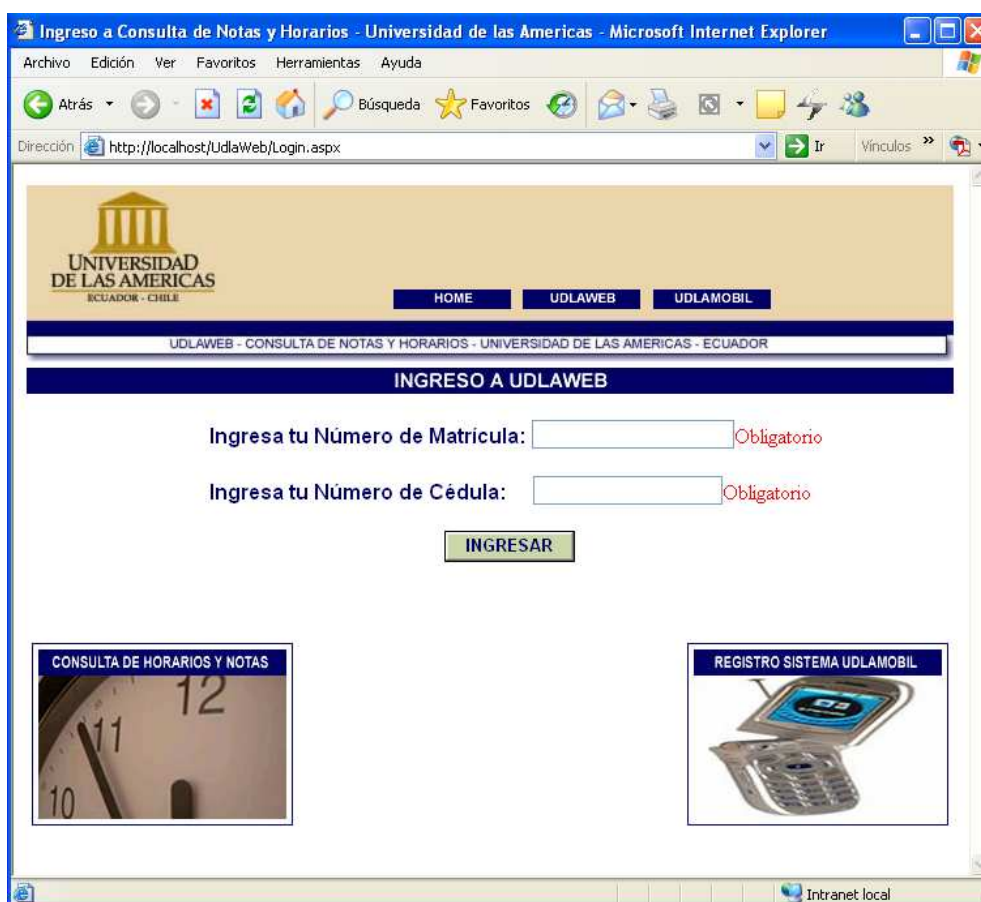


Figura 1.1. Error de Ingreso

De la misma forma si los valores ingresados son erróneos o no es un Usuario Válido, se presentará un mensaje de Usuario no Válido de la siguiente manera.

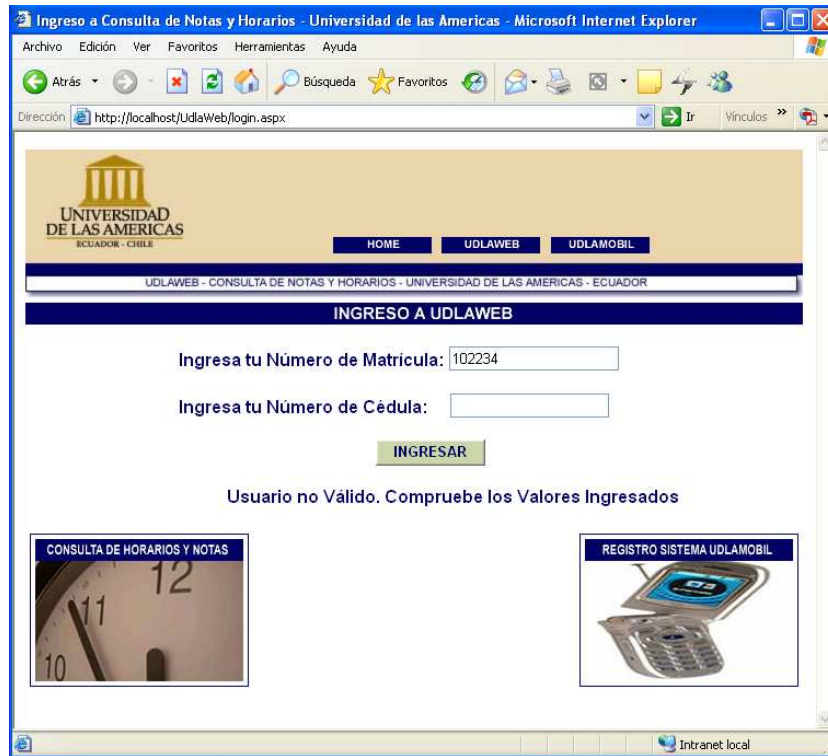


Figura 1.2. Error de Ingreso

Opciones. (Figura 2)



Figura 2. Opciones de Consulta

Esta es la Página en la cual se desplegarán las opciones disponibles para el estudiante. En primera instancia se necesita seleccionar el Año Lectivo y el Periodo lectivo del cual se tomará la Información.

Año Lectivo: Año Lectivo actual del cual se tomará la información. Campo de selección para que el estudiante escoja el año lectivo.

Periodo Lectivo: Periodo Actual que se encuentra cursando el estudiante. Esto es 1 para el primer semestre lectivo, 2 para el segundo semestre lectivo y 3 para el periodo de verano. Campo de selección para que el estudiante escoja el periodo del cual va a realizar la consulta.

Aceptar: Control que valida los datos ingresados y nos despliega las opciones disponibles para el alumno.

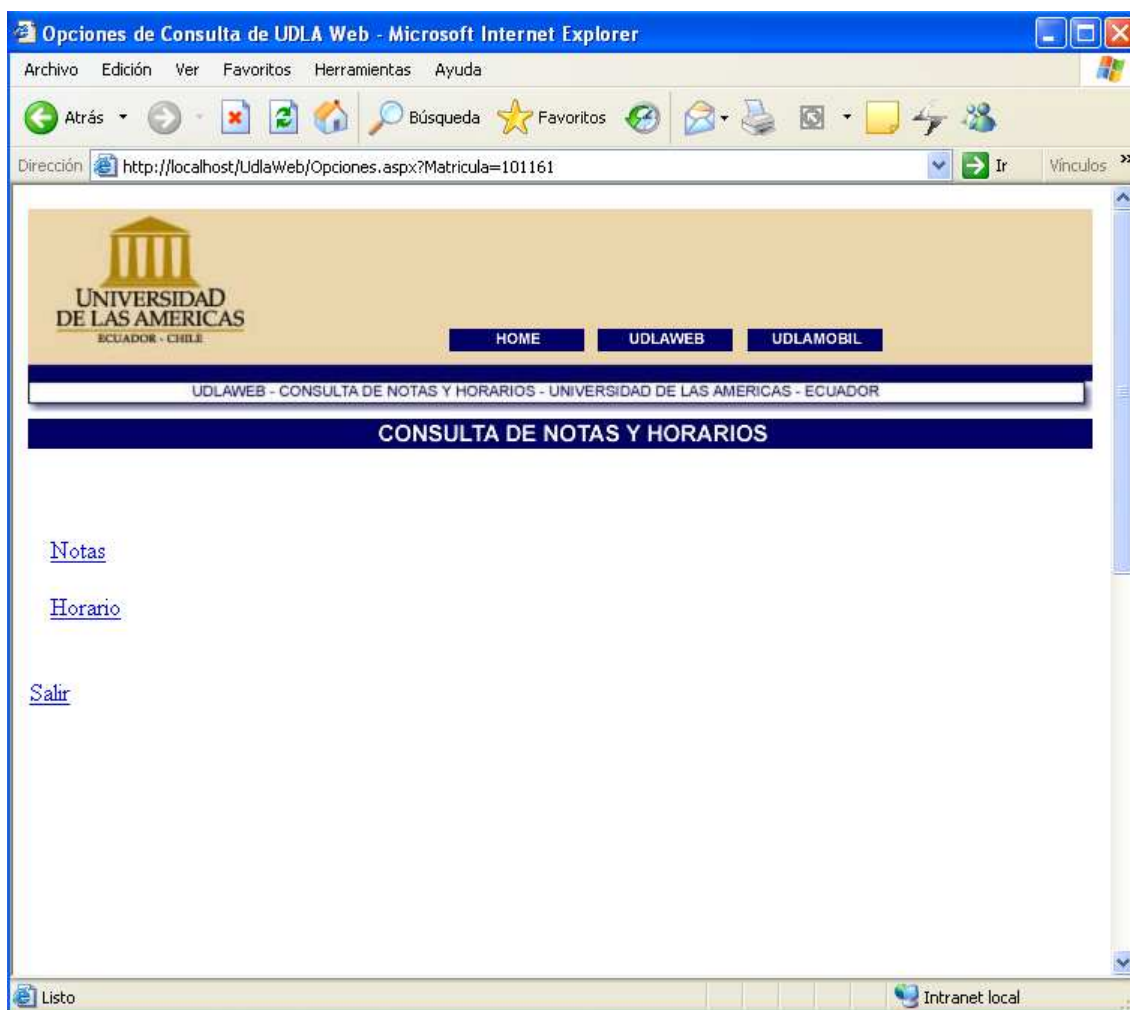


Figura 3. Menú principal

Al ingresar estos valores, se desplegarán todas las opciones de las que el estudiante puede hacer uso. Estas opciones son las siguientes:

- Notas
 - Por Materia
 - Cátedra 1
 - Cátedra 2
 - Controles
 - Examen Final
 - Recuperación
 - Todas las Notas

- Horario

Para la Opción de Notas por Materia se desplegarán todas las materias que el estudiante se encuentra cursando actualmente, y de las cuales podrá revisar las notas que estén pasadas hasta el momento de la petición. **(Figura 4)**

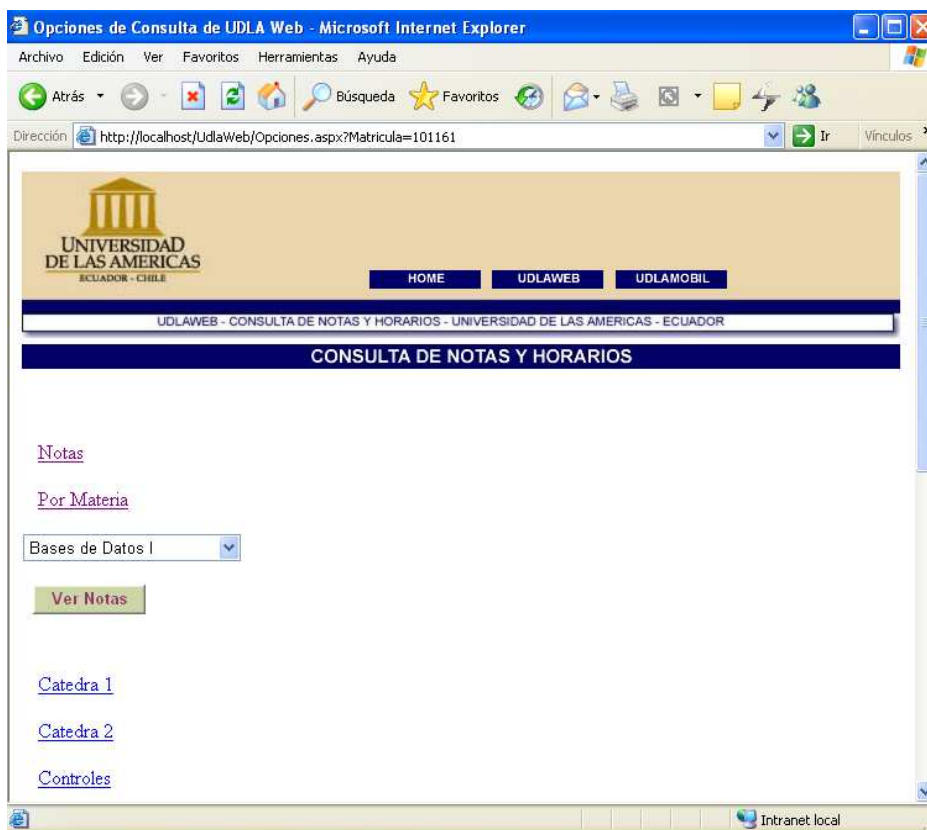


Figura 4. Opción de consulta de notas

Las Notas serán desplegadas en forma de tablas con la información requerida. (Figura 5)



Figura 5. Despliegue de notas

Para las opciones de Tipo de Nota, es decir Cátedra 1, Cátedra 2, Controles, etc. Se desplegará las notas específicamente de ese periodo para todas las materias que el estudiante esté cursando.

Mientras que para la opción de Todas las Notas, se desplegarán todas las notas de todas las materias, que se encuentren pasadas hasta el momento de la petición.

Por último para la opción de Horarios, se desplegará la siguiente información (Figura 6):

Opciones de Consulta de UDLA Web - Microsoft Internet Explorer

Archivo Edición Ver Favoritos Herramientas Ayuda

Atrás Búsqueda Favoritos Multimedia

Dirección http://localhost/udlaWeb/Opciones.aspx?Matricula=101161 Ir Vinculos

Notas

Horario

Matricula: 101161 Cedula: 501828792 Alumno: Guillermo Vargas Alvarez

Horario para el presente periodo

Hora	Lunes	Martes	Miercoles	Jueves	Viernes	Sabado
1
2
3
4	EIN207	..
5
6
7	..	AEA109
8
9
10	ESP776
11

Asignaturas

EIN207 - Bases de Datos I
 ESP776 - Herramientas de Software
 AEA109 - Redes I

Salir

Listo Intranet local

Figura 6. Despliegue de horarios

La información contendrá las siglas de las asignaturas, distribuidas según los módulos horarios y el día o días correspondientes, y después se tendrá una lista con el significado de las siglas, es decir el nombre de las asignaturas.

En todas las opciones antes presentadas se tiene disponible en el lado izquierdo de la página, un control de salida, mediante el cual el estudiante podrá abandonar la sesión en el caso de que lo considerase necesario.

MANUAL DE USUARIO DE UDLA MOBIL

INTRODUCCIÓN

UDLA Mobil es la Página Web mediante la cual los estudiantes podrán acceder al sitio Web de la Universidad de la Américas, y podrán registrarse para acceder a los servicios de envío de Horarios y Notas vía correo electrónico o ya sea vía SMS a sus celulares.

El registro en esta página será obligatorio para los estudiantes que deseen obtener estas ventajas.

REQUERIMIENTOS BÁSICOS

- Computador PIII 550Mhz o Superior.
- 64Mb de memoria RAM
- Tener instalado un navegador de Internet como Internet Explorer

DESCRIPCIÓN DE LA APLICACIÓN

Login. (Figura 7)

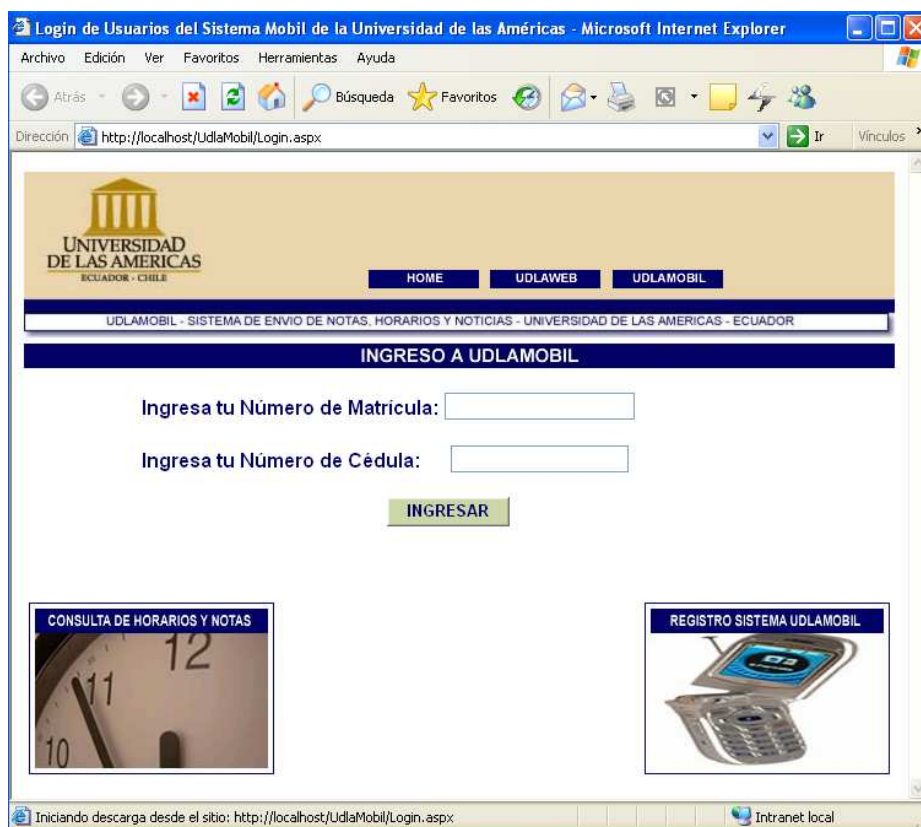


Figura 7. Página de Ingreso a UdlaMobil

Esta página es la que permite al estudiante el ingreso al sitio Web de la universidad de las Américas, consta de dos campos:

Número de Matrícula: Número de Matrícula en la Universidad de las Américas del estudiante que desea Ingresar. Campo tipo texto y de longitud máxima de 6 caracteres.

Número de Cédula: Número de identificación Personal del Estudiante que desea Ingresar. Campo tipo texto y de longitud máxima de 10 caracteres. Se debe ingresar el valor sin espacios ni guión.

Ingresar: Control que valida los datos ingresados y permite la búsqueda del estudiante en la Base de Datos SAES. Realizada la comprobación de la existencia del estudiante, y realiza la existencia del registro del estudiante como Usuario Mobil, si ya

existiere el registro, el estudiante es direccionado a una página para actualizar sus datos o para eliminar su registro, en caso contrario es direccionado a una página para realizar el registro del usuario.

Errores de Ingreso

En el caso de que no se haya ingresado valor alguno en los campos de Matrícula y Cédula, se presenta un mensaje que indica que los dos valores son obligatorios para realizar la correspondiente identificación del alumno, de la siguiente manera.

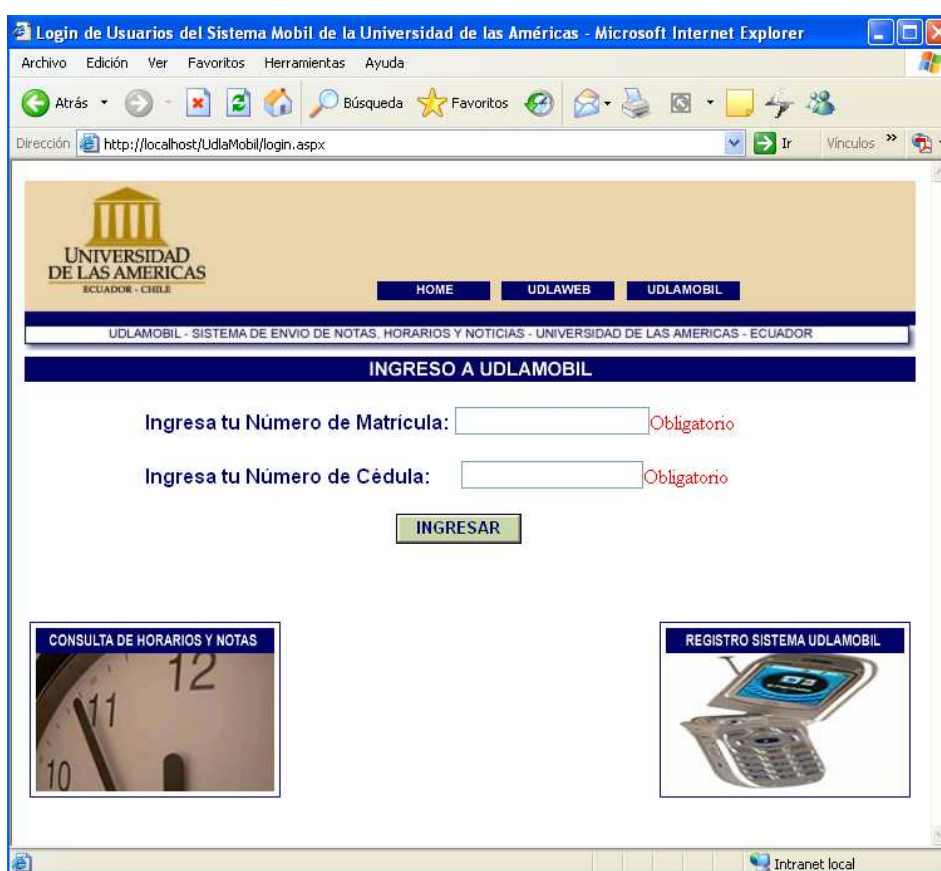


Figura 7.1. Error de Ingreso

De la misma forma si los valores ingresados son erróneos o no es un Usuario Válido, se presentará un mensaje de Usuario no Válido de la siguiente manera.

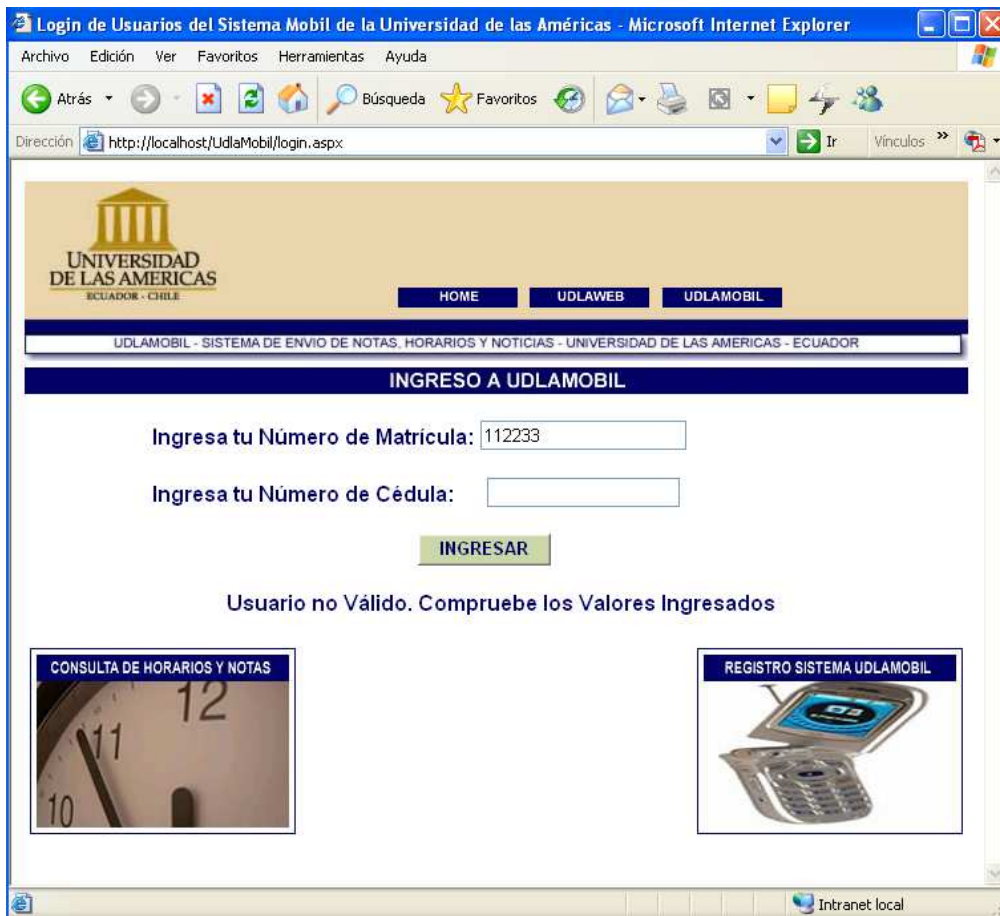



Figura 7.2. Error de Ingreso

Ingreso Datos Usuario Nuevo.

