



**PONTIFICIA
UNIVERSIDAD
CATÓLICA
DEL ECUADOR
SEDE AMBATO**
SERÉIS MIS TESTIGOS

ESCUELA DE INGENIERÍA DE SISTEMAS

Tema:

“ESTUDIO COMPARATIVO DE LAS TECNOLOGÍAS ASP, AJAX, J2EE, J2ME Y .NET PARA EL DESARROLLO DE APLICACIONES MÓVILES, APLICADO AL DESARROLLO DE UNA APLICACIÓN MÓVIL WEB PARA REGISTRO DE CALIFICACIONES Y ASISTENCIA DE LOS ALUMNOS DE LA ESCUELA DE INGENIERIA DE SISTEMAS DE LA P.U.C.E.S.A. APLICADA AL AÑO ACADÉMICO 2010”

DISERTACIÓN DE GRADO PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERA DE SISTEMAS Y COMPUTACIÓN

Línea de Investigación:

ELABORACIÓN DE PROPUESTAS DE DESARROLLO DE SOFTWARE PARA DISPOSITIVOS MÓVILES

Autor:

VERÓNICA DEL ROCÍO CHILQUINGA RIVERA

Director:

ING. MSc. GALO MAURICIO LÓPEZ SEVILLA

Ambato – Ecuador

Marzo 2011

PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR SEDE

AMBATO

ESCUELA DE INGENIERIA DE SISTEMAS

HOJA DE APROBACIÓN

Tema:

“ESTUDIO COMPARATIVO DE LAS TECNOLOGÍAS ASP, AJAX, J2EE, J2ME Y .NET PARA EL DESARROLLO DE APLICACIONES MÓVILES, APLICADO AL DESARROLLO DE UNA APLICACIÓN MÓVIL WEB PARA EL REGISTRO DE CALIFICACIONES Y ASISTENCIA DE LOS ALUMNOS DE LA ESCUELA DE INGENIERIA DE SISTEMAS DE LA P.U.C.E.S.A APLICADA AL AÑO ACADEMICO 2010.”

Autor:

VERÓNICA DEL ROCÍO CHILQUINGA RIVERA

Galo Mauricio López Sevilla Ing. MSc. f. _____

DIRECTOR DE LA DISERTACIÓN

Ricardo Patricio Medina Chicaiza Ing. MSc. f. _____

CALIFICADOR

Marco Polo Silva Segovia Ing. MSc. f. _____

CALIFICADOR

Santiago Acurio, Ing. MSc. f. _____

DIRECTOR ESCUELA DE SISTEMAS

Pablo Poveda, Ab. f. _____

SECRETARIO GENERAL PUCESA

DECLARACIÓN DE AUTENTICIDAD Y RESPONSABILIDAD

Yo, Verónica del Rocío Chilibuina Rivera portador de la cédula de ciudadanía No. 050315383-5 declaro que los resultados obtenidos en la investigación que presento como informe final, previo la obtención del título de Ingeniera de Sistemas y Computación son absolutamente originales, auténticos y personales.

En tal virtud, declaro que el contenido, las conclusiones y los efectos legales y académicos que se desprenden del trabajo propuesto de investigación y luego de la redacción de este documento son y serán de mi sola y exclusiva responsabilidad legal y académica.

Verónica del Rocío Chilibuina Rivera

CI. 050315383-5

RESUMEN

La revolución del fenómeno de la telefonía móvil se va extendiendo masivamente, dando lugar a la utilización cada vez más de dispositivos móviles, convirtiéndose en un elemento de uso necesario en la vida cotidiana de toda persona.

La potencia de los dispositivos móviles los hace adecuados para la ejecución de aplicaciones nativas o aplicaciones móviles web. El desarrollo de aplicaciones móviles permite estar a la vanguardia en comunicaciones y sistemas de información.

En el entorno educativo se ha propuesto el desarrollo de una aplicación móvil web para el registro de la asistencia y de las calificaciones de los alumnos de la Escuela de Ingeniería de Sistemas de la PUCESA aplicada al año académico 2010. Previo un estudio comparativo de las tecnologías ASP, AJAX, J2EE, J2ME y .NET.

Con la ejecución del proyecto se adquirió conocimientos y experiencia teórica y práctica, que ayudo a la elección de la plataforma .NET, entorno de desarrollo integrado que incluye herramientas para desarrollar aplicaciones para dispositivos inteligentes.

El Diseñador de ASP.NET Mobile complementa a ASP.NET y a .NET Framework, permitiendo generar aplicaciones Web para teléfonos móviles, PDA y localizadores. Este diseñador está integrado en el IDE de Visual Studio. Puede crear aplicaciones Web móviles, utilizar el diseñador móvil para modificar un formulario Web móvil y, a continuación, generar y ejecutar la aplicación, todo desde Visual Studio.

ABSTRACT

The revolution of mobile telephony phenomenon is spreading massively, leading to increasing the use of mobile devices, becoming a necessary element for everyone's everyday life.

The power of mobile devices makes them suitable for the execution of native applications or mobile web applications. The development of mobile applications allows us to be updated in communications and information systems.

In the educational environment the development of a mobile web application for the registration of care and skills of students in the School of Systems Engineering PUCESA applied to the academic year 2010 has been proposed. Taking into account a comparative study of technology ASP, AJAX, J2EE, J2ME and .NET.

With the implementation of the project theoretical knowledge and practical experience that helped to elect .NET platform was acquired. It is an integrated development environment that includes tools to develop applications for smart devices.

The ASP.NET Mobile Designer ASP.NET complements .NET Framework, enabling the generation of Web applications for mobile phones, PDAs and pagers. This designer is integrated into the Visual Studio. It can create mobile Web applications, using the mobile designer to modify a mobile Web form, and then build and run the application, everything from Visual Studio.

TABLA DE CONTENIDOS

DECLARACIÓN DE AUTENTICIDAD Y RESPONSABILIDAD	III
RESUMEN	IV
ABSTRACT	V
TABLA DE CONTENIDOS	VI
TABLA DE GRÁFICOS	XI
CAPITULO I	2
PROYECTO DE LA INVESTIGACIÓN	2
1.1. Antecedentes	2
1.2. Significado del Problema	2
1.3. Definición del Problema	3
1.4. Planteamiento del Tema	3
1.5. Delimitación del Tema	3
1.6. Objetivos	4
1.6.1. Objetivo General	4
1.6.2. Objetivos Específicos	5
1.7. Metodología de Trabajo	5
1.8. Justificación	6
CAPITULO II	8
TECNOLOGÍAS ASP. AJAX. J2EE. J2ME Y .NET EN EL DESARROLLO DE APLICACIONES MÓVILES	8
2.1. Aplicaciones Móviles	8
2.1.1. Introducción	8
2.1.2. ¿Qué es un Dispositivo Móvil?	10
2.1.3. Tipos de Dispositivos	10
2.1.4. Características de los Dispositivos Móviles	12
2.1.5. Tipos de Aplicaciones Móviles	13

2.1.6.	<i>Diseño de una Web Móvil</i>	13
2.1.7.	<i>Consideraciones de Diseño en Aplicaciones Móviles</i>	15
2.1.7.1.	<i>Patrones de Uso</i>	15
2.1.7.1.1.	<i>Tiempo de Inicio</i>	15
2.1.7.1.2.	<i>Enfocar en el objetivo</i>	16
2.1.7.2.	<i>Interface de usuario</i>	16
2.1.7.2.1.	<i>Interacción con datos</i>	16
2.1.7.2.2.	<i>Influencia del contexto en el uso de la aplicación</i>	18
2.1.7.3.	<i>Interconexión de Dispositivos</i>	19
2.1.7.3.1.	<i>Seguridad</i>	20
2.2.	<i>Tecnología ASP</i>	21
2.2.1.	<i>Servidores web. ¿Qué es IIS?</i>	22
2.2.2.	<i>Características de ASP</i>	23
2.2.3.	<i>Ventajas y Desventajas ASP</i>	24
2.2.4.	<i>Objetos ASP</i>	25
2.2.4.1.	<i>Objetos ASP incluidos</i>	25
2.2.5.	<i>Objetos ADO</i>	26
2.2.5.1.	<i>Objetos ADO incluidos</i>	26
2.2.6.	<i>Componentes ASP</i>	27
2.2.7.	<i>ASP.NET</i>	27
2.2.7.1.	<i>Fundamentos de ASP.NET</i>	28
2.2.7.2.	<i>Particularidades del Lenguaje</i>	29
2.2.8.	<i>Comparación ASP y ASP.NET</i>	30
2.3.	<i>Tecnología AJAX</i>	32
2.3.1.	<i>Características de la plataforma AJAX</i>	33
2.3.2.	<i>Arquitectura de AJAX</i>	34
2.3.3.	<i>Ventajas y Desventajas de AJAX</i>	36
2.3.4.	<i>Las Tecnologías AJAX</i>	37

2.4.	<i>Tecnología .NET</i>	40
2.4.1.	<i>Plataforma .net</i>	41
2.4.2.	<i>Visual Studio .NET</i>	43
2.4.3.	<i>Características de .NET</i>	44
2.4.4.	<i>Ventajas y Desventajas de .NET</i>	45
2.4.5.	<i>Framework.NET</i>	46
2.4.6.	<i>Arquitectura de la Solución</i>	47
2.5.	<i>Introducción a Java</i>	49
2.5.1.	<i>Plataforma Java</i>	50
2.5.1.1.	<i>Lenguaje Java</i>	50
2.5.1.2.	<i>Máquina virtual</i>	50
2.5.1.3.	<i>Bibliotecas</i>	51
2.5.2.	<i>Ediciones de Java</i>	51
2.5.2.1.	<i>Tecnología Java 2 Platform, Standard Edition (J2SE)</i>	53
2.6.	<i>Tecnología Java 2 Platform, Enterprise Edition (J2EE)</i>	55
2.6.1.	<i>¿Por qué J2EE?</i>	56
2.6.2.	<i>Arquitectura J2EE</i>	57
2.6.3.	<i>Características de J2EE</i>	59
2.6.4.	<i>Ventajas y desventajas de J2EE</i>	60
2.6.5.	<i>Componentes de J2EE</i>	61
2.6.6.	<i>Servicios J2EE</i>	64
2.7.	<i>Tecnología Java 2 Platform, Micro Edition (J2ME)</i>	67
2.7.1.	<i>Características de J2ME</i>	67
2.7.2.	<i>Arquitectura de J2ME</i>	68
2.7.2.1.	<i>Máquina Virtual</i>	69
2.7.2.2.	<i>Configuraciones</i>	74
2.7.2.3.	<i>Perfiles</i>	79
2.7.2.4.	<i>Paquetes Opcionales</i>	85

2.7.3.	<i>Ventajas y desventajas de J2ME</i>	90
CAPITULO III		91
DESARROLLO DE LA APLICACIÓN MÓVIL WEB PARA EL REGISTRO DE LAS CALIFICACIONES Y ASISTENCIA DE LOS ALUMNOS DE LA ESCUELA DE INGENIERÍA DE SISTEMAS DE LA PUCESA		91
3.1.	<i>Análisis Comparativo de las Tecnologías ASP, AJAX, J2ME, J2EE y .NET como resultado de la investigación.</i>	91
3.2.	<i>Ventajas y desventajas de las Tecnologías AJAX, ASP, .NET, J2EE, J2ME</i>	94
3.3.	<i>Proceso de Análisis</i>	96
3.3.1.	<i>Identificación del Problema.</i>	96
3.3.2.	<i>Descripción de Requerimientos de la Aplicación Móvil Web</i>	97
3.3.3.	<i>Metas</i>	97
3.3.4.	<i>Funciones</i>	98
3.3.5.	<i>Definición de Actores</i>	98
3.3.6.	<i>Datos</i>	99
3.3.7.	<i>Casos de Uso</i>	100
3.3.8.	<i>Diagrama de Casos de Uso</i>	107
3.3.9.	<i>Diagrama de Secuencia</i>	108
3.3.9.1.	<i>Inicio de Sesión</i>	108
3.3.9.2.	<i>Cerrar Sesión</i>	110
3.3.9.3.	<i>Registro Calificaciones y Asistencia</i>	112
3.3.9.4.	<i>Cambio de Password</i>	114
3.4.	<i>Proceso de Diseño</i>	115
3.4.1.	<i>Diseño global de la Aplicación Móvil Web.</i>	115
3.4.2.	<i>Diseño de la Base de Datos</i>	116
3.4.3.	<i>Diagrama Entidad Relación</i>	116
3.4.4.	<i>Definición de Procesos</i>	118
3.4.5.	<i>Seguridad del Sistema</i>	118

3.4.6.	<i>Diseño de Interfaz</i>	119
3.5.	<i>Proceso de Implementación</i>	123
3.5.1.	<i>Plataforma de Desarrollo .NET</i>	123
3.5.2.	<i>Diseñador de ASP.NET Mobile</i>	123
3.5.3.	<i>Formularios Web Forms</i>	123
3.5.4.	<i>Herramientas utilizadas en la implementación</i>	124
3.5.5.	<i>Requerimientos de Hardware y Software</i>	126
3.5.5.1.	<i>Requerimientos mínimos</i>	126
3.6.	<i>Proceso de Pruebas</i>	127
3.6.1.	<i>Pruebas de Usabilidad del Servidor: "Sistemasbd"</i>	128
3.6.2.	<i>Código de la Aplicación</i>	129
CAPITULO IV		133
VALIDACIÓN Y VERIFICACIÓN DE LOS RESULTADOS		133
4.1.	<i>Análisis General</i>	133
4.2.	<i>Validaciones</i>	133
4.3.	<i>Conclusiones</i>	136
4.4.	<i>Recomendaciones</i>	138
BIBLIOGRAFÍA		139
GLOSARIO DE TÉRMINOS		142
ABREVIATURAS UTILIZADAS		149
ANEXOS		156

TABLA DE GRÁFICOS

Tabla 1.1. Cuadro de Ventajas y Desventajas de ASP	25
Tabla 1.2. Cuadro Comparativo entre ASP y ASP.NET	31
Figura 2.1. Tecnologías agrupadas bajo el concepto de AJAX	33
Figura 2.2. Arquitectura de AJAX	35
Tabla 1.3. Cuadro de Ventajas y Desventajas de AJAX	37
Figura 2.3. Las Capas de la Plataforma .NET	42
Tabla 1.4. Cuadro de Ventajas y Desventajas de .NET	46
Figura 2.4. Ejecución Administrada en el Common Language Runtime	48
Figura 2.5. Java 2, Standard Edition	53
Figura 2.6. Estructura J2EE	55
Figura 2.7. Arquitectura J2EE	58
Tabla 1.5. Cuadro de Ventajas y Desventajas de J2EE	61
Figura 2.8. Arquitectura de J2ME	68
Figura 2.9. Estructura de la Arquitectura de J2ME	69
Figura 2.10. Pre-verificación de clases en J2ME	72
Tabla 2.6. Configuración, Máquina Virtual y algunos ejemplos	75
Tabla 1.7. Librerías de Configuración CDC	77
Tabla 1.8. Librerías de Configuración CLDC	79
Figura 2.11. Entorno de ejecución para J2ME	80
Tabla 1.9. Librerías del Foundation Profile	81
Tabla 1.10. Librerías del Personal Profile	82
Tabla 1.11. Librerías del perfil MIDP	85
Tabla 1.12. Cuadro de Ventajas y Desventajas de J2ME	90
Tabla 1.13. Cuadro comparativo de las características de las tecnologías ASP, AJAX, J2EE, J2ME y .NET	93
Tabla 1.14. Cuadro Comparativo de las ventajas y desventajas de las tecnologías	96

Tabla 1.15. Descripción del actor de los Casos de Uso	99
Tabla 1.16. Curso de Eventos Registro de las Calificaciones	101
Tabla 1.17. Curso de los Eventos Registro de la Asistencia	102
Tabla 1.18. Curso de los Eventos Cambiar Password	104
Tabla 1.19. Curso de los Eventos Iniciar Sesión	105
Figura 2.12. Diagrama de Casos de Uso	107
Figura 2.13. Inicio de Sesión	108
Tabla 1.21. Diagrama de Secuencias iniciar sesión	109
Figura 2.14. Cerrar Sesión	110
Tabla 1.22. Diagrama de Secuencias iniciar sesión	111
Figura 2.15. Registro Calificaciones	112
Tabla 1.23. Diagrama de Secuencias Registro Calificaciones	113
Figura 2.16. Cambiar Passsword	114
Tabla 1.24. Diagrama de Secuencias Cambiar Password	114
Figura 2.17. Arquitectura del sistema	115
Figura 2.18. Diagrama Entidad Relación	117
Figura 2.19. Página Principal	119
Figura 2.20. Inicio de Sesión	119
Figura 2.21. Autenticación de Usuario	120
Figura 2.22. Menú de opciones de la Aplicación	120
Figura 2.23. Cambiar Password	121
Tabla 1.25. Requerimientos mínimos para PC servidor	126
Tabla 1.26. Requerimientos mínimos para Dispositivo	126
Figura 1.27. Relaciones de las Tablas de la Base de Datos	128

CAPITULO I

PROYECTO DE LA INVESTIGACIÓN

1.1. Antecedentes

La evolución tecnológica en los últimos años ha permitido el crecimiento del mercado de dispositivos móviles con capacidades ricas en recursos multimedia, audio, video, fotografías, conectividad y comunicación.

A lo largo de la historia, las telecomunicaciones acompañaron el desarrollo, tanto en la actividad económica como en los aspectos sociales y culturales. Contribuyendo al crecimiento de la producción y a la mejora de las condiciones políticas de cada época.

En el Ecuador en particular su crecimiento exponencial, hace prever que pronto, prácticamente todos los ecuatorianos podrían tener acceso a comunicación móvil. Actualmente en el Ecuador se han planteado algunos proyectos interesantes con el uso de dispositivos móviles así tenemos:

El INEC (Instituto Nacional de Estadísticas y Censos del Ecuador) implementa la digitalización en el levantamiento de datos para la construcción de encuestas, a través de los Dispositivos Móviles de Captura de Datos (DMC). Los DMC son dispositivos electrónicos móviles que permiten aplicar procesos de validación y crítica, lo que a su vez garantiza la calidad de la información.

El proyecto MEDMOVIL, procura desarrollar una aplicación capaz de integrarse a un sistema de evoluciones clínicas, en coordinación con SoftCase Cia. Ltda., de manera que el médico pueda acceder por medio de dispositivos móviles. Una de las opciones de acceso puede ser el caso de un Pocket PC.

El proyecto CALLEJERO es el primer portal Wap en versión beta, diseñado especialmente para el uso en dispositivos BlackBerry en la ciudad de Cuenca. Con él se podrá localizar direcciones, puntos de interés, negocios, comercios y servicios públicos. Para esto será necesario que el dispositivo cuente con capacidad Wap.

Las soluciones móviles están mostrando sus beneficios para la gestión de las empresas en la mejora de la productividad, en la creación de nuevos servicios. Los retos actuales son desarrollar nuevos sistemas que permitan cubrir las necesidades y exigencias actuales en las distintas áreas, una de ellas la educativa.

1.2. Significado del Problema

La falta de una aplicación móvil web para el registro de las calificaciones y asistencia de los alumnos de la Escuela de Sistemas de la PUCESA ha generado la iniciativa de realizar un estudio comparativo de las tecnologías ASP, AJAX, J2EE, J2ME y .NET,

para lograr cubrir la necesidad de proporcionar al docente una herramienta segura y de fácil utilización al momento de efectuar dicho registro.

1.3. Definición del Problema

- ✓ Falta de un estudio comparativo entre las tecnologías ASP, AJAX, J2EE, J2ME y .NET que permitan determinar los alcances y limitantes en el diseño de aplicaciones móviles.
- ✓ No existe una aplicación móvil web que permita al docente de la Escuela de Sistemas de la PUCESA el registro de las calificaciones y asistencia de los alumnos mediante uso de algún dispositivo móvil (PDA, celular).

1.4. Planteamiento del Tema

Estudio Comparativo de las tecnologías ASP, AJAX, J2EE, J2ME y .NET para el desarrollo de aplicaciones móviles, aplicado al desarrollo de una aplicación móvil web para el Registro de Calificaciones y Asistencia de los Alumnos de la Escuela de Sistemas aplicada al año académico 2010.

1.5. Delimitación del Tema

- a) El lenguaje de programación se especificará previo un estudio y análisis de las tecnologías ASP, AJAX, J2EE, J2ME y .NET, permitiendo determinar cuál brinda un mejor entorno de desarrollo, que cubra con los objetivos de proporcionar una aplicación móvil web segura y de fácil uso en el ingreso de la información.

- b) La interfaz brindará un entorno amigable de fácil uso con la finalidad de que los docentes puedan efectuar el registro de las calificaciones y asistencia de sus alumnos sin complicaciones, para seguridad el ingreso a la aplicación será a través de usuario y password.
- c) La aplicación móvil web para el registro de las calificaciones y asistencia de los alumnos se la desarrollará para la Escuela de Sistemas de la PUCESA con la base de datos “sistemasdb” la cual esta implementada actualmente en el servidor de la universidad.
- d) Las pruebas de funcionalidad se las realizará mediante el uso de un celular, mediante el cual se llevara a cabo el registro de las faltas y notas de cada alumno.
- e) El estudio comparativo de las tecnologías será basado en la obtención de información que permita conocer las características, ventajas y desventajas de cada tecnología para elegir la más factible para el desarrollo de aplicaciones web móviles.

1.6. Objetivos

1.6.1. Objetivo General

Realizar un estudio comparativo de las tecnologías ASP, AJAX, J2EE, J2ME y .NET para el desarrollo de aplicaciones móviles.

1.6.2. Objetivos Específicos

- ✓ Realizar un análisis de las características, ventajas y desventajas resaltantes de las plataformas ASP, AJAX, J2EE, J2ME y .NET.
- ✓ Investigar los tipos de aplicaciones móviles y determinar su uso.
- ✓ Definir la tecnología o tecnologías que mejor se adapten al desarrollo del diseño de la aplicación móvil web.
- ✓ Diseñar una aplicación móvil web para el registro de calificaciones y asistencia de los alumnos de la Escuela de Ingeniería de Sistemas de la PUCESA.

1.7. Metodología de Trabajo

Investigación Bibliográfica: Porque está basada en un conjunto de técnicas y estrategias para localizar, identificar y acceder a documentos que contengan la información.

Metodología de Programación: Porque es un conjunto de normas, métodos y anotaciones que nos indican la forma de programar. Cada lenguaje de programación sigue una metodología distinta.

Metodología Inductiva: Porque la inducción es un tipo de razonamiento que nos lleva de lo particular a lo general, se basa en la observación de un fenómeno o problema y posteriormente se realizan investigaciones y estudios que conducen a la generalización que sería la solución.

Método Dialéctico: Porque propone que todos los fenómenos sean estudiados en sus relaciones con otros y en su estado de continuo cambio, ya que nada existe como un objeto aislado.

Por su carácter dinámico exponen no solamente los cambios cuantitativos, sino los radicales o cualitativos.

1.8. Justificación

El tema desarrollado será útil para guiar y cubrir el nivel de interés en el ámbito tecnológico, permitiendo adquirir y fortalecer nuevos conocimientos y destrezas de manera práctica.

Los retos actuales generan la constante necesidad de investigar nuevas tecnologías como ASP, AJAX, J2EE, J2ME y .NET para desarrollar aplicaciones web móviles, comprendiendo que muchas aplicaciones nacen por el afán e iniciativa de investigar para aprender y emprender grandes ideales.