Si el Alumno es un Alumno de la Universidad de las Américas pero no está registrado aún en UDLA Mobil, aparecerá la siguiente pantalla (**Figura 8**)

IngresoDatos - Microsoft Internet Explorer
 Archivo Edición Ver Favoritos Herramientas Ayuda
 Dirección <http://localhost/UdlaMobil/IngresoDatos.aspx?Matricula=125492> Ir Vínculos >>


 HOME UDLAWEB UDLAMOBIL
 UDLAMOBIL - SISTEMA DE ENVIO DE NOTAS, HORARIOS Y NOTICIAS - UNIVERSIDAD DE LAS AMERICAS - ECUADOR
REGISTRO DE USUARIO NUEVO

* Número de Matrícula:
 * Número de Cédula:
 * Nombres:
 * Apellidos:
 Teléfono Domicilio:
 Teléfono Trabajo:
 Teléfono Celular:
 Proveedor Celular:
 Dirección:

Intranet local

Figura 8. Ingreso de Usuario Nuevo

Dentro de esta opción, el estudiante podrá realizar el registro formal de sus datos, y escoger las opciones que este desee para utilizar los servicios de UDLA Mobil, los cuales, son desplegados de la siguiente forma (**Figura 9**)

IngresoDatos - Microsoft Internet Explorer
 Archivo Edición Ver Favoritos Herramientas Ayuda
 Dirección <http://localhost/UdlaMobil/IngresoDatos.aspx?Matricula=125492> Ir Vínculos >>

Desea Recibir su Horario via mail: Si No
 Desea Recibir sus Notas via mail: Si No
 Desea Recibir sus Notas via Celular: Si No
 Desea Recibir Noticias de la UDLA via mail: Si No
 Desea Recibir Noticias de la UDLA via Celular: Si No
 En que formato desea recibir información via mail: Texto Html

Intranet local

Figura 9. Servicios de Udlamobil

Una vez realizado el registro de los datos del Nuevo Usuario Mobil, aparecerá una pantalla de Bienvenida, confirmando el ingreso del Alumno al Sistema UDLA Mobil, de la siguiente forma (Figura 10)



Figura 10. Página de Bienvenida a Udlamobil

Los campos de la página de registro son campos tipo texto en su mayor parte, utilizados para la recopilación de la información necesaria.

Numero de Matrícula: Número de matrícula del estudiante de la Universidad de las Américas. Campo auto llenado con la información de la Base de datos SAES, para un Alumno especificado. Campo tipo texto, de solo lectura.

Número de Cédula: Número de identificación personal del Alumno. Campo auto llenado con la información de la Base de datos SAES, para un Alumno especificado. Campo tipo texto, de solo lectura.

Nombres: Nombres del Alumno. Campo auto llenado con la información de la Base de datos SAES, para un Alumno especificado. Campo tipo texto, de solo lectura.

Apellidos:	Apellidos del Alumno. Campo auto llenado con la información de la Base de datos SAES, para un Alumno especificado. Campo tipo texto, de solo lectura.
Teléfono Domicilio:	Número telefónico del domicilio del Alumno. Campo tipo texto y longitud máxima de 9 caracteres.
Teléfono Trabajo:	Número telefónico del lugar de trabajo del Alumno, si lo tuviere. Campo tipo texto y longitud máxima de 9 caracteres.
Teléfono Celular:	Número telefónico del dispositivo celular del Alumno. Campo tipo texto y longitud máxima de 9 caracteres.
Proveedor Celular:	Campo de Selección con los nombres de las tres empresas distribuidoras de servicios de telefonía móvil existentes actualmente en el país.
Dirección:	Dirección del Domicilio del Estudiante. Campo tipo Texto y longitud máxima de 50 caracteres.
E-mail Primario:	Dirección de correo electrónico primario del Alumno. Campo tipo texto y longitud máxima de 35 caracteres.
E-mail Secundario:	Dirección de correo electrónico secundario del Alumno. Campo tipo texto y longitud máxima de 35 caracteres.
E-mail Alterno:	Dirección de correo electrónico alterno del Alumno. Campo tipo texto y longitud máxima de 35 caracteres.
Facultad:	Nombre de la facultad a la que pertenece el Alumno en la Universidad de las Américas. Campo tipo texto y longitud máxima de 20 caracteres.
Escuela:	Código de la escuela a la que pertenece el Alumno en la Universidad de las Américas. Campo auto llenado con la información de la Base de datos SAES, para un Alumno especificado. Campo tipo texto y de solo lectura.

Desea Recibir su Horario vía mail: Campo de selección, indica si el usuario desea habilitar esta opción o no.

Desea Recibir sus Notas vía mail: Campo de selección, indica si el usuario desea habilitar esta opción o no.

Desea Recibir sus Notas vía celular: Campo de selección, indica si el usuario desea habilitar esta opción o no.

Desea Recibir Noticias de la UDLA vía mail: Campo de selección, indica si el usuario desea habilitar esta opción o no.

Desea Recibir Noticias de la UDLA vía celular: Campo de selección, indica si el usuario desea habilitar esta opción o no.

En qué formato desea recibir información vía mail: Campo de selección, indica el formato en el cual el usuario recibirá la información en sus correos electrónicos. Este puede ser formato Html o formato texto.

Aceptar: Ingresa los Datos del Usuario en la Base de Datos MOBIL y presenta la página de Bienvenida al usuario al Sistema UDLA Mobil.

Cancelar: Cancela el Ingreso del usuario y presenta la página de Login.

Actualización Datos.

Si el Alumno ya es un usuario registrado en UDLA Mobil, al momento de hacer su Login, el sistema lo direcciona hacia la página de Actualización de Datos de Usuario. En donde el Usuario puede actualizar sus datos o eliminar su cuenta de Usuario Mobil, de la siguiente manera (**Figura 11**):

ActualizaciónDatos - Microsoft Internet Explorer

Archivo Edición Ver Favoritos Herramientas Ayuda

Dirección <http://localhost/UdlaMobil/ActualizacionDatos.aspx?Matricula=125492>

UNIVERSIDAD DE LAS AMERICAS ECUADOR - CHILE

HOME UDLAWEB UDLAMOBIL

UDLAMOBIL - SISTEMA DE ENVIO DE NOTAS, HORARIOS Y NOTICIAS - UNIVERSIDAD DE LAS AMERICAS - ECUADOR

ACTUALIZACION DE DATOS DE USUARIO

Si deseas prescindir de los servicios de UdlaMobil presiona aquí para eliminar tu cuenta.

[Eliminar](#)

[Salir](#)

* Número de Matrícula:

* Número de Cedula:

* Nombres:

* Apellidos:

Teléfono Domicilio:

Teléfono Trabajo:

Teléfono Celular:

Proveedor Celular:

Dirección:

* E-mail Primario:

Listo Intranet local

Figura 11. Actualización de datos de usuario

Los campos aquí presentados son los mismos que fueron ingresados en el Ingreso de Usuario Nuevo, con la Adición de siguiente control.

Eliminar: Comando para eliminar al usuario de la Base de Datos MOBIL.

En todas las opciones antes presentadas se tiene disponible en el lado izquierdo de la página, un control de salida, mediante el cual el estudiante podrá abandonar la sesión en el caso de que lo considerase necesario.

MANUAL DE USUARIO DE UDLA WAP

INTRODUCCIÓN

UDLA Wap es la aplicación mediante la cual los estudiantes de la Universidad de las Américas podrán acceder a la información de notas y horarios desde cualquier teléfono celular, que tenga capacidad de navegación por Internet Móvil, y activado el servicio. Esta alternativa constituye una nueva solución, para obtener el acceso a la información sin necesidad de un computador y de una conexión a Internet convencional.

Este es un nuevo servicio que Universidad de las Américas ofrece a los estudiantes como una alternativa para llegar a la información desde cualquier parte del mundo.

REQUERIMIENTOS BÁSICOS

- Teléfono Celular
- Micro-navegador de Internet disponible en el teléfono celular
- Tener habilitada la opción de Navegación en Internet

DESCRIPCIÓN DE LA APLICACIÓN

Login (Figura 12)



Figura 12. Página de Ingreso a UdlaWap

La página de ingreso permite el acceso del estudiante al sistema. Los siguientes son los parámetros de ingreso:

Matrícula: El número de matrícula del estudiante.

Cédula: El número de cédula del estudiante.

Al activar el control "Entrar", el sistema ingresa estos dos parámetros y los busca en la base de datos. El número de matrícula se mantiene como parámetro, para consultas posteriores.

Error de Ingreso

En el caso que no se encuentre las credenciales ingresadas dentro de la base de datos, el sistema presenta la siguiente pantalla (**Figura 13**):



Figura 13. Error de Ingreso

Ingreso al Sistema



Figura 14. Ingreso al Sistema

Una vez validado el usuario (**Figura 14**), el sistema solicita los siguientes parámetros de consulta:

Año: El año en el que se va a basar la consulta de notas u horarios.

Período: El período en el que se va a basar la consulta de notas u horarios.

El Menú Principal



Figura 15. Menú Principal

La aplicación está destinada a la consulta de notas y horarios. Es así como se ha estructurado el menú principal (**Figura 15**), de modo que el estudiante pueda acceder en la misma sesión a toda la información.

El usuario puede escoger las siguientes opciones:

- Notas.- El usuario puede ingresar al módulo Notas, y realizar consultas.
- Horarios.- El usuario puede ingresar al módulo Horarios, y realizar consultas.
- Volver.- El usuario puede regresar e ingresar nuevos parámetros de consulta.

Consulta de Notas

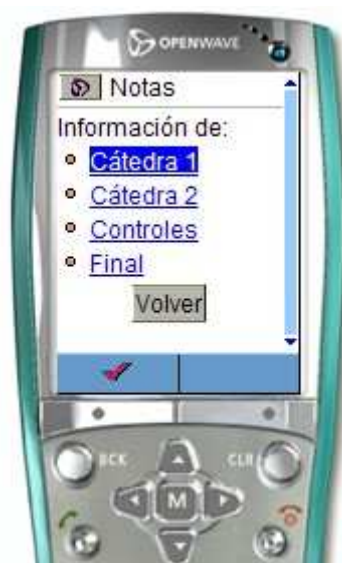


Figura 16. Consulta de Notas

El módulo Notas está destinado a la consulta de las calificaciones del alumno, según los parámetros ingresados al iniciar la sesión (**Figura 16**). El resultado de la consulta será presentado en la misma pantalla, de acuerdo a la opción que haya sido escogida.

La interfaz presenta al usuario las opciones:

- Cátedra 1: Presenta las calificaciones de todas las materias, correspondientes a Cátedra 1.
- Cátedra 2: Presenta las calificaciones de todas las materias, correspondientes a Cátedra 2.
- Controles: Presenta las calificaciones promedio de todas las pruebas, trabajos, deberes, que conforma la nota de Controles.
- Final: Presenta las calificaciones de todas las materias, correspondientes al examen final.

Consulta de Horarios



Figura 17. Consulta de Horarios

El módulo Horarios está destinado a la consulta de horarios del alumno (**Figura 17**). El resultado de la consulta será presentado en la misma pantalla, de acuerdo a la opción que haya sido escogida.

Al ingresar al módulo, la interfaz listará automáticamente todas las materias que el alumno cursa, según los parámetros ingresados al iniciar la sesión de la aplicación. Esta lista sirve a su vez como un grupo de opciones, sobre las cuales el estudiante puede interactuar para realizar la consulta respectiva de horario por materia.

En todas las opciones antes presentadas, se tiene la alternativa de seleccionar el control volver, el cual presentará en pantalla la la página anteriormente visitada.

MANUAL DE USUARIO DE CONFIGURADOR

INTRODUCCIÓN

Configurador es una Aplicación Windows, la cual fue construida para permitir al Administrador del Sistema la configuración y la ejecución de los procedimientos automáticos que se realizarán a lo largo de los periodos de clases en la Universidad de las Américas, como son el Envío de horarios, noticias y de notas vía E-mail, desde el servidor de la Universidad, a través de un Servidor de Correo Electrónico, hacia los Estudiantes registrados (UDLAMobil).

REQUERIMIENTOS BÁSICOS

- Computador PIII 550Mhz o Superior.
- 64Mb de memoria RAM
- 100MB Libres en espacio de Disco Duro

DESCRIPCIÓN DE LA APLICACIÓN

Presentación (Figura 18).



Figura 18. Presentación del Sistema Configurador

Login. (Figura 19).

The image shows a login window titled 'Login'. It has a blue header bar with a close button (X) in the top right corner. The main area is light beige. There are two input fields: 'Usuario:' with the text 'Admin' and 'Contraseña:' with a masked password '*****'. Below the fields is a button labeled 'Aceptar'.

Figura 19. Pantalla de Ingreso al Sistema Configurador

Esta pantalla es la que permite el ingreso a la Aplicación de Configuración, consta de dos campos:

Usuario: Nombre del Usuario que ingresa a la aplicación Configurador. Es un campo de tipo texto y tiene una longitud máxima de 15 caracteres.

Contraseña: Contraseña para el usuario que ingresa a la aplicación Configurador. Es un campo de tipo texto y tiene una longitud máxima de 15 caracteres. Adicionalmente este campo se encuentra en formato de Password, para evitar la visualización del mismo.

Aceptar: Control que ejecuta el ingreso del usuario ingresado. Aquí se verifica la existencia del usuario en la base de datos y se permite o deniega el acceso, según sea el caso.

Errores de Ingreso

En el caso de que el servidor de base de datos no se encuentre disponible se producirá el siguiente mensaje de error:

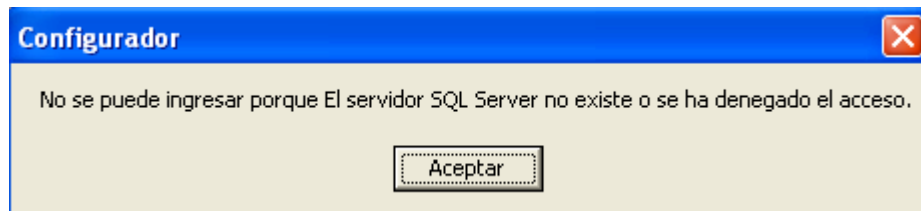


Figura 19.1. Error de Ingreso

Por otra parte si el usuario que desea ingresar al sistema no es un usuario válido, el sistema desplegará el siguiente mensaje de error:

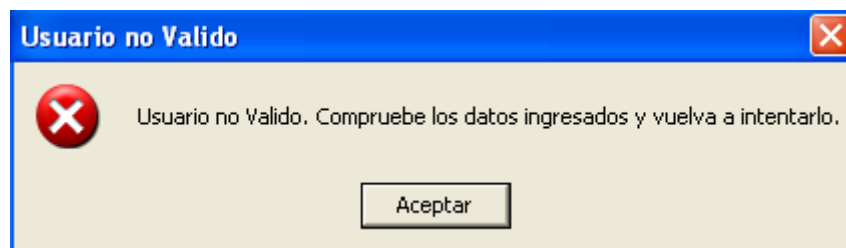


Figura 19.2. Error de Ingreso

Principal (Figura 20)

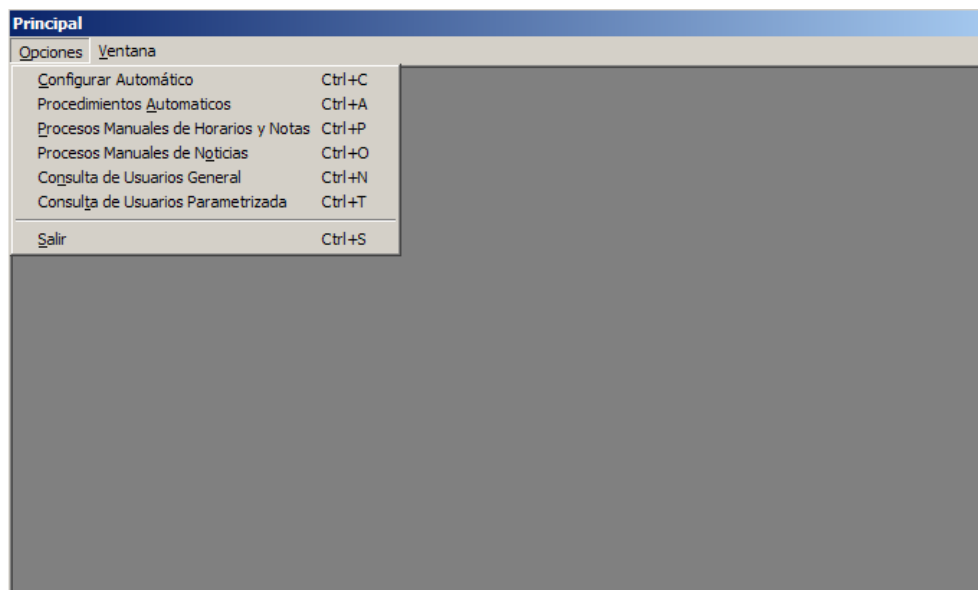


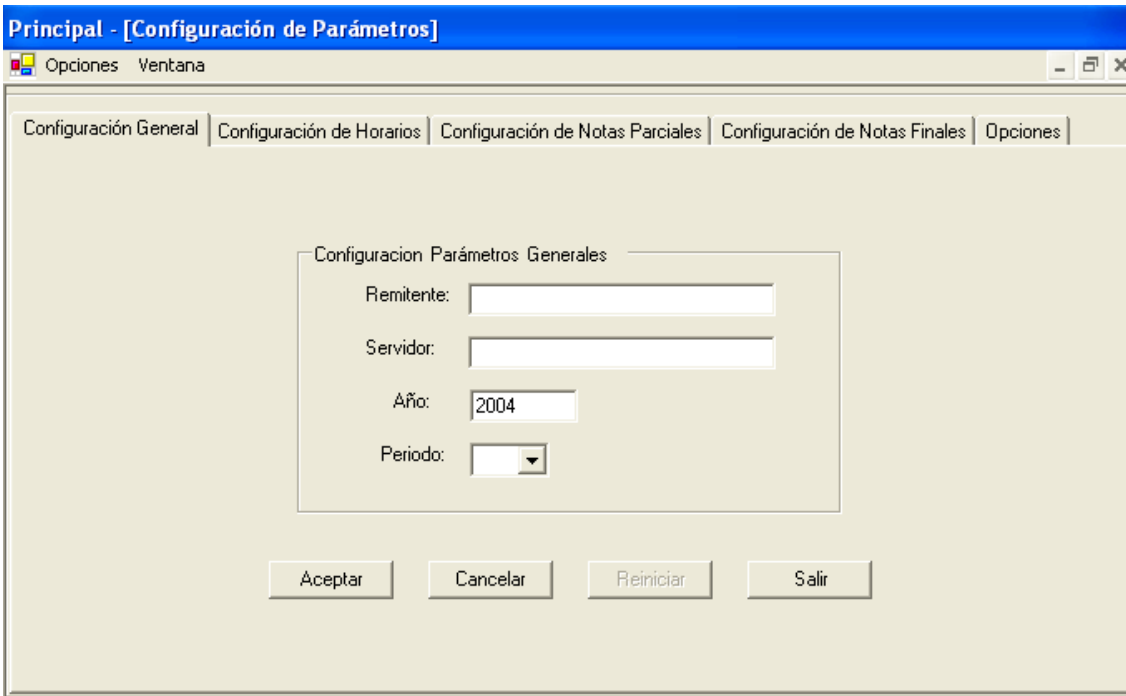
Figura 20. Menú Principal

Este es un Formulario MIDI, por medio del cual se podrá visualizar todas las opciones de la Aplicación.

Dentro de esta ventana, tenemos en el Menú Opciones, lo siguiente:

- Configurar Automático
- Procedimientos Automáticos
- Procesos Manuales de Horarios y Notas
- Procesos Manuales de Noticias
- Consulta de Usuarios General
- Consulta de Usuarios Parametrizada y
- Salir

Configurar Automático (Figura 21)



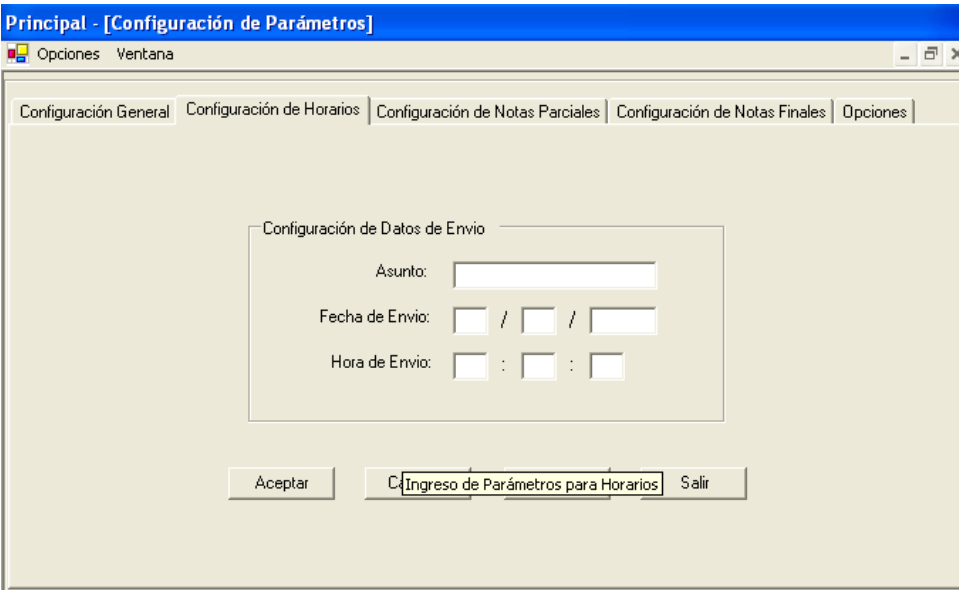
The image shows a screenshot of a software window titled "Principal - [Configuración de Parámetros]". The window has a menu bar with "Opciones" and "Ventana". Below the menu bar, there are several tabs: "Configuración General", "Configuración de Horarios", "Configuración de Notas Parciales", "Configuración de Notas Finales", and "Opciones". The "Configuración General" tab is selected. Inside the window, there is a section titled "Configuración Parámetros Generales" containing four input fields: "Remitente:" (text box), "Servidor:" (text box), "Año:" (text box with "2004" entered), and "Periodo:" (dropdown menu). Below these fields are four buttons: "Aceptar", "Cancelar", "Reiniciar", and "Salir".

Figura 21. Configuración General

La primera opción presentada es la de Configuración General, dentro de la cual se realizará el ingreso de parámetros generales para toda la Aplicación como son el Remitente, el Servidor de Correo Electrónico, el Año y el Periodo con el cual se va a trabajar.

Remitente: Nombre del Remitente que constará en los Correos Electrónicos que se envíe.
Campo de tipo texto y tiene una longitud máxima de 20 caracteres.

- Servidor:** Nombre del Servidor de Correo Electrónico por medio del cual se realizará el envío de datos a los estudiantes. Campo de tipo texto y tiene una longitud máxima de 30 caracteres.
- Año:** Año para el cual se tomará como parámetro de búsqueda de información de los estudiantes. Campo de tipo texto y tiene una longitud máxima de 4 caracteres. Adicionalmente este campo tiene como valor por defecto el año actual.
- Periodo:** Periodo académico del cual se tomará la información de los estudiantes. Campo de Selección con los valores correspondientes al periodo. Esto es 1 para el primer periodo académico, 2 para el segundo, y 3 para el periodo de verano.
- Aceptar:** Control que valida el ingreso de datos en los campos antes explicados y si son correctos los almacena en variables globales de la Aplicación.
- Cancelar:** Control que vacía los valores de los campos ingresados y deja listo para que se vuelva a ingresar nuevos valores.
- Reiniciar:** Control que pone en cero las variables Globales de la Aplicación y permite el ingreso de Nuevos valores para estas variables.
- Salir:** Control que cierra la opción Configurar Automático y nos regresa a la Pantalla Principal.



The screenshot shows a window titled "Principal - [Configuración de Parámetros]". The window has a menu bar with "Opciones" and "Ventana". Below the menu bar, there are several tabs: "Configuración General", "Configuración de Horarios", "Configuración de Notas Parciales", "Configuración de Notas Finales", and "Opciones". The "Configuración de Horarios" tab is active. Inside the window, there is a section titled "Configuración de Datos de Envío" which contains three input fields: "Asunto:" (a text box), "Fecha de Envío:" (three separate boxes for day, month, and year separated by slashes), and "Hora de Envío:" (three separate boxes for hour, minute, and second separated by colons). At the bottom of the window, there are three buttons: "Aceptar", "Ingreso de Parámetros para Horarios", and "Salir".

Figura 22. Configuración de Horarios

La segunda opción es la de Configuración de Horarios (**Figura 22**), mediante la cual se puede ingresar los parámetros necesarios para realizar el envío automático de Horarios vía correo electrónico a los estudiantes.

Asunto: Texto que se incluirá en los mensajes de correo electrónico como objeto del mensaje. Campo tipo texto y de longitud máxima de 20 caracteres.

Fecha de Envío: Consta de tres campos tipo de texto, que sirven para ingresar el primero el día en formato numérico de dos dígitos, el segundo la fecha en formato numérico de dos dígitos y el último el año en formato numérico de cuatro dígitos.

Hora de Envío: Consta de tres campos tipo texto, que sirven, para ingresar, el primero la hora en formato de 24 horas en dos dígitos, el segundo el minuto en formato numérico de dos dígitos y el último los segundos en formato numérico de dos dígitos.

Aceptar: Control que valida el ingreso de datos en los campos antes explicados y si son correctos los almacena en variables globales de la Aplicación.

Cancelar: Control que vacía los valores de los campos ingresados y deja listo para que se vuelva a ingresar nuevos valores.

Reiniciar: Control que pone en cero las variables Globales de la Aplicación y permite el ingreso de Nuevos valores para estas variables.

Salir: Control que cierra la opción Configurar Automático y nos regresa a la Pantalla Principal.

Principal - [Configuración de Parámetros]

Opciones Ventana

Configuración General | Configuración de Horarios | Configuración de Notas Parciales | Configuración de Notas Finales | Opciones

Configuración de Datos de Envío

Asunto:

Fecha de Envío Cátedra1: / / Fecha de Envío Cátedra2: / /

Hora de Envío Cátedra1: : : Hora de Envío Cátedra2: : :

Fecha de Envío Controles: / /

Hora de Envío Controles: : :

Fecha de Envío Examen Final: / / Fecha de Envío Recuperación: / /

Hora de Envío Examen Final: : : Hora de Envío Recuperación: : :

Aceptar Cancelar Reiniciar Salir

Figura 23. Configuración de Notas Parciales

La tercera opción es la de Configuración de Notas Parciales (**Figura 23**), mediante la cual se puede ingresar los parámetros necesarios para realizar el envío automático de las Notas por cada tipo vía correo electrónico a los estudiantes, esto es parametrizar el envío de notas para Cátedra 1, Cátedra 2, Controles, etc.

Asunto: Texto que se incluirá en los mensajes de correo electrónico como objeto del mensaje. Campo tipo texto y de longitud máxima de 20 caracteres.

Fecha de Envío: Consta de tres campos tipo de texto, que sirven para ingresar el primero el día en formato numérico de dos dígitos, el segundo la fecha en formato numérico de dos dígitos y el último el año en formato numérico de cuatro dígitos.

Hora de Envío: Consta de tres campos tipo texto, que sirven, para ingresar, el primero la hora en formato de 24 horas en dos dígitos, el segundo el minuto en formato numérico de dos dígitos y el último los segundos en formato numérico de dos dígitos.

Los campos descritos anteriormente son ingresados para cada tipo de Nota, esto es para Cátedra1, Cátedra2, Controles, Examen Final y Recuperación.

- Aceptar:** Control que valida el ingreso de datos en los campos antes explicados y si son correctos los almacena en variables globales de la Aplicación.
- Cancelar:** Control que vacía los valores de los campos ingresados y deja listo para que se vuelva a ingresar nuevos valores.
- Reiniciar:** Control que pone en cero las variables Globales de la Aplicación y permite el ingreso de Nuevos valores para estas variables.
- Salir:** Control que cierra la opción Configurar Automático y nos regresa a la Pantalla Principal.

Figura 24. Configuración de Notas Finales

La Cuarta opción es la de Configuración de Notas Finales (**Figura 24**), mediante la cual se puede ingresar los parámetros necesarios para realizar el envío automático de las Notas al final del semestre vía correo electrónico a los estudiantes.

Asunto: Texto que se incluirá en los mensajes de correo electrónico como objeto del mensaje. Campo tipo texto y de longitud máxima de 20 caracteres.

Fecha de Envío: Consta de tres campos tipo de texto, que sirven para ingresar el primero el día en formato numérico de dos dígitos, el segundo la fecha en

formato numérico de dos dígitos y el último el año en formato numérico de cuatro dígitos.

Hora de Envío: Consta de tres campos tipo texto, que sirven, para ingresar, el primero la hora en formato de 24 horas en dos dígitos, el segundo el minuto en formato numérico de dos dígitos y el último los segundos en formato numérico de dos dígitos.

Aceptar: Control que valida el ingreso de datos en los campos antes explicados y si son correctos los almacena en variables globales de la Aplicación.

Cancelar: Control que vacía los valores de los campos ingresados y deja listo para que se vuelva a ingresar nuevos valores.

Reiniciar: Control que pone en cero las variables Globales de la Aplicación y permite el ingreso de Nuevos valores para estas variables.

Salir: Control que cierra la opción Configurar Automático y nos regresa a la Pantalla Principal.

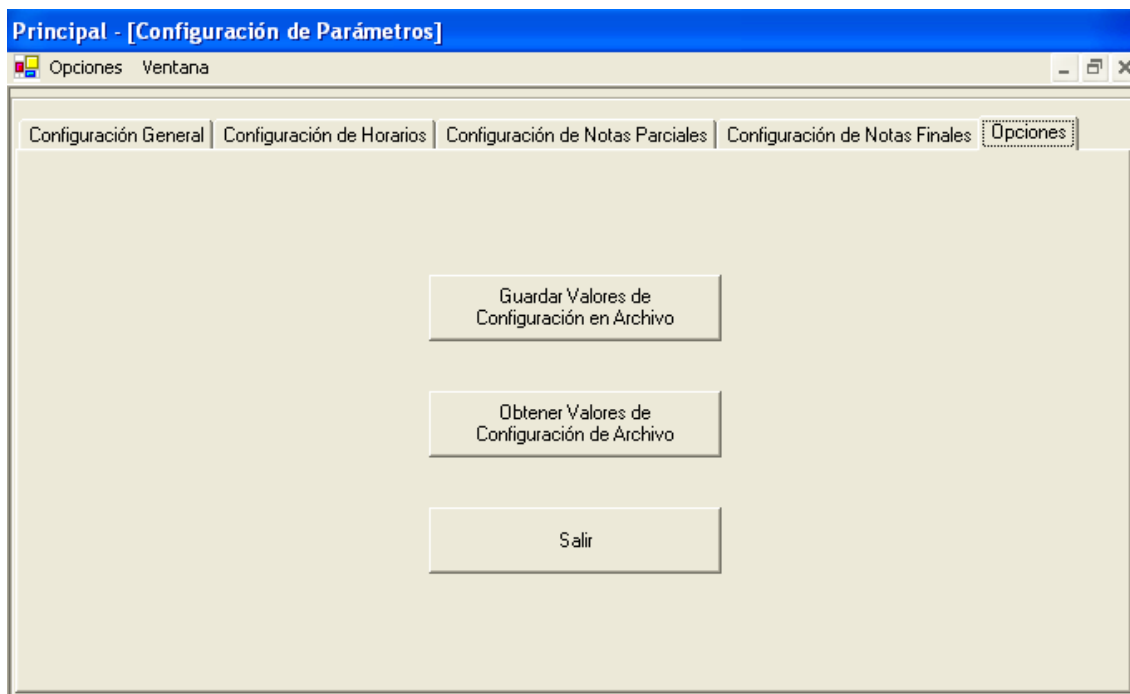


Figura 25. Opciones Adicionales

La última opción presentada es la de Opciones, mediante la cual se podrá almacenar los datos ingresados como parámetros de configuración en un archivo de texto, o a su vez obtener dichos valores desde un archivo de texto.

Guardar Valores de Configuración en Archivo: Control mediante el cual se podrá almacenar los valores ingresados en un archivo de texto.

Obtener Valores de Configuración de Archivo: Control mediante el cual se podrá obtener los valores de configuración desde un archivo de texto.

Salir: Control que cierra la opción Configurar Automático y nos regresa a la Pantalla Principal.

Procedimientos Automáticos (Figura 26)

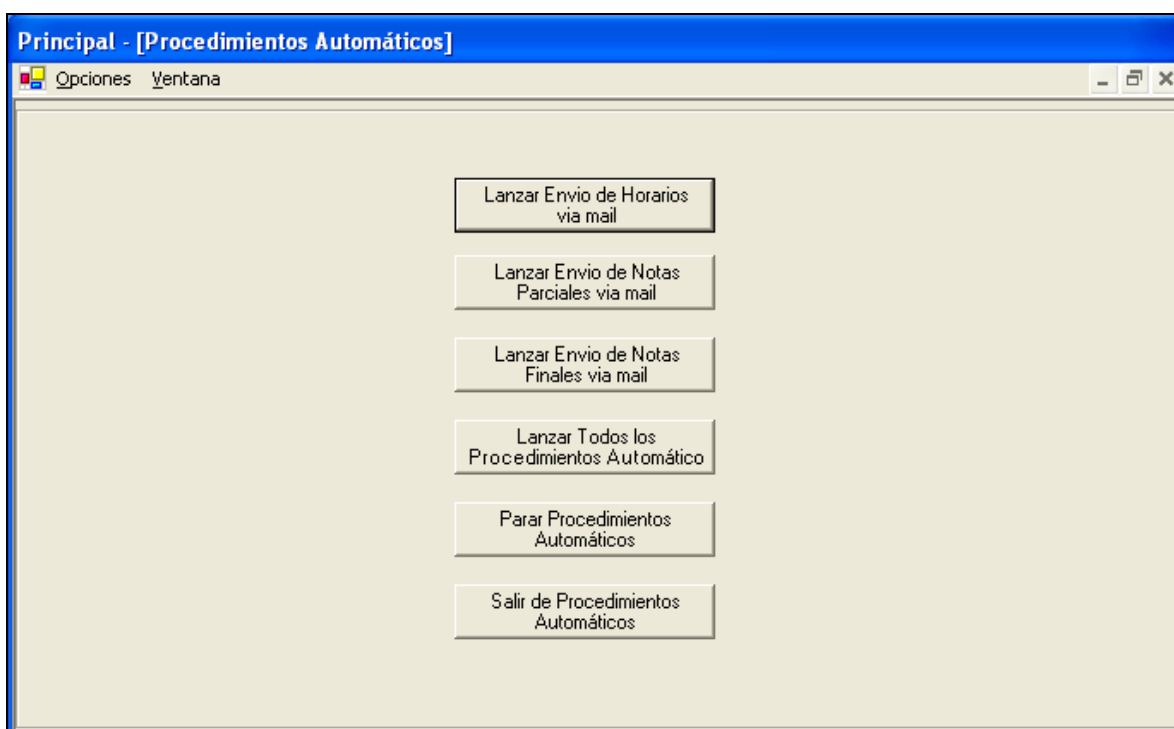


Figura 26. Procedimientos Automáticos

Esta opción es la encargada de realizar la llamada a los procedimientos automáticos, como son el envío de horarios vía correo electrónico y el envío de notas tanto parciales como finales de la misma forma, hacia todos los estudiantes de la universidad, claro, estudiantes que antes deben haber realizado el registro en UDLAMobil.

Tenemos seis opciones:

- Lanzar Envío de Horarios vía mail
 - * Mediante esta opción se puede dejar automatizado el procedimiento de envío de Horarios a los estudiantes. Se mantiene un Temporizador, controlando el cumplimiento de la fecha ingresada como parámetro, y entonces, se envía los horarios a los estudiantes, y se termina la automatización. Esta opción solo realiza el envío de Horarios en la fecha ingresada como parámetro y después libera el control de Temporizador.

- Lanzar Envío de Notas Parciales vía mail
 - * Mediante esta opción se puede dejar automatizado el procedimiento de envío de Notas Parciales a los estudiantes. Se mantiene un Temporizador, controlando el cumplimiento de la fecha ingresada como parámetro de cada tipo, esto es Cátedra 1, Cátedra 2, etc., y entonces, se envía las Notas a los estudiantes, y se termina la automatización. Esta opción solo realiza el envío de Notas en cada determinada fecha y después libera el control de Temporizador.

- Lanzar Envío de Notas Finales vía mail
 - * Mediante esta opción se puede dejar automatizado el procedimiento de envío de Notas Finales a los estudiantes. Se mantiene un Temporizador, controlando el cumplimiento de la fecha ingresada como parámetro, y entonces, se envía las Notas Finales a los estudiantes, y se termina la automatización. Esta opción solo realiza el envío de Notas Finales en la fecha ingresada como parámetro y después libera el control de Temporizador.

- Lanzar Todos los Procedimientos Automáticos
 - * Mediante esta opción, se realiza la automatización de todos los procedimientos automáticos: Esta opción mantiene al temporizador activo durante toda la ejecución de cada uno de los procedimientos automáticos, hasta que se cumplan todos. Es decir este procedimiento, realiza el envío de horarios, de

notas parciales y de notas finales vía correo electrónico a los estudiantes sin necesidad de ejecutar cada uno de ellos por separado, luego de realizar todas estas acciones, el Temporizador es liberado.

- Salir de Procedimientos Automáticos
 - * Mediante esta opción, salimos de Procedimientos Automáticos y nos regresa a la Pantalla Principal.

Descripción de Errores

En el caso de que los valores de configuración vistos en la primera opción (Configurar Automático), no hayan sido ingresados o tomados desde un archivo externo, se presentará el siguiente mensaje de error.

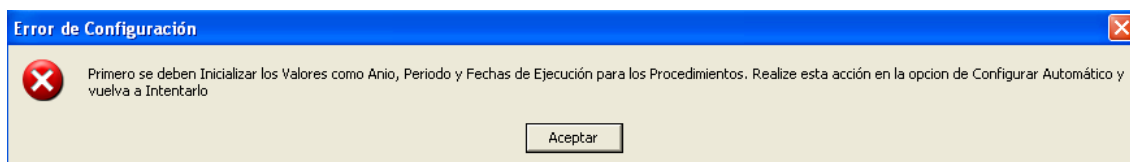


Figura 26.1. Error de Configuración

Procesos Manuales de Horarios y Notas (Figura 27)

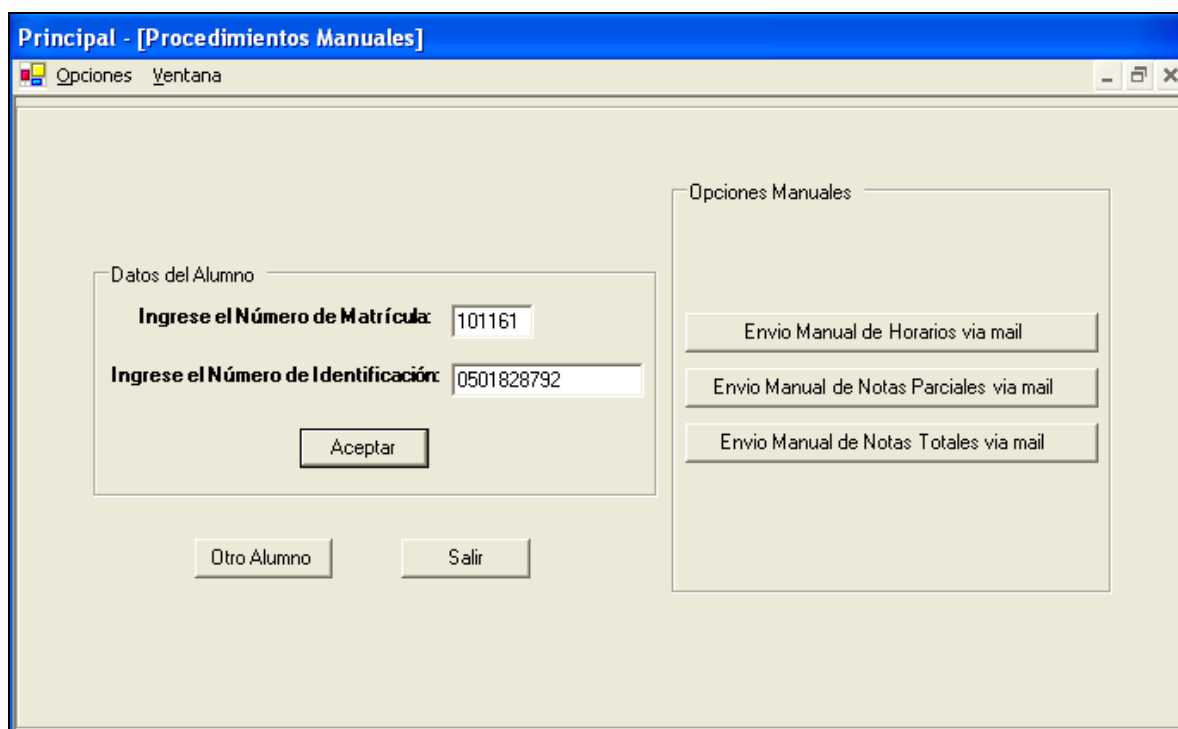


Figura 27. Procesos Manuales de horarios y notas

Esta Opción nos permite realizar el envío manual de Horarios, Notas Parciales y Notas Finales a un estudiante determinado por el número de matrícula y el número de cédula. Las Notas parciales que serán enviadas, son las del periodo actual, es decir según la fecha en la que se vaya a enviar, podría ser las notas de Cátedra 1, Cátedra 2, etc.

Número de Matrícula: Número de Matrícula en la Universidad de las Américas del estudiante que desea que se le envíe su Horario o Notas. Campo tipo texto y de longitud máxima de 6 caracteres.

Número de Cédula: Número de identificación Personal del Estudiante que desea que se le envíe su Horario o Notas. Campo tipo texto y de longitud máxima de 10 caracteres. Se debe ingresar el valor sin espacios ni guión.

Aceptar: Control que permite la búsqueda del estudiante en la Base de Datos y nos despliega las opciones que tenemos para realizar, la cuales son las siguientes:

- Envío Manual de Horarios vía mail
 - * Esta Opción nos permite que se realice el envío del horario del estudiante de quien se ingresó los datos en ese momento.

- Envío Manual de Notas Parciales vía mail
 - * Esta Opción nos permite que se realice el envío de las notas del estudiante de quien se ingresó los datos en ese momento. Las notas corresponden al Parcial actual que se esté cursando en la fecha de envío, es decir pueden ser las notas correspondientes a Cátedra 1, Cátedra 2, etc.

- Envío de Notas Finales vía mail
 - * Esta Opción nos permite que se realice el envío de todas las notas que se encuentren pasadas hasta el momento del envío, del estudiante de quien se ingresó los datos.

Otro Alumno: Control que encera los campos ingresados, libera recursos de la búsqueda anterior y nos permite realizar acciones para otros estudiantes.

Salir: Control que nos saca del Proceso Actual y nos envía a la Pantalla Principal.

Procesos Manuales de Noticias (Figura 28)

Figura 28. Ingreso de Noticias

La primera Opción presentada es Ingreso de Noticia, mediante la cual se nos permite realizar el ingreso de una noticia, la cual podrá ser enviada posteriormente a los estudiantes, por medio de sus correos electrónicos o podría ser dirigida a los celulares vía SMS.

Título de la Noticia: Es el título por el cual se va a reconocer a la noticia. Campo tipo texto con una longitud máxima de 20 caracteres.

Contenido: Es el cuerpo de la noticia, es la noticia en sí. Campo tipo texto con una longitud máxima de 110 caracteres. Las noticias deben ser resumidas, debido a que estas podrían ser enviadas vía SMS a los celulares de los estudiantes, y estos únicamente aceptan 150 caracteres incluyendo el remitente y el número de celular.

Fecha de Caducidad: Es una fecha en la cual las noticias perderán su vigencia, con el objetivo de clasificar las noticias y distinguirlas para su envío. Campo tipo texto que consta de tres partes, la primera es para el día, en formato numérico y longitud máxima de dos caracteres, el segundo corresponde al mes, en formato numérico y longitud máxima de dos caracteres, y el tercero corresponde al año, en formato numérico y longitud máxima de cuatro caracteres.

Ingresar: Control que valida la información ingresada y la almacena en la Base de Datos.

Cancelar: Control que encera los campos de ingreso y deja listo para ingresar una nueva noticia.

Salir: Control para salida del proceso Actual y nos envía a la Pantalla Principal.

The screenshot shows a software window titled "Principal - [Procedimiento Manual de Noticias]". The window has a menu bar with "Opciones" and "Ventana". Below the menu bar is a tabbed interface with five tabs: "Ingreso de Noticia", "Actualización de Noticia" (which is selected), "Consulta de Noticias", "Eliminación de Noticias", and "Ejecución de Procedimiento de Noticias".

The "Actualización de Noticia" tab contains a form with the following fields and controls:

- A section titled "Seleccione la Noticia" containing a dropdown menu for "Título de la Noticia" with the value "Suspensión de Clases" and a "Ver Noticia" button below it.
- A section for editing the news item with the following fields:
 - "Título de la Noticia": Text box containing "Suspensión de Clases".
 - "Contenido": Text area containing "Las clases del día 15/10/2004 han sido suspendidas por efectos del temporal".
 - "Fecha de Ingreso": Date picker showing 30 / 9 / 2004.
 - "Fecha de Caducidad": Date picker showing 16 / 10 / 2004.
- At the bottom of the form are three buttons: "Actualizar", "Cancelar", and "Salir".

Figura 29. Actualización de Noticias

Mediante esta opción se puede desplegar una noticia seleccionada, para realizar los cambios necesarios en la misma.

Título de la Noticia: Campo de selección por título de noticia, para escoger una determinada noticia y desplegar su contenido. Campo tipo Lista para desplegar las Noticias existentes en la Base de Datos. Campo de solo lectura.

Título de la Noticia: Campo al cual se podrá hacer las modificaciones necesarias. Campo tipo texto y longitud máxima de 20 caracteres.

Contenido: Campo que contiene la noticia en sí y al cual se podrá realizar las pertinentes modificaciones. Campo tipo texto y longitud máxima de 110 caracteres.

Fecha de Ingreso: Fecha en la cual se realizó el ingreso de la Noticia. Campo tipo texto que consta de de tres partes, la primera es para el día, en formato numérico y longitud máxima de dos caracteres, el segundo corresponde al mes, en formato numérico y longitud máxima de dos caracteres, y el tercero corresponde al año, en formato numérico y longitud máxima de cuatro caracteres.

Fecha de Caducidad: Fecha en la cual la Noticia sale de vigencia. Campo tipo texto que consta de de tres partes, la primera es para el día, en formato numérico y longitud máxima de dos caracteres, el segundo corresponde al mes, en formato numérico y longitud máxima de dos caracteres, y el tercero corresponde al año, en formato numérico y longitud máxima de cuatro caracteres.

Actualizar: Control por medio del cual se almacenarán los cambios realizados a la noticia respectiva en la Base de Datos.

Cancelar: Control por medio del cual se cancelarán todos los cambios realizados a la noticia, siempre y cuando los cambios no hayan sido ya aceptados con anterioridad, por medio del control Actualizar.

Salir: Control que cierra el proceso actual y direcciona hacia a la pantalla Principal.

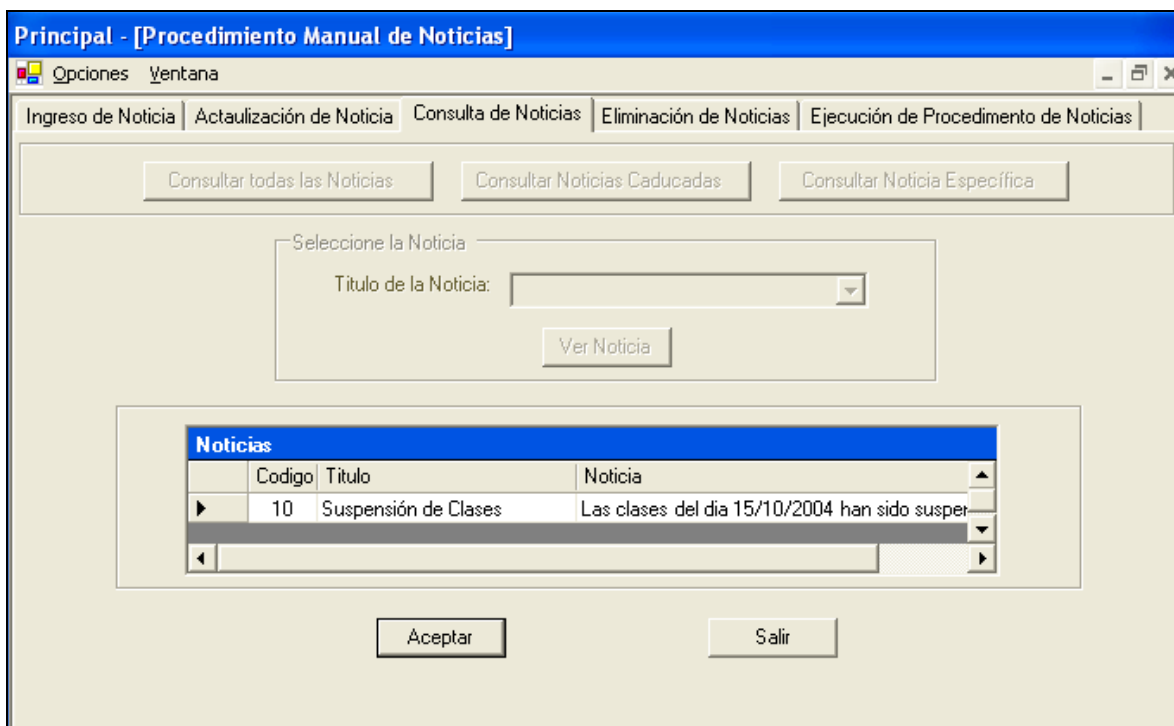


Figura 30. Consulta de Noticias

Mediante esta opción (**Figura 30**) se puede realizar la consulta de las noticias existentes en la Base de Datos. La consulta se puede hacer de tres diferentes formas.

- Consultar Todas las Noticias
 - * De esta manera se desplegarán todas las noticias existentes en la Base de Datos, sin distinción alguna.
- Consultar Noticias Caducadas
 - * Aquí se desplegarán las noticias que hayan salido de su periodo de vigencia, es decir que hayan pasado la fecha de caducidad, para la cual fueron configuradas.
- Consultar Noticia Específica
 - * Este método de consulta, permite seleccionar una noticia específica, de una lista desplegable, que contiene los títulos de las noticias, y ver la información de esta noticia en particular.

Titulo de la Noticia: Nos permite seleccionar el título de la noticia de la cual necesitamos desplegar la información. Campo tipo Lista que contiene los títulos de

todas las noticias existentes en la Base de Datos, Campo de solo lectura.

Noticias: Información de las Noticias. Campo tipo Grilla, que despliega toda la información acerca de las Noticias existentes.

Aceptar: Control que habilita las opciones de consulta, y nos permite seleccionar el tipo de consulta que se va a realizar.

Salir: Control que cierra el proceso actual y direcciona hacia a la pantalla Principal.