Teniendo en cuenta que la tecnología ha extendido sus alcances, y que ya no es limitado el uso de una computadora personal, sino que se ha extendido al uso de dispositivos móviles como PDA, celulares entre otros, optimizando los recursos y facilitando procesos para lograr obtener beneficios que el servicio móvil ofrece a todos los sectores uno de ellos es el sector educativo.

Por lo tanto el avance de la tecnología ha originado la necesidad de desarrollar una aplicación móvil web para el registro de calificaciones y asistencia de los alumnos de la

Escuela de Sistemas de la Pontificia Universidad Católica del Ecuador Sede Ambato, en la cual se enfoca en permitir facilitar y agilizar las actividades del docente al momento del registro de las notas y asistencia de sus alumnos, logrando desarrollar una herramienta útil.

CAPITULO II

TECNOLOGÍAS ASP. AJAX. J2EE. J2ME Y .NET EN EL DESARROLLO DE APLICACIONES MÓVILES

2.1. Aplicaciones Móviles

2.1.1. Introducción

El rango de dispositivos móviles va desde dispositivos dedicados a tareas específicas, como los teléfonos celulares, hasta aquellos dispositivos de propósito general, como notebooks. Cada uno presenta diferentes conjuntos de desafíos para el diseño de aplicaciones móviles. Algunos de estos desafíos compartidos por la mayoría de los dispositivos móviles incluyen:

- ✓ La ubicación física del dispositivo y la configuración pueden cambiar impredeciblemente a medida que el dispositivo está conectado o desconectado de la red o se mueve entre dos puntos de conexión. La arquitectura de aplicación móvil debe soportar una operación consistente operando tanto online como offline y

proveer una conectividad continua mientras el dispositivo se mueve entre puntos de conexión.

- ✓ Los dispositivos energizados por baterías pueden operar por un tiempo limitado sin recargar o reemplazar las mismas. La arquitectura de una aplicación móvil debe ser diseñada para administrar la energía limitada de las baterías, mediante el uso de estrategias que prolonguen la vida útil al reducir el consumo sin sacrificar la performance del sistema.

Los dispositivos pequeños dedicados para un fin específico pueden tener limitaciones adicionales como un pantalla de tamaño reducido o un limitado almacenamiento o poder de cómputos. Una arquitectura de aplicaciones móviles debe proveer soporte para un amplio rango de dispositivos.

Las redes inalámbricas que utilizan tecnología Wi-Fi operan a velocidades de una red de área local, estas permiten a los usuarios estar conectados sin tener ninguna conexión física, eliminar los cargos por el acceso a través de redes públicas y dejar de lado las limitaciones de la conexión a través de un modem y una línea telefónica. Sin embargo las redes inalámbricas son susceptibles a externas interferencias lo que potencialmente afecta su confiabilidad, disminuye el ancho de banda efectivo, y presenta desafíos para una seguridad eficiente.¹

¹ <http://postgrado.info.unlp.edu.ar/Carrera/Magister/Ingenieria%20de%20Software/Tesis/Yorio.pdf>

2.1.2. ¿Qué es un Dispositivo Móvil?

Los dispositivos móviles son aquellos suficientemente pequeños para ser transportados y usados durante su transporte. Normalmente se sincronizan con un sistema de sobremesa para actualizar aplicaciones y datos.

Las características básicas de este tipo de dispositivos son:

- ✓ Son aparatos pequeños.
- ✓ Tienen capacidad de procesamiento.
- ✓ Pueden conectarse a la red.
- ✓ Tienen memoria limitada.
- ✓ Están diseñados específicamente para una función, pero que pueden llevar a cabo otras más generales.
- ✓ Normalmente se asocian al uso individual de una persona, tanto en posesión como en operación, el cual puede adaptarlos a su gusto.
- ✓ La mayoría de estos aparatos pueden ser transportados en el bolsillo del propietario.²

2.1.3. Tipos de Dispositivos

Para competir en el mercado, cada fabricante equipa a sus dispositivos con diversas características esperando atraer al mayor número de usuarios disponibles.

El dispositivo más común actualmente es el teléfono móvil, seguido de los conocidos PDAs (Personal Data Assistant).

² http://leo.ugr.es/J2ME/INTRO/intro_4.htm

Todos los dispositivos disponen en mayor o menor medida de posibilidades de desarrollo. Los fabricantes permiten que terceras partes puedan desarrollar aplicaciones que los usuarios puedan instalar y disfrutar en los dispositivos.

La aplicación más común encontrada en estos dispositivos suele ser un navegador Web, adaptado por supuesto al tamaño y forma del aparato en cuestión.

Una aplicación Web adaptable a navegadores móviles tiene un mercado potencial de usuarios enorme, ya que se estima que el número de teléfonos móviles en el mundo supera los 1000 millones de unidades.³

Entre los tipos de dispositivos móviles tenemos:

- ✓ Notebooks.
- ✓ PDAs
- ✓ Cell Phones
- ✓ SmartPhones
- ✓ Teléfonos Celulares.
- ✓ Rugged Devices
- ✓ UMPC

³ <http://aulas.campusmvp.com>

2.1.4. Características de los Dispositivos Móviles

Algunas de las características que hacen que estos dispositivos sean diferentes de los ordenadores de sobremesa son los siguientes:

- ✓ Funcionalidad limitada.
- ✓ No necesariamente extensible y actualizable.
- ✓ En pocos años el usuario deberá cambiarlo.
- ✓ Más barato.
- ✓ Menos complicado en su manejo.
- ✓ Fácil de aprender su operación.
- ✓ No se requieren usuarios expertos.

Algunos de estos dispositivos son los siguientes:

- ✓ Paginadores.
- ✓ Comunicadores de bolsillo.
- ✓ Teléfonos con pantalla para Internet (Internet Screen Phones).
- ✓ Sistemas de navegación de automóviles.
- ✓ Sistemas de entretenimiento.
- ✓ Sistemas de televisión e Internet (WebTV).
- ✓ Teléfonos móviles.
- ✓ Organizadores y asistentes personales digitales (PDA).⁴

⁴ http://leo.ugr.es/J2ME/INTRO/intro_4.htm

2.1.5. Tipos de Aplicaciones Móviles

Entendemos por aplicación móvil Web la que necesita de un navegador web o browser como iExplorer Mobile, Mínimo u Opera para ejecutarse. Aplicación y datos pueden residir remotamente en un servidor u obtenerse del mismo dispositivo móvil. Aplicación nativa es aquella que se instala en el propio dispositivo como cualquier otra aplicación y se desarrolla utilizando un lenguaje de programación compatible con el sistema operativo del dispositivo (Symbian) o de un framework de desarrollo (Java ME, .Net Compact Framework).

Las aplicaciones nativas requieren un mayor esfuerzo de desarrollo, tanto en horas como en especialización del equipo. El dispositivo y los lenguajes utilizados son más limitados y complejos que el entorno servidor o desktop. Siempre que sea posible un desarrollo por terceros, ya que algunos sistemas operativos móviles no lo permiten.

La Web es más sencilla de programar, permite que las actualizaciones sean transparentes al usuario y el desarrollador tiene todo el control de la misma al residir en servidor.⁵

2.1.6. Diseño de una Web Móvil

Para el diseño y desarrollo de una Web Móvil se puede concluir que:

- ✓ Hay que tener siempre en cuenta las características específicas de los dispositivos móviles, como el tamaño de la pantalla o la ausencia de teclado y ratón.

⁵<http://ispamat.wordpress.com/2007/05/09/aplicacion-movil-%C2%BFweb-o-nativa/>

- ✓ Se debe tratar de simplificar el contenido para proporcionar sólo aquel que pueda ser relevante para los usuarios de teléfonos móviles y pueda cubrir sus necesidades, dependiendo la lógica de negocio de la organización y de la naturaleza del sitio que se pretende desarrollar. Y por otro lado se debe descartar el contenido que no sea útil o necesario.
- ✓ El diseño de la arquitectura de información debe ser lo más simple posible, para conseguir una navegación sencilla y que el usuario no se sienta perdido en el sitio. Además, el contenido de una página no debe ser muy extenso y en caso de que fuese necesario se distribuirá a lo largo de varias páginas.
- ✓ Se debe utilizar el lenguaje XHTML-MP como lenguaje de etiquetas para la realización de las páginas orientadas a móviles ya que es lenguaje adoptado por el estándar WAP 2.0. Y además, se deben utilizar las hojas de estilo Wireless CSS para separar la presentación del contenido.
- ✓ No se debe abusar del uso de recursos externos ni de imágenes, ya que puede suponer un gasto extra de descarga y de tiempo. Además, es posible que el dispositivo no de soporte o se visualice incorrectamente.
- ✓ Se debe evitar el uso de scripts por no estar soportados por la mayoría de dispositivos. También hay que tratar de evitar el uso de tablas para albergar el contenido. En su lugar, se deben utilizar listas verticales.
- ✓ Cuando se trata de realizar una versión móvil de un sitio Web existente la mejor opción es realizar las páginas específicas del móvil a parte y redireccionar a los usuarios de los dispositivos móviles a dichas páginas. Siempre dejando la posibilidad

de consultar la versión de escritorio y viceversa por si se accede a través de un dispositivo de altas prestaciones que pueda soportarlo.⁶

2.1.7. Consideraciones de Diseño en Aplicaciones Móviles

2.1.7.1. Patrones de Uso

Los patrones de uso difieren fundamentalmente entre dispositivos móviles y PCs. Los usuarios tienden estar sentados frente a sus computadoras de escritorio en sesiones de larga duración, utilizando un teclado, pantallas de gran tamaño, un disco rígido, otros periféricos, para crear y editar grandes cantidades de información. En contraste, las personas utilizan los dispositivos móviles con mayor frecuencia pero en periodos de corta duración, comúnmente para revisar o actualizar pequeñas partes de información. Por ejemplo buscar un número telefónico o revisar su próximo compromiso.

2.1.7.1.1. Tiempo de Inicio

Un tiempo de inicio corto es una característica importante para las aplicaciones móviles. Dado que los usuarios tienden a utilizar frecuentemente sus dispositivos móviles en sesiones de corta duración, se hace imperativo que la aplicación se encuentre lista para ser empleada en el menor tiempo posible.

Una técnica habitual para acelerar el tiempo de inicio es la de guardar la última sesión del usuario para luego devolverlo al mismo punto donde la dejó. Esto puede ser válido para algunas aplicaciones no todas.

⁶ <http://majuelo.usal.es/diaweb20/archivos/10008835resumenMovil.pdf>

2.1.7.1.2. Enfocar en el objetivo

Enfocar en el objetivo es otra característica común de las aplicaciones móviles. La aplicación debe tener bien definido cuáles es el conjunto de tareas que realiza; las cuales deben poder hacerse con el menor número de ingreso de comandos por parte del usuario, y lo más rápido posible. La importancia del foco en el objetivo es puesta de manifiesto, en los dispositivos móviles, con la existencia de botones especiales que realizan tareas específicas, por ejemplo el botón de acceso a la aplicación de mensajería de un teléfono celular.

Durante el desarrollo de una aplicación móvil, es necesario esforzarse en identificar las tareas comunes a realizar para que las mismas sean sencillas y absolutamente posibles de llevar a cabo por el usuario.

2.1.7.2. Interface de usuario

Hoy en día esperan dispositivos móviles que los ayuden a administrar la información, analizar y seleccionar datos, y realizar transacciones, sin tener que esperar a regresar a sus computadoras de escritorio. Por estos motivos, las interfaces con las que el usuario debe interactuar juegan un papel importante al momento de tomar decisiones de diseño.

2.1.7.2.1. Interacción con datos

Existen dos aspectos a tener en cuenta en los dispositivos móviles, referidos a interacción con datos:

- ✓ **Características de la pantalla.** Debe presentar la información en un formato legible. Esto es en parte función del tamaño de la pantalla, este debe ser lo

suficientemente grande para permitir el uso de menús, check boxes, botones, gráficos y otras técnicas de información. Y por otra parte es función de cuan eficientemente es utilizada la pantalla. Solamente los controles relativamente inmediatos a la información que se está mostrando, deben estar visibles, el resto deberían permanecer ocultos hasta que sean necesarios.

- ✓ **Modos de ingreso de datos disponibles.** Para un trabajo eficiente de la información que se está mostrando los usuarios deben disponer de una variedad de modos de ingreso. Actualmente se puede optar por plataformas que soportan entradas infrarrojas, note pad, teclados en pantalla, touch screen, etc. La aplicación en si necesita ser diseñada para trabajar rápidamente en una pantalla de dimensiones reducidas.

El desarrollo de una aplicación para un fin específico puede utilizar un enfoque similar. T9 es un ejemplo de una buena solución a una restricción de un dispositivo móvil. Permite el ingreso rápido de texto en un teléfono celular el cual tiene las 12 teclas características de un teclado telefónico.

T9 predice que quiere escribir el usuario en base a un diccionario y sugiere la siguiente letra o letras. Produce un ahorro de tiempo, ya que no se debe esperar para ingresar otra letra, aun en el caso de que esté en la misma tecla. Así mismo tiene la posibilidad de aprender aquellas expresiones que el usuario utiliza con frecuencia e incorporarlas al diccionario. Es una solución pensada exclusivamente a un problema propio de una aplicación móvil, que si bien no es adecuada para escribir un libro, lo es para el ingreso de mensajes de texto.

2.1.7.2.2. Influencia del contexto en el uso de la aplicación

Es de destacar como el contexto en el que son utilizadas las aplicaciones móviles influye en la forma que los usuarios interactúan con la misma.

Supóngase la situación donde el usuario con su dispositivo móvil se encuentran en un ambiente público o ruidoso. La alternativa de dotar a la aplicación con la capacidad de ser operada a través del reconocimiento de voz no siempre es una buena idea aun en los casos donde tecnológicamente sea posible. Por otra parte es también importante que el uso del dispositivo móvil no moleste a las personas de su entorno.

Una pantalla “touch-screen” ofrece un rico entorno para aplicaciones de navegación, pero si desarrollan interfaces pequeñas con gran cantidad de opciones, las mismas serán difíciles de utilizar en un entorno móvil, donde existen saltos y vibraciones como por ejemplo dentro un vehículo en movimiento.

Otro punto a tener en cuenta es la forma de operación habitual del dispositivo. Varios dispositivos fueron pensados para ser operados con una sola mano. La mayoría de los teléfonos celulares adhieren a este criterio. Algunos otros fueron pensados para que sean sostenidos con una mano y operados con la otra, tal es el caso de la mayoría de las PDAs. Las laptops generalmente requieren dos manos y una superficie plana para un uso eficiente. Una aplicación móvil debe seguir el paradigma del dispositivo para el que será desarrollada, por ejemplo no construir una aplicación que requiera dos manos para su uso si la misma correrá en un teléfono celular, el cual las personas típicamente lo operan con solo una mano.

2.1.7.3. Interconexión de Dispositivos

El método de conexión que el dispositivo móvil pueda establecer con la red y con otros dispositivos, tendrá una influencia directa sobre las consideraciones de diseño.

Dentro de las posibilidades de conexión a través de cables o sin ellos, se tiende al uso de redes inalámbricas, especialmente dado la proliferación que estas han tenido en los últimos tiempos.

En la actualidad los métodos comúnmente disponibles son:

1. Infrarrojo (IR) o conexión por cable entre dos dispositivos móviles o entre el dispositivo móvil y el teléfono celular.
2. Cuna de sincronización (Synchronization Cradle) conectada a un computadora de escritorio o directamente a la red corporativa
3. Conexión Dial-Up a la red corporativa
4. Conexión inalámbrica a la red corporativa
5. Conexión de red inalámbrica a través de un teléfono celular u otro dispositivo con capacidades inalámbricas
6. LAN inalámbrica (IEEE802.11b) dentro de edificios de oficinas, aeropuertos, hoteles, etc.
7. Redes inalámbricas de área personal (PAN: Personal Área Network) basadas en el estándar Bluetooth

2.1.7.3.1. Seguridad

La seguridad es una de las mayores preocupaciones tanto para entornos móviles como no móviles.

Una solución de seguridad para aplicaciones móviles debe tratar todo el trayecto móvil, es decir desde los sistemas back-end hasta el dispositivo del cliente. Bajo este concepto se puede decir que la seguridad en una aplicación móvil tiene tres preocupaciones principales:

- **Privacidad de la mensajería.** Asegurar que los datos sensibles no estén comprometidos durante su transmisión. Esto comúnmente es realizado con el uso de criptografía.
- **Autenticación.** Asegurar que la identidad de todos los usuarios involucrados en la comunicación es correcta. Usualmente alcanzado con el uso de certificados digitales o técnicas de usuario / password
- **Seguridad en el dispositivo.** Asegurar la protección de los datos almacenados en el dispositivo móvil, usualmente logrado a través del uso de password de protección y encriptación.

Según la arquitectura de la aplicación móvil, se dispone de distintos mecanismos de seguridad ya provistos por sistemas comerciales, a los cuales se puede adherir.⁷

⁷ <http://postgrado.info.unlp.edu.ar/Carrera/Magister/Ingenieria%20de%20Software/Tesis/Yorio.pdf>

2.2. Tecnología ASP

Las siglas ASP corresponden a las palabras Active Server Pages (Páginas Activas en el Servidor). ASP es una tecnología desarrollada por Microsoft para crear páginas web de contenido dinámico apoyándose en scripts ejecutados en el servidor. Su funcionamiento se basa, principalmente, sobre servidores Microsoft con Internet Information Server para Windows NT o 2000, y en caso de contar con un sistema operativo Windows 95 o 98 se utiliza un servidor web personal, especialmente el Personal Web Server.

Se trata básicamente de un lenguaje de tratamiento de textos (scripts), basado en Basic, y que se denomina VBScript (Visual Basic Script). Se utiliza casi exclusivamente en los servidores Web de Microsoft (Internet Information Server y Personal Web Server). Los scripts ASP se ejecutan, por lo tanto, en el servidor y puede utilizarse conjuntamente con HTML y Javascript para realizar tareas interactivas y en tiempo real con el cliente.

Con ASP se pueden realizar fácilmente páginas de consulta de bases de datos, funciones sencillas como obtener la fecha y la hora actual del sistema servidor, cálculos matemáticos simples, etc.

Lo interesante de este modelo tecnológico es poder utilizar diversos componentes ya desarrollados como algunos controles ActiveX así como componentes del lado del servidor, tales como CDONTS, por ejemplo, que permite la interacción de los scripts con el servidor SMTP que integra IIS.

Se facilita la programación de sitios web mediante varios objetos integrados, como por ejemplo un objeto de sesión basada en cookies, que mantiene las variables mientras se pasa de página a página.

ASP es una tecnología dinámica funcionando del lado del servidor, lo que significa que cuando el usuario solicita un documento ASP, las instrucciones de programación dentro del script son ejecutadas para enviar al navegador únicamente el código HTML resultante. La ventaja principal de las tecnologías dependientes del servidor radica en la seguridad que tiene el programador sobre su código, ya que éste se encuentra inicialmente en los archivos del servidor que al ser solicitado a través del web, es ejecutado, por lo que los usuario no tienen acceso más que a la página resultante en su navegador.

2.2.1. Servidores web. ¿Qué es IIS?

Internet Information Services (o Server), IIS, es una serie de servicios para los ordenadores que funcionan con Windows. Originalmente era parte del Option Pack para Windows NT.

Luego fue integrado en otros sistemas operativos de Microsoft destinados a ofrecer servicios, como Windows 2000 o Windows Server 2003. Windows XP Profesional incluye una versión limitada de IIS. Los servicios que ofrece son: FTP, SMTP, NNTP y HTTP/HTTPS.

Este servicio convierte a un ordenador en un servidor de internet o Intranet, es decir que en los ordenadores que tienen este servicio instalado se pueden publicar páginas web tanto local como remotamente (servidor web).

El servidor web se basa en varios módulos que le dan capacidad para procesar distintos tipos de páginas, por ejemplo Microsoft incluye los de Active Server Pages (ASP) y ASP.NET. También pueden ser incluidos los de otros fabricantes, como PHP o Perl.

Existen otros servidores web que pueden usarse como alternativa a este que es de propiedad de Microsoft Corporation, por ejemplo Apache, Cherokee que son desarrollados en Software libre y otros muchos.⁸

2.2.2. Características de ASP

- ✓ ASP es totalmente gratuito para Microsoft Windows NT o Windows 95/98.
- ✓ El código ASP se puede mezclar con el código HTML en la misma página (no es necesario compilarlo por separado).
- ✓ El código ASP se puede escribir con un simple editor de textos como el Bloc de notas de Windows o UltraEdit.
- ✓ Cómo el código ASP se ejecuta en el servidor, y produce como salida código HTML puro, su resultado es entendible por todos los navegadores existentes.
- ✓ Mediante ASP se pueden manipular bases de datos (consultas, actualizaciones, borrados, etc.) de prácticamente cualquier plataforma, con tal de que proporcione un driver OLEDB u ODBC.
- ✓ ASP permite usar componentes escritos en otros lenguajes (C++, Visual Basic, Delphi), que se pueden llamar desde los guiones ASP.
- ✓ Sin modificar la instalación, los guiones ASP se pueden programar en JScript o VBScript (este último es el más usado porque mas programadores lo dominan), pero también existen otros lenguajes, como Perlscript y Rexx, que se pueden emplear para programar ASP.

⁸ http://tgp0607.awardspace.com/Recursos_ASP.pdf

- ✓ Se ha portado a la plataforma Java por Chili!Soft y Halcyon Software, lo que permite que ASP sea usado en casi cualquier sistema operativo.⁹

2.2.3. Ventajas y Desventajas ASP

Ventajas	Desventajas
Permite acceder a bases de datos de una forma sencilla y rápida.	Al ser VBScript un lenguaje propietario de Microsoft, sólo es procesado por defecto por los servidores de esta casa comercial, ya sea en las versiones de Internet Information Server (IIS) o en las de Personal Web Server (PWS).
Las páginas se generan dinámicamente mediante el código de scripts.	La mayoría de los componentes no son gratuitos.
El código de script se ejecuta en el servidor.	
Desde una página ASP se pueden ejecutar servidores OLE en el servidor de web, lo que abre un abanico de nuevas posibilidades sólo accesibles previamente usando CGI (Common Gateway Interface - Interfaz de entrada común) y filtros ISAPI (Internet Server Application Programming Interface): acceso a base de datos, acceso a fichero, envío de correo, etc.	
El código ASP se puede escribir con un simple editor de textos.	
No se depende del navegador que se emplee.	
Es muy versátil para la utilización de plantillas	

⁹ http://gplsi.dlsi.ua.es/~slujan/asp/Introduccion_a_ASP.htm

(templates) para la construcción de páginas Web dinámicas.	
Existe muchos Websites que brindan material que se puede utilizar como soporte ante cualquier tipo de desconocimiento, o duda en el desarrollo de aplicaciones ASP	

Tabla 1.1. Cuadro de Ventajas y Desventajas de ASP

Fuente: Investigador

2.2.4. Objetos ASP

ASP no es un lenguaje de programación por sí solo. Técnicamente, ASP se compone de objetos que son llamados desde VBScript o JScript para realizar funciones altamente útiles, como capturar datos enviados por los usuarios, responder a entrada de usuarios, administrar aplicaciones y sesiones y administrar el servidor.

2.2.4.1. Objetos ASP incluidos

Los objetos incluidos en ASP son los siguientes:

- ✓ **Objetos Request (Solicitud) y Response (Respuesta).** Estos objetos capturan datos que ingresan del usuario y responden con texto procesado.
- ✓ **Objetos Application (Aplicación) y Session (Sesión).** Estos objetos permiten al diseñador configurar la funcionalidad del sitio Web como una aplicación cuyas variables tienen alcance sobre todo el conjunto de páginas en uso, y como una sesión cuyas variables tienen un alcance sobre las páginas en uso por parte de un usuario individual.