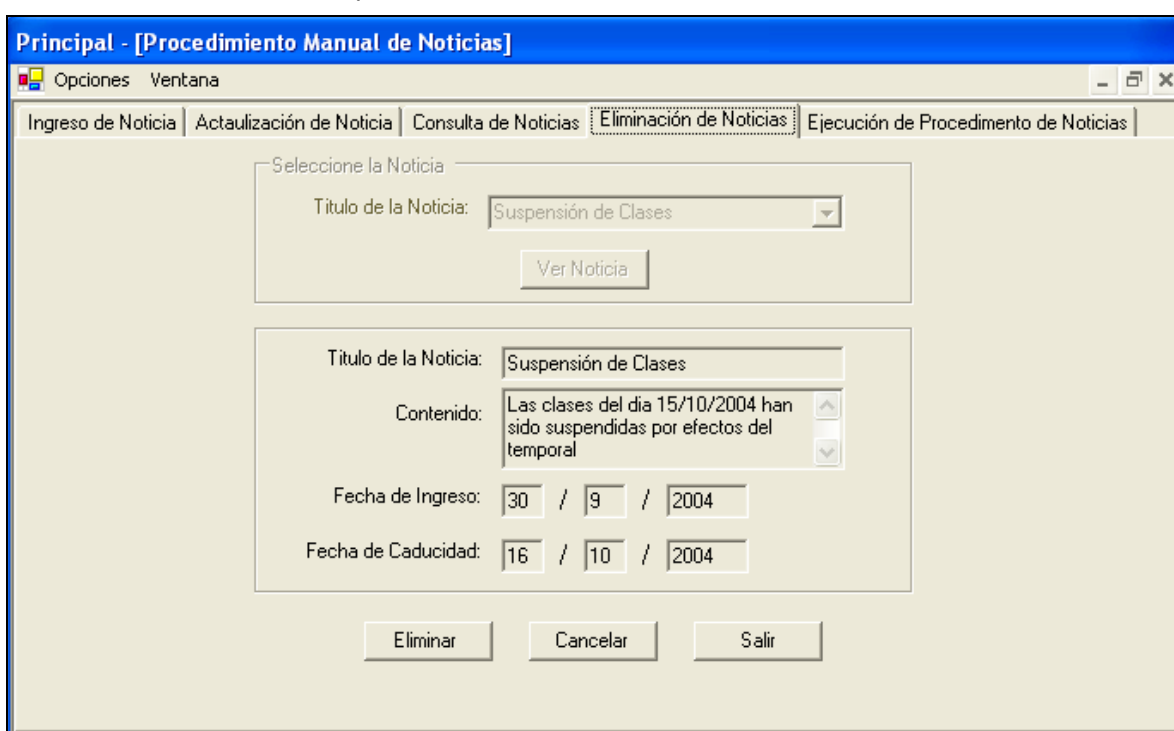


Figura 31. Eliminación de Noticias

Opción mediante la cual se puede realizar la eliminación de una noticia específica seleccionada por el usuario.

Titulo de la Noticia: Campo de selección por título de noticia, para escoger una determinada noticia y desplegar su contenido. Campo tipo Lista para desplegar las Noticias existentes en la Base de Datos. Campo de solo lectura.

Titulo de la Noticia: Campo que contiene el título de la Noticia seleccionada. Campo tipo texto y de solo lectura.

- Contenido:** Campo que contiene la noticia en sí. Campo tipo texto y de solo lectura.
- Fecha de Ingreso:** Fecha en la cual se realizó el ingreso de la Noticia. Campo tipo texto que consta de de tres partes, la primera es para el día, en formato numérico, el segundo corresponde al mes, en formato numérico, y el tercero corresponde al año, en formato numérico. Campo de solo lectura
- Fecha de Caducidad:** Fecha en la cual la Noticia sale de vigencia. Campo tipo texto que consta de de tres partes, la primera es para el día, en formato numérico, el segundo corresponde al mes, en formato numérico, y el tercero corresponde al año, en formato numérico. Campo de solo lectura
- Eliminar:** Control por medio del cual se elimina la noticia respectiva de la Base de Datos.
- Cancelar:** Control por medio del cual se cancelarán la eliminación de la noticia, siempre y cuando no haya sido ya eliminada con anterioridad, por medio del control Eliminar.
- Salir:** Control que cierra el proceso actual y direcciona hacia a la pantalla Principal.

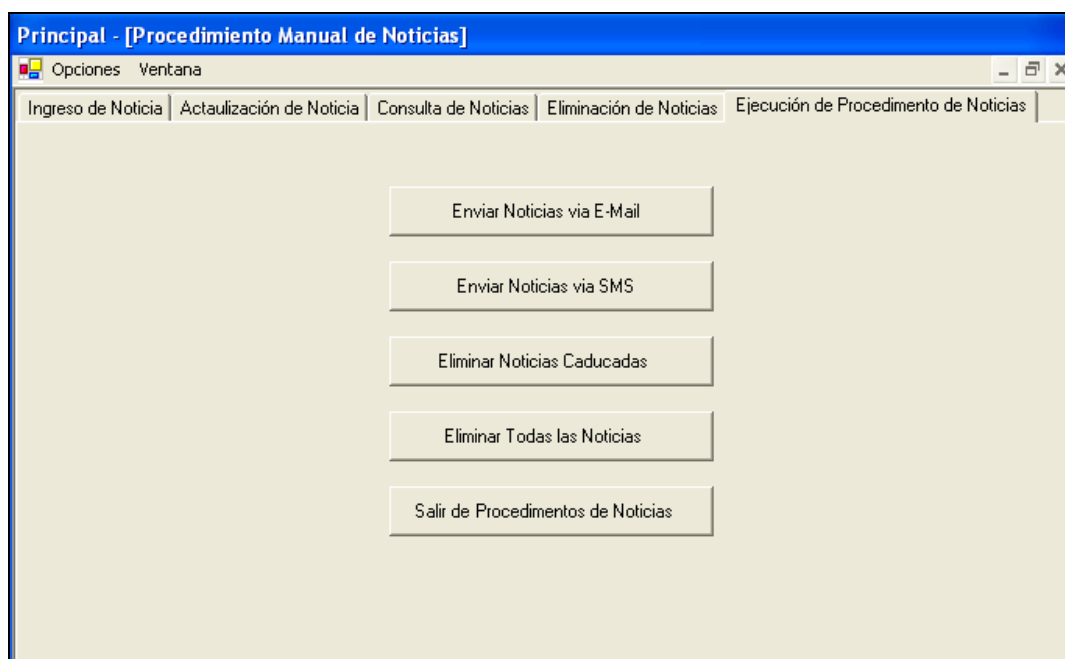
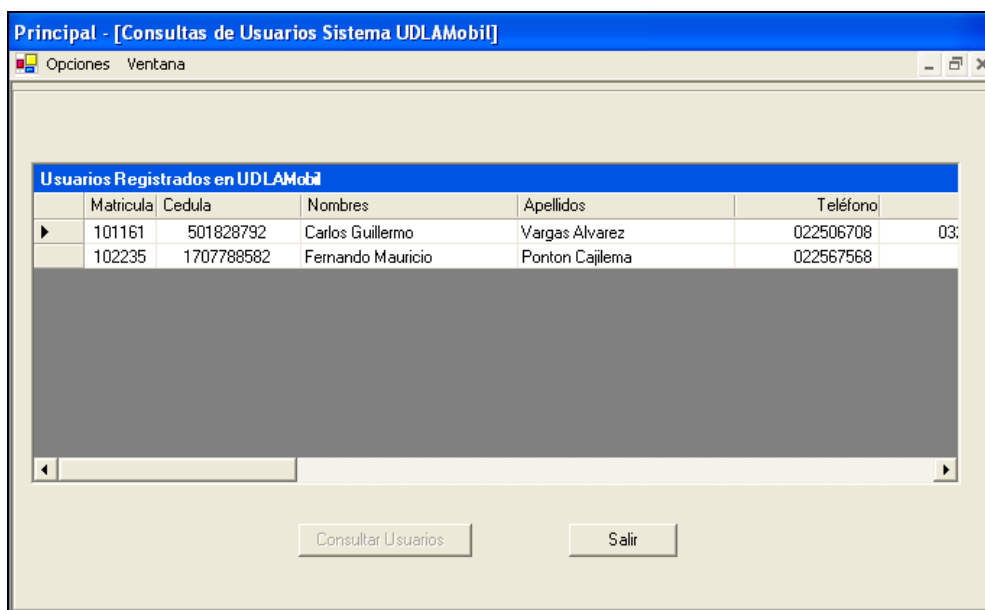


Figura 32. Ejecución de Procedimientos de Noticias

Opción mediante la cual se realiza el envío de las noticias ya sea vía correo electrónico o vía SMS. Adicionalmente aquí se puede realizar la eliminación de las noticias que ya están caducadas, o por su defecto la eliminación de todas las noticias.

- Enviar Noticias vía E-mail
 - * De esta manera, se enviarán las noticias de una fecha determinada hacia sus destinatarios de correo electrónico.
- Enviar Noticias vía SMS
 - * Por medio de esta opción, se hace uso de un Servicio Web de una empresa Integrador, para enviar a esta la información de noticias, y esta a su vez la pase hacia los celulares de los estudiantes registrados.
- Eliminar Noticias Caducadas
 - * De esta manera, se puede eliminar todas las noticias que hayan salido de su periodo de vigencia.
- Eliminar Todas las Noticias
 - * Esta opción permite realizar un vaciado masivo de todas las noticias contenidas en la Base de Datos.
- Salir de Procedimientos de Noticias
 - * Control que cierra el proceso actual y direcciona hacia a la pantalla Principal.

Consulta de Usuarios General (Figura 33)



Usuarios Registrados en UDLAMobil					
	Matricula	Cedula	Nombres	Apellidos	Teléfono
▶	101161	501828792	Carlos Guillermo	Vargas Alvarez	022506708 03
	102235	1707788582	Fernando Mauricio	Ponton Cajilema	022567568

Figura 33. Consulta de Usuarios General

Esta opción nos permite desplegar toda la información de todos los Usuarios Registrados en el sistema UDLAMobil.

Consultar Usuarios: Comando que despliega en una Grilla toda la información de los usuarios registrados en UDLAMobil

Salir: Control que cierra el proceso actual y direcciona hacia a la pantalla Principal.

Consulta de Usuarios Parametrizada (Figura 34)

Figura 34. Consulta de Usuarios Parametrizada

Por medio de esta opción se puede consultar todos los datos de un Usuario Registrado Específico.

Número de Matrícula: Número de matricula en la Universidad de las Américas de un estudiante especificado. Campo tipo texto y longitud máxima de seis caracteres.

Número de Cédula: Número de Identificación personal del estudiante. Campo tipo texto y longitud máxima de diez caracteres. Número de Cédula no debe ser ingresado ni con espacio ni con guión.

Buscar: Control que valida la información ingresada en los campos pedidos y despliega la información del estudiante a continuación.

Cancelar: Control que libera recursos utilizados y permite la búsqueda de información para un nuevo Usuario Registrado.

Salir: Control que cierra el proceso actual y direcciona hacia a la pantalla Principal.

Todos los campos que se despliegan son acerca de la información solicitada y disponible para el Usuarios Especificado. Todos estos Campos son tipo texto y de solo lectura.

Salir

Esta es la Opción de salida de la Aplicación Configurador. Esta opción libera todos los recursos utilizados en la aplicación, encera las variables globales, y también libera los Temporizadores utilizados.

MANUAL TECNICO

INDICE

BASE DE DATOS	158
MOBIL	158
Diagrama entidad relación	158
Tablas	158
Consultas	160
Procedimientos almacenados	161
Diccionario de Datos	162
Tablas	162
Procedimientos Almacenados	167
Componentes	170
Descripción de Componentes	170
Componente ComponentesComunes	170
Componente Componentes	174
Componente ComponentesVb	190
Componente Gestor	194
DESCRIPCION DE FUNCIONES Y PROCEDIMIENTOS	201
Librería de Clases Gestor	201
Librería de Clases Componentes	202
Librería de Clases ComponentesVB	205
Librería de Clases ComponentesComunes	206
Servicio Web UDLAServicio	207
Servicio Web UDLAServicioSMS	210
Aplicación Windows Configurador	211
DESCRIPCION DE VARIABLES GLOBALES	211

BASE DE DATOS

MOBIL

Diagrama entidad relación

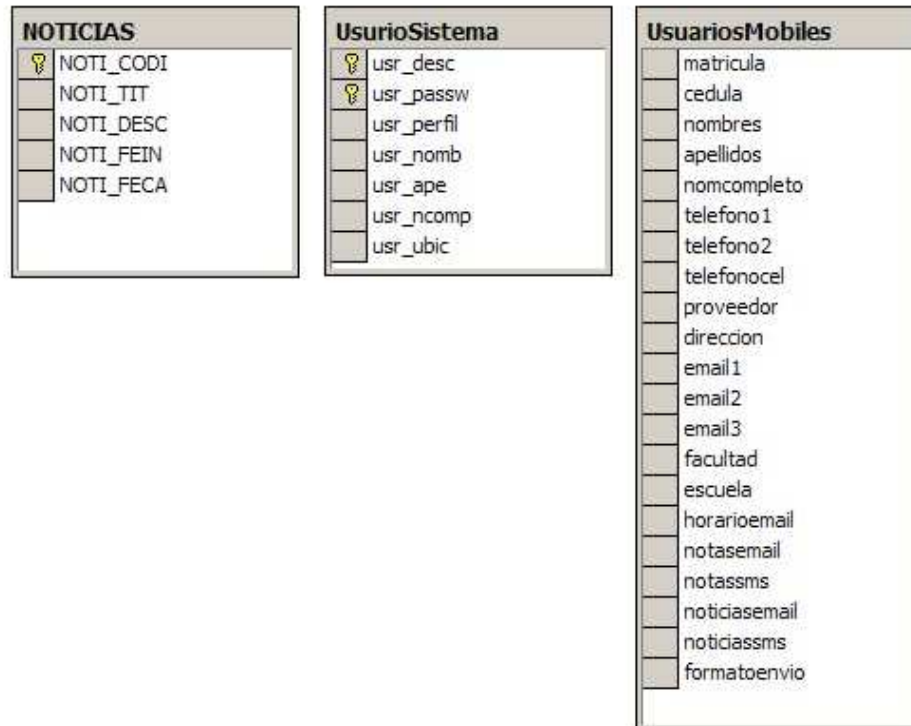


Figura 1. DER base de datos MOBIL

Tablas

Tabla: NOTICIAS

Columnas

Nombre	Tipo	Tamaño
NOTI_CODI	Entero largo	4
NOTI_TIT	Texto	30
NOTI_DESC	Texto	120
NOTI_FEIN	Fecha/Hora	8
NOTI_FECA	Fecha/Hora	8

Tabla: UsuariosMobiles**Propiedades**

DateCreated:	04/10/2004 15:52:46	GUID:	{guid {3E697625- 1CE9-4512- 2EC39E535A04}}
LastUpdated:	04/10/2004 22:11:25	RecordCount:	2
Updatable:	Verdadero		

Columnas

Nombre	Tipo	Tamaño
matricula	Doble	8
cedula	Entero largo	4
nombres	Texto	20
apellidos	Texto	20
nomcompleto	Texto	50
telefono1	Texto	10
telefono2	Texto	10
telefonocel	Texto	10
proveedor	Texto	20
direccion	Texto	50
email1	Texto	35
email2	Texto	35
email3	Texto	35
facultad	Texto	20
escuela	Texto	3
horarioemail	Texto	2
notasemail	Texto	2
notassms	Texto	2
noticiasemail	Texto	2
noticiassms	Texto	2
formatoenvio	Texto	10

Tabla: UsuarioSistema**Columnas**

Nombre	Tipo	Tamaño
usr_desc	Texto	15
usr_passw	Texto	15
usr_perfil	Texto	20
usr_nomb	Texto	20
usr_ape	Texto	20
usr_ncomp	Texto	50
usr_ubic	Texto	50

Consultas**Consulta: consDatosEnvio****Columnas**

Nombre	Tipo	Tamaño
matricula	Doble	8
telefonocel	Texto	10
email1	Texto	35
horarioemail	Texto	2
notasemail	Texto	2
notassms	Texto	2
noticiasemail	Texto	2
noticiassms	Texto	2
formatoenvio	Texto	10

Consulta: consDatosUsuario**Columnas**

Nombre	Tipo	Tamaño
matricula	Doble	8
cedula	Entero largo	4
nombres	Texto	20
apellidos	Texto	20
nomcompleto	Texto	50
telefono1	Texto	10
telefono2	Texto	10
telefonocel	Texto	10
proveedor	Texto	20
direccion	Texto	50
email1	Texto	35
email2	Texto	35
email3	Texto	35
facultad	Texto	20
escuela	Texto	3
horarioemail	Texto	2
notasemail	Texto	2
notassms	Texto	2
noticiasemail	Texto	2
noticiassms	Texto	2
formatoenvio	Texto	10

Consulta: consNoticias**Columnas**

Nombre	Tipo	Tamaño
NOTI_CODI	Entero largo	4
NOTI_TIT	Texto	30
NOTI_DESC	Texto	120
NOTI_FEIN	Fecha/Hora	8
NOTI_FECA	Fecha/Hora	8

Consulta: consUsuario**Propiedades**

DateCreated:	04/10/2004 22:17:29	DefaultView:	Hoja de datos
DOL:	Datos binarios largos 4E10-47F1-B0D9-4A63A4282F0A}}	GUID:	{guid {E724B796-
LastUpdated:	04/10/2004 22:17:29	MaxRecords:	0
ODBCTimeout:	60	OrderByOn:	Falso
Orientation:	De izquierda a derecha	RecordLocks:	Sin bloquear
RecordsAffected:	0	RecordsetType:	Dynaset
ReturnsRecords:	Verdadero	Type:	0
Updatable:	Verdadero		

Columnas

Nombre	Tipo	Tamaño
matricula	Doble	8
cedula	Entero largo	4

Consulta: consUsuarioSistema**Columnas**

Nombre	Tipo	Tamaño
usr_desc	Texto	15
usr_passw	Texto	15

Procedimientos almacenados

Name	Code	Func
SpActualizarDatosUsuario	SpActualizarDatosUsuario	No
SpActualizarNoticia	SpActualizarNoticia	No
SpEliminarNoticia	SpEliminarNoticia	No
SpEliminarNoticiasCaducadas	SpEliminarNoticiasCaducadas	No
SpEliminarNoticiasTodo	SpEliminarNoticiasTodo	No
SpEliminarUsuario	SpEliminarUsuario	No
SpIngresarNoticia	SpIngresarNoticia	No
SpInsertarUsuarioMobil	SpInsertarUsuarioMobil	No
SpSacarDatosEnvio	SpSacarDatosEnvio	No
SpSacarDatosEnvioManual	SpSacarDatosEnvioManual	No
SpSacarDatosUsuario	SpSacarDatosUsuario	No
SpSacarNoticias	SpSacarNoticias	No
SpSacarNoticiasCaducadas	SpSacarNoticiasCaducadas	No
SpSacarNoticiaSeleccionada	SpSacarNoticiaSeleccionada	No
SpSacarNoticiasTodo	SpSacarNoticiasTodo	No
SpSacarTodo	SpSacarTodo	No
SpSacarUsuario	SpSacarUsuario	No
SpVerifUsuarioSistema	SpVerifUsuarioSistema	No

Diccionario de Datos

Tablas

NOTICIAS

NOTI_CODI

Nombre: NOTI_CODI

Descripción: Código de la Noticia

Tipo de Dato: Integer

Longitud: 4

NOTI_TIT

Nombre: NOTI_TIT

Descripción: Título descriptivo para la Noticia

Tipo de Dato: Varchar

Longitud: 30

NOTI_DESC

Nombre: NOTI_DESC

Descripción: Cuerpo o contenido de la noticia

Tipo de Dato: Varchar

Longitud: 120

NOTI_FEIN

Nombre: NOTI_FEIN

Descripción: Fecha de ingreso de la noticia

Tipo de Dato: Date

Longitud: 8

NOTI_FECA

Nombre: NOTI_FECA

Descripción: Fecha de fin de la vigencia de la noticia

Tipo de Dato: Date

Longitud: 8

UsuarioSistema

Usr_desc

Nombre: usr_desc

Descripción: Nombre descriptivo del usuario para el sistema Configurator
Tipo de Dato: Varchar
Longitud: 15

Usr_passw

Nombre: usr_passw
Descripción: Contraseña para el ingreso del usuario al sistema Configurator
Tipo de Dato: Varchar
Longitud: 15

Usr_perfil

Nombre: usr_perfil
Descripción: Perfil del usuario. Existente inicialmente el de Administrador
Tipo de Dato: Varchar
Longitud: 20

Usr_nombre

Nombre: usr_nombre
Descripción: Nombres propios del usuario del sistema Configurator
Tipo de Dato: Varchar
Longitud: 20

Usr_ape

Nombre: usr_ape
Descripción: Apellidos propios del usuario del sistema Configurator
Tipo de Dato: Varchar
Longitud: 20

Usr_ncomp

Nombre: usr_ncomp
Descripción: Nombres completos del Usuario del sistema Configurator
Tipo de Dato: Varchar
Longitud: 50

Usr_ubic

Nombre: usr_ubic
Descripción: Referencia del departamento al que pertenece el Usuario
Tipo de Dato: Varchar
Longitud: 50

UsuariosMobiles

Matricula

Nombre: matricula

Descripción: Número de matrícula en la Universidad de las Américas del usuario

Tipo de Dato: Double

Longitud: 8

Cedula

Nombre: cedula

Descripción: Número de identificación personal del usuario

Tipo de Dato: Integer

Longitud: 4

Nombres

Nombre: nombres

Descripción: Nombres propios del usuario

Tipo de Dato: Varchar

Longitud: 20

Apellidos

Nombre: apellidos

Descripción: Apellidos propios del usuario

Tipo de Dato: Varchar

Longitud: 20

Nomcompleto

Nombre: nomcompleto

Descripción: Nombres Completos del usuario

Tipo de Dato: Varchar

Longitud: 50

Telefono1

Nombre: telefono1

Descripción: Número de teléfono principal del usuario

Tipo de Dato: Varchar

Longitud: 10

Telefono2

Nombre: telefono2
Descripción: Número de teléfono alternativo del usuario
Tipo de Dato: Varchar
Longitud: 10

Telefonocel

Nombre: telefonocel
Descripción: Número de teléfono celular del usuario
Tipo de Dato: Varchar
Longitud: 10

Proveedor

Nombre: proveedor
Descripción: Nombre del Proveedor de telefonía celular del usuario
Tipo de Dato: Varchar
Longitud: 20

Direccion

Nombre: direccion
Descripción: Dirección del domicilio del usuario
Tipo de Dato: Varchar
Longitud: 50

Email1

Nombre: email1
Descripción: Dirección de correo electrónico principal del usuario
Tipo de Dato: Varchar
Longitud: 35

Email2

Nombre: email2
Descripción: Dirección de correo electrónico secundario del usuario
Tipo de Dato: Varchar
Longitud: 35

Email3

Nombre: email3
Descripción: Dirección de correo electrónico alternativo del usuario

Tipo de Dato: Varchar

Longitud: 35

Facultad

Nombre: Facultad

Descripción: Nombre de la facultad a la que pertenece el usuario

Tipo de Dato: Varchar

Longitud: 20

Escuela

Nombre: escuela

Descripción: Código de la escuela a la que pertenece el usuario

Tipo de Dato: Varchar

Longitud: 3

Horarioemail

Nombre: horarioemail

Descripción: Dato verificador para el envío de horarios por medio de correo electrónico

Tipo de Dato: Varchar

Longitud: 2

Notasemail

Nombre: notasemail

Descripción: Dato verificador para el envío de notas por medio de correo electrónico

Tipo de Dato: Varchar

Longitud: 2

Notassms

Nombre: notassms

Descripción: Dato verificador para el envío de notas por medio de mensajes escritos

Tipo de Dato: Varchar

Longitud: 2

Noticiasemail

Nombre: noticiasemail

Descripción: Dato verificador para el envío de noticias de la Universidad de las Américas por medio de correo electrónico

Tipo de Dato: Varchar

Longitud: 2

Noticiassms

Nombre: noticiassms

Descripción: Dato verificador para el envío de noticias de la Universidad de las Américas por medio de mensajes escritos

Tipo de Dato: Varchar

Longitud: 2

Formatoenvio

Nombre: formatoenvio

Descripción: Formato en el cual desea recibir el usuario sus correos electrónicos.

Tipo de Dato: Varchar

Longitud: 10

Procedimientos Almacenados**SpActualizarDatosUsuario**

Nombre: SpActualizarDatosUsuario

Descripción: Procedimiento almacenado que ejecuta la actualización de los datos del usuario en la tabla UsuariosMóviles de la base de datos MOBIL

SpActualizarNoticia

Nombre: SpActualizarNoticia

Descripción: Procedimiento almacenado que ejecuta la actualización de datos de las noticias en la tabla NOTICIAS de la base de datos MOBIL

SpEliminarNoticia

Nombre: SpEliminarNoticia

Descripción: Procedimiento almacenado que ejecuta la eliminación de una noticia especificada en la tabla NOTICIAS de la base de datos MOBIL

SpEliminarNoticiasCaducadas

Nombre: SpEliminarNoticiasCaducadas

Descripción: Procedimiento almacenado que ejecuta la eliminación de todas las noticias que han salido de vigencia en la tabla NOTICIAS de la base de datos MOBIL

SpEliminarNoticiasTodo

Nombre: SpEliminarNoticiasTodo

Descripción: Procedimiento almacenado que ejecuta la eliminación de todas las noticias de la tabla NOTICIAS de la base de datos MOBIL

SpEliminarUsuario

Nombre: SpEliminarUsuario

Descripción: Procedimiento almacenado que ejecuta la eliminación de un usuario determinado de la tabla UsuariosMobiles de la base de datos MOBIL

SpIngresarNoticia

Nombre: SpIngresarNoticia

Descripción: Procedimiento almacenado que ejecuta la inserción de una noticia en la tabla NOTICIAS de la base de datos MOBIL

SpInsertarUsuarioMobil

Nombre: SpInsertarUsuarioMobil

Descripción: Procedimiento almacenado que ejecuta la inserción de un usuario en la tabla UsuariosMobiles de la base de datos MOBIL

SpSacarDatosEnvio

Nombre: SpSacarDatosEnvio

Descripción: Procedimiento almacenado que realiza la extracción de los datos necesarios para realizar los envíos masivos correspondientes a todos los usuarios

SpSacarDatosEnvioManual

Nombre: SpSacarDatosEnvioManual

Descripción: Procedimiento almacenado que realiza la extracción de los datos necesarios para realizar los envíos correspondientes a un usuario en particular

SpSacarDatosUsuario

Nombre: SpSacarDatosUsuario

Descripción: Procedimiento almacenado que realiza la extracción de los datos de los usuarios para consultas

SpSacarNoticias

Nombre: SpSacarNoticias

Descripción: Procedimiento almacenado que realiza la extracción de las noticias para consultas

SpSacarNoticiasCaducadas

Nombre: SpSacarNoticiasCaducadas

Descripción: Procedimiento almacenado que realiza la extracción de las noticias que han salido de su periodo de vigencia para consultas

SpSacarNoticiaSeleccionada

Nombre: SpSacarNoticiaSeleccionada

Descripción: Procedimiento almacenado que realiza la extracción de una noticia en particular para consultas

SpSacarNoticiasTodo

Nombre: SpSacarNoticiasTodo

Descripción: Procedimiento almacenado que realiza la extracción de todas las noticias para consultas

SpSacarTodo

Nombre: SpSacarTodo

Descripción: Procedimiento almacenado que realiza la extracción de toda la información referente a todos los usuarios para consulta

SpSacarUsuario

Nombre: SpSacarUsuario

Descripción: Procedimiento almacenado que realiza la verificación de la existencia de un usuario en la tabla UsuariosMóviles de la base de datos MOBIL

SpVerifUsuarioSistema

Nombre: SpVerifUsuarioSistema

Descripción: Procedimiento almacenado que realiza la verificación de la existencia de un usuario con respecto de sus contraseña en la tabla UsuarioSistema de la base de datos MOBIL

Componentes

Descripción de Componentes

Componente ComponentesComunes

Documentación

Componente que agrupa las clases Conexión, Envío y Seguridades.

Clases asignadas	Conexion	Envio	Seguridades
-------------------------	--------------------------	-----------------------	-----------------------------

Configuración

Modelado de datos

dmlItem	False	DMName	
IsDatabase	False	TargetDatabase	
Location		IsTableSpace	False
TableSpaceType		IsDeault	False
BufferPool		ExtentSize	1
PrefetchSize	1	PageSize	4
ManagedBy		ContainerList	



Clase Conexion

Documentación

Clase que permite la conexión entre las bases de datos el resto de componentes. Contiene las cadenas de conexión necesarias, tanto a la base de datos SAES como a la base de datos MOBIL.

Componente Asignado	ComponentesComunes
----------------------------	--------------------

Métodos

Nombre	Definición	Clase
 miCon	sqlConnection  miCon ()	Conexion

 miConM	sqlConnection	 miConM ()	Conexion
--	---------------	---	--------------------------

Método sqlConnection  miCon (), en Clase [Conexion](#)

Documentación

Método que retorna la cadena de conexión a la base de datos SAES.

Tipo de dato de retorno

sqlConnection

Argumentos

Ninguno

Método sqlConnection  miConM (), en Clase [Conexion](#)

Documentación

Método que retorna la cadena de conexión a la base de datos MOBIL.

Tipo de dato de retorno

sqlConnection

Argumentos

Ninguno



Clase Seguridades



Documentación

Clase que se encarga de validar las credenciales que el usuario ingresa para acceder al sistema. Controla el acceso a la información requerida.

Componente Asignado	ComponentesComunes
----------------------------	--------------------

Métodos

Name	Signatura	Class
 VerUsuario	String  VerUsuario (Float mat, Int id)	Seguridades

 VerUsuarioM	String  VerUsuarioM (float mat, int id)	Seguridades
---	--	-----------------------------

Método String  VerUsuario (Float mat, Int id), en Clase [Seguridades](#)

Documentación

Método que retorna una cadena con el número de matrícula del alumno.

Dato de retorno

String


Argumentos

mat

Número de matrícula del alumno.

id

Cédula de identidad del alumno.

Método String  VerUsuarioM (float mat, int id), en Clase [Seguridades](#)

Documentación

Método que retorna el número de matrícula del alumno en la base de datos MOBIL.

Dato de retorno

String

Argumentos

mat

Número de matrícula del alumno.

id

Número de cédula del alumno.



Clase Envio

Documentación

Clase genérica que se encarga del envío de correo electrónico vía SMTP (Simple Mail Transfer Protocol)

Componente Asignado	ComponentesComunes
----------------------------	------------------------------------

Métodos

Name	Signatura	Class
 EnviarMail	void  EnviarMail (string de, string para, string asun, string mens, string form, string serv)	Envio

Método void EnviarMail

(string de, string para, string asun, string mens, string form, string serv), en Clase [Envio](#)

Documentación

Método que envía correo electrónico.

Dato de retorno

Void

Argumentos

de

El remitente del correo electrónico.

para

El destinatario del correo electrónico.

asun

El asunto o el tema del correo electrónico.

mens

El mensaje del correo electrónico.

form

El formato del correo electrónico (texto o Html)

serv

El servidor del correo electrónico.

Componente Componentes**Documentación**

Componente que agrupa las clases Alumnos, General, Materias, Notas, Noticias, y UsuarioMobil.

Clases asignadas	Alumnos	General	Materias
Notas	Noticias	UsuarioMobil	

Configuración**Modelado de datos**

dmlItem	False	DMName	
IsDatabase	False	TargetDatabase	
Location		IsTableSpace	False
TableSpaceType		IsDeault	False
BufferPool		ExtentSize	1
PrefetchSize	1	PageSize	4
ManagedBy		ContainerList	

Clase Alumnos**Documentación**

Clase destinada a obtener la información completa de un alumno.

Componente Asignado	Componentes
---------------------	-----------------------------

Métodos

Name	Signatura	Class
 SacarAlumnos	DataSet  SacarAlumnos (float nmat)	Alumnos

Método DataSet  SacarAlumnos (float nmat), en Clase [Alumnos](#)

Documentation

Método para obtener la información por alumno.

Dato de retorno

DataSet

Argumentos

nmat

Número de matrícula del alumno.

Clase General

Documentación

Clase destinada a dar formato a un texto determinado.

Componente Asignado	Componentes
---------------------	-----------------------------

Métodos

Name	Signatura	Class
 FormatoMensaje	String  FormatoMensaje (String Mensaje)	General

Operation String  FormatoMensaje (String Mensaje), en Clase [General](#)

Documentación

Método que convierte una cadena de texto a minúsculas.

Dato de retorno

String

Argumentos

Mensaje

Cadena de caracteres a ser convertidas a minúsculas.

Clase Materias


Documentación

Clase destinada a la obtención de materias por alumno.

Componente Asignado	Componentes
---------------------	-----------------------------

Métodos

Name	Signatura	Class
 SacarMaterias	DataSet  SacarMaterias (Float mat, Int anio, Char per)	Materias

Método DataSet  SacarMaterias (Float mat, Int anio, Char per), en Clase [Materias](#)

Documentación

Método destinado a la obtención de materias por estudiante.

Dato de retorno

DataSet

Argumentos

mat

Número de matrícula del estudiante como parámetro de consulta.

anio

Año como parámetro de consulta.

per

Período como parámetro de consulta.







Clase Notas


Documentación

Clase destinada a la obtención de calificaciones.

Componente Asignado	Componentes
----------------------------	-----------------------------

Métodos

Name	Signatura	Class
 SacarNotas	DataSet  SacarNotas (Float mat, Char tiponota, Int anio, Char per)	Notas
 SacarNotasMateria	DataSet  SacarNotasMateria (Float mat, String asig, Int anio, Char per)	Notas
 SacarNotasTodo	DataSet  SacarNotasTodo (Float mat, Int anio, Char per)	Notas

Método DataSet  SacarNotas (Float mat, Char tiponota, Int anio, Char per), en Clase [Notas](#)

Documentación

Método para obtener las notas por alumno dentro de un año, período y tipo de nota determinados.

Dato de retorno

DataSet

Argumentos**mat**

Número de matrícula del estudiante como parámetro de consulta.

tiponota

Tipo de nota a obtener como parámetro de consulta.

anio

Año como parámetro de consulta.

per

Período como parámetro de consulta.

Método DataSet 

SacarNotasMateria (Float mat, String asig, Int anio, Char per), en Clase [Notas](#)

Documentación

Método para obtener las notas por alumno dentro de un año, período y asignatura determinados.

Dato de retorno

DataSet

Argumentos**mat**

Número de matrícula del estudiante como parámetro de consulta.

asig


Código de la asignatura como parámetro de consulta.

anio

Año como parámetro de consulta.

per

Período como parámetro de consulta.

Método DataSet  **SacarNotasTodo (Float mat, Int anio, Char per), en Clase [Notas](#)**

Documentación

Método para obtener todas las notas del alumno en un año y período de tiempo determinado.

Dato de retorno

DataSet

Argumentos**mat**

Número de matrícula del estudiante.

anio

Año como parámetro de consulta.

per



Período como parámetro de consulta.

Class Noticias**Documentación**

Clase destinada al manejo de las noticias.

Componente Asignado	Componentes
----------------------------	-----------------------------

Métodos

Name	Signatura	Class
 SacarNoticias	DataSet  SacarNoticias (DateTime Fecha)	Noticias

SacarNoticiaSeleccionada	DataSet  SacarNoticiaSeleccionada (int Codigo)	Noticias
SacarNoticiasTodo	DataSet  SacarNoticiasTodo ()	Noticias
SacarNoticiasCaducadas	DataSet  SacarNoticiasCaducadas (DateTime Fecha)	Noticias
IngresarNoticia	void  IngresarNoticia (String Titulo, String Noticia, DateTime FechaIng, DateTime FechaCad)	Noticias
ActualizarNoticia	void  ActualizarNoticia (int Codigo, String Titulo, String Noticia, DateTime FechaIng, DateTime FechaCad)	Noticias
EliminarNoticia	void  EliminarNoticia (Int Codigo)	Noticias
EliminarNoticiasCaducadas	void  EliminarNoticiasCaducadas (DateTime Fecha)	Noticias
EliminarNoticiasTodo	void  EliminarNoticiasTodo ()	Noticias

Método DataSet  SacarNoticias (DateTime Fecha), en Clase [Noticias](#)

Documentación

Método que obtiene en un dataset todas las noticias de acuerdo a una fecha determinada.

Dato de retorno

DataSet

Argumentos

Fecha

Fecha como parámetro de consulta.

Método DataSet  SacarNoticiaSeleccionada (int Codigo), en Clase [Noticias](#)

Documentación

Método que obtiene en un dataset una noticia en particular.

Dato de retorno

DataSet

Argumentos**Codigo**

El identificador de la noticia.

Método DataSet  **SacarNoticiasTodo ()**, en Clase [Noticias](#)

Documentación

Método que obtiene en un dataset todas las noticias.

Dato de retorno

DataSet

Argumentos

Ninguno

Método DataSet  **SacarNoticiasCaducadas (DateTime Fecha)**, en Clase [Noticias](#)

Documentación

Método que obtiene en un dataset todas las noticias caducadas de acuerdo a una fecha determinada.

Dato de retorno

DataSet

Argumentos**Fecha**

La fecha como parámetro de consulta.

Método void  **IngresarNoticia (String Titulo, String Noticia, DateTime FechaIng, DateTime FechaCad), en Clase [Noticias](#)**

Documentación

Método para insertar una noticia.

Dato de retorno

Void

Argumentos

Titulo

Inserta el encabezado de la noticia.

Noticia


Inserta el cuerpo de la noticia.

FechaIng

Inserta la fecha de ingreso de la noticia.

FechaCad

Inserta la fecha de caducidad de la noticia.

Método void  **ActualizarNoticia (int Codigo, String Titulo, String Noticia, DateTime FechaIng, DateTime FechaCad), en Clase [Noticias](#)**

Documentación

Método para actualizar los datos de una noticia en particular.

Dato de retorno

Void

Argumentos

Codigo

Actualiza el código de la noticia.

Titulo

Actualiza el encabezado de la noticia.

Noticia


Actualiza el cuerpo de la noticia.

FechaIng

Actualiza la fecha de ingreso de la noticia.

FechaCad

Actualiza la fecha de caducidad de la noticia.

Método void  **EliminarNoticia (Int Codigo), en Clase [Noticias](#)**

Documentación

Método para eliminar una noticia en particular.

Dato de retorno

Void

Argumentos**Codigo**

El código de la noticia.

Método void  **EliminarNoticiasCaducadas (DateTime Fecha), en Clase [Noticias](#)**

Documentación

Método para eliminar todas las noticias caducadas.

Dato de retorno

Void

Argumentos

Fecha

Fecha de referencia para la eliminación de las noticias caducadas.

Método void  EliminarNoticiasTodo (), en Clase [Noticias](#)

Documentación

Método para eliminar todas las noticias.

Dato de retorno

Void

Argumentos



Ninguno

Clase UsuarioMobil

Documentación

Clase destinada para el manejo de la base de datos MOBIL.

Componente Asignado [Componentes](#)

Name	Signature	Class
SacarDatos	DataSet SacarDatos (Float mat)	UsuarioMobil
SacarTodo	DataSet SacarTodo ()	UsuarioMobil
InsertaNuevo	 id InsertaNuevo (Float mat, Int ced, String nom, String ape, String tel1, String tel2, String telcel, String prov, String dir, String em1, String em2, String em3, String fac, String esc, String hoem, String notem, String notisms, String notiformen)	UsuarioMobil
ActualizarDatos	 id ActualizarDatos (Float mat, Int ced, String nom, String ape, String tel1, String tel2, String telcel, String prov, String dir, String em1, String em2, String em3, String fac, String esc, String hoem, String notem, String notisms, String notiformen)	UsuarioMobil
EliminarUsuario	void EliminarUsuario (float mat, Int ced)	UsuarioMobil

Método DataSet  **SacarDatos** (Float mat), en Clase [UsuarioMobil](#)

Documentation

Método que obtiene los datos de un estudiante en particular.

Dato de retorno

DataSet

Arguments

mat

Número de matrícula del estudiante.

Método DataSet  **SacarTodo** (), en Clase [UsuarioMobil](#)

Documentación


Método que devuelve todos los datos de usuarios móviles.

Dato de retorno

DataSet

Argumentos

Ninguno

Método void  **InsertaNuevo** (Float mat, Int ced, String nom, String ape, String tel1, String tel2, String telcel, String prov, String dir, String em1, String em2, String em3, String fac, String esc, String hoem, String notem, String notsms, String notiem, String notisms, String formen), en Clase [UsuarioMobil](#)

Documentación

Método que inserta los datos de un usuario nuevo en la base de datos de UsuarioMobil.

Dato de retorno

Void

Argumentos**mat**

Inserta el número de matrícula de estudiante.

ced

Inserta el número de cédula del estudiante.

nom

Inserta el nombre del estudiante.

ape

Inserta el apellido del estudiante.

tel1

Inserta el número de teléfono principal del estudiante.

tel2

Inserta el número de teléfono secundario del estudiante.

telcel

Inserta el número de teléfono celular del estudiante.

prov

Inserta el proveedor de telefonía celular a quien pertenece el número de celular del estudiante.

dir

Inserta la dirección del estudiante.

em1

Inserta email primario del estudiante.

em2

Inserta email secundario del estudiante.

em3

Inserta email alternativo del estudiante.

fac

Inserta la facultad a la que pertenece el estudiante.

esc

Inserta la escuela a la que pertenece el estudiante.

hoem

Inserta si el estudiante registrado desea recibir su horario via email.

notem

Inserta si el estudiante registrado desea recibir sus calificaciones via email.

notsms

Inserta si el estudiante registrado desea recibir sus calificaciones via sms.

notiem


Inserta si el estudiante registrado desea recibir noticias via email.

notisms

Inserta si el estudiante registrado desea recibir noticias via sms.

formen

Inserta en qué formato el estudiante desea recibir su información, texto o Html.

Método void  ActualizarDatos (Float mat, Int ced, String nom, String ape, String tel1, String tel2, String telcel, String prov, String dir, String em1, String em2, String em3, String fac, String esc, String hoem, String notem, String notsms, String notiem, String notisms, String formen), en Clase [UsuarioMobil](#)

Documentación

Método que actualiza los datos de un estudiante registrado.

Dato de retorno

Void

Argumentos**mat**

Actualiza el número de matrícula de estudiante.

ced

Actualiza el número de cédula del estudiante.

nom

Actualiza el nombre del estudiante.

ape

Actualiza el apellido del estudiante.

tel1

Actualiza el número de teléfono principal del estudiante.

tel2

Actualiza el número de teléfono secundario del estudiante.

telcel

Actualiza el número de teléfono celular del estudiante.

prov

Actualiza el proveedor de telefonía celular a quien pertenece el número de celular del estudiante.

dir

Actualiza la dirección del estudiante.

em1

Actualiza email primario del estudiante.

em2

Actualiza email secundario del estudiante.

em3

Actualiza email alterno del estudiante.

fac

Actualiza la facultad a la que pertenece el estudiante.

esc

Actualiza la escuela a la que pertenece el estudiante.

hoem

Actualiza si el estudiante registrado desea recibir su horario via email.

notem

Actualiza si el estudiante registrado desea recibir sus calificaciones via email.

notsms

Actualiza si el estudiante registrado desea recibir sus calificaciones via sms.

notiem

Actualiza si el estudiante registrado desea recibir noticias via email.

notisms

Actualiza si el estudiante registrado desea recibir noticias via sms.

formen

Actualiza en qué formato el estudiante desea recibir su información, texto o Html.

Método void  **EliminarUsuario (float mat, Int ced), en Clase [UsuarioMobil](#)**

Documentación

Método que elimina los datos de un estudiante registrado.

Dato de retorno

Void

Argumentos

mat

Número de matrícula del estudiante.

ced

Número de cédula del estudiante.

Componente ComponentesVb

Documentación

Componente que agrupa las clases FormatMsg y Horarios

Clases asignadas	FormatMsg	Horarios
-------------------------	---------------------------	--------------------------

Configuración

Modelado de datos

dmlItem	False	DMName	
IsDatabase	False	TargetDatabase	
Location		IsTableSpace	False
TableSpaceType		IsDeault	False
BufferPool		ExtentSize	1
PrefetchSize	1	PageSize	4
ManagedBy		ContainerList	





Clase FormatMsg


Documentación

Clase destinada a dar forma a los mensajes a ser enviados por correo electrónico. Los métodos reciben datasets, los cuales son transformados a texto, listos para ser enviados por email.

Componente Asignado	ComponentesVb
----------------------------	-------------------------------

Métodos

Name	Signatura	Class
FormatMsgText	String  FormatMsgText (DataSet DS)	FormatMsg
FormatMsgHtml	String  FormatMsgHtml (DataSet DS)	FormatMsg
SendNotasTablaHtml	String  SendNotasTablaHtml (DataSet DS)	FormatMsg
SendCargaTablaHtml	String  SendCargaTablaHtml (DataSet DS)	FormatMsg

Método String  FormatMsgText (DataSet DS), en Clase [FormatMsg](#)

Documentación

Método genérico para convertir un dataset en formato Texto


Dato de retorno

String

Arguments

DS

Dataset genérico.

Método String  FormatMsgHtml (DataSet DS), en Clase [FormatMsg](#)

Documentation

Método genérico para convertir un dataset en formato Html

Dato de retorno

String

Arguments**DS**

Dataset genérico.

Método String  **SendNotasTablaHtml (DataSet DS), en Clase [FormatMsg](#)**

Documentación

Método que devuelve las notas del alumno en formato Html.

Dato de retorno

String

Arguments**DS**

Dataset que contiene las notas del alumno.

Método String  **SendCargaTablaHtml (DataSet DS), en Clase [FormatMsg](#)**

Documentación

Método que devuelve la carga horaria en formato Html.

Dato de retorno

String

Argumentos**DS**

Dataset que contiene la carga horaria del alumno.





Clase Horarios


Documentación

Clase que se encarga de retornar los horarios de los alumnos

Assigned Components	ComponentesVb
----------------------------	-------------------------------

Métodos

Name	Signatura	Class
 SacarHorario	DataSet  SacarHorario (decimal nmat, integer anio, Char per)	Horarios
 SacarHorarioPorMateria	DataSet  SacarHorarioPorMateria (decimal nmat, integer anio, char per, string asig)	Horarios

Método DataSet  SacarHorario (decimal nmat, integer anio, Char per), en Clase [Horarios](#)

Documentación

Método que obtiene el conjunto de datos del horario completo del estudiante

Dato de retorno

DataSet

Argumentos

nmat


Número de matrícula del estudiante.

anio

El año como parámetro de consulta

per

El período como parámetro de consulta.

Método DataSet  **SacarHorarioPorMateria**
(decimal nmat, integer anio, char per, string asig), en Clase [Horarios](#)

Documentación

Método que obtiene el conjunto de datos del horario del estudiante por materia

Dato de retorno

DataSet

Argumentos**nmat**

Número de matrícula del estudiante.

anio

Año como parámetro de consulta.

per

Período como parámetro de consulta.

asig

Código de la asignatura como parámetro de consulta.

Componente Gestor**Documentación**

El componente Gestor agrupa la clase "maestra" del sistema, se apoya en el resto de componentes para lograr la configuración de todo el proyecto.

Clases asignadas	Gestor
-------------------------	------------------------

Configuración

Modelado de datos

dmlItem	False	DMName	
IsDatabase	False	TargetDatabase	
Location		IsTableSpace	False
TableSpaceType		IsDeault	False
BufferPool		ExtentSize	1
PrefetchSize	1	PageSize	4
ManagedBy		ContainerList	









Clase Gestor





Documentación

Clase encargada de manejar los componentes. Representa la lógica del sistema.

Componente Asignado	Gestor
----------------------------	------------------------

Métodos

Name	Signatura	Class
 EnvioMasivoNotasMail	void  EnvioMasivoNotasMail (Char tnota, Int anio, Char per, String de, String asun, String serv)	Gestor
 EnvioManualNotasMail	void  EnvioManualNotasMail (nmat, ced, tnota, anio, per, de, asun, serv)	Gestor
 EnvioMasivoHorariosMail	void  EnvioMasivoHorariosMail (Int anio, Char per, String de, String asun, String serv)	Gestor
 EnvioManualHorariosMail	void  EnvioManualHorariosMail (Float nmat, Int ced, Int anio, Char per,	Gestor

	String de, String asun, String serv)	
 EnvioMasivoNoticiasMail	void  EnvioMasivoNoticiasMail (DateTime Fecha, String de, String asun, String serv)	Gestor
 EnvioMasivoNoticiasSMS	void  EnvioMasivoNoticiasSMS (DateTime Fecha, String de)	Gestor

Método void  **EnvioMasivoNotasMail**

(Char tnota, Int anio, Char per, String de, String asun, String serv), en Clase [Gestor](#)

Documentación

Método para enviar notas masivamente por correo electrónico.

Dato de retorno

Void

Argumentos

tnota

Tipo de nota como parámetro de consulta.

anio

Año como parámetro de consulta.

per

Período como parámetro de consulta.

de

Remitente como parámetro de envío de correo electrónico.

asun

Mensaje como parámetro de envío de correo electrónico.

serv

Servidor como parámetro de envío de correo electrónico.

Método void  **EnvioManualNotasMail**

(**nmat**, **ced**, **tnota**, **anio**, **per**, **de**, **asun**, **serv**), en Clase [Gestor](#)

Documentación

Método para enviar notas de forma manual por correo electrónico.

Dato de retorno

Void

Argumentos**nmat**

Número de matrícula del estudiante como parámetro de consulta.

ced

Cédula del estudiante como parámetro de consulta.

tnota

Tipo de nota como parámetro de consulta.

anio

Año como parámetro de consulta.

per

Período como parámetro de consulta.

de

Remitente como parámetro de envío de correo electrónico.

asun

Asunto como parámetro de envío de correo electrónico.

serv

Servidor como parámetro de envío de correo electrónico.

Método void  **EnvioMasivoHorariosMail**

(Int anio, Char per, String de, String asun, String serv), en Clase [Gestor](#)

Documentación

Método para enviar horarios de forma masiva por correo electrónico.

Dato de retorno

Void

Argumentos

anio

Año como parámetro de consulta.

per

Período como parámetro de consulta.

de

Remitente como parámetro de envío de correo electrónico.

asun

Asunto como parámetro de envío de correo electrónico.

serv

Servidor como parámetro de envío de correo electrónico.

Método void  **EnvioManualHorariosMail**

(Float nmat, Int ced, Int anio, Char per, String de, String asun, String serv),
en Clase [Gestor](#)

Documentation

Método para enviar horarios de forma manual por correo electrónico.

Dato de retorno

Void

Argumentos**nmat**

Número de matrícula del estudiante como parámetro.

ced

Número de cédula del estudiante como parámetro.

anio

Año como parámetro de consulta.

per

Período como parámetro de consulta.

de

Remitente como parámetro de envío de correo electrónico.

asun

Asunto como parámetro de envío de correo electrónico.

serv

Servidor como parámetro de envío de correo electrónico.