- ✓ **Objeto Server (Servidor).** Este objeto permite al desarrollador interactuar con el servidor, codificar HTML y los URL, fijar timeouts para scripts, y crear instancias de otros objetos y componentes dentro de la aplicación.

2.2.5. Objetos ADO

ADO (ActiveX Data Object) es el término utilizado para uno de los más valiosos componentes que ASP puede llamar. Los objetos ADO permiten acceso directo y conveniente a la base de datos y otras fuentes de datos, y esta funcionalidad sirve para muchos propósitos comunes en los sitios Web dinámicos.

2.2.5.1. Objetos ADO incluidos

ADO incluye los siguientes objetos:

- ✓ **Objeto Connection.** Este objeto crea una conexión entre sus scripts ASP y una base de datos o fuente de datos.
- ✓ **Objeto Command.** Este objeto permite ejecutar comandos específicos sobre una fuente de datos.
- ✓ **Objeto Recordset.** Este objeto proporciona el control sobre un conjunto de registros y campos de una fuente de datos, de manera que pueda realizar todas las operaciones comunes sobre registros disponibles cuando trabaja con una tabla de base de datos, como buscar, editar, agregar, borrar, etc.

2.2.6. Componentes ASP

Como se menciona anteriormente, ADO es un componente ASP. Los componentes son valiosos porque ofrecen la funcionalidad programada previamente que, de otra forma, le tomaría muchas horas de esfuerzo repetirla por sí solo, aun suponiendo que tuviera excelentes habilidades de programación. Los componentes incluidos son esencialmente gratuitos, y el costo de la mayoría de componentes de terceros es relativamente bajo. Los componentes de terceros generalmente se les prueban y depura para efectos de confiabilidad, y exponen métodos y propiedades que puede llamar y utilizar realmente, al igual que los objetos incorporados.

Los motores de script proporcionan acceso a su propio conjunto de objetos, en especial los objetos Dictionary (Diccionario), FileSystem Objetc, Drive (Unidad), Folder (Carpeta) y File (Archivo). De manera general, a estos objetos se les llama objetos Script de librería para tiempo de ejecución.¹⁰

2.2.7. ASP.NET

ASP.NET es parte del entorno de trabajo de .NET. Como programador, interactuará utilizando los tipos apropiados en la biblioteca de clases para escribir programas y diseñar formularios web. Cuando un cliente solicita una página, se ejecuta el servicio ASP.NET (dentro del entorno del CRL), después ejecuta su código y crea una página HTML final para enviarla al cliente.¹¹

¹⁰ Fundamentos de Programación en ASP 3.0

¹¹ Alcazar, 2009, ASP.NET Manual de Referencia

2.2.7.1. Fundamentos de ASP.NET

El entorno necesario para poder desarrollar aplicaciones ASP.NET es el nuevo producto de Microsoft: .NET Framework. Este entorno de programación permite tratar ASP.NET como un lenguaje del tipo orientado a objetos. En este punto es donde podemos afirmar que ASP.NET rompe completamente con las anteriores versiones de ASP los puntos fundamentales de la nueva estructura ASP.NET son:

- ✓ Básicamente, los lenguajes para programar ASP.NET son VB.NET, JScript y el nuevo Visual C#, aunque realmente existen más de 20 (Perl, .NET, Cobol.NET, etc)
- ✓ ASP.NET forma parte de la estructura .NET (lenguajes orientados a objetos) y no es una versión ASP 4 (lenguajes interpretados).
- ✓ ASP.NET crea aplicaciones Web rápidas, escalables, manejables y flexibles, pero, por encima de todo, son fáciles de entender y codificar.
- ✓ El código de las aplicaciones ASP.NET se compila a través del motor CLR, que compila justo a tiempo (JIT, Just In Time, análogo a Java). Optimiza y almacena la compilación en memoria cache. Recuerde que el paso intermedio consistía en traducirlo a un lenguaje común MSIL.
- ✓ Los parámetros de configuración se almacenan en archivos de tipo XML, porque es de lectura universal y se pueden generar con cualquier editor de textos.
- ✓ La seguridad de las aplicaciones ASP.NET es muy adaptable a las necesidades de cada situación, pues se basa en un conjunto de esquemas de autorización que puede configurarse ampliamente.

- ✓ ASP.NET puede acceder al potente grupo de librerías y clases que contiene .NET Framework para configurar transacciones TCP/IP y DNS (Domain Name System), a través de XML y con los servicios Web.

2.2.7.2. Particularidades del Lenguaje

Los tres lenguajes suministrados por Microsoft para la programación de aplicaciones Web con ASP.NET son JScript, VB.NET y V.C#. Otros fabricantes han enunciado mucho más: Python, Cobol.NET, Perl.NET, etc.

JScript ha sido modificado para comportarse como un lenguaje orientado a objetos.

VB.Net reemplaza VBScript como base en la programación ASP. El potencial de desarrollo, evidentemente, ha aumentado muchísimo ,pues no se ve limitado al reducido número de funciones que poseía el lenguaje script.

El nuevo lenguaje C# se ha descrito como J++ mark 2, pero no es cierto. La estructura de programación se parece mucho más a la del lenguaje C++. Se ha mejorado este último eliminando los bugs (errores) y la optimización final se correspondería con C#.

Gracias al entorno .NET Framework, una aplicación Web escrita en un lenguaje determinado puede heredar funciones escritas en otros lenguajes. A su vez, dicha aplicación puede extenderse o referenciarse por otras aplicaciones escritas en otros lenguajes. En definitiva gracias al lenguaje común de compilación MSIL, la

comunicación entre objetos y aplicaciones dentro de la arquitectura .NET no implica ningún problema añadido.¹²

2.2.8. Comparación ASP y ASP.NET

ASP	ASP.NET
Código desorganizado (mezcla de lenguajes). No hay una distinción entre el contenido de una página y su comportamiento.	Proporciona un modelo de desarrollo tanto para aplicaciones basadas y no basadas en el Web. Su estructura es más organizada.
La información de configuración se almacena en la metabase del servidor Web, Internet Information Server, por lo que hace difícil mover una aplicación de una máquina a otra.	La información de configuración se almacena en archivos textuales de configuración en formato XML denominados config.web, que pueden ser ubicados en los mismos directorios que los archivos de aplicación. Utiliza una arquitectura de configuración jerárquica. En ASP.NET, todos los archivos que un servidor Web necesita están ubicados bajo la carpeta raíz del sitio Web.
Es habitual escribir una gran cantidad de código para resolver necesidades sencillas.	Posee un modelo declarativo a la programación Web: los controles de servidor funcionan en una página Web solo con declararlos. Cuando se carga la página ASP.NET, se instancian los controles listados en la página ASP y es responsabilidad del control emitir código HTML que el navegador pueda entender.
Solo utiliza lenguajes Script (VBscript y Jscript) los cuales son interpretados y no	Soporta múltiples lenguajes compilados (C#, VB.Net, C++.net) y fuertemente tipados, incluyendo

¹² Pratdepadua, 2009, Domine ASP.NET

son fuertemente tipados.	(Scripts) que ya son compilados antes de su ejecución, lo que permite una ganancia en el rendimiento.
El único tipo de autenticación que podemos utilizar es la autenticación Windows.	Permite varios tipos de identificación y autenticación de usuario: Windows, Passport y Cookies. El servidor ejecuta el código como si el usuario estuviese presente.
Soporta un archivo declarativo global global.asax para las directivas de programa a nivel de aplicación, eventos, declaraciones de objetos globalmente accesibles y el estado de la aplicación.	Similar a ASP, ASP.NET soporta un archivo declarativo global global.asax, pero ahora este archivo soporta más de 15 eventos nuevos.
La Web, es un modelo sin estado y sin correlación entre peticiones HTTP. Esto dificulta el proceso de desarrollo de aplicaciones Web, ya que las aplicaciones generalmente necesitan mantener el estado entre varias peticiones. ASP.NET mejora y extiende los servicios de gestión de estado introducidos por ASP. El estado de aplicación, es específico de la instancia de una aplicación y no es Persistente	Proporciona tres tipos de estado a las aplicaciones Web: aplicación, sesión y usuario. El estado de aplicación, como en ASP, es específico de la instancia de una aplicación y no es persistente. El estado de sesión, se almacena en un proceso separado, puede configurarse para ser almacenado en una base de datos, en un proceso o en una máquina separada. Es útil cuando se implanta una granja de servidores. El estado de usuario es similar al estado de sesión, pero no expira y es persistente.
Las aplicaciones ASP tienen una extensión .asp	Las páginas ASP.NET utilizan la extensión .aspx. se puede utilizar tanto páginas ASP como ASP.NET en un mismo sitio Web sin la obligación de reconvertirlas a páginas ASP.NET.

Tabla 1.2. Cuadro Comparativo entre ASP y ASP.NET

Fuente: Investigador

2.3. Tecnología AJAX

El termino AJAX hace referencia a un mecanismo de combinación de tecnologías y estándares de cliente, consistente en la solicitud asíncrona de datos al servidor desde una página Web y la utilización de éstos para actualizar una parte de la misma, sin obligar al navegador a realizar una recarga completa la página.

Además de transmitirse una menor cantidad de información por la red, al enviarse en la respuesta únicamente los datos a actualizar, la realización de la operación en modo asíncrono permite al usuario de la aplicación seguir trabajando sobre la interfaz mientras tiene lugar el proceso de recuperación de los datos actualizados; como se puede sospechar, esto supone una enorme reducción del tiempo de espera para el usuario, percibiendo éste la sensación de estar trabajando sobre una aplicación de escritorio.¹³

En realidad, el término AJAX es un acrónimo de Asynchronous JavaScript + XML, que se puede traducir como "JavaScript asíncrono + XML".

El artículo define AJAX de la siguiente forma:

“Ajax no es una tecnología en sí mismo. En realidad, se trata de varias tecnologías independientes que se unen de formas nuevas y sorprendentes.”

Las tecnologías que forman AJAX son:

- ✓ XHTML y CSS, para crear una presentación basada en estándares.
- ✓ DOM, para la interacción y manipulación dinámica de la presentación.

¹³ Martín Sierra, Antonio J., 2009, Ajax en J2EE

- ✓ XML, XSLT y JSON, para el intercambio y la manipulación de información.
- ✓ XMLHttpRequest, para el intercambio asíncrono de información.
- ✓ JavaScript, para unir todas las demás tecnologías.

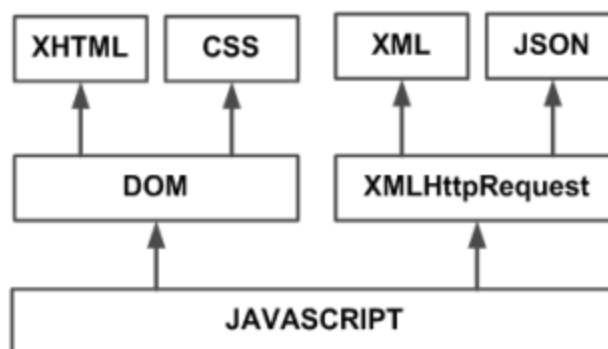


Figura 2.1. Tecnologías agrupadas bajo el concepto de AJAX

Fuente: < <http://www.librosweb.es/ajax/capitulo1.html> >

Desarrollar aplicaciones AJAX requiere un conocimiento avanzado de todas y cada una de las tecnologías anteriores.¹⁴

2.3.1. Características de la plataforma AJAX

- ✓ Las aplicaciones son más interactivas, responden a las interacciones del usuario más rápidamente, al estilo aplicaciones de escritorio.
- ✓ Estas aplicaciones tienen un aspecto (look and feel) muy similar a las aplicaciones de escritorio tradicionales sin depender de plugins o características específicas de los navegadores.

¹⁴ <http://www.librosweb.es/ajax/capitulo1.html>

- ✓ Se reduce el tamaño de la información intercambiada
- ✓ Muchas micro-peticiones, pero el flujo de datos global es inferior
- ✓ Se libera de procesamiento a la parte servidora (se realiza en la parte cliente)
- ✓ AJAX actualiza porciones de la página en vez de la página completa.¹⁵

2.3.2. Arquitectura de AJAX

Dar al usuario una buena experiencia en el uso de las aplicaciones web es uno de los puntos que permiten que este retorne a visitarlo. Todos hemos experimentado las aplicaciones web en las que para cargar un dato necesitamos esperar a que toda la página se cargue para obtener resultados, imaginarse recargar toda la página por un dato específico no es realmente agradable, gracias AJAX podemos solventar esto.

Lo que Ajax utiliza JavaScript combinado con XML para recargar la información del servidor sin refrescar la página, lo que mejora la experiencia del usuario y el ancho de banda requerido para el paso de información disminuye

¹⁵ <http://www.monografias.com/trabajos43/ajax/ajax2.shtml#ajax>

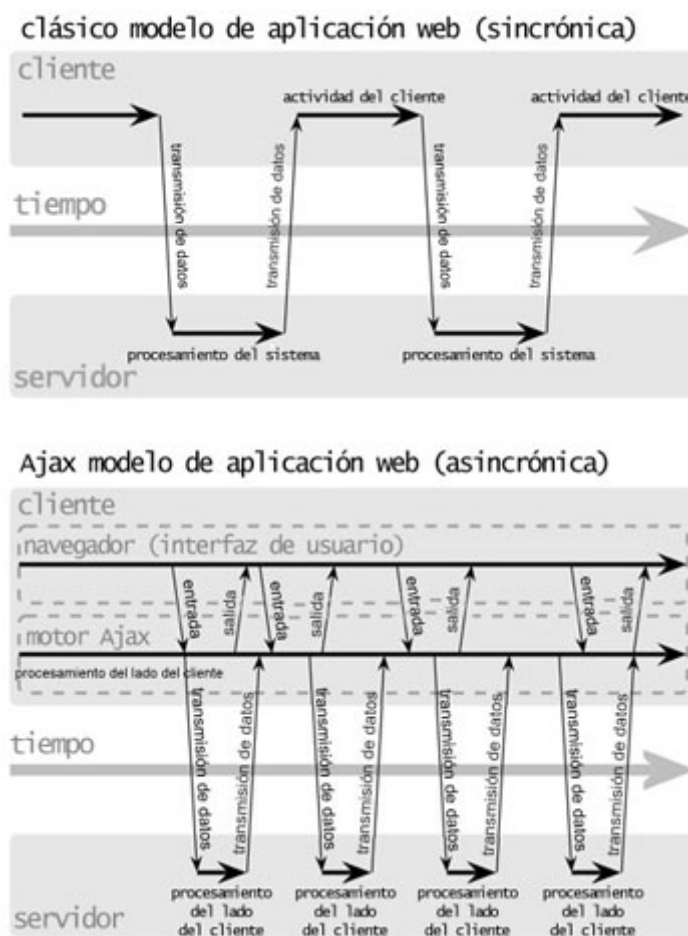


Figura 2.2. Arquitectura de AJAX

Fuente: < <http://faustol.wordpress.com/2007/12/12/arquitectura-ajax/>>

Podemos observar la diferencia entre el modelo clásico y un modelo con AJAX.

Dentro de los casos en los cuales es aconsejable la utilización de AJAX se tiene:

- ✓ Validación de información de formularios
- ✓ Autocompletar información (correos electrónicos, nombres, etc.)
- ✓ Operaciones en donde se maneja maestro detalle
- ✓ Barras de progreso, mapas, entre otros.

Existen varias alternativas para implantar AJAX, dentro de las que más se recomienda son las librerías de cliente para Java Script, por el mayor control del lado del cliente, así como navegadores, facilidad de desarrollo, etc.¹⁶

2.3.3. Ventajas y Desventajas de AJAX

Ventajas	Desventajas
Todos los componentes y estándares son de código abierto y utiliza tecnologías ya existentes.	Mantenimiento de código: El código de AJAX tiene la tendencia a ponerse complicado mientras crezca el proyecto
Complemento en las capas básicas de la arquitectura cliente-servidor.	Usabilidad, los botones del navegador “Atrás” y “Refrescar” pueden funcionar mal en algunas ocasiones, ya que aunque el contenido cambia dinámicamente, la pagina sigue siendo la misma
Causa-efecto el flujo de navegación lo va creando el usuario, se evita una capa de control de flujo (J2EE)	Dificultad en encontrar errores (debug), el procesamiento lógico de las aplicaciones AJAX se encuentra entre el servidor y el cliente.
Implementa validaciones en el lado servidor por motivos de seguridad, con las aplicaciones tradicionales.	Posibles problemas de seguridad, AJAX por ser algo nuevo puede ser susceptible a los ataques de hackers, si no se sabe usarlo correctamente.
Mayor interactividad al enviar peticiones al servidor Web para traer solamente los datos que son necesarios.	Accesibilidad, existe el riesgo de que algunos usuarios no puedan usar aplicaciones AJAX ya que se basan en JavaScript.
El resultado es una interfaz con mayor velocidad de respuesta, fluida y en menor tiempo de	Problemas con navegadores antiguos que no implementan esta tecnología

¹⁶ <http://faustol.wordpress.com/2007/12/12/arquitectura-ajax/>

procesamiento en el servidor Web.	
Los métodos de carga asincrónicos reducen el cuello de botella. Este método utiliza múltiples conexiones en vez de la única que usan muchas páginas Web.	No funciona si el usuario tiene desactivado el JavaScript en su navegador.
Se adiciona una capa lógica de presentación para gestionar los eventos que se transforman en llamados al servidor actualizando la interfaz.	Requiere programadores que conozcan todas las tecnologías que intervienen en AJAX.
Se han desarrollado muchos toolkit, librerías y frameworks	
Portabilidad porque no se requiere ningún plug-in o software adicional	
Es independiente del tipo de tecnología de servidor que se utilice	

Tabla 1.3. Cuadro de Ventajas y Desventajas de AJAX

Fuente: Investigador

2.3.4. Las Tecnologías AJAX

Básicamente, el mecanismo de funcionamiento de una aplicación AJAX consiste en realizar peticiones HTTP de forma asíncrona al servidor desde la página cliente, utilizando los datos recibidos en la respuesta para actualizar determinadas parte de la página.

Para poder realizar estas operaciones, las aplicaciones AJAX se apoyan en las siguientes tecnologías:

- ✓ XHTML y CSS. La interfaz gráfica de una aplicación AJAX es una página Web cargada en un navegador, XHTML y CSS son los dos estándares definidos por el

W3C para la construcción de páginas Web: mientras que XHTML se basa en la utilización de un conjunto demarcas o etiquetas para la construcción de la página, el estándar CSS define una serie de propiedades de estilo que pueden aplicarse sobre las etiquetas XHTML, a fin de mejorar sus capacidades de presentación.

- ✓ JavaScript. Como cualquier otro tipo de programa informático, las aplicaciones AJAX deben ser escritas en algún lenguaje de programación, este lenguaje es JavaScript.
- ✓ Los bloques de código Java Script que forman la aplicación AJAX se encuentran embebidos dentro de la propia página Web o bien en archivos independientes con extensión .js que son referenciado desde ésta. Estos bloques de código son traducidos y ejecutados por el navegador, bien cuando se produce la carga de la pagina en él bien cuando tiene lugar alguna acción en el usuario sobre la misma. Mediante código JavaScript las aplicaciones AJAX pueden realizar solicitudes al servidor desde la página Web, recuperar los datos enviados en la respuesta y modificar el aspecto de la interfaz.
- ✓ XML. Aunque podría utilizarse cualquier otro formato basado en texto, cuando la aplicación del servidor tiene que enviar una serie de datos de forma estructurada al cliente, XML resulta la solución más práctica y sencilla, ya que podemos manipular fácilmente un documento de estas características y extraer sus datos desde el código JavaScript cliente.
- ✓ DOM. El Modelo de Objeto de Documento (DOM) proporciona un mecanismo estándar para acceder desde código a la información contenida en un documento de texto basado en etiquetas.

Mediante el DOM, tanto páginas Web como documentos XML pueden ser tratados como un conjunto de objetos organizados de forma jerárquica, cuyas propiedades y métodos pueden ser utilizados desde código JavaScript para modificarse el contenido de la página en un caso, o para leer los datos de la respuesta en otro.

- ✓ El objeto XMLHttpRequest. Se trata del componente fundamental de una página AJAX. A través de sus propiedades y métodos es posible lanzar peticiones en modo asíncrono al servidor y acceder a la cadena de texto enviada en la respuesta. Para poderlo utilizar, deberá ser previamente instanciado desde la aplicación.
- ✓ Las anteriores tecnologías y componentes constituyen el núcleo fundamental de AJAX, sin embargo, estas aplicaciones se apoyan también para su funcionamiento en los siguientes estándares y tecnologías:
 - HTTP. Al igual que el propio navegador, el objeto XMLHttpRequest utiliza el protocolo HTTP para realizar las solicitudes al servidor, manejando también las respuestas recibidas mediante este protocolo.
 - Tecnologías de servidor. Una aplicación AJAX no tendría sentido sin la existencia de un programa en el servidor que atendiera las peticiones enviadas desde la aplicación y devolviera resultados al mismo.

Existen numerosas tecnologías que pueden emplearse para programar en el servidor, como J2EE, ASP.NET o PHP.¹⁷

¹⁷ Martin Sierra, Antonio J., 2009, Ajax en J2EE

2.4.Tecnología .NET

.NET es la plataforma de Microsoft para servicios Web XML, la siguiente generación de software que conecta nuestro mundo de información, dispositivos y personas de una manera unificada y personalizada.

La Plataforma .NET permite la creación y uso como servicios de aplicaciones, procesos y sitios Web basados en XML, que compartan y combinen información y funcionalidad entre ellos por diseño, en cualquier plataforma o dispositivo inteligente, para proveer soluciones a la medida para empresas e individuos.

La Plataforma .NET incluye una familia de productos integral, construida en estándares de la industria y de Internet, que provee servicios Web XML para cada aspecto del desarrollo (herramientas), administración (servers), uso (bloques ensamblables y clientes inteligentes) y experiencias (ricas experiencias de usuario). .NET será parte de las aplicaciones, herramientas, y servers de Microsoft que usa actualmente, así como será parte de nuevos productos que extienden las capacidades de los servicios Web XML a todas sus necesidades de negocio.

Este trabajo examina lo que significa la visión Microsoft® .NET para profesionales de tecnología y de negocios y para el futuro de las aplicaciones de software. Describe la visión .NET y explica las ventajas competitivas que brindará a los negocios.

2.4.1. Plataforma .net

La plataforma .NET es una capa de software que se coloca entre el Sistema Operativo (SO) y el programador y que abstrae los detalles internos del SO.

Microsoft define la plataforma .NET como un entorno para la construcción, desarrollo y ejecución de servicios web y otras aplicaciones que consiste en tres partes fundamentales: el Common Language Runtime (entorno de ejecución), las Framework Classes (clases de la plataforma) y ASP.NET.

El Common Language Runtime (CLR) es el entorno de ejecución de la plataforma .NET, y constituye su núcleo. El CLR es el entorno en el que se ejecutan nuestras aplicaciones .NET. Estas aplicaciones pueden escribirse en cualquiera de los múltiples lenguajes que ofrece .NET (Visual C#.Net, Visual Basic.NET...) que en lugar de compilarse a código máquina (que es lo más habitual) se compila a un lenguaje intermedio llamado Microsoft Intermediate Language o MSIL (Lenguaje Intermedio de Microsoft). El MSIL es el único lenguaje que el CLR comprende. Esta característica permite, por ejemplo, utilizar un fragmento de código en una aplicación sin depender del lenguaje en el que está escrito.