Método void  **EnvioMasivoNoticiasMail**

(DateTime Fecha, String de, String asun, String serv), en Clase [Gestor](#)

Documentación

Método para enviar noticias de forma masiva por correo electrónico.

Dato de retorno

Void

Argumentos**Fecha**

Fecha como parámetro de consulta.

de


Remitente como parámetro de envío de correo electrónico.

asun

Asunto como parámetro de envío de correo electrónico.

serv

Servidor como parámetro de envío de correo electrónico.

Método void  **EnvioMasivoNoticiasSMS (DateTime Fecha, String de), en Clase [Gestor](#)**

Documentación

Método para enviar noticias de forma masiva por mensajes de texto de celular.

Dato de retorno

Void

Argumentos**Fecha**

Fecha como parámetro de consulta,

de

Remitente como parámetro de envío de mensaje.

DESCRIPCION DE FUNCIONES Y PROCEDIMIENTOS

En el Diseño de las Aplicaciones se han utilizado las siguientes funciones y procedimientos definidos por el usuario:

Librería de Clases Gestor

Clase SeguridadesG

Función	Descripción
VerUsuarioSistema	Verifica la existencia de un Usuario del Sistema Configurator. Tiene como parámetros de entrada un Usr del tipo string y un Passw del tipo String. Utiliza el Store Procedure SpVerifUsuarioSistema. Retorna un valor de 1 si el usuario existe y 0 si no existe.

Clase Envios

Función	Descripción
EnvioMasivoNotasMail	Realiza el envío de notas parciales por vía correo electrónico. Tiene como parámetros de entrada tnota del tipo char, anio del tipo int, per del tipo char, de del tipo string, asun del tipo string y serv del tipo string. Utiliza el Store Procedure SpSacarDatosEnvio.
EnvioMasivoNotasMail	Realiza el envío de notas finales por vía correo electrónico. Tiene como parámetros de entrada anio del tipo int, per del tipo char, de del tipo string, asun del tipo string y serv del tipo string. Utiliza el Store Procedure SpSacarDatosEnvio.
EnvioManulNotasMail	Realiza el envío de notas parciales para un alumno específico por vía correo electrónico. Tiene como parámetros de entrada nmat del tipo flota, ced del tipo int, tnota del tipo char, anio del tipo int, per del tipo char, de del tipo string, asun del tipo string y serv del tipo string. Utiliza el Store Procedure SpSacarDatosEnvio.
EnvioManulNotasMail	Realiza el envío de notas finales para un alumno específico por vía correo electrónico. Tiene como parámetros de entrada nmat del tipo flota, ced del tipo int, anio del tipo int, per del tipo char, de del tipo string, asun del tipo string y

Función	Descripción
EnvioMasivoHorariosMail	serv del tipo string. Utiliza el Store Procedure SpSacarDatosEnvio. Realiza el envío de Horario por vía correo electrónico. Tiene como parámetros de entrada anio del tipo int, per del tipo char, de del tipo string, asun del tipo string y serv del tipo string. Utiliza el Store Procedure SpSacarDatosEnvio.
EnvioManualHorariosMail	Realiza el envío de Horario para un alumno específico por vía correo electrónico. Tiene como parámetros de entrada nmat del tipo flota, ced del tipo int anio del tipo int, per del tipo char, de del tipo string, asun del tipo string y serv del tipo string. Utiliza el Store Procedure SpSacarDatosEnvio.
EnvioMasivoNoticiasMail	Realiza el envío de Noticias por vía correo electrónico. Tiene como parámetros de entrada Fecha del tipo DateTime, de del tipo string, asun del tipo string y serv del tipo string. Utiliza el Store Procedure SpSacarDatosEnvio.
EnvioMasivoNoticiasSMS	Realiza el envío de Noticias por SMS. Tiene como parámetros de entrada Fecha del tipo DateTime y de del tipo string. Utiliza el Store Procedure SpSacarDatosEnvio. Esta Función debe hacer referencia hacia el Servicio Web de la empresa Integradora, con la cual se hará el envío de datos hacia los celulares.

Librería de Clases Componentes

Clase Alumnos

Función	Descripción
SacarAlumnos	Obtiene la información de un alumno determinado por su número de matrícula. Tiene como parámetros de entrada, nmat del tipo float. Utiliza el Store Procedure SpSacarDatosAlumno. Retorna un DataSet con la información solicitada.

Clase General

Función	Descripción
FormatoMensaje	Transforma un mensaje en un solo tipo de escritura. Tiene como parámetro de entrada, Mensaje del tipo string. Retorna un string con el mensaje formateado.
Clase Materias	
Función	Descripción
SacarMaterias	Obtiene la información de materias de un alumno determinado por su número de matrícula. Tiene como parámetros de entrada, nmat del tipo float, anio del tipo int y per del tipo char. Utiliza el Store Procedure SpSacarMateriasPorAlumno. Retorna un DataSet con la información solicitada.
Clase Notas	
Función	Descripción
SacarNotas	Obtiene la información de notas por tipo de nota de un alumno determinado por su número de matrícula. Tiene como parámetros de entrada, nmat del tipo float, tiponota del tipo char, anio del tipo int y per del tipo char. Utiliza el Store Procedure SpSacarNotas. Retorna un DataSet con la información solicitada.
SacarNotasMateria	Obtiene la información de notas por materia de un alumno determinado por su número de matrícula. Tiene como parámetros de entrada, nmat del tipo float, asig del tipo string, anio del tipo int y per del tipo char. Utiliza el Store Procedure SpSacarNotasMateria. Retorna un DataSet con la información solicitada.
SacarNotasTodo	Obtiene la información de notas de un alumno determinado por su número de matrícula. Tiene como parámetros de entrada, nmat del tipo float, anio del tipo int y per del tipo char. Utiliza el Store Procedure SpSacarNotas. Retorna un DataSet con la información solicitada.
Clase Noticias	

Función	Descripción
SacarNoticiaSeleccionada	Obtiene la información de una noticia determinada por su código. Tiene como parámetros de entrada, Código del tipo int. Utiliza el Store Procedure SpSacarNoticiaSeleccionada. Retorna un DataSet con la información solicitada.
SacarNoticiasTodo	Obtiene la información de todas las noticias. No tiene parámetros de entrada, Utiliza el Store Procedure SpSacarNoticiasTodo. Retorna un DataSet con la información solicitada.
SacarNoticiasCaducadas	Obtiene la información de las noticias que estén fuera del periodo de vigencia. Tiene como parámetros de entrada, Fecha del tipo DateTime. Utiliza el Store Procedure SpSacarNoticiasCaducadas. Retorna un DataSet con la información solicitada.
IngresarNoticia	Realiza el ingreso de una noticia en la tabla NOTICIAS de la Base de Datos MOBIL. Tiene como parámetros de entrada, Título del tipo string, Noticia del tipo string, Fecha_Ingreso del tipo string, Fecha_Caducidad del tipo string. Utiliza el Store Procedure SpIngresarNoticia
ActualizarNoticia	Realiza la actualización de una noticia en la tabla NOTICIAS de la Base de Datos MOBIL. Tiene como parámetros de entrada, Código del tipo int, Título del tipo string, Noticia del tipo string, Fecha_Ingreso del tipo string, Fecha_Caducidad del tipo string. Utiliza el Store Procedure SpActualizarNoticia.
EliminarNoticia	Realiza la eliminación de una noticia de la tabla NOTICIAS de la Base de Datos MOBIL. Tiene como parámetros de entrada, Código del tipo int. Utiliza el Store Procedure Sp EliminarNoticia.
EliminarNoticiasCaducadas	Realiza la eliminación de las noticias que han salido de vigencia de la tabla NOTICIAS de la Base de Datos MOBIL. Tiene como parámetros de entrada, Fecha del tipo DateTime. Utiliza el Store Procedure SpEliminarNoticiasCaducadas.
EliminarNoticiasTodas	Realiza la eliminación de todas las noticias de la tabla NOTICIAS de la Base de Datos MOBIL. No tiene parámetro

Función	Descripción
	de entrada. Utiliza el Store Procedure SpEliminarNoticiasTodo.

Clase UsuarioMobil

Función	Descripción
SacarDatos	Obtiene un listado de los datos de un usuario registrado especificado por su numero de matrícula. Tiene como parámetro de entrada, mat del tipo float. Utiliza el Store Procedure SpSacarDatosUsuario. Retorna un DataSet con la información solicitada.
SacarTodo	Obtiene un listado de los datos de todos los usuarios registrados. No tiene parámetro de entrada. Utiliza el Store Procedure SpSacarTodo. Retorna un DataSet con la información solicitada.
InsertaNuevo	Realiza el ingreso de un usuario mobil en la tabla UsuariosMobiles en la Base de Datos MOBIL. Tiene como parámetros de entrada toda la información del usuario Mobil. Utiliza el Store Procedure SpInsertarUsuarioMobil.
ActualizarDatos	Realiza la actualización de los datos de un usuario mobil en la tabla UsuariosMobiles en la Base de Datos MOBIL. No tiene parámetros de entrada, toda la información del usuario Mobil. Utiliza el Store Procedure SpActualizarDatosUsuario.
EliminarUsuario	Realiza la eliminación de los datos de un usuario mobil de la tabla UsuariosMobiles de la Base de Datos MOBIL. Tiene como parámetros de entrada, mat del tipo float y ced del tipo int. Utiliza el Store Procedure SpEliminarUsuario.

Librería de Clases ComponentesVB

Clase Horarios

Función	Descripción
---------	-------------

Función	Descripción
SacarHorario	Obtiene la información de los datos que generan el horario de un alumno determinado por su número de matrícula. Tiene como parámetros de entrada, nmat del tipo Decimal, anio del tipo Integer y per del tipo Char. Utiliza el Store Procedure spSacarCarga.
SacarHorarioMateria	Obtiene la información de los datos para una materia y su respectivo horario para un alumno determinado por su número de matrícula. Tiene como parámetros de entrada, Amat del tipo Decima, anio del tipo Integer, per del tipo Char y asig del tipo string. Utiliza el Store Procedure spSacarCargaMateria.

Librería de Clases ComponentesComunes

Clase Conexión

Función	Descripción
miCon	Obtiene la cadena de conexión a la Base de Datos SAES. Retorna un SqlConnection con la conexión a la BDD.
miConM	Obtiene la cadena de conexión a la Base de Datos MOBIL. Retorna un SqlConnection con la conexión a la BDD.
miConParametrizada	Obtiene la cadena de conexión para una Base de Datos Ingresada por el Usuario. Tiene como parámetro de entrada StringConexion del tipo string. Retorna un SqlConnection con la conexión a la BDD.

Clase Envio

Función	Descripción
EnviarMail	Realiza el envío de datos por correo electrónico a un dirección determinada y mediante un servidor de correo determinado. Tiene como parámetros de entrada de del tipo string, para del tipo string, asun del tipo string, mens del tipo string, form del tipo string y serv del tipo string.

Clase Seguridades

Función	Descripción
VerUsuario	Verifica la existencia de un Alumno en la Base de Datos SAES. Tiene como parámetros de entrada mat del tipo float y id del tipo int. Utiliza el Store Procedure SpSacarAlumno. Retorna el valor de mat si el usuario existe y 0 si no existe.
VerUsuarioM	Verifica la existencia de un Usuario Registrado en la Base de Datos MOBIL. Tiene como parámetros de entrada mat del tipo float y id del tipo int. Utiliza el Store Procedure SpSacarUsuario. Retorna el valor de mat si el usuario existe y 0 si no existe.

Servicio Web UDLAServicio

Función	Descripción
Sacar_Materias	Obtiene un listado de las materias que se encuentran asignadas a un Alumno especificado por su numero de matrícula. Tiene como parámetros de entrada Matricula del tipo float, Anio del tipo int, Periodo del tipo char. Retorna un DataSet con la información solicitada.
Sacar_Datos_Alumno	Obtiene un listado de los datos de un alumno especificado por su numero de matrícula. Tiene como parámetro de entrada Matricula del tipo float. Retorna un DataSet con la información solicitada.
Sacar_Datos_Usuario	Obtiene un listado de los datos de un usuario registrado especificado por su numero de matrícula. Tiene como parámetro de entrada Matricula del tipo float. Retorna un DataSet con la información solicitada.
Sacar_Datos_Todos_Usuarios	Obtiene un listado de los datos de todos los usuarios registrados. Retorna un DataSet con la información solicitada.
Sacar_Notas_Por_Tipo	Obtiene un listado de los datos de notas por tipo de nota, esto puede ser Cátedra 1, Cátedra 2, etc. Tiene como parámetros de entrada, Matricula del tipo float, Tipo_Nota del tipo char, Anio del tipo int y Periodo del tipo char. Retorna un DataSet con la información seleccionada.

Función	Descripción
Sacar_Notas_Por_Materia	Obtiene un listado de los datos de todas las notas de una materia seleccionada. Tiene como parámetros de entrada, Matricula del tipo float, Asignatura del tipo string, Anio del tipo int y Periodo del tipo char. Retorna un DataSet con la información seleccionada.
Sacar_Notas_Todo	Obtiene un listado de los datos de todas las notas para un alumno seleccionado. Tiene como parámetros de entrada, Matricula del tipo float, Anio del tipo int y Periodo del tipo char. Retorna un DataSet con la información seleccionada.
Ingresar_Nuevo_Usuario_Mobil	Realiza el ingreso de un usuario mobil en la tabla UsuariosMobiles en la Base de Datos MOBIL. Tiene como parámetros de entrada toda la información del usuario Mobil.
Actualizar_Datos_Usuario_Mobil	Realiza la actualización de los datos de un usuario mobil en la tabla UsuariosMobiles en la Base de Datos MOBIL. Tiene como parámetros de entrada toda la información del usuario Mobil.
Eliminar_Usuario_Mobil	Realiza la eliminación de los datos de un usuario mobil de la tabla UsuariosMobiles de la Base de Datos MOBIL. Tiene como parámetros de entrada, mat del tipo float y ced del tipo int.
Enviar_Mail	Realiza el envío de información hacia una dirección de correo electrónico, mediante un servidor determinado. Tiene como parámetros de entrada, de del tipo string, para del tipo string, asun del tipo string, mens del tipo string, form del tipo string y serv del tipo string.
Generar_Coneccion_Manual	Genera un string de conexión mediante el ingreso de una cadena con la que se reconoce al servidor y su ubicación. Tiene como parámetros de entrada, StringConexion del tipo string. Retorna un string con la cadena de conexión.
Obtener_Coneccion_SAES	Obtiene la cadena de conexión predeterminada para la Base de Datos SAES. Retorna un string con la cadena de conexión.
Obtener_Coneccion_MOBIL	Obtiene la cadena de conexión predeterminada para la Base de Datos MOBIL. Retorna un string con la cadena

Función	Descripción
Sacar_Horario_Alumno	<p data-bbox="687 275 842 302">de conexión.</p> <p data-bbox="687 320 1370 533">Obtiene un listado de los datos que generan el horario para un alumno determinado por su número de matrícula. Tiene como parámetros de entrada, Matricula del tipo float. Anio del tipo int, Periodo del tipo char. Retorna un DataSet con la información seleccionada.</p>
Formato_Texto	<p data-bbox="687 551 1370 674">Genera un string de un DataSet enviado. Tiene como parámetro de entrada un DataSet. Retorna un string con la información en tipo de texto.</p>
Formato_Html	<p data-bbox="687 692 1370 815">Genera un string de un DataSet enviado. Tiene como parámetro de entrada un DataSet. Retorna un string con la información en tipo de Html.</p>
Formato_Tabla_Html	<p data-bbox="687 833 1370 994">Genera un string de un DataSet enviado con la información de Horarios. Tiene como parámetro de entrada un DataSet. Retorna un string con la información en tipo de Html.</p>
Formato_Tabla_Notas_Html	<p data-bbox="687 1012 1370 1173">Genera un string de un DataSet enviado con la información de Notas. Tiene como parámetro de entrada un DataSet. Retorna un string con la información en tipo de Html.</p>
Verificar_Alumno	<p data-bbox="687 1191 1370 1449">Verifica la existencia de un Alumno en la Base de Datos SAES. Tiene como parámetros de entrada mat del tipo float y un id del tipo int. Utiliza la Función VerUsuario de la clase Seguridades de la Librería de Clases ComponentesComunes. Retorna el valor de mat si el usuario existe y 0 si no existe.</p>
Verificar_Usuario_Mobil	<p data-bbox="687 1467 1370 1724">Verifica la existencia de un Usuario Registrado en la Base de Datos MOBIL. Tiene como parámetros de entrada mat del tipo float y un id del tipo int. Utiliza la Función VerUsuarioM de la clase Seguridades de la Librería de Clases ComponentesComunes. Retorna el valor de mat si el usuario existe y 0 si no existe.</p>
Ingresar_Noticia	<p data-bbox="687 1742 1370 1953">Realiza el ingreso de una noticia en la tabla NOTICIAS de la Base de Datos MOBIL. Tiene como parámetros de entrada Titulo del tipo string, Noticia del tipo string, Fecha_Ingreso del tipo string, Fecha_Caducidad del tipo string.</p>

Función	Descripción
Actualizar_Noticia	Realiza la actualización de una noticia en la tabla NOTICIAS de la Base de Datos MOBIL. Tiene como parámetros de entrada Código del tipo int, Título del tipo string, Noticia del tipo string, Fecha_Ingreso del tipo string, Fecha_Caducidad del tipo string.
Eliminar_Noticia	Realiza la eliminación de una noticia de la tabla NOTICIAS de la Base de Datos MOBIL. Tiene como parámetros de entrada Código del tipo int.
Eliminar_Noticias_Caducadas	Realiza la eliminación de las noticias que han salido de vigencia de la tabla NOTICIAS de la Base de Datos MOBIL. Tiene como parámetros de entrada Código del tipo int.
Eliminar_Todas_Noticias	Realiza la eliminación de todas las noticias de la tabla NOTICIAS de la Base de Datos MOBIL.
Sacar_Noticia_Seleccionada	Realiza la extracción de la información referente a una noticia determinada. Tiene como parámetros de entrada Código del tipo int. Retorna un DataSet con el contenido de la Noticia.
Sacar_Noticias_Todas	Realiza la extracción de la información referente a todas las noticias existentes. Retorna un DataSet con el contenido de las Noticias.
Sacar_Noticias_Caducadas	Realiza la extracción de la información referente a las noticias que han salido de vigencia. Tiene como parámetros de entrada Fecha del tipo string. Retorna un DataSet con el contenido de la Noticia.

Función	Descripción
Alistar_Notas	Este método realiza primero el formateo y la comprobación de idoneidad del mensaje enviado por medio del celular a la empresa integradora, para tenerlo listo para el requerimiento de información de la Base de Datos. Seguidamente, comprueba la existencia del alumno en la Base de Datos SAES y la existencia del usuario en la Base de Datos MOBIL. A continuación verifica si el usuario tiene habilitada la opción de notas por medio del celular, y procede a la extracción de los datos requeridos y los deja listos, enviándolos en formato XML a la empresa integradora, para que sea ella quien los presente a los usuarios que han solicitado el servicio. Tiene como parámetros de entrada Mensaje del tipo string. Retorna un string con contenido XML con los datos de las notas si ese fuere el caso, o un string con un mensaje según sea el caso del error.

Aplicación Windows Configurator

Clase ValidacionFechas

Función	Descripción
VerificarDia	Verifica si un valor ingresado como valor para día es correcto o no, dependiendo del mes y del año, si es bisiesto o no. Tiene como parámetros de entrada Mes del tipo Integer y Bisiesto del tipo Integer. Retorna el numero de días que tiene el mes especificado.
AnioViciesto	Verifica si un año determinado es bisiesto o no. Tiene como parámetros de entrada Anio del tipo Integer. Retorna un 1 si el año es bisiesto y un 0 si no lo es.

DESCRIPCION DE VARIABLES GLOBALES

Variable	Descripción
Anio	Variable para el ingreso del año determinado por el usuario. Variable Tipo Integer.
Periodo	Variable para el ingreso del Periodo determinado por el usuario. Variable Tipo Char.
Servidor	Variable para el ingreso del Servidor de Correo Electrónico determinado por el usuario. Variable Tipo String.
Remitente	Variable para el ingreso del Remitente de los correos determinado por el usuario. Variable Tipo String.
AsuntoH	Variable para el ingreso del Asunto para los correos de Horarios determinado por el usuario. Variable Tipo String.
FechaH	Variable para el ingreso de la Fecha de envío de los correos de Horarios determinado por el usuario. Variable Tipo DateTime.
AsuntoNF	Variable para el ingreso del Asunto para los correos de Notas Finales determinado por el usuario. Variable Tipo String.
FechaNF	Variable para el ingreso de la Fecha de envío de los correos de Notas Finales determinado por el usuario. Variable Tipo DateTime.
AsuntoNP	Variable para el ingreso del Asunto para los correos de Notas Parciales determinado por el usuario. Variable Tipo String.
FechaNP(5)	Variable tipo arreglo para el ingreso de las Fecha de envío de los correos de Notas Parciales determinado por el usuario. Variable Tipo DateTime.
Parcial(5)	Variable tipo arreglo para el dato del parcial para el envío de los correos de Notas Parciales determinada por código. Variable tipo Char.
ControlVG	Variable para control del ingreso de variables globales para Parámetros Generales. Variable tipo string.
ControlVH	Variable para control del ingreso de variables globales para Parámetros de Horarios. Variable tipo string.
ControlVNP	Variable para control del ingreso de variables globales para Parámetros de Notas Parciales. Variable tipo string.
ControlVNF	Variable para control del ingreso de variables globales para Parámetros de Notas Finales. Variable tipo string.
ControlPLanzados	Variable para el control de la existencia de Procedimientos

Variable	Descripción
ControlEnvio	Automáticos. Variable tipo string. Variable para el control del Envío de Mails para el lanzamiento de todos los procedimientos automáticos. Variable tipo string.
ControlEnvioP	Variable para el control del Envío de Mails para el lanzamiento de los procedimientos automáticos parciales. Variable tipo string.
ControlDG	Variable para el control de la creación de estilos para el DataGrid. Variable tipo string.

MANUAL DE INSTALACION

INDICE

Instalación De Los Servicios Web

216

<i>Instalación De Las Aplicaciones Web Y Wap</i>	222
<i>Instalación De La Aplicación Windows Sistema Configurator</i>	229
<i>Instalación De las Bases De Datos</i>	234

A continuación se detalla todos los pasos a seguirse para realizar la instalación de cada una de las aplicaciones que se tiene, estas son:

- Servicios Web

- Aplicaciones Web y WAP
- Aplicación Windows

El primer paso será la instalación de los Servicios Web.

Instalación de los Servicios Web

Primeramente se debe ubicar la carpeta en donde se encuentran los Instaladores para el Servicio Web que se desea instalar, en este caso es la carpeta Servicio Web UDLAServicio. Ubicamos el Icono de Setup.exe y damos doble clic sobre el mismo.

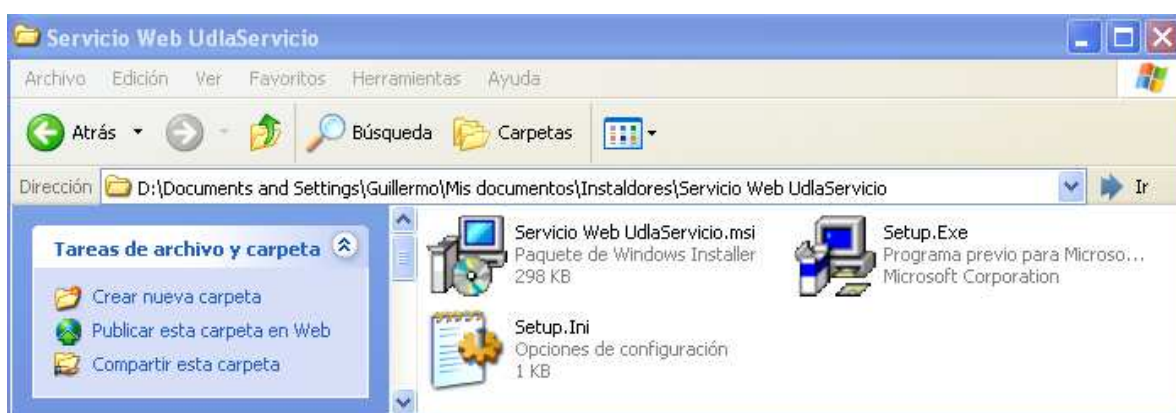


Figura 1. Instaladores para UDLAServicio

Lo primero que aparece, es una ventana con un mensaje de que se está preparando la instalación, de la siguiente forma:

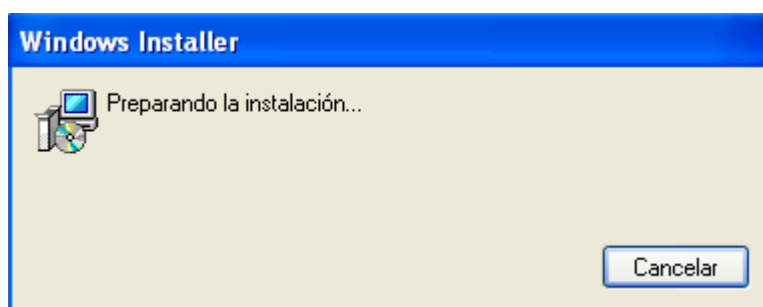


Figura 2. Preparación de la Instalación

A continuación se presenta la pantalla inicial de la instalación en la cual se detalla la aplicación que se está instalando y advertencias de ley.



Figura 3. Pantalla de Bienvenida al Instalador

Al dar clic en el botón siguiente se presentará una ventana en la cual se podrá escoger el directorio virtual en el cual se desea instalar este Servicio, o dejar el que viene predeterminado, en este caso es UDLAServicio.

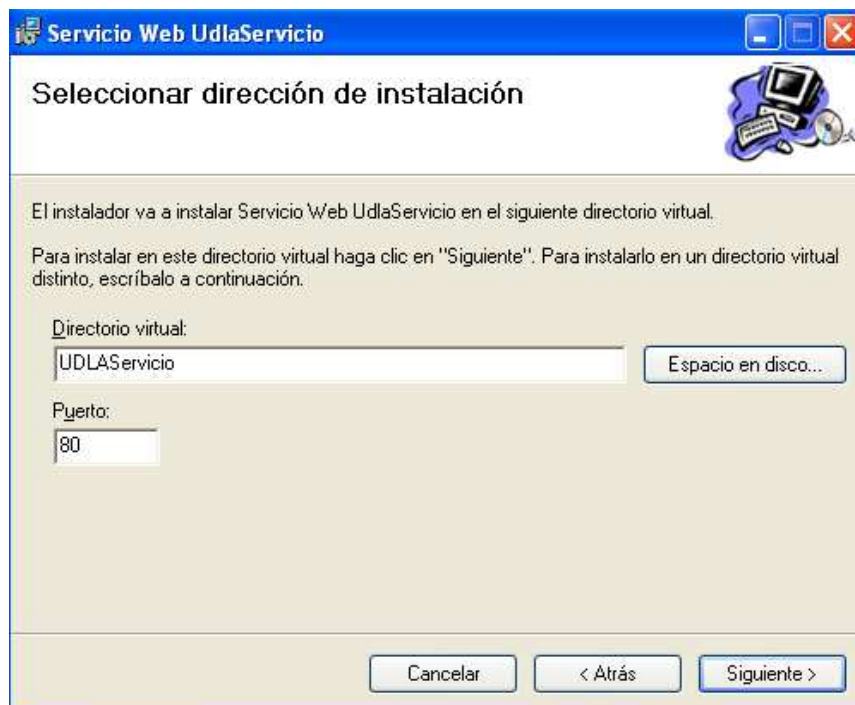


Figura 4. Ubicación de la Instalación

A continuación se tiene la ventana de confirmación de la instalación, en la cual se indica que se procederá ya con la instalación del servicio según lo que se ha seleccionado antes.

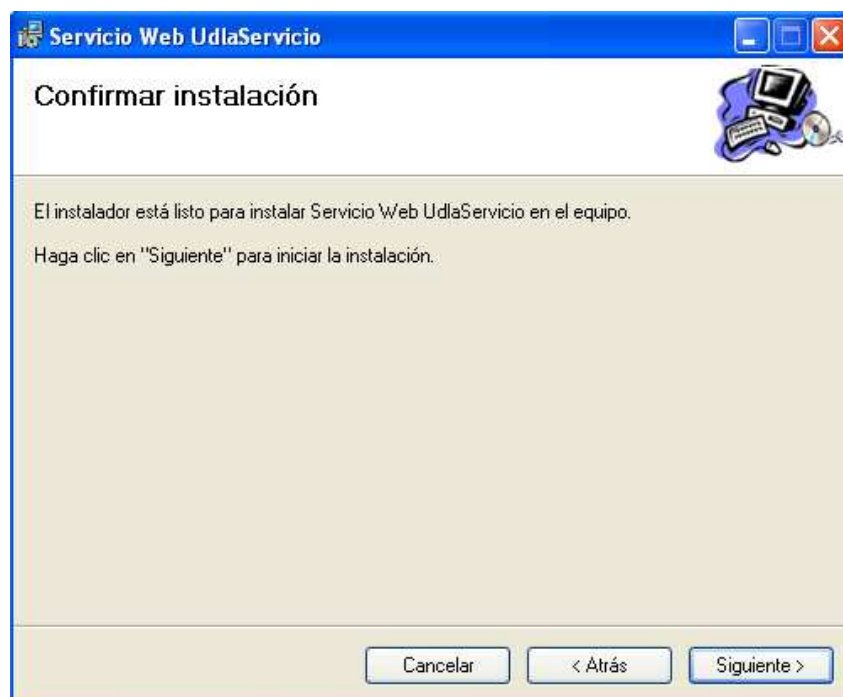


Figura 5. Confirmación de la Instalación

Al dar clic en el botón de Siguiete, se dará paso a la instalación propiamente dicha del Servicio Web, para la cual se presenta una ventana en la cual se podrá ver una barra de progreso según se vaya ejecutando la misma.

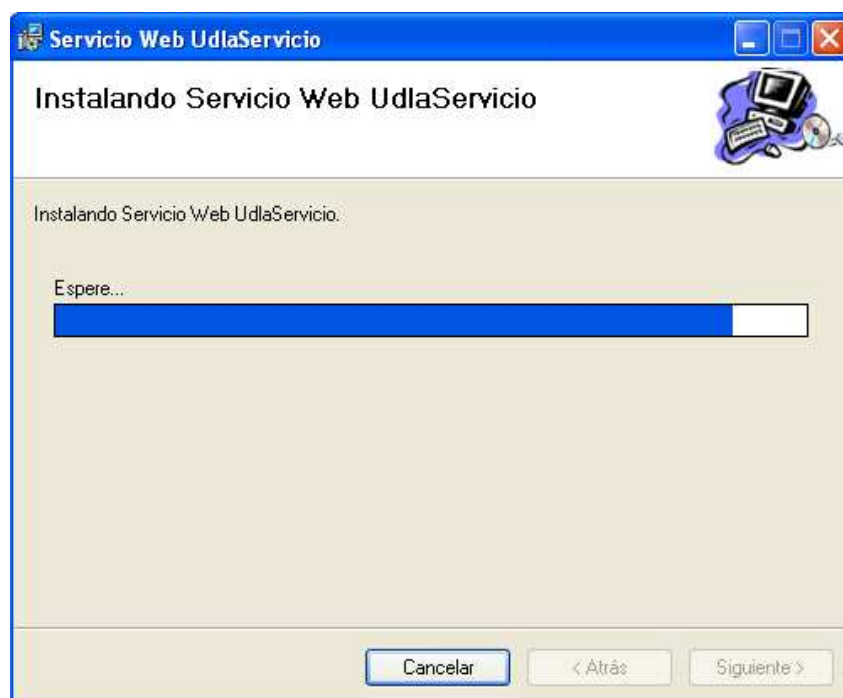


Figura 6. Progreso de la Instalación

Al realizarse por completo el proceso de instalación, se presenta una pantalla de Instalación Completada.

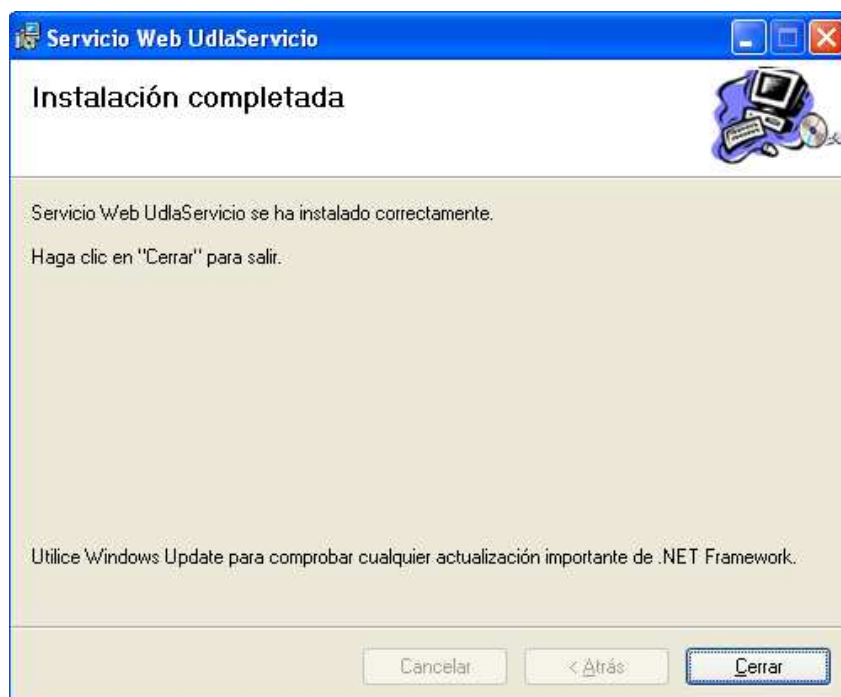


Figura 7. Instalación completada

Una vez realizada la instalación del Servicio Web, para poder comprobar que se lo haya realizado correctamente, dentro del menú Inicio de la Barra de Herramientas de Windows, se selecciona Panel de Control.



Figura 8. Menú Inicio

Ya dentro del Panel de Control, escogemos la opción Rendimiento y Mantenimiento.



Figura 9. Opción Rendimiento y Mantenimiento

Seguidamente se selecciona la opción Herramientas Administrativas.



Figura 10. Opción Herramientas Administrativas

Y aquí buscamos y se selecciona la opción Servicios de Internet Information Server.



Figura 11. Opción de Internet Information Server

Aquí se puede verificar la creación del directorio virtual con el nombre que se detallo durante la instalación, en este caso el directorio virtual es llamado UDLAServicio.

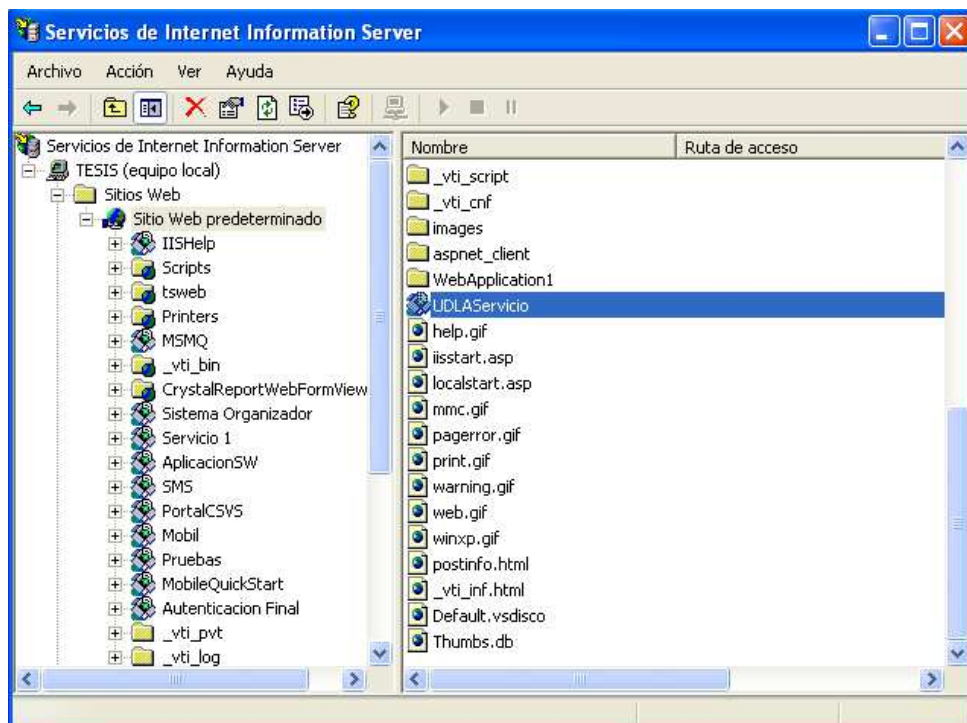


Figura 12. Confirmación de Instalación

Por último se puede digitar dentro del Internet Explorer "http://localhost/UDLAServicio/UDLAServicio.asmx", para ejecutar el Servicio Web y Probar su funcionalidad.



Figura 13. Prueba de Servicio Web UDLAServicio

Este procedimiento se llevará a cabo para cada uno de los Servicios Web que se tenga, en este caso se lo deberá volver a hacer para el Servicio Web UDLAServicioSMS, obteniendo como resultado la siguiente pantalla.

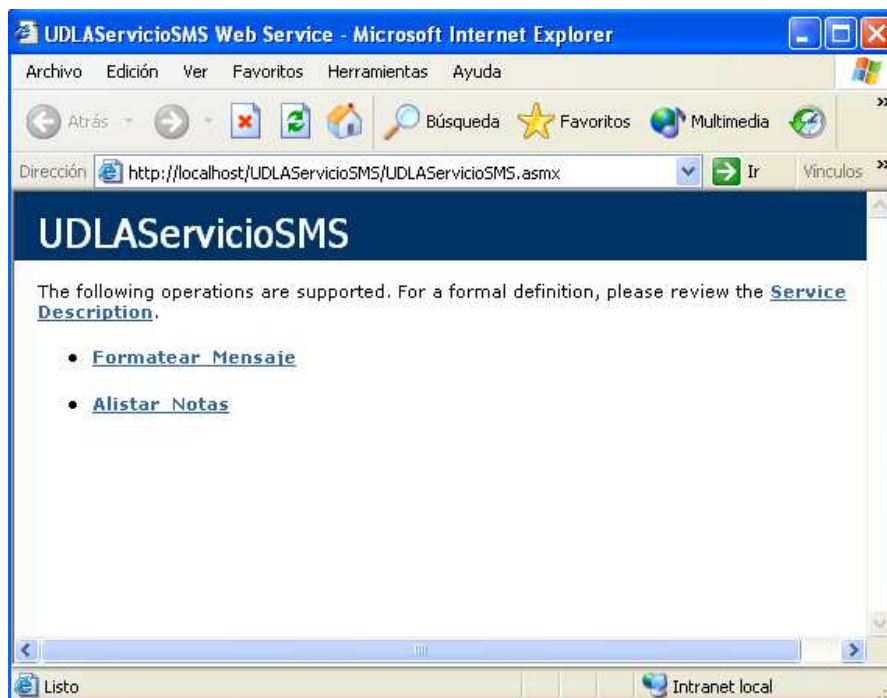


Figura 14. Prueba de Servicio Web UDLAServicioSMS

Instalación de las Aplicaciones Web y WAP

Primeramente se debe ubicar la carpeta en donde se encuentran los Instaladores para la Aplicación Web que se desea instalar, en este caso es la carpeta Aplicación UdlaWeb. Ubicamos el Icono de Setup.exe y damos doble clic sobre el mismo.

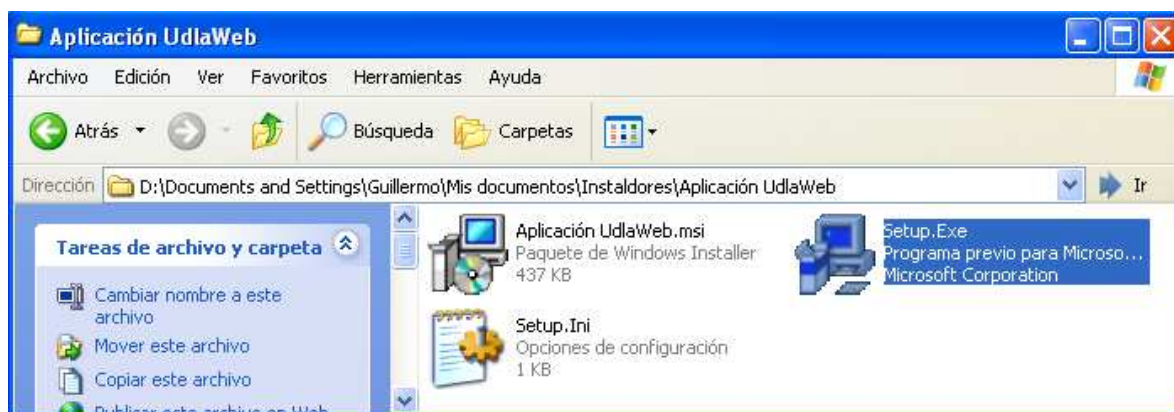


Figura 15. Instaladores para UdlaWeb

Lo primero que aparece, es una ventana con un mensaje de que se está preparando la instalación, de la siguiente forma:

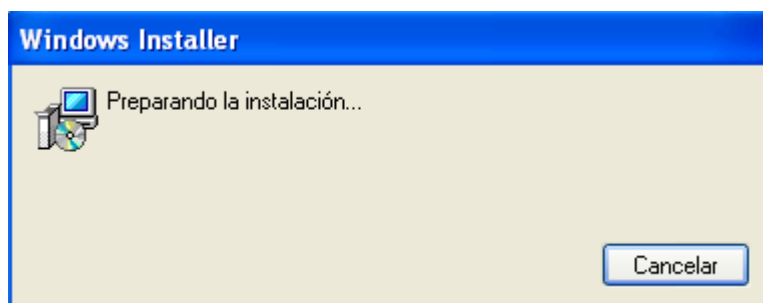


Figura 16. Preparación de la Instalación

A continuación se presenta la pantalla inicial de la instalación en la cual se detalla la aplicación que se está instalando y advertencias de ley.

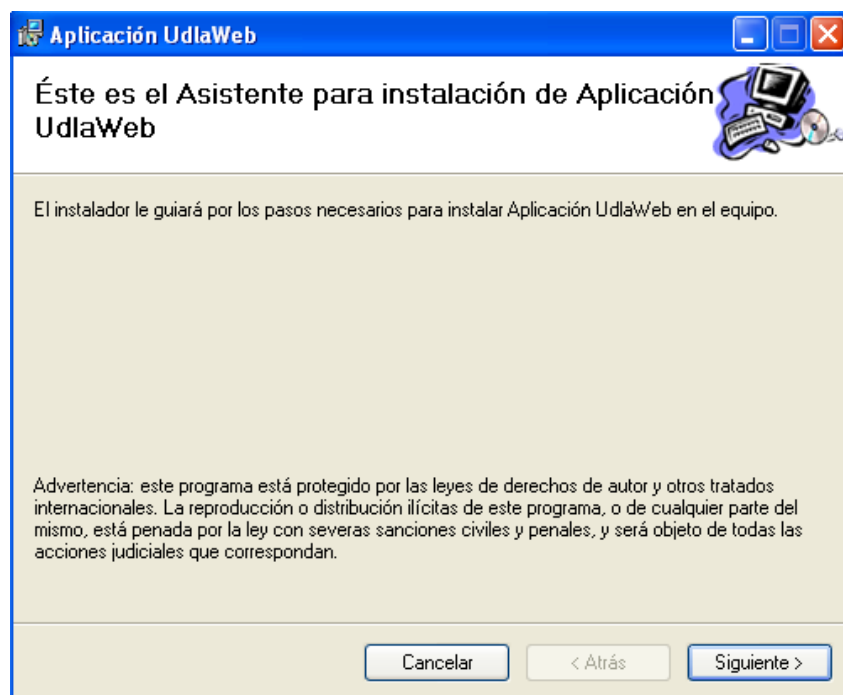


Figura 17. Pantalla de Bienvenida al Instalador

Al dar clic en el botón siguiente se presentará una ventana en la cual se podrá escoger el directorio virtual en el cual se desea instalar la Aplicación, o dejar el que viene predeterminado, en este caso es UdlaWeb.

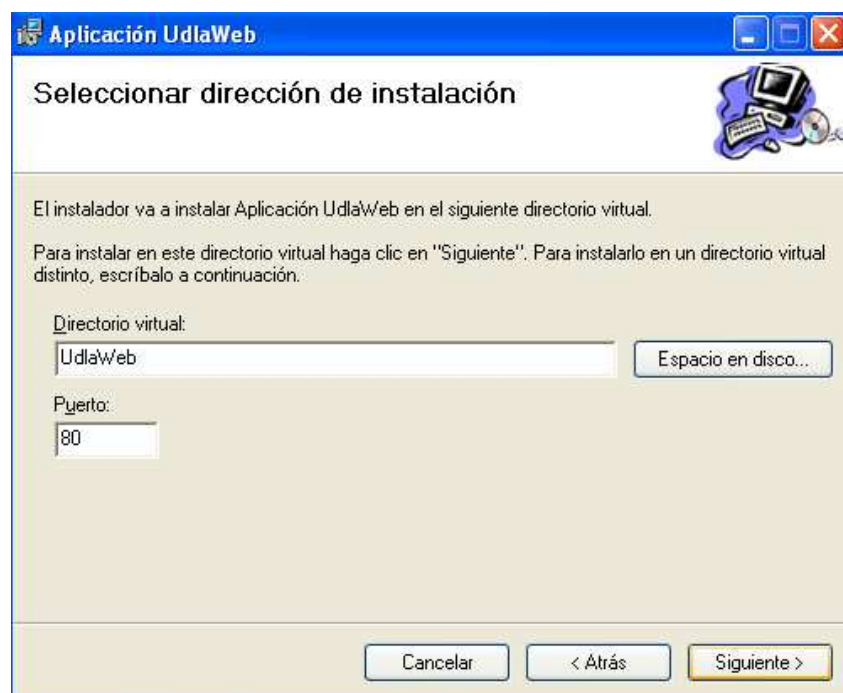


Figura 18. Ubicación de la Instalación

A continuación se tiene la ventana de confirmación de la instalación, en la cual se indica que se procederá ya con la instalación de la aplicación según lo que se ha seleccionado antes.

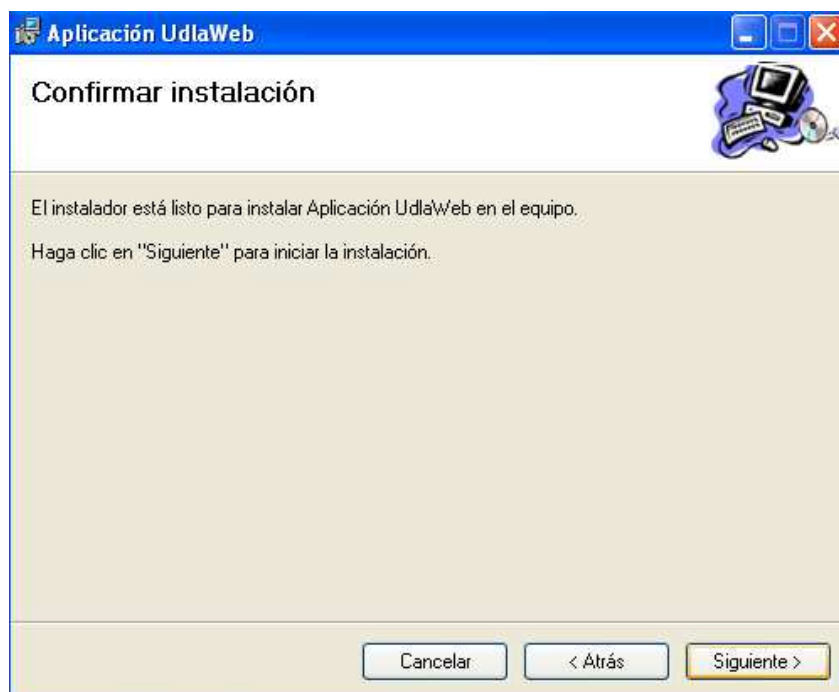


Figura 19. Confirmación de la Instalación

Al dar clic en el botón de Siguiente, se dará paso a la instalación propiamente dicha de la Aplicación Web o WAP, para la cual se presenta una ventana en la cual se podrá ver una barra de progreso según se vaya ejecutando la misma.

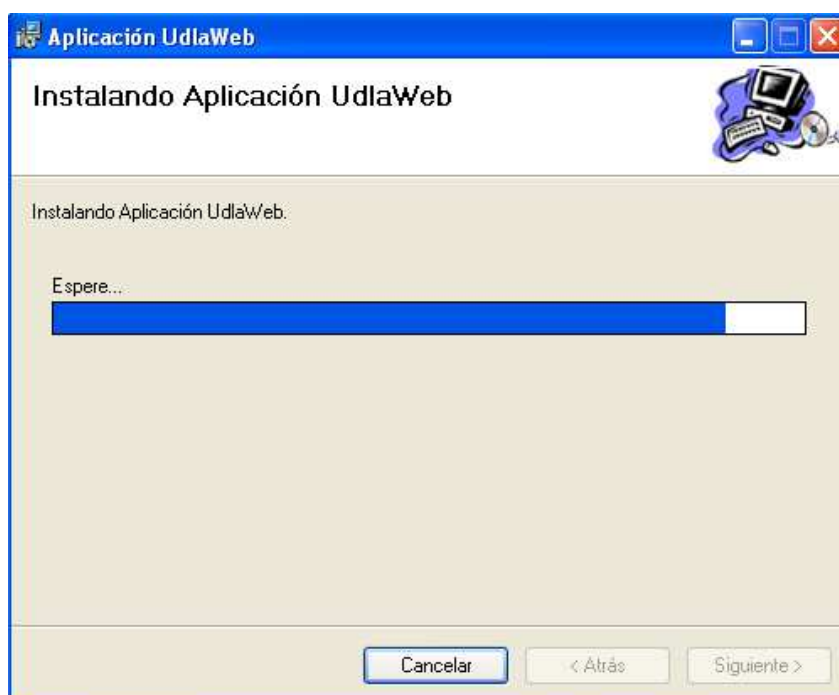


Figura 20. Progreso de la Instalación

Al realizarse por completo el proceso de instalación, se presenta una pantalla de Instalación Completada.



Figura 21. Instalación completada

Una vez realizada la instalación de la Aplicación Web o WAP, para poder comprobar que se lo haya realizado correctamente, dentro del menú Inicio de la Barra de Herramientas de Windows, se selecciona Panel de Control.