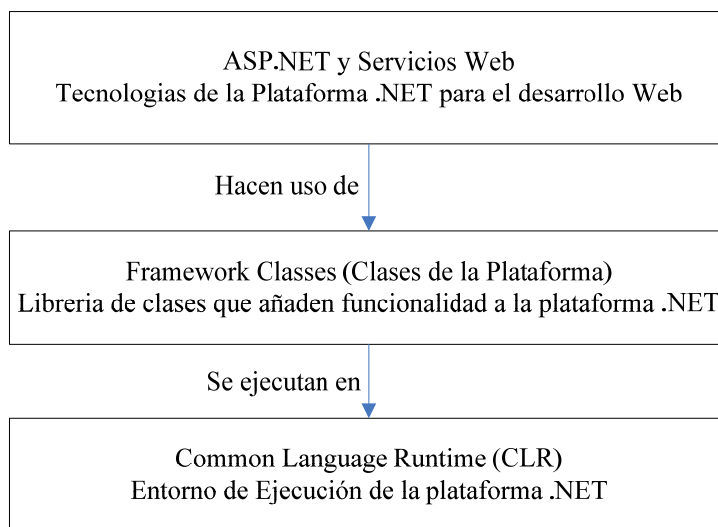


Figura 2.3. Las Capas de la Plataforma .NET

Fuente: Investigador

Las Framework Classes forman otra de las capas que constituyen la plataforma .NET. Esta capa provee al programador de servicios, estructuras y modelos de objetos para datos ADO.Net (siguiente generación de ADO), entrada/salida, seguridad, manejo de documentos XML... ASP.NET es la parte más importante de la capa superior de la plataforma .NET.

Para los programadores web ASP.NET es mucho más que una nueva versión de la tecnología ASP ya que supone una nueva idea y forma de programar aplicaciones Web. ASP.NET provee una plataforma más robusta para el desarrollo de aplicaciones, y ofrece mayores beneficios. A diferencia de ASP, los ASP.NET permiten separar limpiamente la lógica de la aplicación de la interfaz. De esta manera, el programador puede centrarse exclusivamente en la lógica de la aplicación sin preocuparse de los

detalles de la interfaz. ASP.NET además incorpora un nuevo concepto en el desarrollo de tecnologías Internet: los Servicios Web.

Estos servicios representan un paso más hacia la descentralización del software en la red y de hecho, son un factor clave para el desarrollo de una web orientada a objetos. Los servicios Web permiten a los desarrolladores construir aplicaciones combinando recursos locales y remotos para una solución distribuida e integrada. La comunicación a través de la web se hace utilizando el protocolo SOAP, lo cual no supone ningún problema para el desarrollador ya que es la plataforma .NET la que se encarga de tratarlo.

2.4.2. Visual Studio .NET

Visual Studio .NET es la herramienta que Microsoft distribuye junto a la plataforma que permite construir y desarrollar aplicaciones .NET. Esta nueva versión no revoluciona la anterior sino que se limita a añadir una serie de nuevas características y funciones. Es una mezcla de los diferentes entornos que Microsoft utilizaba hasta ahora (Visual Basic 6 IDE, Visual InterDev.).

La principal diferencia respecto a versiones anteriores es que Microsoft utiliza exactamente el mismo entorno para todos los lenguajes incluidos en la plataforma. De hecho, este entorno está creado para poder manejar proyectos que usen más de un lenguaje a la vez, teniendo en cuenta la característica multilenguaje de la plataforma.

El Visual Studio.NET incluye los siguientes lenguajes de programación:

- ✓ Visual Basic.NET es la adaptación de Visual Basic a la plataforma .NET

- ✓ Visual C++.NET permite tanto escribir código adaptado a la plataforma .NET (código gestionado o compilado a MSIL) como código C++ nativo (código no gestionado o no compilado a MSIL).
- ✓ Visual C#.NET (C Sharp) es un nuevo lenguaje de programación orientado a objetos con el que se ha desarrollado parte de la plataforma .NET.
- ✓ Visual J#.NET (J Sharp) es la adaptación de Visual J++ para la plataforma .NET.

2.4.3. Características de .NET

Las características fundamentales de esta plataforma son las siguientes:

- ✓ Portabilidad: Debido a la abstracción del programador respecto al SO, una aplicación .NET puede ser ejecutada en cualquier SO de cualquier máquina que disponga de una versión de la plataforma. .Net tan sólo está disponible para la familia Windows aunque se está desarrollando una versión para Linux de Corel.
- ✓ Multilenguaje: Cualquier lenguaje de programación puede adaptarse a la plataforma .NET y ejecutarse en ella.
- ✓ Interoperabilidad: La interoperabilidad entre diferentes trozos de código escritos en diferentes lenguajes es total.¹⁸
- ✓ Plataforma abierta: añadir nuevos lenguajes (Cobol, Perl) o nuevas herramientas.
- ✓ Permite el desarrollo de aplicaciones empresariales.
- ✓ Modelo de programación único para todo tipo de aplicaciones y dispositivos de Hardware.
- ✓ Se integra fácilmente con aplicaciones existentes.¹⁹

¹⁸ <http://people.cs.uchicago.edu/~borja/pubs/revistaeside2002.pdf>

2.4.4. Ventajas y Desventajas de .NET

Ventajas	Desventajas
Interoperabilidad Multilenguaje. .NET Soporta aplicaciones con componentes en múltiples lenguajes lo que permite integrar desarrolladores de distintos perfiles.	Mantenimiento en múltiples lenguajes. Mantener un proyecto en múltiples lenguajes es costoso. Si una aplicación está realizada en varios lenguajes se necesitan expertos en varios lenguajes para entenderla y mantenerla, aumentando los costos.
Documentación. .NET ofrece mucha documentación de ayuda (herramientas, depuradores, editores) incluida en la IDE y de soporte. Esto simplifica el desarrollo y la implementación.	.NET no es multiplataforma. La plataforma .NET sólo está disponible para la familia Windows.
Rendimiento. Todos los códigos que se ejecutan en el ambiente .NET son compilados, lo cual proporciona un gran rendimiento a diferencia de versiones interpretadas.	Licencias. Es un código cerrado, no hay licencias libres. La infraestructura para desarrollar en .NET representa un alto costo para las empresas.
Rápido aprendizaje por parte de los desarrolladores. Es sencillo de aprender por la documentación y el soporte de ayuda.	Portabilidad. Hay productos que ofrecen mucha más portabilidad que .NET, que sólo está preparada para ejecutarse sobre plataformas Microsoft (Windows).
Movilidad. Las aplicaciones pueden ser desplegadas en una amplia variedad de dispositivos.	
Escalabilidad y flexibilidad. Escalabilidad es la capacidad de un sistema para soportar más carga de	

¹⁹ <http://people.cs.uchicago.edu/~borja/pubs/revistaeside2002.pdf>

<p>trabajo, usualmente debida al aumento de usuarios que lo utilizan. .NET ofrece métodos de escalabilidad como la carga balanceada que permite a un cluster de servidores (varios servidores) colaborar y dar un servicio de forma simultánea.</p> <p>En cuanto a la flexibilidad: el modo de programación que se emplea permite agregar nuevos módulos sin modificar la aplicación en su totalidad.</p>	
<p>Seguridad. .NET da respaldo para ejecutar código no seguro.</p>	
<p>Estándar abierto. La plataforma .NET está basada en estándares (HTML, XML, SOAP, WSDL, UDDI).²⁰</p>	

Tabla 1.4. Cuadro de Ventajas y Desventajas de .NET

Fuente: Investigador

2.4.5. Framework.NET

La plataforma .Net es la propuesta de Microsoft para el desarrollo de aplicaciones completamente orientadas a objetos, seguras, sencillas de instalar y multiplataforma; no es un lenguaje de programación: es un conjunto de tecnologías de software que permite el desarrollo de aplicaciones de escritorio (Windows), Web (ASP.Net), bibliotecas de código (componentes reutilizables: DLL), dispositivos móviles, controles de usuario, acceso a datos, reportes y mucho más, que se ejecutan bajo el .Net Framework. Es

²⁰ http://www.qualitrain.com.mx/index2.php?option=com_content&do_pdf=1&id=149

completamente extensible: podemos heredar de sus clases base y modificar su comportamiento o implementar sus interfaces para desarrollar clases completamente nuevas.

El entorno de programación de la plataforma .Net, Visual Studio, es uno de los más potentes en cuanto a facilidad de uso y proporciona una gran cantidad de herramientas para mejorar la productividad del programador. Es independiente del lenguaje de programación, por lo que podemos realizar componentes en un lenguaje y utilizarlo en otro, sin complicaciones ni pérdida de rendimiento, debido a su sistema de tipos comunes y a su arquitectura de lenguaje intermedio (MIL).²¹

2.4.6. Arquitectura de la Solución

.NET Framework es el producto de sistema operativo de Microsoft que proporciona soluciones prefabricadas a esos problemas de programación, La clave de la estructura es el código administrado. El código administrado se ejecuta en el entorno denominado Common Language Runtime (CLR), que proporciona un conjunto de servicios más rico que el sistema operativo estándar Win 32, como se muestra en la Figura 2.4 .El entorno CLR es la potencia superior que tenemos que aplicar al código para que se enfrente al salvaje mundo en que se ha convertido la programación moderna de Internet.²²

²¹ <http://img.redusers.com/imagenes/libros/lpcu113/capitulogratis.pdf>

²² Sánchez, 2009, Así es Microsoft .NET

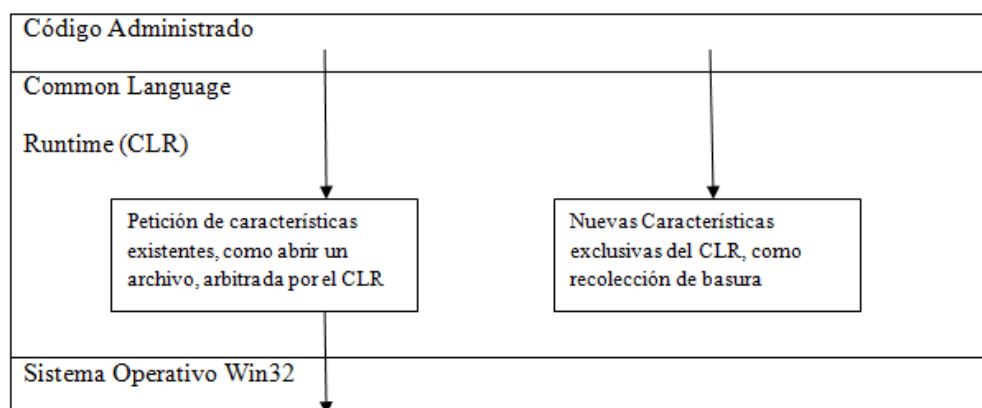


Figura 2.4. Ejecución Administrada en el Common Language Runtime

Fuente: Investigador

2.5. Introducción a Java

Java surgió en 1991 cuando un grupo de ingenieros de Sun Microsystems trataron de diseñar un nuevo lenguaje de programación destinado a electrodomésticos. La reducida potencia de cálculo y memoria de los electrodomésticos llevó a desarrollar un lenguaje sencillo capaz de generar código de tamaño muy reducido.

Debido a la existencia de distintos tipos de CPUs y a los continuos cambios, era importante conseguir una herramienta independiente del tipo de CPU utilizada. Desarrollaron un código “neutro” que no dependía del tipo de electrodoméstico, el cual se ejecutaba sobre una “máquina hipotética o virtual” denominada Java Virtual Machine (JVM). Era la JVM quien interpretaba el código neutro convirtiéndolo a código particular de la CPU utilizada. Esto permitía lo que luego se ha convertido en el principal lema del lenguaje: “Write Once, Run Everywhere”. A pesar de los esfuerzos realizados por sus creadores, ninguna empresa de electrodomésticos se interesó por el nuevo lenguaje.

Como lenguaje de programación para computadores, Java se introdujo a finales de 1995. El principal objetivo del lenguaje Java es llegar a ser el “nexo universal” que conecte a los usuarios con la información, esté ésta situada en el ordenador local, en un servidor de Web, en una base de datos o en cualquier otro lugar.

La compañía Sun describe el lenguaje Java como “simple, orientado a objetos, distribuido interpretado, robusto, seguro, de arquitectura neutra, portable, de altas prestaciones, multitarea y dinámico”.

2.5.1. Plataforma Java

Una plataforma es el entorno de hardware o software en el cual se ejecuta un programa. A diferencia de las plataformas tradicionales como Linux, Windows y Solaris, la plataforma Java está basada sólo en software que se ejecuta sobre otras plataformas basadas en hardware. Por esto la plataforma Java y su software puede ser utilizado sobre variados sistemas operativos y hardware.

La plataforma Java está constituida de tres componentes: el lenguaje, la máquina virtual y las bibliotecas.

2.5.1.1. Lenguaje Java

El lenguaje Java es un lenguaje de propósito general, de alto nivel, que utiliza el paradigma de orientación a objetos. Su sintaxis y tipos están basados principalmente en C++, sin embargo, las diferencias principales con éste son la administración de memoria, siendo ejecutada por la máquina virtual automáticamente y no por el código de cada programa, y el soporte de procesos livianos o threads a nivel del lenguaje, que ayuda a controlar la sincronización de procesos paralelos. Estas características dan al lenguaje Java las propiedades de robustez y seguridad, evitando por ejemplo problemas de buffer overflow utilizados en ataques a sistemas.

2.5.1.2. Máquina virtual

Los programas escritos en Java son compilados como archivos ejecutables de una máquina virtual llamada Java Virtual Machine (JVM). Existen implementaciones de esta máquina para múltiples plataformas, permitiendo ejecutar en diferentes arquitecturas el

mismo programa ya compilado. La característica de independencia de la plataforma hace posible el libre intercambio de software desarrollado en Java sin necesidad de modificaciones, lo que ha sido llamado “Write once, Run anywhere”.

Java es un lenguaje compilado e interpretado a la vez. Compilado ya que previo a su ejecución un programa debe ser transformado a un lenguaje intermedio, llamado Java bytecodes. Interpretado porque cada programa luego debe ser procesado y ejecutado por alguna implementación de la JVM específica a la plataforma.

2.5.1.3. Bibliotecas

El conjunto de bibliotecas del lenguaje es conocido como la Java Application Programming Interface (Java API) que es un gran conjunto de componentes que proporcionan diferentes herramientas para el desarrollo de programas Java. La API de Java está agrupada en conjuntos de bibliotecas relacionadas conocidas como paquetes, que contienen grupos de elementos básicos de Java, llamados clases e interfaces.

2.5.2. Ediciones de Java

La plataforma Java ha sido dividida en tres ediciones distintas según sus diferentes objetivos:

- ✓ **Java 2 Platform, Standard Edition (J2SE):** Conjunto de herramientas para desarrollar aplicaciones Java que incluye APIS, JDK, JRE, JVM, ejemplos de códigos fuente. Maneja el núcleo de toda la funcionalidad del lenguaje Java. Satisface requerimientos específicos para aplicaciones particulares como applets,

incluyendo paquetes opcionales para extensión de seguridad en sockets usado en comercio electrónico.

- ✓ **Java 2 Platform, Enterprise Edition (J2EE):** Proporciona un conjunto de herramientas y librerías necesarias para extender a java en tecnologías de server side. J2EE es un super conjunto de J2SE. Usado para el desarrollo a nivel empresarial y en entornos de aplicaciones servidor, incorporando nuevas tecnologías tales como los servlets, Enterprise JavaBeans (EJBs), XML y Java Server Pages.

- ✓ **Java 2 Platform, Micro Edition (J2ME):** Es un ambiente de desarrollo y ejecución diseñado para poner software hecho en Java en consumibles electrónicos y dispositivos embebidos. Subconjunto de J2SE usado para programación de aplicaciones y necesidades combinadas tales como:
 - Consumidores y fabricantes de dispositivos móviles con recursos limitados.
 - Proveedores de servicio que desarrollan contenido personal sobre estos dispositivos.
 - Creadores de contenido que ajustan el contenido para dispositivos con recursos limitados.

Estas ediciones de la Plataforma Java estandarizan un conjunto de tecnologías que pueden ser usadas con un producto particular:

- Máquinas virtuales Java que se ajustan dentro de un amplio rango de dispositivos.
- Librerías y APIs especializadas para cada tipo de dispositivo, y
- Herramientas para el desarrollo y configuración de dispositivos.

2.5.2.1. Tecnología Java 2 Platform, Standard Edition (J2SE)

Java 2, Standard Edition (J2SE) es la edición principal de la plataforma Java sobre la cual se basan las demás ediciones. Provee las capacidades de desarrollo y ejecución de software escrito en lenguaje Java. Esta constituido de dos módulos principales (ver Figura 2.5):

- ✓ Software Development Kit (J2SE SDK), conocido inicialmente como Java Development Kit (JDK), proporciona el software necesario para desarrollar programas en Java como es el compilador, el debugger y las bibliotecas con las funcionalidades del lenguaje.
- ✓ Java Runtime Environment (JRE), contiene sólo el ambiente necesario y las bibliotecas principales para ejecutar software escrito en Java.

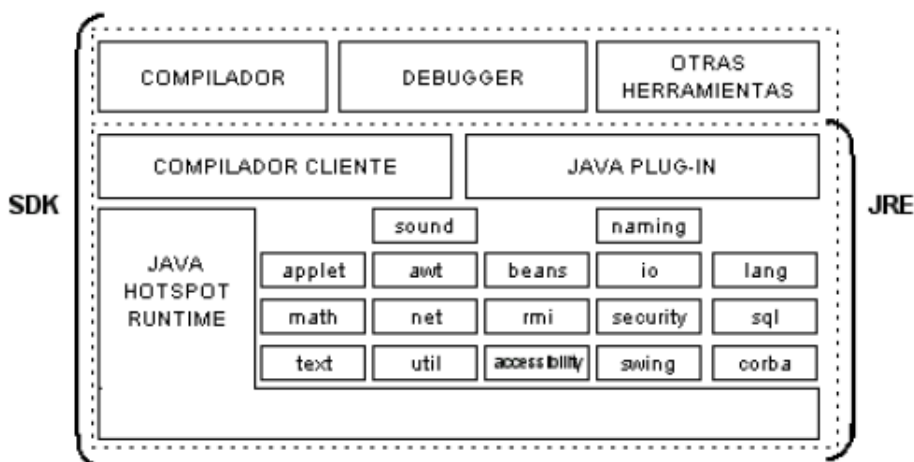


Figura 2.5. Java 2, Standard Edition

Fuente: <<http://www.dcc.uchile.cl/~jbarrios/J2EE/MemoriaJ2EE.pdf>>

J2SE incluye herramientas y APIs para desarrollar aplicaciones con interfaz gráfica, acceso a base de datos, acceso a directorios, seguridad, entrada/salida, programación en red y varias otras funcionalidades.

Las bibliotecas principales son clases que se encuentran dentro de los paquetes `java.*` y las bibliotecas con extensiones estándares se encuentran como clases dentro de los paquetes `javax.*`. Para poder nombrar bibliotecas de aplicaciones desarrollados por terceros, se utiliza la convención de invertir el nombre de dominio de Internet del desarrollador y luego separar por funcionalidades, así existe por ejemplo `org.w3c.*` y `org.apache.*` entre otros.

2.6. Tecnología Java 2 Platform, Enterprise Edition (J2EE)

Java 2, Enterprise Edition (J2EE) es una especificación que define una plataforma para crear aplicaciones empresariales utilizando un modelo de multicapas, dividiendo la aplicación en diferentes niveles, cada uno especializándose en una tarea en particular.

La figura 2.6 ilustra la composición de J2EE. Su estructura está basada en J2SE y un conjunto de sus APIs, a la cual J2EE aporta la especificación de componentes, containers y las APIs para los servicios de transacciones, mensajería, envío de correos y conectores de recursos externos.

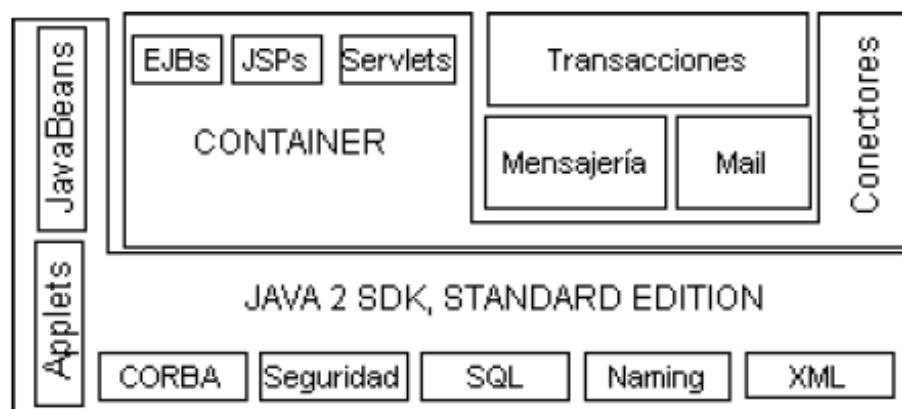


Figura 2.6. Estructura J2EE

Fuente: < <http://www.dcc.uchile.cl/~jbarrios/J2EE/MemoriaJ2EE.pdf> >

La especificación de J2EE se encuentra en la versión 1.3, publicada en Agosto de 2001, y está compuesta de un conjunto de paquetes que, si bien algunos son opcionales para J2SE, son requeridos para la plataforma J2EE.

2.6.1. ¿Por qué J2EE?

Con la aparición de las aplicaciones web que dependían cada vez más de tecnologías de servidor como es el middleware, los departamentos de informática de las empresas necesitaban alguna forma sustentable de desarrollar aplicaciones y el middleware relacionado que fueran portables y escalables.

Era necesario diseñar estas aplicaciones de forma que pudieran atender a miles de usuarios en forma simultánea, 24 horas al día, siete días a la semana, sin ningún tiempo de inactividad. Uno de los desafíos para construir una aplicación compleja de este tipo es poder diseñar y probarla. J2EE simplifica la creación de las aplicaciones empresariales, ya que la funcionalidad se encapsula en los componentes de J2EE. Esto permite a los diseñadores y programadores dividir la aplicación de acuerdo con sus funciones, las cuales se distribuyen entre los componentes del servidor que se construyen con J2EE.

J2EE es una tecnología versátil, ya que los componentes de las aplicaciones que se construyen con J2EE se comunican entre sí detrás de las cámaras mediante métodos estándar de comunicación como son HTTP, SSL, XML, RMI e IIOP.

Todos los programas J2EE están escritos en Java, lo cual permite a las empresas utilizar los programadores Java con los que ya cuenta para crear programas que funcionan en cada una de las capas de la infraestructura multicapa. Ya no es necesario que las empresas busquen programadores para escribir programas que se conecten con componentes específicos de cada proveedor. Esto además acorta el ciclo de desarrollo de programas complejos que requieren de hilos y de sincronización, ya que J2EE incluye todas las interfaces y bibliotecas necesarias para gestionar estas cuestiones complejas.

Los Enterprise JavaBeans, los servlets y las JavaServer Pages son los componentes clave de J2EE. Además de ello, J2EE incluye siete servicios más.

2.6.2. Arquitectura J2EE

La especificación de J2EE define su arquitectura basándose en los conceptos de capas, containers, componentes, servicios y las características de cada uno de éstos. Las aplicaciones J2EE son divididas en cuatro capas: la capa cliente, la capa web, la capa negocio y la capa datos. La Figura 2.7 representa estas capas y las componentes relacionadas.

Capa Cliente

Esta capa corresponde a lo que se encuentra en el computador del cliente. Es la interfaz gráfica del sistema y se encarga de interactuar con el usuario. J2EE tiene soporte para diferentes tipos de clientes incluyendo clientes HTML, applets Java y aplicaciones Java.

Capa Web

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

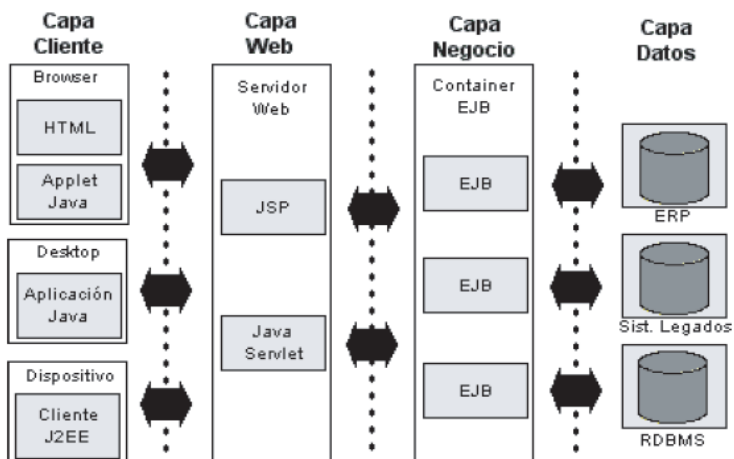


Figura 2.7. Arquitectura J2EE

Fuente: < <http://www.dcc.uchile.cl/~jbarrios/J2EE/MemoriaJ2EE.pdf> >

Capa Negocio

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

Capa Datos

Esta capa es responsable del sistema de información de la empresa o Enterprise Information System (EIS) que incluye bases de datos, sistema de procesamiento datos,

sistemas legados y sistemas de planificación de recursos. Esta capa es el punto donde las aplicaciones J2EE se integran con otros sistemas no J2EE o con sistemas legados.²³

2.6.3. Características de J2EE

- ✓ Multiplataforma, a través de la Máquina Virtual Java las aplicaciones J2EE pueden ser ejecutadas bajo cualquier entorno como Windows, Linux, Unix, Mac, etc.
- ✓ Tipo de tecnología Estándar.
- ✓ J2EE permite construir aplicaciones empresariales de alta disponibilidad.
- ✓ Acceso a base de datos JDBC SQL/J
- ✓ Lenguaje de programación Java.
- ✓ La plataforma J2EE es la mejor opción para proyectos de gran tamaño, que requieren robustez y escalabilidad.
- ✓ Arquitectura multi-capa

²³ <http://www.dcc.uchile.cl/~jbarrios/J2EE/MemoriaJ2EE.pdf>

2.6.4. Ventajas y desventajas de J2EE

Ventajas	Desventajas
<p>Soporte de múltiples sistemas operativos. Al ser una plataforma basada en el lenguaje Java, es posible desarrollar arquitecturas basadas en J2EE utilizando cualquier sistema operativo donde se pueda ejecutar una máquina virtual Java.</p>	<p>Depende de un único lenguaje. La plataforma J2EE depende exclusivamente del lenguaje Java. Sólo se puede utilizar este lenguaje para desarrollar aplicaciones lo que puede suponer un gran problema si nuestro equipo no dispone de los conocimientos suficientes o tiene otras preferencias.</p>
<p>Organismo de control. La plataforma J2EE está controlada por el JCP, un organismo formado por más de 500 empresas. Entre las empresas que lo forman están todas las más importantes del mundo informático (SUN, IBM, Oracle, SAP, HP, AOL, etc.) lo que garantiza la evolución de la misma.</p>	<p>Complejidad. Aunque no es una plataforma tan compleja como CORBA, no existe un VB .NET en Java. La creación de aplicaciones bajo J2EE requiere normalmente desarrolladores más experimentados que los necesarios para desarrollar bajo .NET</p>
<p>Competitividad. Muchas empresas crean soluciones basadas en J2EE y que ofrecen características como rendimiento, precio, etc., muy diferentes. De este modo el cliente tiene una gran cantidad de opciones a elegir.</p>	<p>Heterogeneidad. Existe una gran heterogeneidad en las soluciones de desarrollo. No existe en J2EE un símil a Visual Studio .NET. La gran cantidad de herramientas disponibles causa confusión dentro de los desarrolladores y puede crear dependencias dentro de las empresas.</p>
<p>Madurez. Creada en el año 1997, J2EE tiene ya varios años de vida y una gran cantidad de proyectos importantes a sus espaldas.</p>	<p>Incremento del tiempo de desarrollo</p>

Soluciones libres: En la plataforma J2EE es posible crear arquitecturas completas basadas única y exclusivamente en productos de software libre. Normalmente disponen de varias soluciones libres para cada una de las partes de su arquitectura. ²⁴	
Capas. Al tener las capas separadas existe poco acoplamiento entre las mismas, de modo que es mucho más fácil hacer modificaciones en ellas sin que interfiera en las demás.	
Estandarización de la solución y consecuente posibilidad de reutilización.	

Tabla 1.5. Cuadro de Ventajas y Desventajas de J2EE

Fuente: Investigador

2.6.5. Componentes de J2EE

Servlets

Es una componente web de J2EE desarrollada con el objetivo de procesar requerimientos de un cliente o requests y generar respuestas con contenidos web dinámicos. Para ser ejecutados es necesaria la utilización de un servidor que de soporte a servlets y su container.

²⁴ http://www.uhu.es/josel_alvarez/NvasTecnProg/recursos/articulo-j2ee.pdf

Por ejemplo, un servlet puede procesar los datos desde un formulario en HTML, mantener un registro de la transacción, actualizar una base de datos, contactar algún sistema remoto y retornar un documento dinámico o redirigir a otro servlet.

JavaServer Pages

JavaServer Pages (JSP) es una componente de J2EE para construir fácilmente aplicaciones con contenido web como HTML, DHTML, XHTML y XML, en forma dinámica, con gran poder y flexibilidad. JSP se basa en los siguientes conceptos:

- ✓ Plantillas o templates. Una parte importante del contenido está compuesto por una plantilla. Típicamente en esta plantilla se encuentran fragmentos HTML o de texto.
- ✓ Contenido dinámico. JSP provee una forma simple de agregar datos dinámicos a cada plantilla al permitir incrustar instrucciones de programación en éste. El lenguaje es generalmente Java, aunque se puede utilizar otro lenguaje que sea soportado por el container JSP.
- ✓ Encapsulación de funcionalidad. JSP provee dos formas distintas para encapsular funcionalidad: componentes JavaBeans y bibliotecas de etiquetas o taglibs.

JavaServer Pages fue creado con el objetivo de proveer un método declarativo y centrado en la presentación para crear servlets. Además de los beneficios que proveen los servlets, JSP ofrece la posibilidad de desarrollar rápidamente servlets donde los contenidos y la presentación se encuentran separados, y reutilizar código con el uso de una arquitectura basada en componentes.

Una página JSP es un documento de texto que describe como procesar un request para crear una respuesta, mezclando datos estáticos de plantillas con acciones dinámicas. JSP ofrece además los siguientes beneficios principales:

- ✓ Separación de roles. JSP permite la separación de los roles de diseñador y desarrollador. El diseñador crea presentaciones para cada página ubicando contenidos estáticos y dinámicos pensando en los usuarios. El desarrollador escribe componentes que interactúen con los objetos del servidor para proveer la información y lógica del negocio.
- ✓ Reutilización de componentes. JSP permite la utilización de componentes reutilizables como JavaBeans, EJB y taglibs, lo cual se traduce en mayor eficiencia en el desarrollo y menor probabilidad de inconsistencias en una aplicación.

Enterprise JavaBeans

Enterprise JavaBeans (EJB) es una arquitectura que permite la creación de componentes de aplicaciones distribuidas y orientadas a transacciones. Las aplicaciones escritas utilizando EJB son escalables, transaccionales y multiusuarios.

Las características esenciales de EJB son:

- ✓ Contiene la lógica del negocio que opera con el Enterprise Information System (EIS).
- ✓ Las instancias son creadas y manejadas por el container EJB.
- ✓ Puede ser configurado editando sus parámetros de entorno vía archivos XML.
- ✓ Las características de seguridad y transacciones se encuentran separadas de las clases EJB, lo que permite la operación de aplicaciones externas y middlewares.

2.6.6. Servicios J2EE

J2EE especifica los siguientes servicios estándares, junto con las APIs necesarias para la utilización por parte de cada componente. Algunos de estos servicios actualmente son provistos por J2SE.

1. HTTP y HTTPS: Protocolos estándares utilizados para comunicaciones web y para comunicaciones seguras sobre Secure Socket Layer (SSL), respectivamente. La API para clientes está definida por el paquete `java.net.*` y la API para servidor está definida por las clases de `servlets` y `JSP`.
2. JDBC: Una API estándar para acceder a los recursos de una base de datos relacional de una forma independiente del proveedor. Esta API consta de dos partes, una interfaz para ser utilizada por las componentes y una interfaz de proveedores para definir drivers específicos. Oficialmente JDBC no es un acrónimo, aunque comúnmente se utiliza el nombre de `Java Database Connectivity`.
3. `JavaMail`: Una API que permite crear aplicaciones Java para mensajería y envío de correo electrónico en forma independiente de la plataforma y del protocolo a utilizar.
4. `JavaBeans Activation Framework (JAF)`: API que proporciona un framework de activación que es utilizado por otros paquetes, como `JavaMail`. Los desarrolladores pueden utilizar JAF para determinar el tipo de un trozo arbitrario de datos, accederlo, descubrir las operaciones disponibles en él e instanciar el bean apropiado para ejecutar esas operaciones. Por ejemplo, `JavaMail` usa JAF para determinar qué tipo de objeto instanciar dependiendo del `Mime-Type` del objeto.

5. Remote Method Invocation-Internet Inter-ORB Protocol (RMI-IIOP): Está compuesto de APIs que permiten la programación distribuida a través de Java RMI. Los protocolos utilizados para la comunicación pueden ser JRMP (protocolo nativo de RMI), o IIOP (protocolo de CORBA).
6. Java Interface Definition Language (JavaIDL): Permite a las aplicaciones actuar como clientes de servicios CORBA, invocando objetos externos utilizando el protocolo IIOP.
7. Java Transaction API (JTA): Permite el manejo de transacciones. Las aplicaciones pueden utilizar JTA para iniciar, cerrar o abortar transacciones. Además permite al container comunicarse con monitores transaccionales y administradores de recursos.
8. Java Message Service (JMS): Es una API que se utiliza para comunicarse con un Message-Oriented Middleware (MOM) en una forma independiente al proveedor para permitir mensajería del tipo punto a punto y del tipo publicar/subscribe entre sistemas.
9. Java Naming and Directory Interface (JNDI): Una API estándar para el registro y acceso de servicios y objetos. Incluye soporte para LDAP (Lightweight Directory Access Protocol), COS (CORBA Object Services), Naming Service y Java RMI Registry.
10. Java API for XML Parsing (JAXP): Permite a las aplicaciones la utilización de documentos XML a través de las APIs estándares SAX, DOM y XSLT.
11. J2EE Connector Architecture (JCA): Una API de J2EE que permite agregar recursos nuevos a cualquier producto J2EE. La arquitectura Connector define un contrato entre un servidor J2EE y un adaptador de recursos para permitir esto.

12. Java Authentication and Authorization Service (JAAS): Proporciona una forma para la identificación de usuarios y su autorización para acceder a recursos de la aplicación. Implementa una versión en Java del estándar Pluggable Authentication Module (PAM).²⁵

²⁵ <http://www.dcc.uchile.cl/~jbarrios/J2EE/MemoriaJ2EE.pdf>

2.7. Tecnología Java 2 Platform, Micro Edition (J2ME)

La edición J2ME lanzada en 1999 por la Sun Microsystems tiene una aparición tardía debido a que las necesidades de los usuarios de dispositivos móviles cambio de manera drástica en los últimos años. J2ME se orienta al desarrollo de aplicaciones para pequeños dispositivos con limitaciones gráficas, de procesamiento y memoria (teléfonos móviles, PDAs, Beepers, etc.). J2ME consta de un conjunto de especificaciones para definir una colección de plataformas, apropiadas para un subconjunto del amplio rango de dispositivos para usuarios existentes en el mercado.

2.7.1. Características de J2ME

- ✓ Simplicidad, resulta sencillo de usar para los consumidores, permite el fácil desarrollo para programadores y es accesible para proveedores de servicios que son los que brindan las soluciones de software.
- ✓ Portabilidad, ofrece un marco de desarrollo de servicios en todo momento y en todo lugar. Es decir, se desarrollan aplicaciones que pueden correr en distintos dispositivos con características similares.
- ✓ Accesibilidad, provee de bases para desarrollar aplicaciones inteligentes y dinámicas en red. Lo que significa que se pueden descargar programas desde la red.
- ✓ Seguridad, las aplicaciones no tienen acceso al sistema operativo.
- ✓ Se desarrolla J2ME como un ambiente altamente optimizado que permite la ejecución de programas Java en esta rama de pequeños dispositivos.²⁶

²⁶ http://catarina.udlap.mx/u_dl_a/tales/documentos/lis/rivera_d_g/capitulo3.pdf

2.7.2. Arquitectura de J2ME

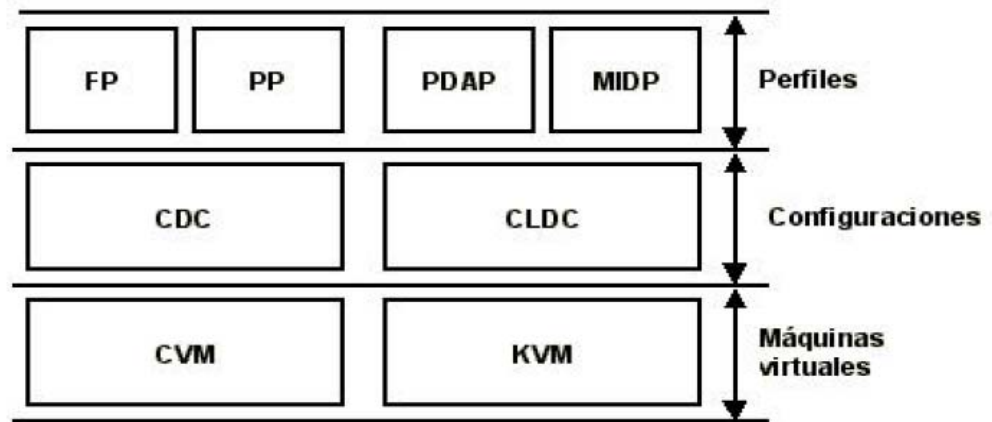


Figura 2.8. Arquitectura de J2ME

Fuente: < <http://www.slideshare.net/arpal986/aplicaciones-moviles-empresariales-con-servicios-web-y-j2me> >

La arquitectura J2ME está proyectada para ser modular y escalable, además que puede soportar los tipos de desarrollo demandados por el consumidor e incorporados en los mercados. El entorno de J2ME proporciona un conjunto de Maquinas Virtuales Java, optimizadas para diferentes tipos de procesamiento.

Existen cuatro conceptos esenciales, que conforman el corazón de la arquitectura de J2ME:

1. Máquina Virtual.
2. Configuración.
3. Perfil.
4. Paquetes Opcionales.

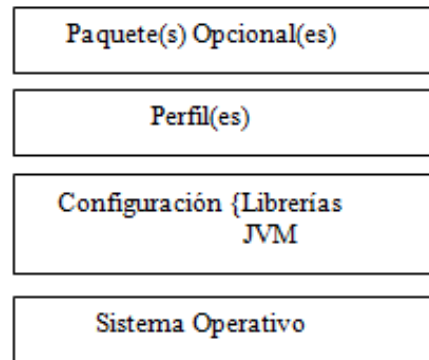


Figura 2.9. Estructura de la Arquitectura de J2ME

Fuente: Investigador

2.7.2.1. Máquina Virtual

La Máquina virtual Java tiene como función principal interpretar código intermedio (bytecode) procedentes de la pre-compilación a código ejecutable por la plataforma, permitiendo así efectuar llamadas al sistema operativo subyacente para que este evalúe el código y establezca reglas de seguridad para que el lenguaje Java se ejecute. Esto proporciona al programa independencia a la plataforma con respecto al hardware y el sistema operativo del dispositivo, lo cual la convierte en interoperable.

A diferencia de las ediciones de Java J2SE y J2EE, J2ME utiliza una máquina virtual con capacidades reducidas debido a las características intrínsecas de los teléfonos celulares con el fin de realizar una implementación menos exigente, suprimiendo el soporte a algunas características del lenguaje Java pertenecientes a J2SE y J2EE. De acuerdo al tipo de configuración se utiliza una máquina virtual distinta. La Máquina

Virtual de la configuración CLDC se denomina KVM y la de la configuración CDC CVM.

KVM (Kilobyte Virtual Machine): Es la máquina virtual más pequeña desarrollada por la Sun, compacta y portable diseñada específicamente para un grupo de dispositivos pequeños con recursos restringidos. Su nombre hace referencia a la baja ocupación de memoria que utiliza. KVM es apropiado para microprocesadores de 16/32 bits RISC/CISC y con un total de memoria asignada de no más de unos cientos de kilobytes (algunas veces menos de 128 kilobytes). KVM fue diseñada para ser:

- ✓ Pequeña, con una carga de memoria entre los 40Kb y los 80 Kb, dependiendo de la plataforma y las opciones de compilación.
- ✓ Alta portabilidad.
- ✓ Modulable.
- ✓ Lo más completa y rápida posible y sin sacrificar características para las que fue diseñada.

El hecho de no contar con todas las capacidades de memoria, posee algunas limitaciones con respecto a la clásica Máquina Virtual Java (JVM):

1. No hay soporte para la JNI (Interfaz Nativa de Java) debido a los recursos limitados de memoria.
2. No existen cargadores de clases (class loaders) definidos por el usuario. Existen únicamente los predefinidos.
3. No se realiza la implementación del método `Object.finalize()`, el cual permite la finalización de instancias de clases.

4. Es necesario el uso de objetos de tipo Colección que permitan agrupar varios hilos y almacenar cada uno en el ámbito de la aplicación.
5. La capacidad en el manejo de excepciones es limitada, ya que este tipo de eventos son manejados de forma independiente por las APIs propias del dispositivo.
6. No se manejan las referencias débiles, no se apunta a objetos que puedan ser adquiridos por el recolector de basura.

Debido a que el verificador de clases de J2SE es muy pesado para la verificación de clases en J2ME.

La implementación del verificador de clases en KVM requiere de al menos 10 kB de código binario y menos de 100 bytes de RAM dinámica en tiempo de ejecución para clases típicas. El verificador realiza solamente un escaneo lineal del código intermedio, sin la necesidad de un algoritmo de flujo de datos iterativo costoso. Así el nuevo verificador de clases opera en dos fases como se ilustra en la Figura 2.10. En la fase número 1, las clases tienen que ser ejecutadas a través de una herramienta de pre-verificación para poder remover ciertos códigos intermedios y aumentar las clases con atributos adicionales StackMap aumentando la velocidad de verificación en tiempo de ejecución. La fase de preverificación es realizada típicamente en una estación de trabajo que el desarrollador usa para escribir y compilar aplicaciones.

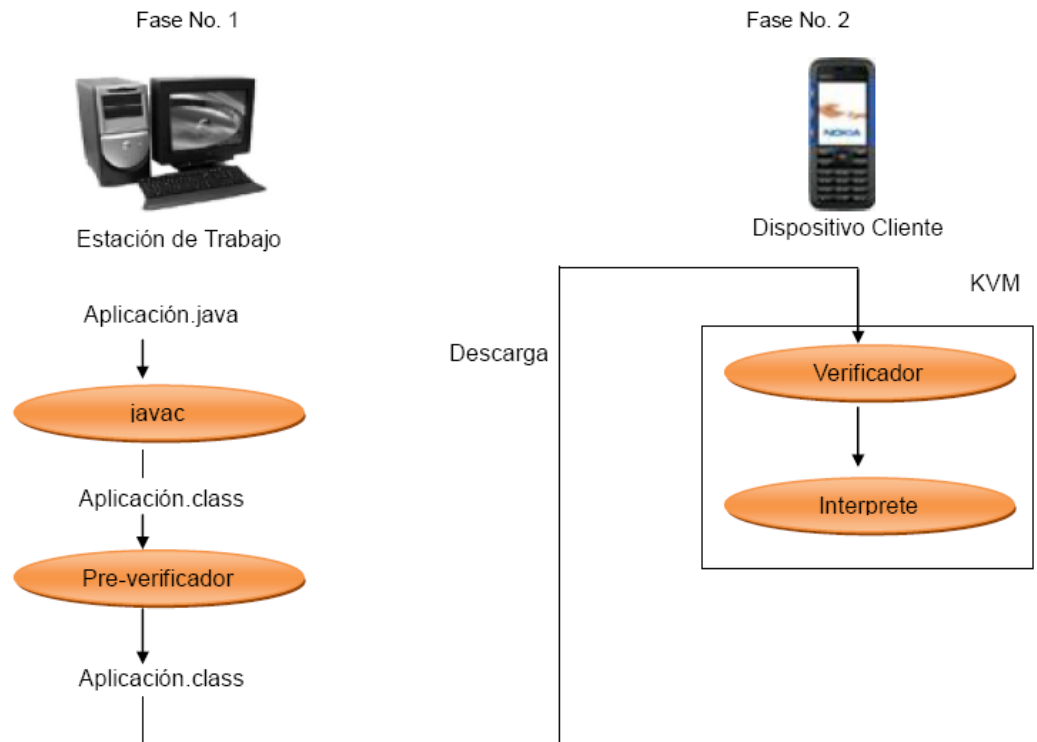


Figura 2.10. Pre-verificación de clases en J2ME

Fuente: < <http://www.slideshare.net/arpal986/aplicaciones-moviles-empresariales-con-servicios-web-y-j2me> >

En la fase número 2, el componente verificador en tiempo de ejecución de la máquina virtual usa los atributos adicionales `StackMap` generados por el pre-verificador para realizar la preverificación de la clase actual de forma eficiente. A través de estas fases se realizan las siguientes comprobaciones:

- ✓ Observar que el código no sobrepase los límites de la pila de la Máquina Virtual.
- ✓ Comprobar que no se utilizan variables locales antes de ser inicializadas.
- ✓ Comprobar que se respetan los campos, métodos y los modificadores de control de acceso a clases.

CVM (Compact Virtual Machine): Máquina Virtual para la configuración CDC. Orientado a dispositivos electrónicos con procesadores de 32 bits de gama alta y alrededor de 2 Mb ó más de memoria RAM, con conexiones de red de forma intermitente o permanente a través de TCP/IP. Sus principales características son:

1. Sistema de memoria avanzado
2. Tiempo de espera bajo para el recolector de basura
3. Separación de la funcionalidad de la Máquina Virtual del sistema de memoria.
4. Consta de un Recolector de Basura modularizado.
5. Portabilidad.
6. Sincronización rápida.
7. Ejecución de clases a través de la implementación de la memoria de solo lectura ROM.
8. Soporte nativo de hilos.
9. Las clases tienen baja ocupación de la memoria.
10. Tiene soporte para interfaces y servicios en Sistemas Operativos de Tiempo Real.
11. Conversión de hilos Java a hilos nativos.
12. Soporte para JNI, referencias débiles, RMI (Invocación de Métodos Remotos), Interfaz de Depuración de la Máquina Virtual (JVMDI).
13. Serialización de Objetos.
14. Cargadores de clases definidos por el usuario.

2.7.2.2. Configuraciones

En el entorno J2ME las configuraciones se refieren al conjunto de APIs que permiten el desarrollo de aplicaciones para un grupo de dispositivos. La configuración maneja características básicas, comunes a un grupo de dispositivos o mercado objetivo en cuanto a asignación de memoria y otras capacidades de hardware se refiere. El perfil hace uso o “hereda” una configuración particular. Las configuraciones no permiten la definición de características adicionales ya que este tipo de características son manejadas por los perfiles.