Figura 22. Menú Inicio

Ya dentro del Panel de Control, escogemos la opción Rendimiento y Mantenimiento.



Figura 23. Opción Rendimiento y Mantenimiento

Seguidamente se selecciona la opción Herramientas Administrativas.



Figura 24. Opción Herramientas Administrativas

Y aquí buscamos y se selecciona la opción Servicios de Internet Information Server.

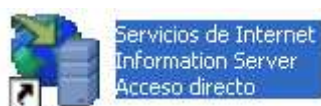


Figura 25. Opción de Internet Information Server

Aquí se puede verificar la creación del directorio virtual con el nombre que se detallo durante la instalación, en este caso el directorio virtual es llamado UdlWeb.

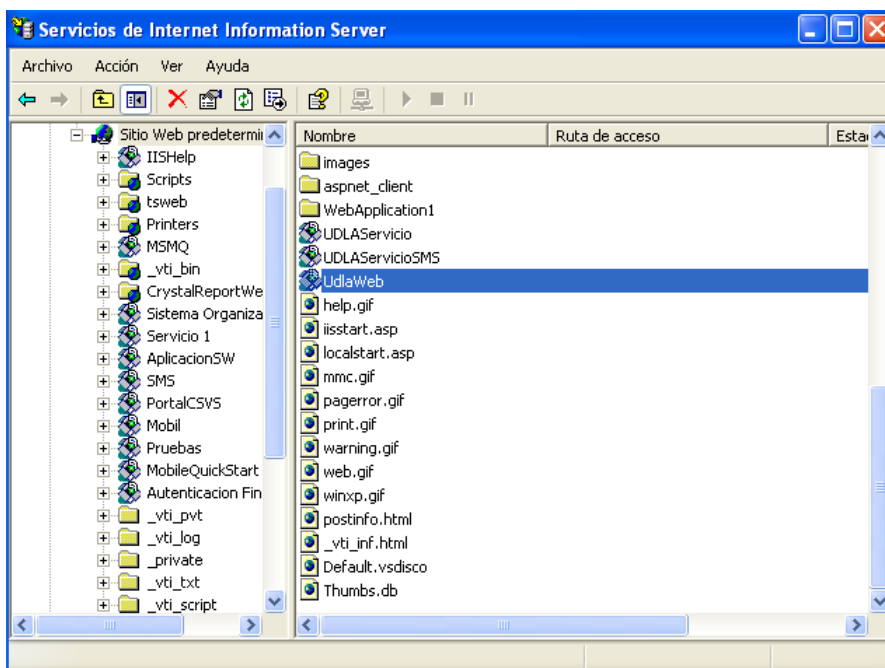


Figura 26. Confirmación de Instalación

Por último se puede digitar dentro del Internet Explorer “http://localhost/UdlaWeb/Login.aspx”, para ejecutar la Aplicación Web y probar su funcionalidad.



Figura 27. Prueba de la Aplicación UdlaWeb

Este procedimiento se llevará a cabo para cada una de las Aplicaciones que se tenga, en este caso se lo deberá volver a hacer para la Aplicación Web UdlaMobil, obteniendo como resultado la siguiente pantalla.



Figura 28. Prueba de la Aplicación UdlaMobil

Por otra parte, para realizar la comprobación de la instalación de la Aplicación WAP, se podrá utilizar un emulador, en este caso es Openwave SDK 6.2.2 y se tiene lo siguiente.



Figura 29. Prueba de la Aplicación UdlawAP

Instalación de la Aplicación Windows Sistema Configurador

Primeramente se debe ubicar la carpeta en donde se encuentran los Instaladores para la Aplicación Windows que se desea instalar, en este caso es la carpeta Sistema Configurador. Ubicamos el Icono de Setup.exe y damos doble clic sobre el mismo.

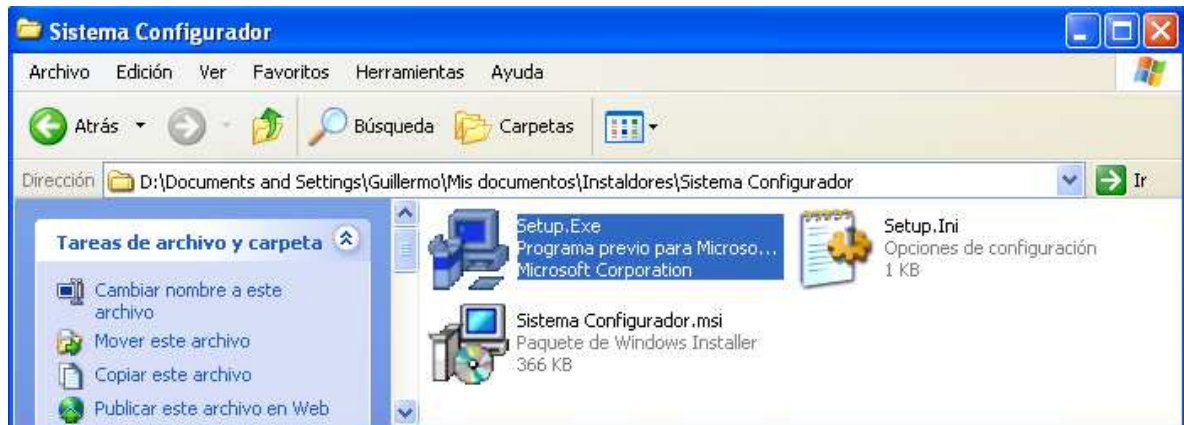


Figura 30. Instaladores para Sistema Configurador

Lo primero que aparece, es una ventana con un mensaje de que se está preparando la instalación, de la siguiente forma:

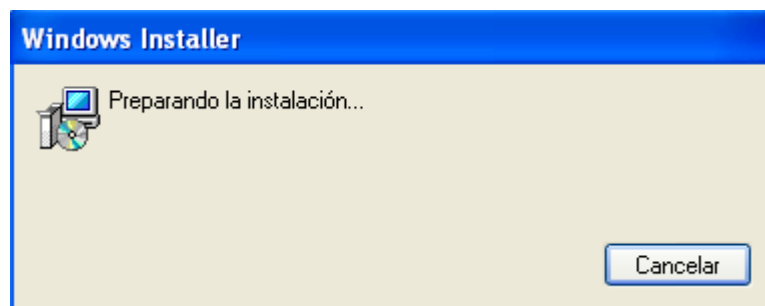


Figura 31. Preparación de la Instalación

A continuación se presenta la pantalla inicial de la instalación en la cual se detalla la aplicación que se está instalando y advertencias de ley.

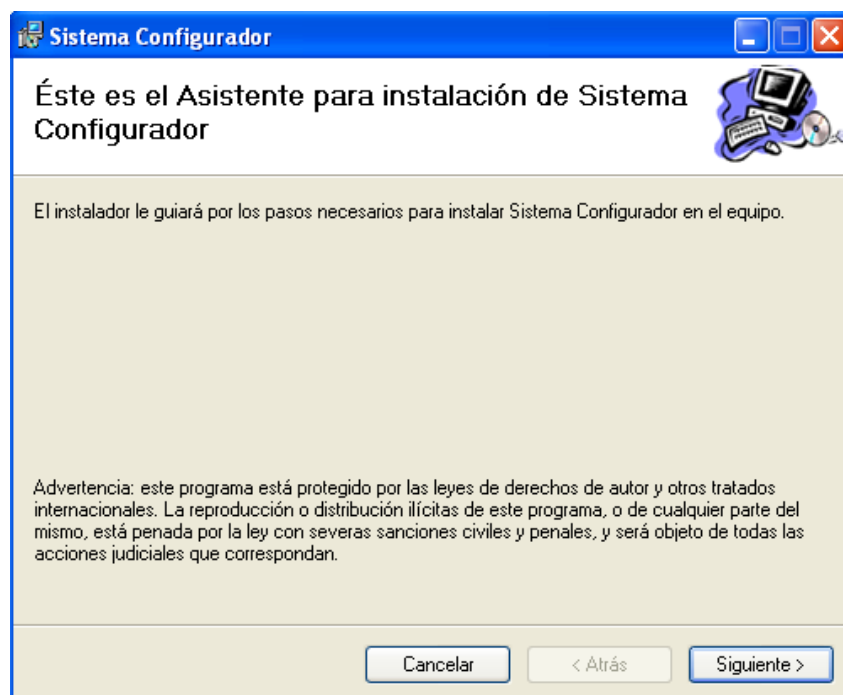


Figura 32. Pantalla de Bienvenida al Instalador

Al dar clic en el botón siguiente se presentará una ventana en la cual se podrá leer un contrato de licencia y aceptación de condiciones, si se selecciona aceptar, se podrá continuar con la instalación, caso contrario la instalación no se realizará.



Figura 33. Contrato de licencia y aceptación

Seguidamente se presenta una ventana en la cual se puede seleccionar la ubicación, en donde se realizará esta instalación, o se puede dejar la que viene predeterminada, que en este caso es "C:\Archivos de Programa\Sistema Configurator\".

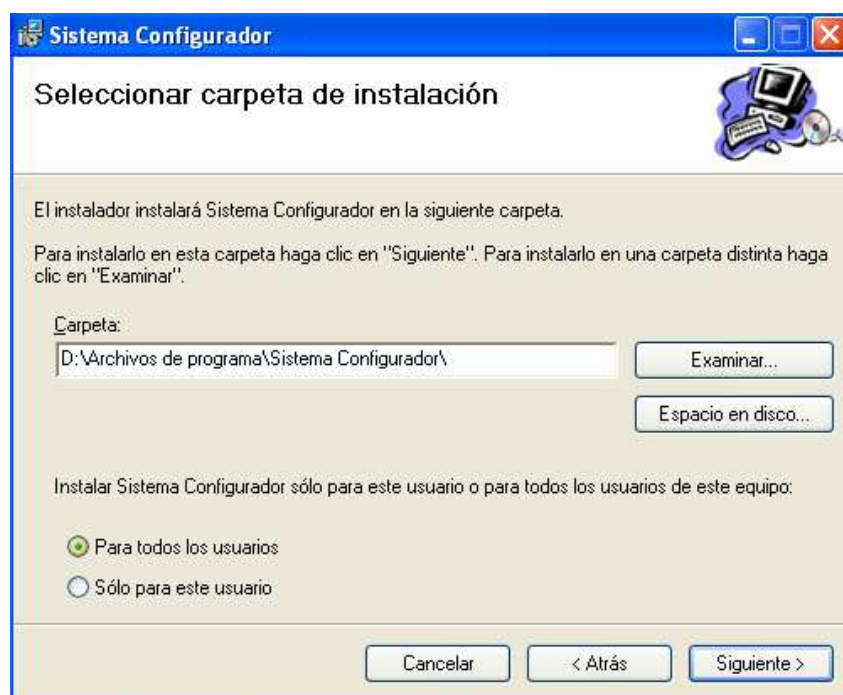


Figura 34. Ubicación de la Instalación

A continuación se tiene la ventana de confirmación de la instalación, en la cual se indica que se procederá ya con la instalación de la aplicación según lo que se ha seleccionado antes.

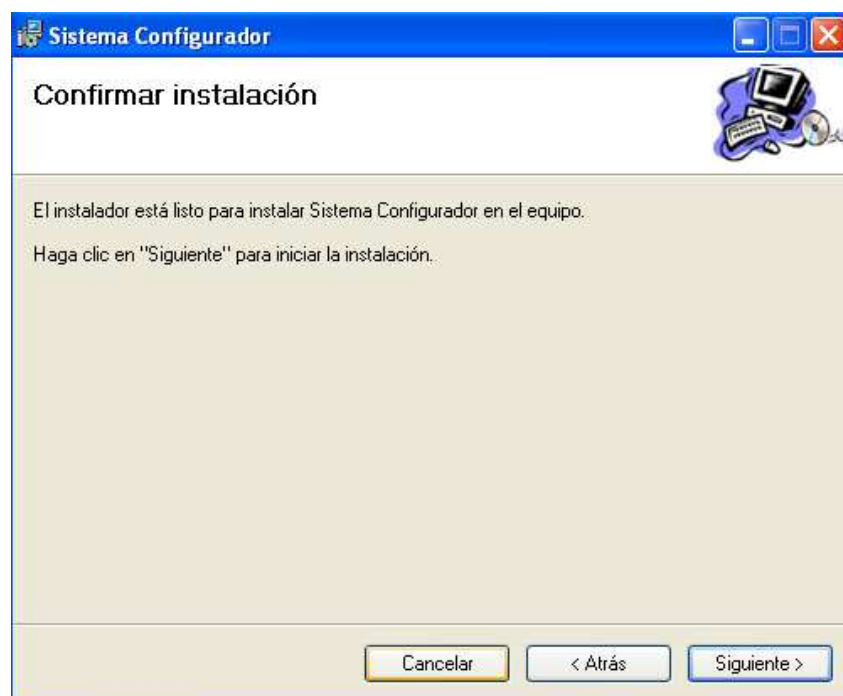


Figura 35. Confirmación de la Instalación

Al dar clic en el botón de Siguiente, se dará paso a la instalación propiamente dicha de la Aplicación Windows, para la cual se presenta una ventana en la cual se podrá ver una barra de progreso según se vaya ejecutando la misma.

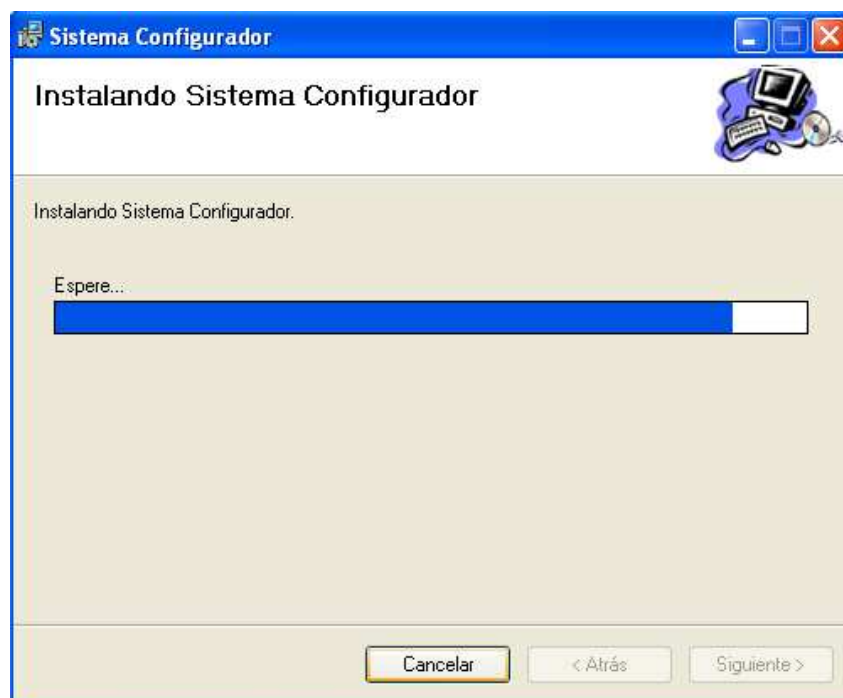


Figura 36. Progreso de la Instalación

Al realizarse por completo el proceso de instalación, se presenta una pantalla de Instalación Completada.

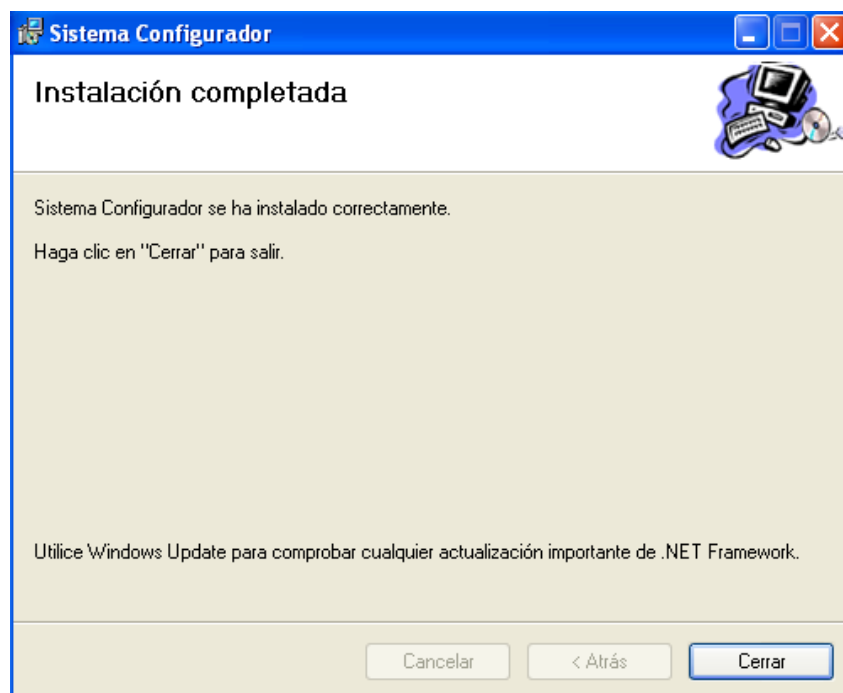


Figura 37. Instalación completada

Una vez realizada la instalación completa de la Aplicación Sistema Configurador, se debe ubicar la carpeta en la que se instaló, esto es primero encontrar la carpeta Archivos de Programa.



Figura 38. Carpeta Archivos de Programa

Seguidamente se ubica la carpeta de la aplicación, que en este caso es Sistema Configurador.



Figura 39. Carpeta de la Aplicación

Se da doble clic sobre esta y se encuentra el archivo Configurador.exe.



Figura 40. Ubicación del Archivo ejecutable

Por último si se hace doble clic sobre el archivo ejecutable Configurador.exe, se podrá probar su funcionamiento.

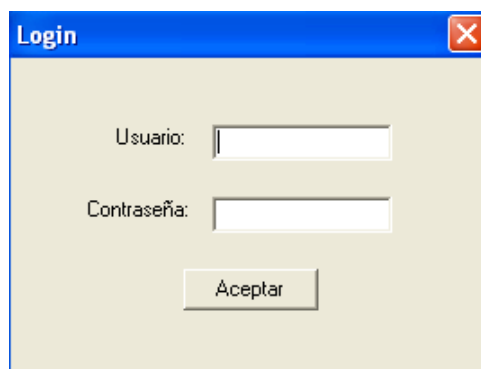


Figura 41. Prueba de funcionamiento de Sistema Configurador

Instalación de las Bases de Datos

La instalación de las Bases de Datos SAES y MOBIL, respectivamente, se lleva a cabo desde el mismo instalador de la Aplicación Sistema Configurator. Durante el proceso de instalación de dicha aplicación, aparece una ventana que solicita el ingreso del nombre del Servidor de Base de Datos y el Usuario del mismo, mediante estos parámetros se genera la cadena de conexión con la cual se ubica a este Servidor dentro del que residirán estas dos Bases de Datos, de la siguiente manera:

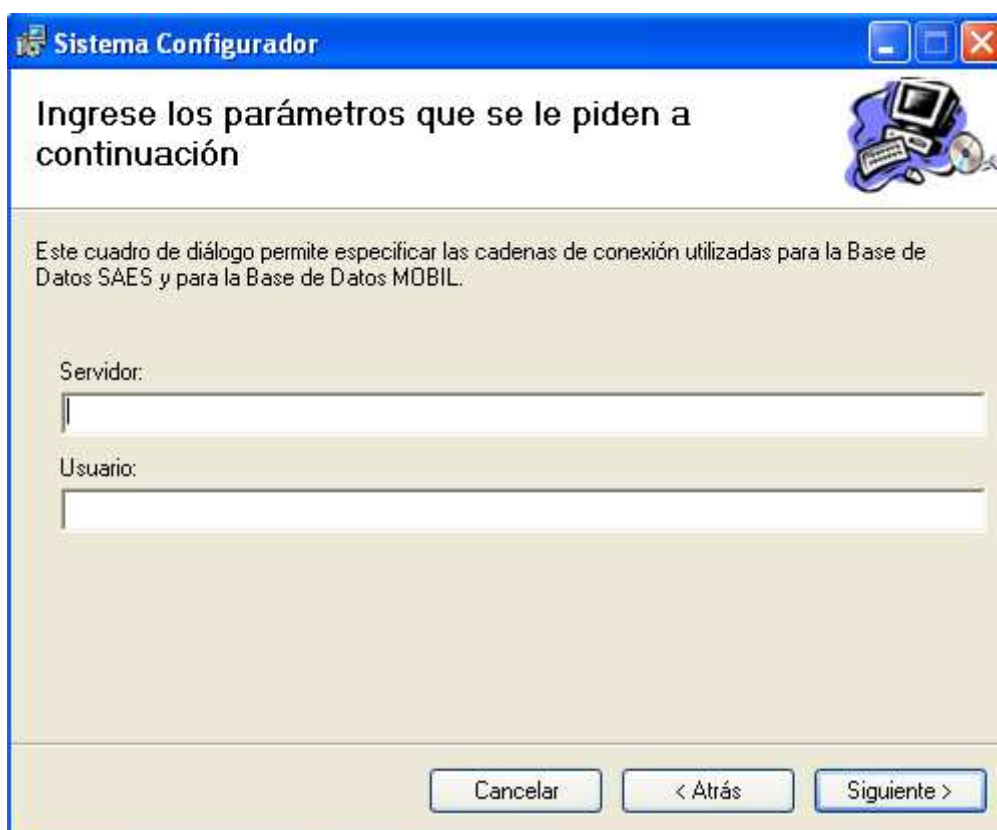


Figura 42. Instalación de las Bases de Datos

La cadena que se ingrese, deberá ser ingresada siempre entre comillas, para que el instalador sepa que esa será la ruta en la cual deben ser instaladas las Bases de Datos SAES y MOBIL respectivamente.

ANEXO X

ANALISIS TARIFARIO

ANALISIS DE COSTOS

Para el análisis tarifario de este trabajo de titulación se tomaron en cuenta los siguientes rubros:

- Costo de mensaje SMS para petición de Notas.
- Costo de envío de mensajes para Noticias.
- Costo de navegación en Internet Móvil.

COSTO DE MENSAJES SMS PARA PETICIÓN DE NOTAS

En el país, se encuentran actualmente trabajando, alrededor de 15 Empresas Integradoras, las cuales tienen como objetivo, actuar como una empresa intermediaria entre las empresas de telefonía celular, ya sea esta Porta, BellSouth o Alegro, y la empresa que requiere el servicio de mensajes vía SMS.

Dentro de este conjunto de Empresas Integradoras, se ha tomado datos referenciales de la Empresa LIRELA, de la cual se pudo obtener la información requerida, mediante conversación con el Sr. Vladimir Valverde, Asesor Comercial de dicha empresa. Conversación por medio de la cual se obtuvo las tres formas como se manejan estos servicios de mensajes vía SMS.

- La primera forma es en la cual la Universidad de las Américas, cubriría todos los costos de los mensajes que los estudiantes realizarían a lo largo del semestre, costo que se debe pagar mensualmente a la empresa integradora, con el fin de que para el estudiante el mensaje no tenga costo directo alguno. En dicho caso la Universidad de las Américas debería incluir este costo, dentro del precio de las Anualidades de los estudiantes.
- La segunda opción, llamada por la LIRELA, opción de Juegos o Negocios, es la que el costo del mensaje pasa a ser debitado directamente de la cuenta celular del estudiante, esto es que el mensaje tiene un costo de 0,35 ctvs + impuestos, los cuales serían descontados del plan o de la tarjeta prepago del estudiante. En este caso, la Universidad de las Américas, tendría lucro, es decir tendría derecho a cierto porcentaje del costo del mensaje, yendo el porcentaje restante para la Empresa Integradora y para la Empresa de Telefonía Celular.
- Por último y la que pareciere ser la más conveniente es en la cual, la Universidad de las Américas, renuncia a su porcentaje de lucro, con el objetivo de reducir los costos de

los mensajes para los estudiantes, es decir de el costo normal de 0,35 ctvs el mensaje, se reduciría a unos 0,15 ctvs el mensaje, siendo este costo distribuido entre la Empresa Integradora y la Empresa de Telefonía Celular.

COSTO DE ENVIO DE MENSJAES PARA NOTICIAS

En lo referente al costo de envío de mensajes para noticias, sería un rubro que la Universidad de las Américas debería cubrir mensualmente. Este rubro depende directamente de la cantidad de mensajes que se vayan a enviar.

Por medio del Sr. Vladimir Valverde se pudo saber que el plan que contiene el menor número de mensajes de envío tiene un costo de 0,07 ctvs por mensaje. Para obtener mas detalles de los planes era necesario una reunión formal entre las partes interesadas.

COSTO DE NAVEGACION EN INTERNET MOVIL

PORTA

Tarifas

Conectividad Móvil - Wap e Internet

Costo mensual 0.5MB: \$3 USD + IVA.

Costo mensual 1MB: \$5 USD + IVA.

Costo mensual 20MB: \$29 USD + IVA.

Costo adicional Kb: \$0.01 USD + IVA.

BELLSOUTH

Tarifas

Costo mensual 512Kb: \$5 USD + IVA.

Costo adicional Kb: \$0.01 USD + IVA.

El costo del servicio Navegación Personal cuando sea contratado por separado del plan tarifario tendrá un costo de \$10 USD + IVA mensuales.

ALEGROPCS

No tiene activo este servicio.

ANEXO XI

CD

CONTENIDO DEL CD

- Documento Trabajo de Titulación
- Instaladores
- Código Fuente