Entre las características soportadas por las configuraciones, se especifica:

- ✓ Soporte para características del lenguaje de programación Java.
- ✓ Soporte para características de la Máquina Virtual.
- ✓ Soporte para las librerías Java y APIs.

Las configuraciones en el momento de definir el entorno software para un conjunto de dispositivos relacionados entre sí por un conjunto de características, tiene en cuenta cosas como:

- ✓ El tipo y cantidad de memoria disponible para ejecución de aplicaciones.
- ✓ El tipo de procesador y velocidad.
- ✓ La forma en la que se conecta el dispositivo a la red.

La implementación de una maquina virtual en J2ME deberá adherirse a las especificaciones definidas por la configuración.

Las configuraciones definen el contrato entre un perfil y la Maquina Virtual Java.

Configuración	Máquina Virtual	Ejemplos: dispositivos
CDC	CVM	Pocket PCs Communicator class decices TV set-top boxes
CLDC	KVM	Cellular phones PDAs Two-way pagers

Tabla 2.6. Configuración, Máquina Virtual y algunos ejemplos

Fuente: Investigador

Actualmente están disponibles dos configuraciones en J2ME:

- ✓ CLDC (Connected Limited Device Configuration)
- ✓ CDC (Connected Device Configuration)

CLDC utiliza la Máquina Virtual Kilobyte (KVM) mientras que CDC utiliza la Máquina Virtual C (CVM)

Configuración de dispositivos con Conexión (CDC): Esta configuración está diseñada para dispositivos con ciertas capacidades computacionales y de memoria soportando un entorno más completo para la Máquina Virtual Java con características similares a las de J2SE, con limitaciones en el ambiente gráfico y memoria del dispositivo. La Máquina Virtual adaptada a esta configuración es la CVM. Esta configuración se enfoca en dispositivos con las siguientes características:

- ✓ Memoria volátil de al menos 2 Mb ó más, incluyendo RAM Y ROM.
- ✓ Procesador de 32 bits.

- ✓ Conexión a algún tipo de red.
- ✓ Manejo de la funcionalidad completa de la Máquina Virtual de Java 2.

Estos rasgos pueden ser encontrados en dispositivos como decodificadores de televisión digital, televisores con internet y sistemas de navegación de automóviles. La configuración cuenta con la siguiente lista de clases de la Edición Estándar de la Plataforma Java 2, J2SE. Incluye soporte para protocolos como HTTP y el manejo de datagramas. La Tabla 1.7. muestra las librerías incorporadas en CDC.

Configuración de Dispositivos Limitados con Conexión (CLDC): Esta configuración es apropiada para dispositivos provistos con conexión pero con limitaciones en cuanto a capacidad gráfica, de cómputo y memoria. Es el bloque básico en donde se soportan los perfiles de la gran mayoría de dispositivos que se apoyan en J2ME como plataforma de desarrollo.

La gran mayoría de estas restricciones está dada debido al uso que se hace de la KVM, necesaria al trabajar con CLDC.

Paquete	Descripción
java.io	Clases e Interfaces de Entrada y Salida
java.lang	Clases básicas del lenguaje
java.lang.ref	Clases para referencias
java.lang.reflect	Clases e interfaces de reflexión
java.math	Paquete para operaciones matemáticas
java.net	Clases e interfaces de red

java.security	Clases e interfaces de seguridad
java.security.cert	Clases para el manejo de certificados de seguridad
java.text	Paquete para texto
java.util	Clases de utilidades estándar
java.util.jar	Clases y utilidades para el manejo de archivos JAR
java.util.zip	Clases y utilidades para el manejo de archivos ZIP y comprimidos
java.microedition.io	Clases e interfaces de conexión genérica para CDC

Tabla 1.7. Librerías de Configuración CDC

Fuente: Investigador

Los dispositivos que hacen uso de esta configuración (en la versión 1.1) deben cumplir con las siguientes características:

1. Al menos 192 kB de memoria total disponible para la plataforma Java,
2. Un procesador de 16 ó 32 bits,
3. Bajo poder de consumo, a menudo operan con suministro de energía limitado, a través de baterías,
4. Conectividad a algún tipo de red, por lo general inalámbrica, con conexión intermitente y con ancho de banda limitado.

Entre los requerimientos de hardware especificados para CLDC se encuentran:

- ✓ Al menos 160 kilobytes de memoria no volátil disponible para la maquina virtual y las librerías CLDC.
- ✓ Al menos 32 kilobytes de memoria volátil disponible para la ejecución de la maquina virtual.

“La CLDC aporta las siguientes funcionalidades a los dispositivos:

- ✓ Un subconjunto del lenguaje Java y todas las restricciones de su Máquina Virtual (KVM).
- ✓ Un subconjunto de las bibliotecas Java del núcleo.
- ✓ Soporte para E/S básica.
- ✓ Soporte para acceso a redes.
- ✓ Seguridad“.

Entre los dispositivos que soportan esta configuración se encuentran los teléfonos celulares, Asistentes Digitales Personales (PDAs), organizadores, decodificadores de televisión digital, y terminales de venta.

A continuación se muestra en la Tabla 1.8 las librerías utilizadas por la configuración CLDC.

Paquetes	Descripción
java.io	Subconjunto de J2SE para manejo de E/S
java.lang	Clases e interfaces de la Máquina Virtual
java.util	Utilidades estándar
javax.microedition.io	Clases de conexión genérica para CLDC

Tabla 1.8. Librerías de Configuración CLDC

Fuente: Investigador

De acuerdo a la tabla anterior, la Configuración CLDC las siguientes áreas:

- ✓ Lenguaje Java y características de la maquina virtual.
- ✓ Librerías del Núcleo Java (java.lang.*, java.util.*).
- ✓ Entrada/Salida (java.io.*).
- ✓ Seguridad.
- ✓ Interconexión.
- ✓ Internacionalización (tipo de codificación de caracteres).

Las demás características adicionales referentes al manejo del ciclo de vida de una aplicación en un dispositivo móvil, interfaz de usuario, manejo de eventos, etc., son administradas por perfiles implementados en la parte superior de CLDC.

2.7.2.3. Perfiles

Los perfiles definen de forma personalizada mediante APIs el manejo del ciclo de vida de la aplicación, interfaz de usuario, etc., extendiendo de esta manera la configuración a

través de la adición de clases, para un tipo específico de dispositivo, un ámbito de aplicación determinado o un segmento del mercado, permitiendo identificar un grupo de dispositivos a través de la funcionalidad que ofrecen y el tipo de aplicaciones que se ejecutan en ellos. Los perfiles surgen debido a la necesidad de soportar los diferentes requerimientos demandados por los consumidores de dispositivos.

Básicamente se considera importante el uso de librerías de interfaz gráfica para la definición de un perfil. Por lo tanto el perfil define rasgos propios de un dispositivo mientras que la configuración hace lo mismo con un conjunto de ellos. Así que a la hora de construir una aplicación se tenga a disposición tanto las APIs del perfil como de la configuración. Además un perfil está constituido bajo las bases de una configuración dotando a esta última de funcionalidades específicas. Se puede observar entonces las características del entorno de ejecución de J2ME, el cual se estructura en capas, observada en la Figura 2.11.

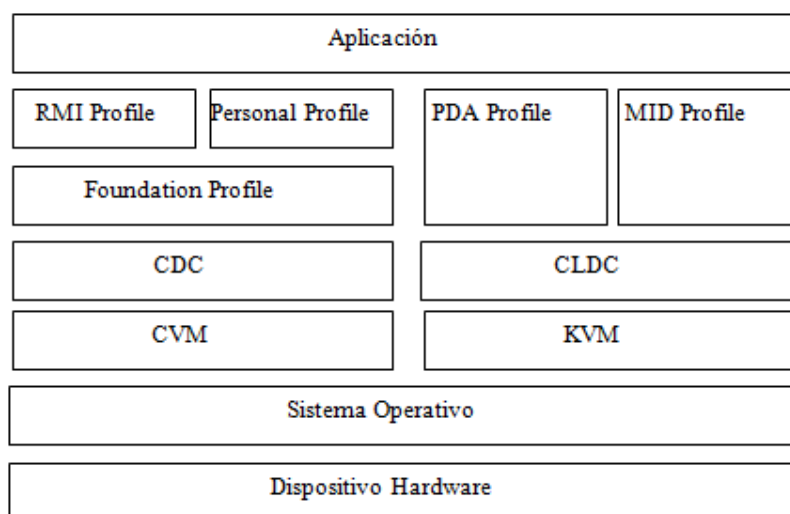


Figura 2.11. Entorno de ejecución para J2ME

Fuente: Investigador

Los perfiles al igual que la máquina virtual están predeterminados por la configuración que se pretende usar en el desarrollo de una aplicación J2ME. Para la configuración CDC se tienen los siguientes perfiles:

Foundation Profile: Orientada a dispositivos móviles que no tienen interfaz gráfica, como por ejemplo, decodificadores de televisión digital. Funciona como base para otros perfiles definidos para CDC. Excluye paquetes que hacen parte de la interfaz gráfica de J2SE, de tal manera que para poder implementar interfaz de usuario en forma gráfica sería necesario la implementación de otro perfil adicional. Las librerías usadas por el Foundation Profile se muestran en la Tabla 1.9

Paquetes	Descripción
java.io	Clases de Lectura y Escritura de J2SE
java.lang	Soporte del lenguaje Java.
java.util	Soporte para zip y otras funcionalidades (Timer)
java.net	Inclusión de sockets para TCP/IP y conexiones vía HTTP
java.text	Soporte para Internacionalización
java.security	Manejo de Códigos y certificados

Tabla 1.9. Librerías del Foundation Profile

Fuente: Investigador

Personal Profile: Extiende al Foundation Profile proporcionando soporte completo para gráficos AWT, permitiendo el manejo de applets y con capacidades web. Este perfil redefine a PersonalJava como un perfil J2ME. Las librerías utilizadas por este perfil se muestran en la Tabla 1.10.

Paquetes	Descripción
java.applet	Sirve para creación de applets
java.awt	Clase para crear Interfaces de Usuario con Manejo de Eventos.
java.awt.datatransfer	Clases e interfaces que permiten la transmisión de datos entre aplicaciones
java.awt.event	Clases e interfaces de manejo de eventos
java.beans	Soporte de JavaBeans
javax.microedition.xlet	Interfaces de comunicación del Personal Profile

Tabla 1.10. Librerías del Personal Profile

Fuente: Investigador

RMI Profile: Requiere de la implementación del Foundation Profile. Soporta un subconjunto de las APIs RMI (Invocación de Métodos Remotos) de J2SE. Ha sido necesaria la eliminación de ciertos rasgos del perfil RMI debido a las limitaciones propias de los dispositivos manejados bajo la configuración CDC. Estas características son:

- ✓ java.rmi.server.disableHTTP.
- ✓ java.rmi.activation.port.
- ✓ java.rmi.loader.packagePrefix.

- ✓ java.rmi.registry.packagePrefix.
- ✓ java.rmi.server.packagePrefix.

Para la configuración CLDC los perfiles más importantes son:

PDA Profile: construido sobre CLDC y abarca PDAs (Asistentes Digitales Personales) de baja gama, con pantalla cuya resolución oscile entre los 20000 pixeles y puntero.

- ✓ **Mobile Information Device Profile (MIDP Perfil de Información de Dispositivo Móvil):** Al igual que el perfil PDA este también está construido sobre CLDC y es el perfil central de esta configuración. Fue la primera configuración establecida para la plataforma J2ME. Este perfil maneja las siguientes características:
 - Capacidad computacional, gráfica y de memoria reducida.
 - Ciclo de vida de la aplicación (definición de la semántica de una aplicación MIDP y como se controla).
 - Almacenamiento persistente.
 - Conexión limitada.
 - Entrada de datos alfanumérica.
 - Transacciones punto a punto seguras (HTTPS).
 - Temporizadores.

Entre los requerimientos de hardware se puede encontrar:

- Tamaño de pantalla: 96 x 54 pixeles con factor 1:1.

- Uno o varios de los siguientes mecanismos de entradas: teclado o pantalla táctil.
- 256 kilobytes de memoria no volátil para la implementación de MIDP.
- 8 kilobytes de memoria no volátil para creación de aplicaciones con datos persistentes.
- 128 kilobytes de memoria volátil para el entorno de ejecución de Java.
- Dos formas posibles de interconexión: de forma inalámbrica, posiblemente intermitente ó con ancho de banda limitado.
- Capacidad para reproducir tonos, a través de un hardware o software.

Los requerimientos de software son los siguientes:

- Un sistema operativo pequeño que administre el hardware de las capas inferiores (que permita el manejo de excepciones e interrupciones).
- Un mecanismo que permita leer y escribir de la memoria no volátil para soportar los requerimientos de las APIs del Record Management Storage (RMS Manejo de Almacenamiento de Registros) que manejan almacenamiento persistente.
- Soporte para APIs que manejen Interconexión de redes Inalámbricas.
- Un mecanismo para la captura de entradas desde el usuario a través de cualquier componente de entrada referido en los requerimientos de hardware.
- Mecanismo para el manejo del ciclo de vida de la aplicación en el dispositivo.

Los paquetes que utiliza el perfil MIDP se pueden observar en la Tabla 1.11.

Paquetes	Descripción
javax.microedition.Icdui	Clases e interfaces para interfaces de usuario.
javax.microedition.rms	Administración del almacenamiento persistente en el dispositivo.
javax.microedition.midlet	Clases para la definición de la aplicación
javax.microedition.io	Manejo de conexión genérica.
java.io	Clases e interfaces de entrada y salida
java.lang	Clases e interfaces de la máquina virtual
java.util	Clase e interfaces de utilidades estándar

Tabla 1.11. Librerías del perfil MIDP

Fuente: Investigador

Las aplicaciones que se realizan utilizando MIDP y la configuración CLDC llevan el nombre de MIDlets.

2.7.2.4. Paquetes Opcionales

Además de las características que traen incorporadas los perfiles y configuraciones, existe la necesidad de adicionar librerías de uso general que no están limitadas a una categoría o familia de dispositivos. Los paquetes opcionales J2ME son un conjunto de APIs estándar, que pueden ser usada para extender un perfil y hacer uso tanto de tecnologías existentes como emergentes.

Estos paquetes opcionales especifican funcionalidad independiente de cualquier perfil, jugando un papel importante en la evolución de los mismos. El desarrollo de estos paquetes opcionales permite que las APIs maduren y sean creadas nuevas versiones a

través de la Comunidad de Procesos de Java (JCP), para luego ser incorporadas en un perfil.

A continuación se muestra en forma detallada algunos de los paquetes opcionales más usados:

API de Mensajería Inalámbrica (WMA, Wireless Messaging API) JSR 120: Define un conjunto de APIs que proporcionan acceso independiente de la plataforma a través de recursos de comunicación inalámbricos, como el Servicio de Mensajería Corta (SMS, Short Message Service). En la versión WMA 1.1 se incluyen arquitecturas que permiten el manejo de seguridad en la comunicación a través del perfil MIDP en su versión 2.0. Puede ser implementado en CLDC y CDC.

API de Contenido Multimedia Móvil (MMAPI, Mobile Media API) JSR 135: Proporciona control de recursos multimedia (audio y video), archivos y gran variedad de tipos de datos multimedia basados en tiempo, a dispositivos con recursos limitados. Al igual que WMA este paquete puede ser incorporado en aplicaciones desarrolladas para las configuraciones CLDC y CDC.

Especificación de Servicios Web en J2ME (WSA, Web Services API) JSR 172: Extiende el concepto de Servicios Web permitiendo que los dispositivos J2ME puedan ser consumidores de servicios Web a través de modelos de programación provenientes de la plataforma estándar de servicios Web. La infraestructura de esta API permite:

- Proporcionar capacidades para el procesamiento de archivos XML.

- Reusar el concepto de servicios Web en el momento de diseñar servicios empresariales para clientes J2ME.
 - Suministrar APIs y convenciones para el desarrollo de clientes J2ME de servicios empresariales.
 - Posibilitar la interacción entre los clientes J2ME con los servicios Web de forma interoperable.
 - Manejar un modelo de programación para la comunicación entre el cliente J2ME y los servicios Web.
 - Implementado exclusivamente en CLDC.
- ✓ **API para Servicios de Seguridad y Certificación en J2ME JSR 177:** Este paquete opcional extiende las características de seguridad para la plataforma J2ME a través del uso de APIs que proporcionen servicios de seguridad, con el fin de que sean suministrados mecanismos eficientes que soporten una amplia variedad de aplicaciones basadas en servicios, entre los que se encuentra, el acceso a redes corporativas, comercio electrónico, etc. Estas APIs de seguridad manejarán aspectos como el cifrado, firmas digitales, y gestión de credenciales de usuario, entre otros. También permiten definir un modelo de acceso que ayuda a las aplicaciones existentes en el dispositivo a comunicarse con una tarjeta inteligente insertada en el dispositivo, definiendo a través de mecanismos flexibles la definición de operaciones seguras entre los proveedores de servicios y el dispositivo. Usado para la configuración CLDC.

- ✓ **API de Localización para J2ME JSR 179:** Especifica que permiten el desarrollo de aplicaciones basadas en localización para dispositivos J2ME. Su propósito es proporcionar una API que genere información acerca de la ubicación geográfica y orientación del dispositivo para las aplicaciones Java. Permite el acceso a bases de datos con mapas almacenados en el terminal móvil. Define interfaces estándar para trabajar con metodologías de posicionamiento como GPS. Utilizado por CLDC y CDC.

- ✓ **Protocolo de Iniciación de Sesión para J2ME (SIP, Session Initiation Protocol) JSR 180:** La configuración que lo utiliza es CLDC. Permite el establecimiento y gestión de sesiones IP multimedia. Este mismo mecanismo puede ser ampliado para el soporte de mensajería multimedia y servicios de juego.

- ✓ **API para Gráficas Móviles en 3D para J2ME JSR 184:** Usado bajo CLDC, permite la generación de gráficos tridimensionales a frecuencias de imagen interactiva en dispositivos con recursos limitados. También se incluye la gestión de escenas 3D, animaciones, mensajes animados, salvapantallas, etc.

- ✓ **API para Bluetooth JSR 82:** Paquete usado en CLDC y MIDP. Proporciona mecanismos estándar para la creación de aplicaciones Bluetooth de modo que puedan ser ejecutadas a través de esta tecnología.

- ✓ **API para Invocación de Métodos Remotos (RMI) JSR 66:** Puede ser implementado en dispositivos con la configuración CDC y CLDC. Permite a dispositivos consumidores interactuar con aplicaciones distribuidas.

- ✓ **Paquete Opcional JDBC para CDC/Foundation Profile JSR 169:** Se puede utilizar en CDC y es un subconjunto de JDBC, que sirve para el procesamiento de datos de repositorio, que por lo general son bases de datos relacionales a través de SQL. Además permite la manipulación de datos tabulares como si fueran JavaBeans.²⁷

²⁷ <http://www.slideshare.net/arpa1986/aplicaciones-moviles-empresariales-con-servicios-web-y-j2me>

2.7.3. Ventajas y desventajas de J2ME

Ventajas	Desventajas
Portabilidad del código. Las aplicaciones se escriben una vez y pueden ejecutarse en cualquier dispositivo de las mismas características.	Dependencia del lenguaje Java
Descarga dinámica, de las aplicaciones sobre la red móvil.	Alejamiento del hardware. No puede accederse a ciertos recursos del teléfono si éste no incorpora el API correspondiente
Aplicaciones locales. Las aplicaciones se descargan al terminal y se ejecutan en local cuando quiera y donde quiera.	
Seguridad. Son aplicaciones autocontenidas, no tienen acceso al sistema operativo del dispositivo.	
Mejora drástica de la experiencia de usuario.	
Comunidad de desarrolladores, una de las más grandes del mundo. ²⁸	

Tabla 1.12. Cuadro de Ventajas y Desventajas de J2ME

Fuente: Investigador

²⁸ <http://www.slideboom.com/presentations/107714/04>

CAPITULO III

DESARROLLO DE LA APLICACIÓN MÓVIL WEB PARA EL REGISTRO DE LAS CALIFICACIONES Y ASISTENCIA DE LOS ALUMNOS DE LA ESCUELA DE INGENIERÍA DE SISTEMAS DE LA PUCESA

2.1. Análisis Comparativo de las Tecnologías ASP, AJAX, J2ME, J2EE y .NET como resultado de la investigación.

En la Tabla. 1.13 se resume las características más relevantes de las tecnologías ASP, AJAX, J2ME, J2EE y .NET, de las cuales se eligió la tecnología ASP.NET para el desarrollo de la aplicación RMW-PUCE “Registro Móvil Web PUCE”.

ANÁLISIS COMPARATIVO					
CARACTERÍSTICAS	AJAX	ASP	.NET	J2EE	J2ME
Compañía propietaria		Microsoft Corp.		Sun Microsystems	
Ambiente de Desarrollo			Requiere de Visual Estudio .Net		Completo
Plataforma		Servidor y Cliente de Microsoft	Solo plataforma Microsoft	Multiplataforma	Multiplataforma
Lenguaje de Programación		Lenguajes Scrip (VBscrip y Jscript)	Multilenguaje (C#, Visual Basic.NET, ASP.NET, C++,C, Java, J#, etc)	Java	
Tipo de Tecnología			Producto	Estándar	
Interpretación			CLR	JRE	
Entorno de Desarrollo			Visual Estudio .Net	Variedad de entornos (Eclipse, Netbeans, etc)	
Acceso a Base de Datos			ADO .NET	JDBC, SQL /J	
SopORTE de Conexión		ODBC	ODBC		
Arquitectura				Multicapa	
Ejecución	Cliente	Cliente-Servidor			
Tecnología	Conjunto de Tecnologías		Multiprototipo	Exclusivamente Empresarial	
Compilador JIT (Just In Time)			Si	Si	Si
Formato del código compilado			Compilador JIT del CLR ejecuta código IL(MSIL)	Compilador JIT de la JVM ejecuta .class, código intermedio (bytecode)	
Paradigma de programación			Orientado a objetos, declarativo, funcional	Orientado a objetos de forma no estricta, ni permite herencia múltiple ni sobrecarga de operadores.	
Premisas de la tecnología			Disponer de la tecnología apropiada para un problema específico Distribuir aplicaciones en forma de servicios	“Write Once, Run Anyware” Escríbelo una vez, ejecútalo donde sea.	

			<p>Comunicar y combinar aplicaciones independientes del modelo de programación, plataforma y lenguaje.</p> <p>Integrar todos los productos Microsoft, desde herramientas de desarrollo al sistema operativo.</p>	<p>Permitir la ejecución de un mismo programa compilado en múltiples sistemas operativos.</p> <p>Utilizar POO</p> <p>Incluir soporte para trabajo en red.</p> <p>Trabajar de forma segura en sistemas remotos.</p> <p>Desarrollar aplicaciones distribuidas.</p>
Orientada al desarrollo de Aplicaciones			Distribuido en múltiples lenguajes de programación , permite crear cualquier tipo de aplicación	<p>Empresariales</p> <p>Móviles</p>

Tabla 1.13. Cuadro comparativo de las características de las tecnologías ASP, AJAX, J2EE, J2ME y .NET

Fuente: Investigador

3.2. Ventajas y desventajas de las Tecnologías AJAX, ASP, .NET, J2EE, J2ME

A continuación se muestra en la Tabla.1.14 las ventajas y desventajas de las tecnologías ASP, AJAX, J2ME, J2EE y .NET, las cuales se analizaron para elegir una de ellas.

	Ventajas	Desventajas
AJAX	<ul style="list-style-type: none"> ✓ De código abierto y utiliza tecnologías existentes. ✓ Complemento en las capas básicas de la arquitectura cliente servidor. ✓ Se evita una capa de control. ✓ Por seguridad implementa validaciones en el lado del servidor. ✓ Mayor interactividad. ✓ Interfaz con mayor velocidad de respuesta ✓ Reduce cuello de botella. ✓ Se adiciona una capa lógica de presentación, ✓ Se han desarrollado muchos toolkit, librerías y frameworks. ✓ Portabilidad, no se requiere de ningún plugin. ✓ Independiente del tipo de tecnología de servidor que se utilice 	<ul style="list-style-type: none"> ✓ Mantenimiento de código. ✓ Usabilidad, los botones “atrás” y “refrescar” pueden funcionar mal. ✓ Dificultad en encontrar errores. ✓ Por ser nuevo puede ser susceptible a los ataques de los hackers. ✓ Problema en la accesibilidad, se basa en JavaScript ✓ Problemas con navegadores antiguos. ✓ Requiere de programadores que conozcan todas las tecnologías que intervienen en AJAX
ASP	<ul style="list-style-type: none"> ✓ Permite acceder a bases de datos. 	<ul style="list-style-type: none"> ✓ Al ser VBScript un lenguaje

	<ul style="list-style-type: none"> ✓ Las páginas se generan dinámicamente mediante el código de scripts. ✓ El código de script se ejecuta en el servidor. ✓ Desde una página ASP se pueden ejecutar servidores OLE en el servidor de web. ✓ El código ASP se puede escribir con un simple editor de textos. ✓ No se depende del navegador que se emplee. ✓ Es muy versátil para la utilización de plantillas (templates). ✓ Existe muchos Websites que brindan material que se puede utilizar como soporte. 	<p>propietario de Microsoft, sólo es procesado por defecto por los servidores de esta casa comercial, ya sea en las versiones de Internet Information Server (IIS) o en las de Personal Web Server (PWS).</p> <ul style="list-style-type: none"> ✓ La mayoría de los componentes no son gratuitos.
.NET	<ul style="list-style-type: none"> ✓ Interoperabilidad Multilenguaje. ✓ Documentación. ✓ Rendimiento. ✓ Rápido aprendizaje. ✓ Movilidad ✓ Escalabilidad y flexibilidad. ✓ Seguridad. ✓ Estándar abierto. 	<ul style="list-style-type: none"> ✓ Mantenimiento en múltiples lenguajes. ✓ .NET no es multiplataforma. ✓ Licencias. ✓ Portabilidad.
J2EE	<ul style="list-style-type: none"> ✓ Soporte de múltiples sistemas 	<ul style="list-style-type: none"> ✓ Depende de un único lenguaje.

	<ul style="list-style-type: none"> operativos. ✓ Organismo de control. ✓ Competitividad. ✓ Madurez. ✓ Soluciones libres. ✓ Capas. ✓ Estandarización de la solución. 	<ul style="list-style-type: none"> ✓ Complejidad. ✓ Heterogeneidad. ✓ Incremento del tiempo de desarrollo
J2ME	<ul style="list-style-type: none"> ✓ Portabilidad del código. ✓ Descarga dinámica. ✓ Aplicaciones locales. ✓ Seguridad. ✓ Mejora drástica de la experiencia de usuario. ✓ Comunidad de desarrolladores. 	<ul style="list-style-type: none"> ✓ Dependencia del lenguaje Java. ✓ Alejamiento del hardware.

Tabla 1.14. Cuadro Comparativo de las ventajas y desventajas de las tecnologías

Fuente: Investigador

3.3. Proceso de Análisis

3.3.1. Identificación del Problema.

Los procesos para el registro de la asistencia y las calificaciones de los alumnos de la Escuela de Sistemas de la PUCESA, se realizan de forma manual, se lleva un registro impreso en hojas de Excel, lo que puede provoca pérdida de tiempo en la actualización

de datos y la falta de información inmediata respecto a las calificaciones y al número de faltas que el alumno posee en un momento específico, o existe el riesgo que el registro se pierda.

3.3.2. Descripción de Requerimientos de la Aplicación Móvil Web

Se identificaron los siguientes requerimientos:

- Definición de hardware y software para el desarrollo
- Identificación de procesos que intervienen en la aplicación
- Seguridad, definir un usuario y un password para cada docente
- Registrar las notas de cada alumno
- Registrar las calificaciones de cada alumno
- Permitir al docente cambiar el password.

3.3.3. Metas

- Conocer las características más relevantes de las tecnologías ASP, AJAX, J2EE, J2ME y .NET en el desarrollo de aplicaciones móviles.
- Desarrollar una aplicación móvil web para el registro de la asistencia y las calificaciones de los alumnos de la Escuela de Sistemas.

3.3.4. Funciones

Registros (Calificaciones y Asistencia de los Alumnos)

F1. Seleccionar el nivel (paralelo-materia)

F2. Seleccionar parcial

F3. Visualizar los datos de los alumnos

F4. Registrar la calificación (2 #enteros, 2 #decimales)

F5. Registrar la asistencia (1= una falta)

F6. Guardar registro

Seguridad

F7. El sistema almacenará los datos de nuevo password de cada docente

F8. El sistema solicitara un nombre de usuario y password para su inicio.

F9. Al concluir las acciones se cierra la sesión.

3.3.5. Definición de Actores

En la aplicación existe el actor docente, es quien utilizara la aplicación móvil web para el registro de las calificaciones y asistencia de los alumnos.

Nombre	Descripción
Docente	Usuario que requiere de un usuario y password para tener acceso a la aplicación y poder efectuar el registro de las calificaciones y asistencia de los alumnos.
Administrador	Persona encargada de supervisar el correcto funcionamiento de la aplicación móvil web.

Tabla 1.15. Descripción del actor de los Casos de Uso

Fuente: Investigador

3.3.6. Datos

Registros (Calificaciones y Asistencia de los Alumnos)

Docente

Datos de los Alumnos

Datos de asistencia

Datos de las calificaciones

Seguridad

Usuario (Docente)

Password

3.3.7. Casos de Uso

En la Figura 2.12 Diagrama de Casos de Uso se visualizan las acciones que el docente realiza, las cuales se mencionan más detalladamente a continuación.

Registros (Calificaciones y Asistencia de los Alumnos)

Registro de Calificaciones de los Alumnos

Nombre: Registrar Calificaciones

Actores: Docente

Propósito: Registrar las calificaciones de cada estudiante

Resumen: El docente selecciona el nivel (paralelo- materia), a continuación el parcial. El registro de las notas llevará el siguiente formato: 00,00 (2 #enteros y 2 #decimales)

Referencias: En el literal 3.3.4 se hace mención una lista de funciones que la aplicación debe cumplir, de las cuales para el registro de las calificaciones de los alumnos se aplican las siguientes F1, F2, F3, F4, F6

Curso de los Eventos:

Actores	Sistema
1. El docente selecciona el nivel (paralelo-materia)	
	2. Visualiza en pantalla los parciales.
3. El docente selecciona el parcial del cual va a registrar las notas	
	4. El sistema despliega el registro de los alumnos.
5. El docente registra las notas siguiendo el formato establecido.	
	6. El sistema guarda los datos ingresado

Tabla 1.16. Curso de Eventos Registro de las Calificaciones

Fuente: Investigador

Cursos alternos

Línea 1. 3. El docente selecciona el nivel (paralelo-materia) o parcial equivocado. Existe la opción para retornar a la pantalla anterior.

Línea 5. En caso de que el docente ingrese la nota al alumno equivocado, la nota no podrá ser modificada, deberá ser notificada al administrador.

Registro de la Asistencia de los Alumnos

Nombre: Registrar asistencia

Actores: Docente

Propósito: Registrar la asistencia de los alumnos

Resumen: El docente selecciona el nivel (paralelo-materia) y parcial. El registro de la asistencia seguirá el siguiente formato: 1= una falta, el número de faltas al final será acumulativo.

Referencias: Para el registro de la asistencia de los alumnos se aplican las siguientes funciones F1, F2, F3, F5, F6 las cuales son detalladas en el literal 3.3.4.

Curso de los Eventos:

Actores	Sistema
1. El docente selecciona nivel (paralelo-materia)	
	2. Visualiza en pantalla los parciales
3. El docente selecciona el nivel-paralelo de la cual va a registrar la asistencia	
	4. El sistema despliega el registro de los alumnos.
5. El docente registra las faltas siguiendo el formato establecido.	
	6. El sistema guarda los datos ingresado

Tabla 1.17. Curso de los Eventos Registro de la Asistencia

Fuente: Investigador

Cursos alternos

Línea 1. 3. El docente selecciona el nivel (paralelo-materia) o parcial equivocado. Existe la opción para retornar al la opción anterior.

Línea 5. En caso de que el docente ingrese la falta al alumno equivocado, el registro no podrá ser modificada, deberá ser notificado al administrador.

Seguridad

Cambiar Password

Nombre: Cambiar Password

Actores: Docente

Propósito: Registrar los datos de nuevo password y confirmación del mismo

Resumen: La primera vez el docente accederá al sistema con su número de cédula, una vez haya accedido a la aplicación el docente elegirá la opción cambiar password, en esta pantalla ingresará el nuevo password y la confirmación del mismo y presionara el botón Cambiar. La base de Datos se actualiza automáticamente.

Referencias: Para la seguridad de la aplicación se emplean las funciones F7, F8. (Véase literal 3.3.4)

Curso de los eventos:

Actores	Sistemas
1. El docente ingresa los datos (nuevo password, confirmación del nuevo password)	
	2. Confirma datos iguales
3. Crea un nuevo password	
	4. Actualiza la base de datos con la información del nuevo password

Tabla 1.18. Curso de los Eventos Cambiar Password**Fuente: Investigador****Cursos alternos:****Línea 2.** Datos no coinciden, visualiza mensaje de error.

Iniciar Sesión

Nombre: Iniciar Sesión

Actores: Docente

Propósito: Iniciar la sesión para acceder al sistema

Resumen: El docente inicia la sesión, el sistema solicita el usuario y password, lo verifica. El docente accede al sistema

Referencias: Para el proceso de inicio de sesión se debe cumplir con la función F8 (véase literal 3.3.4)

Curso de los eventos:

Actores	Sistemas
1. El docente inicia la sesión	
	2. Solicitud escuela a la que pertenece el usuario (ESCUELA DE SISTEMAS), usuario y password
3. Selecciona Escuela (Sistemas)	
4. Ingresa usuario y password	
	5. Verifica los datos
6. Ingresa al sistema	

Tabla 1.19. Curso de los Eventos Iniciar Sesión

Fuente: Investigador

Cursos alternos:

Línea 3. Cuando no se encuentra el usuario y password se visualizara un mensaje de error.

Cerrar Sesión

Nombre: Cerrar Sesión

Actores: Docente

Propósito: Terminar la sesión y conexión del usuario con el servidor

Resumen: El docente cierra la sesión al concluir cualquier acción.

Referencias: Para el proceso de cerrar sesión se debe cumplir con la función F9 (véase literal 3.3.4)

Curso de los eventos:

Actores	Sistemas
1. El docente cierra la sesión	
	2. La aplicación AMW-EIS desconecta al usuario del servidor y cierra la aplicación.

Tabla 1.20. Curso de los Eventos Cerrar Sesión

Fuente: Investigador

3.3.8. Diagrama de Casos de Uso

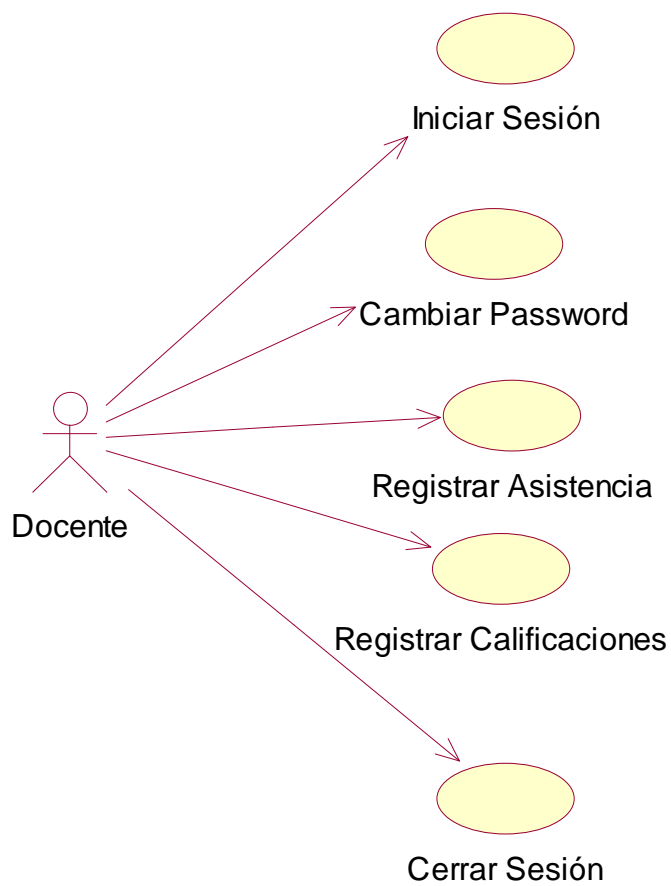


Figura 2.12. Diagrama de Casos de Uso

Fuente: Investigador

3.3.9. Diagrama de Secuencia

3.3.9.1. Inicio de Sesión

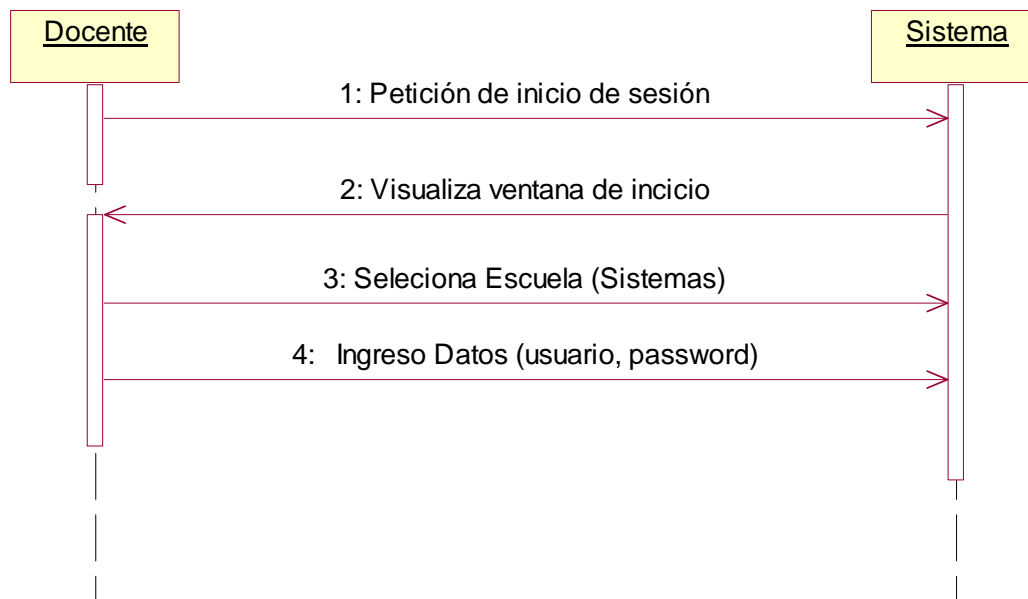


Figura 2.13. Inicio de Sesión

Fuente: Investigador

Nombre:	Iniciar Sesión (usuario, password)
Responsabilidad:	Verificar los datos ingresados de usuario y password, si son correctos acceder al menú principal. En caso de datos incorrectos volver a ingresar los datos
Tipo:	Sistema
Referencias:	Caso de uso iniciar sesión, diagrama de secuencias iniciar sesión.
Excepciones:	Si el sistema no identifica al usuario, el administrador verificará los datos. Si el usuario decide salir de la sesión, se le indicara al sistema al dar clic en la opción Cerrar Sesión.
Notas:	El sistema devuelve la respuesta hasta 5 seg
Salidas:	
Precondiciones:	El dispositivo móvil debe tener acceso a internet, en el browser se digitara la dirección de la aplicación.
Postcondiciones:	Si los datos escuela, usuario y password son válidos accede al menú principal. Si es nuevo usuario, deberá ingresar por primera vez con su número de cédula. Si los datos ingresados son incorrectos, despliega mensaje de error.

Tabla 1.21. Diagrama de Secuencias iniciar sesión

Fuente: Investigador

3.3.9.2. Cerrar Sesión

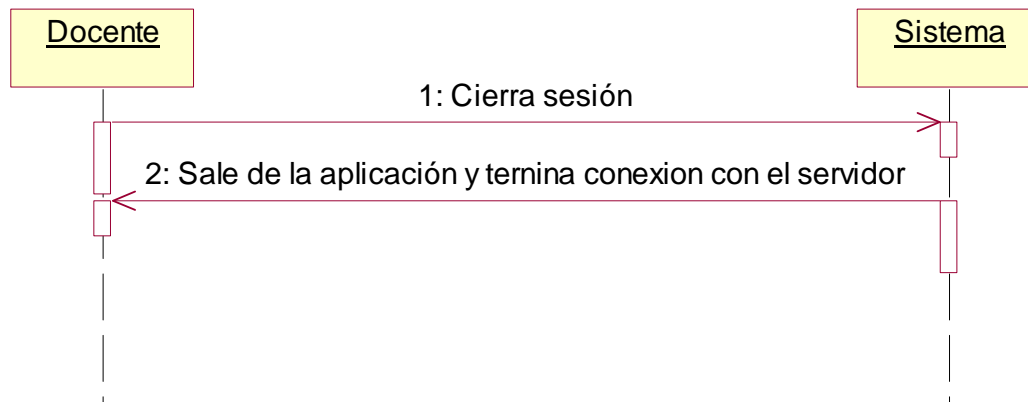


Figura 2.14. Cerrar Sesión

Fuente: Investigador

Nombre:	Cerrar Sesión
Responsabilidad:	Salir de la aplicación y terminar conexión con el servidor.
Tipo:	Sistema
Referencias:	Caso de uso cerrar sesión, diagrama de secuencias cerrar sesión.
Excepciones:	Si el usuario decide salir de la sesión, se le indicara al sistema al dar clic en la opción Cerrar sesión.
Notas:	El sistema devuelve la respuesta hasta 5 seg
Salidas:	
Precondiciones:	El dispositivo móvil debe tener acceso a internet, al salir de la aplicación se re direccionará a otra pantalla.

Tabla 1.22. Diagrama de Secuencias iniciar sesión

Fuente: Investigador

3.3.9.3. Registro Calificaciones y Asistencia

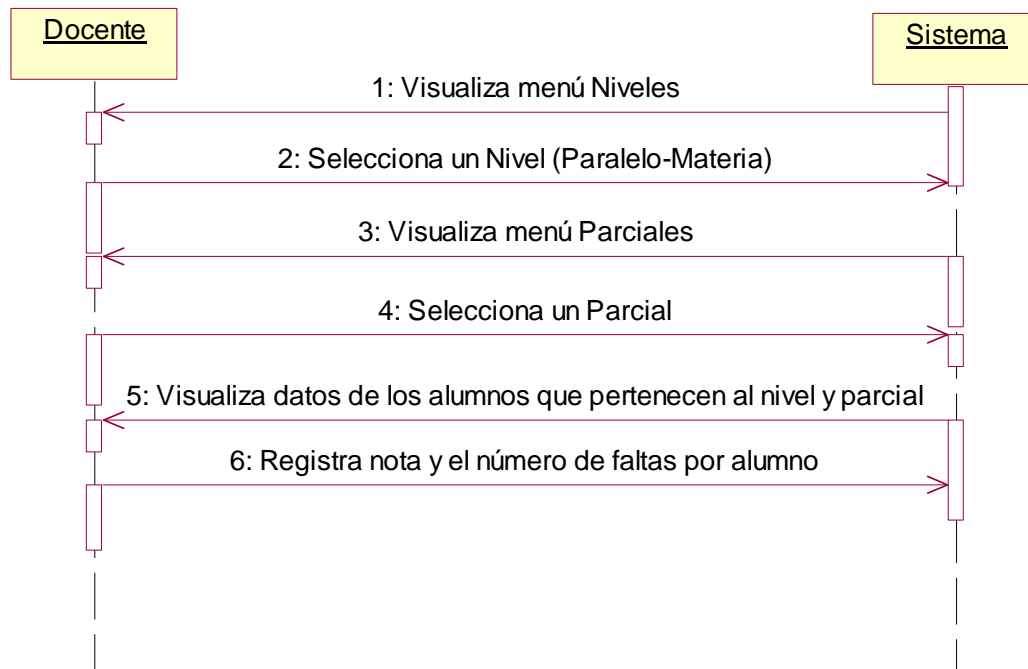


Figura 2.15. Registro Calificaciones

Fuente: Investigador

Nombre:	Registro Calificaciones y Asistencia
Responsabilidad:	Ingresar en cada parcial las notas y el número de faltas de los alumnos.
Tipo:	Sistema
Referencias:	Caso de uso registrar calificaciones, registrar asistencia diagrama de secuencias registrar calificaciones y asistencia.
Excepciones:	
Notas:	El sistema devuelve la respuesta hasta 1 seg
Salidas:	Registro almacenado en la base de datos.
Precondiciones:	El dispositivo móvil debe tener acceso a internet, en el browser se digitara la dirección de la aplicación.
Postcondiciones:	Se crea una instancia de registro que es almacenada en la base de datos al momento de elegir Registrar.

Tabla 1.23. Diagrama de Secuencias Registro Calificaciones

Fuente: Investigador

3.3.9.4. Cambio de Password

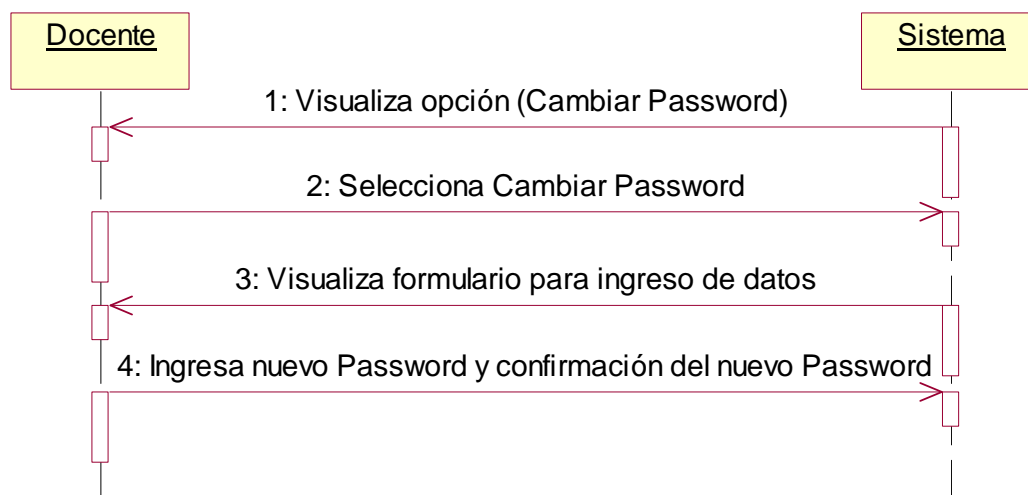


Figura 2.16. Cambiar Password

Fuente: Investigador

Nombre:	Seguridad
Responsabilidad:	Cambiar el password cuando se lo requiera necesario
Tipo:	Sistema
Referencias:	Caso de uso cambiar password.
Notas:	El sistema devuelve la respuesta hasta 1 seg
Salidas:	Registro almacenado en la base de datos.
Precondiciones:	El dispositivo móvil debe tener acceso a internet, en el browser se digitara la dirección de la aplicación.
Postcondiciones:	Se crea una instancia de actualización del campo password en la tabla dba.tbl_profesor de la base de datos al momento de elegir Cambiar.

Tabla 1.24. Diagrama de Secuencias Cambiar Password

Fuente: Investigador

3.4. Proceso de Diseño

El diseño y desarrollo de la aplicación móvil web, se ha llevado a cabo a través de diferentes fases las cuales se describen a continuación.

3.4.1. Diseño global de la Aplicación Móvil Web.

La aplicación móvil web se implemento utilizando una arquitectura cliente-servidor. En el servidor se recepta y almacena los datos de calificaciones y asistencia de los alumnos.

El programa servidor interactúa con el usuario docente, que es quien realiza el registro.

En la Figura 2.17 se muestra la relación entre los componentes mencionados. El docente trabaja directamente con la aplicación móvil web alojada en el servidor por medio del dispositivo móvil. La información registrada a través del dispositivo móvil es almacenada en la base de datos que se encuentra en el servidor.

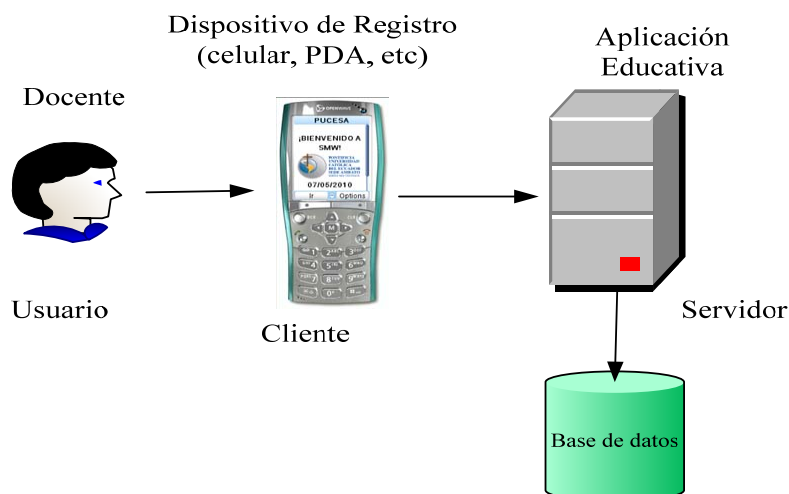


Figura 2.17. Arquitectura del sistema

Fuente: Investigador

3.4.2. Diseño de la Base de Datos

La base de datos empleada en la aplicación es la que actualmente se encuentra implementada en la universidad, la cual fue diseñada con la herramienta Sybase SQL Anywhere 5.0

Server Name: sistemasbd

Database Name: sistemasdb

A continuación se listan las tablas utilizadas en el desarrollo de la aplicación móvil web:

- tbl_alumno
- tbl_hc_detalle
- tbl_hc_enc
- tbl_materia
- tbl_notas_asistencia
- tbl_paralelo
- tbl_parcial
- tbl_pen_ciclo_sisteval
- tbl_prof_mat_np_pen
- tbl_profesor
- tbl_pensum
- tbl_periodo
- tbl_sisteval
- tbl_sisteval_parcial

3.4.3. Diagrama Entidad Relación

En el diagrama entidad-relación se muestran las tablas de la base de datos “**sistemasdb**”, la cual está implementada actualmente en la universidad y con la cual se trabajo para el desarrollo de la aplicación móvil web.

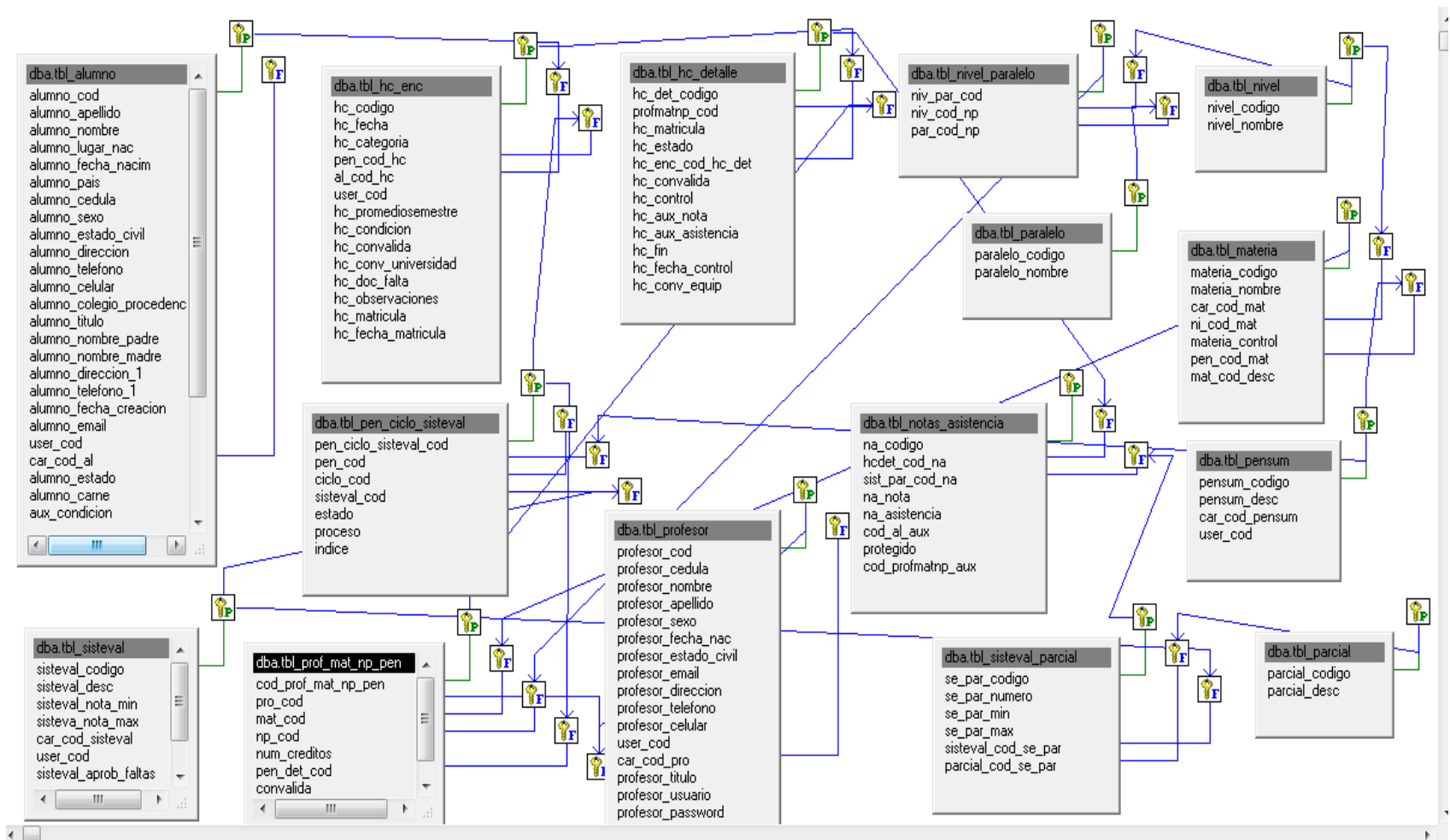


Figura 2.18. Diagrama Entidad Relación

Fuente: Investigador

3.4.4. Definición de Procesos

- ✓ Validar usuarios
- ✓ Registrar el número de faltas.
- ✓ Registrar las calificaciones.
- ✓ Cambiar Password

3.4.5. Seguridad del Sistema

El administrador del sistema será el encargado de crear los usuarios (docentes).

Los docentes accederán al sistema mediante la autenticación de su usuario y password, dicha información será comparada con los datos de la base de datos.

El docente tendrá la opción de cambiar su password.

Al concluir las acciones existe la opción cerrar sesión para salir del sistema.

3.4.6. Diseño de Interfaz

Página Inicial



Figura 2.19. Página Principal

Fuente: Investigador

Inicio de Sesión

Escuela de Sistemas

Iniciar sesión

Usuario:

Password:

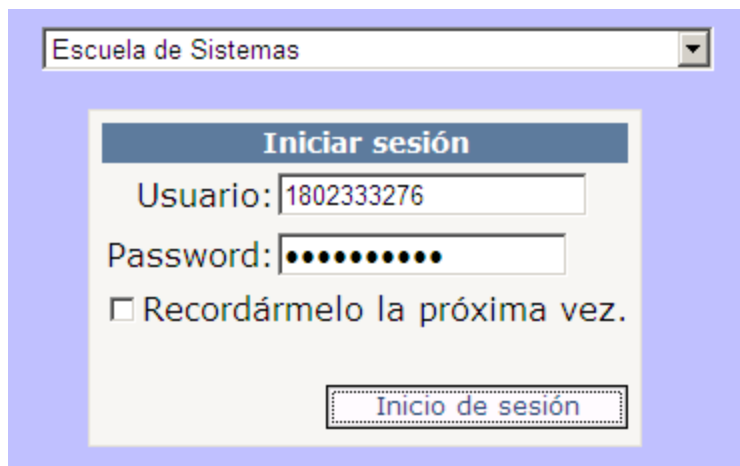
Recordármelo la próxima vez.

Inicio de sesión

Figura 2.20. Inicio de Sesión

Fuente: Investigador

Autenticación de Usuario



Escuela de Sistemas

Iniciar sesión

Usuario: 1802333276

Password: ●●●●●●●●

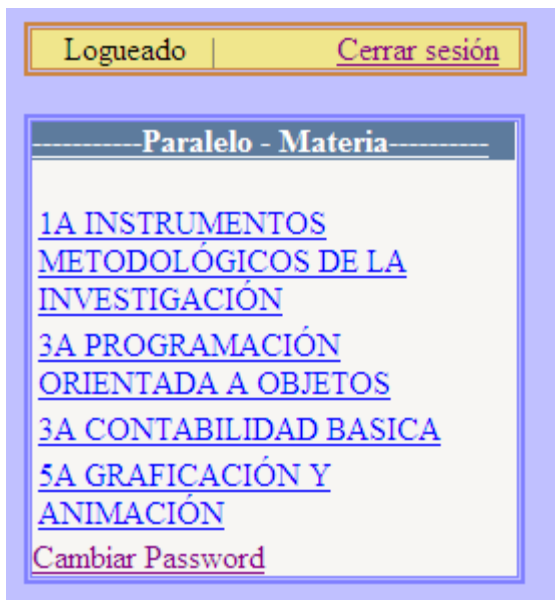
Recordármelo la próxima vez.

Inicio de sesión

Figura 2.21. Autenticación de Usuario

Fuente: Investigador

Menú de opciones de la Aplicación



Logueado | [Cerrar sesión](#)

-----Paralelo - Materia-----

[1A INSTRUMENTOS
METODOLÓGICOS DE LA
INVESTIGACIÓN](#)

[3A PROGRAMACIÓN
ORIENTADA A OBJETOS](#)

[3A CONTABILIDAD BASICA](#)

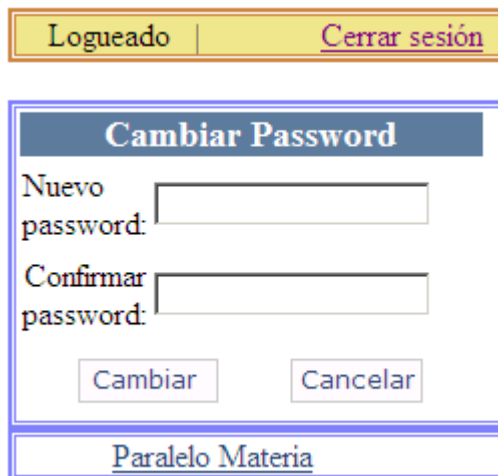
[5A GRAFICACIÓN Y
ANIMACIÓN](#)

[Cambiar Password](#)

Figura 2.22. Menú de opciones de la Aplicación

Fuente: Investigador

Cambiar Password



Logeado | [Cerrar sesión](#)

Cambiar Password

Nuevo password:

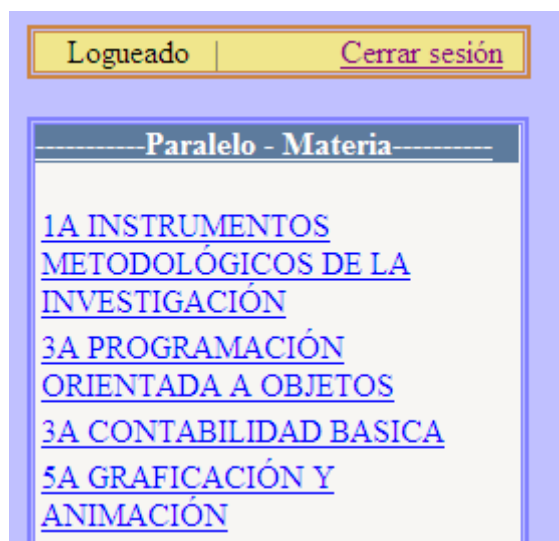
Confirmar password:

[Paralelo Materia](#)

Figura 2.23. Cambiar Password

Fuente: Investigador

Menú de Paralelo Materia



Logeado | [Cerrar sesión](#)

Paralelo - Materia

[1A INSTRUMENTOS METODOLÓGICOS DE LA INVESTIGACIÓN](#)

[3A PROGRAMACIÓN ORIENTADA A OBJETOS](#)

[3A CONTABILIDAD BASICA](#)

[5A GRAFICACIÓN Y ANIMACIÓN](#)

Figura 2.24. Menú de Niveles (Paralelo - Materia)

Fuente: Investigador

Menú de Parciales

Logueado | Cerrar sesión

Parcial

[PRIMER PARCIAL](#)

[SEGUNDO PARCIAL](#)

[FINAL](#)

Paralelo Materia

Figura 2.25. Menú Parciales

Fuente: Investigador

Ventana de Registro de Notas y Faltas

Logueado | Cerrar sesión

3A CONTABILIDAD BASICA--PRIMER PARCIAL

REGISTRO ALUMNO

Apellido: ALTAMIRANO
IBARRA

Nombre: GABRIEL
EDUARDO

Nota:

Falta:

Registrar

Paralelo Materia Parcial

Figura 2.26. Ventana Registro de Notas y Faltas

Fuente: Investigador

3.5. Proceso de Implementación

3.5.1. Plataforma de Desarrollo .NET

Se selecciona la plataforma de desarrollo .Net dado que ofrece las ventajas que se enumeran a continuación:

- ✓ Gran variedad de librerías que automatizan funcionalidades comunes.
- ✓ Librerías que simplifican la comunicación con dispositivos de hardware.
- ✓ Programas que se ejecutan con pocos recursos en el computador.

3.5.2. Diseñador de ASP.NET Mobile

El Diseñador de ASP.NET Mobile complementa a ASP.NET y a .NET Framework, permitiendo generar aplicaciones Web para teléfonos móviles, PDA y localizadores. Este diseñador está integrado en el IDE de Visual Studio. Puede crear aplicaciones Web móviles, utilizar el diseñador móvil para modificar un formulario Web móvil y, a continuación, generar y ejecutar la aplicación, todo desde Visual Studio.

3.5.3. Formularios Web Forms

Los formularios Web Forms son una tecnología ASP.NET que se utiliza para crear páginas Web programables. Los formularios Web Forms se representan como código HTML y secuencias de comandos compatibles con exploradores, lo que permite ver las páginas en cualquier explorador y plataforma. Mediante el uso de formularios Web Forms se pueden crear páginas Web arrastrando y colocando controles en el diseñador y

agregando código posteriormente, de forma parecida a la creación de formularios en Visual Basic.²⁹

3.5.4. Herramientas utilizadas en la implementación

✓ **Rational Rose Enterprise Edition:** Es la mejor elección para el ambiente de modelado que soporte la generación de código a partir de modelos en Ada, ANSI C++, C++, CORBA, Java™/J2EE™, Visual C++® y Visual Basic®. Como todos los demás productos Rational Rose, proporciona un lenguaje común de modelado para el equipo que facilita la creación de software de calidad más rápidamente.³⁰

✓ **Microsoft Office Visio 2003:** Es un software de dibujo vectorial para Microsoft Windows.

Las herramientas que lo componen permiten realizar diagramas de oficinas, diagramas de bases de datos, diagramas de flujo de programas, UML, y más, que permiten iniciar al usuario en los lenguajes de programación.³¹

✓ **Microsoft Project 2007:** Es un programa de la suite Microsoft Office usado para la gestión de proyectos.

Microsoft Project (o MSP) es un software de administración de proyectos diseñado, desarrollado y comercializado por Microsoft para asistir a administradores de

²⁹ www.msdn.microsoft.com

³⁰ http://es.wikipedia.org/wiki/Rational_Software

³¹ http://es.wikipedia.org/wiki/Microsoft_Visio

proyectos en el desarrollo de planes, asignación de recursos a tareas, dar seguimiento al progreso, administrar presupuesto y analizar cargas de trabajo.³²

- ✓ **Microsoft Visual .NET 2005:** Es un entorno de desarrollo integrado (IDE, por sus siglas en inglés) para sistemas operativos Windows. Soporta varios lenguajes de programación tales como Visual C++, Visual C#, Visual J#, ASP.NET y Visual Basic .NET, aunque actualmente se han desarrollado las extensiones necesarias para muchos otros.

Visual Studio permite a los desarrolladores crear aplicaciones, sitios y aplicaciones web, así como servicios web en cualquier entorno que soporte la plataforma .NET (a partir de la versión net 2002). Así se pueden crear aplicaciones que se intercomunican entre estaciones de trabajo, páginas web y dispositivos móviles.³³

³² http://es.wikipedia.org/wiki/Microsoft_Project

³³ http://es.wikipedia.org/wiki/Microsoft_Visual_Studio

3.5.5. Requerimientos de Hardware y Software

3.5.5.1. Requerimientos mínimos

Requisitos Mínimos para PC Servidor.

Requerimientos mínimos	
Procesador	Intel Pentium IV a 2.8 GHz o superior
Sistema Operativo	Microsoft Windows Server 2003
Memoria RAM	512 MB
Disco Duro	40GB
Monitor	Resolución: 800 x 600, 256 colores
Dispositivo de entrada	Teclado y mouse
Servidor Web	Apache 2.2
Base de Datos	Sybase SQL Anywhere 5
Programa	Microsoft Visual Studio 2005 (Framework v2)
Otros	Internet Explorer

Tabla 1.25. Requerimientos mínimos para PC servidor

Fuente: Investigador

Requisitos Mínimos para Dispositivo Móvil

Requerimientos mínimos	
Dispositivo Móvil	Celular, PDA, Notebook, ect.
Internet	Navegador web o browser (iExplorer Mobile, OperaMini)
Conexión a internet	WI-FI

Tabla 1.26. Requerimientos mínimos para Dispositivo

Fuente: Investigador

3.6. Proceso de Pruebas

Esta etapa se realiza la identificación y depuración de fallas en la parte lógica de las interfaces, verificando las validaciones, el funcionamiento correcto de todos los procesos, afinación de detalles, y finalmente asegurando que el sistema cumpla con los objetivos definidos previamente.

3.6.1. Pruebas de Usabilidad del Servidor: “Sistemasbd”

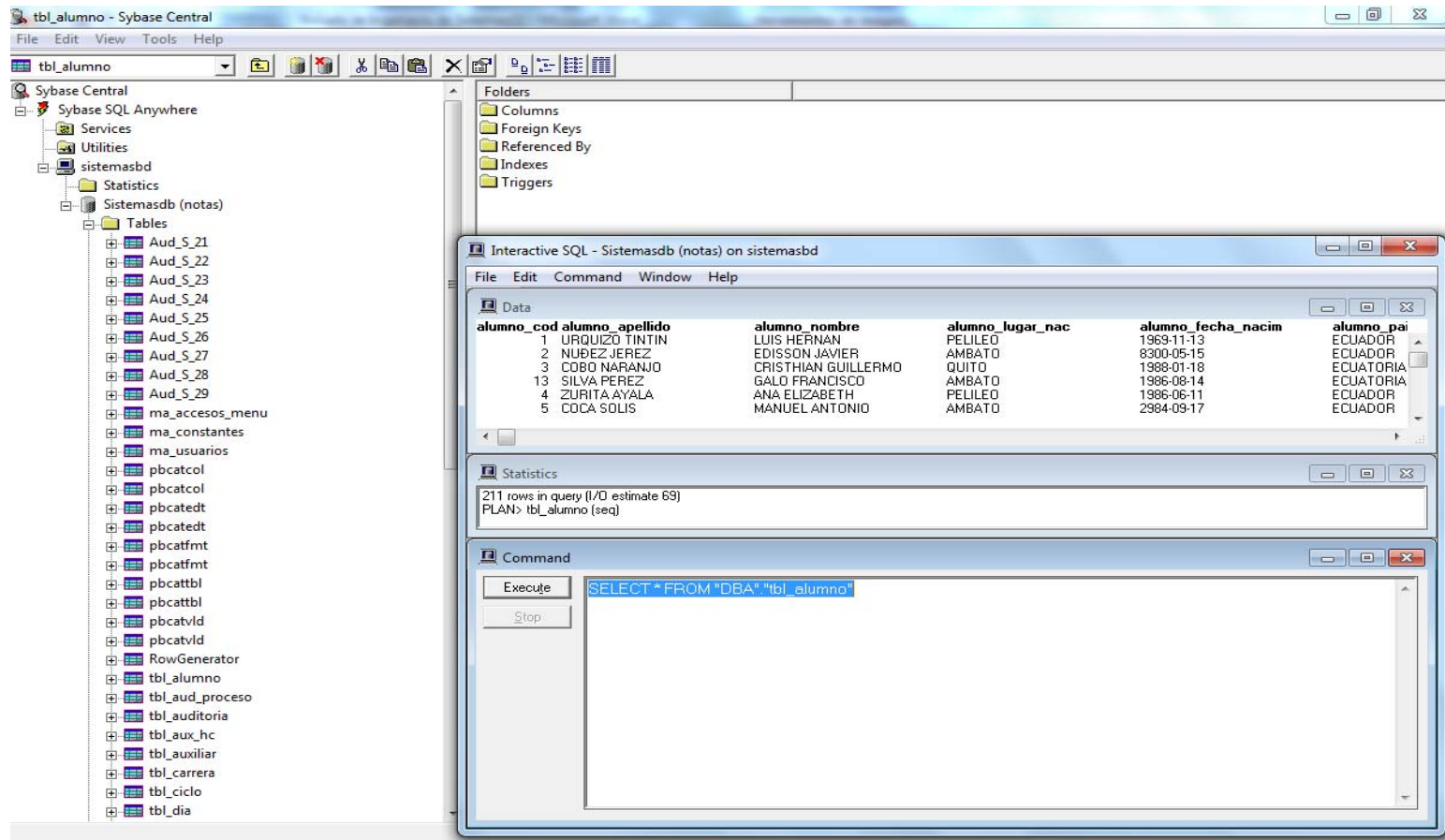


Figura 1.27. Relaciones de las Tablas de la Base de Datos

Fuente: Investigador

3.6.2. Código de la Aplicación

Login.ascx

```

<%@ import Namespace="System.Threading" %>
<%@ import Namespace="System.IO" %>
<%@ import Namespace="System.Data" %>
<%@ import Namespace="System.Data.Odbc" %>
<script runat="server">
protected void Page_Load(object sender, EventArgs e)
{

    string escuela = Page.MapPath("conexion.txt");
    StreamReader sr = new StreamReader(escuela );
    string linea;

    while ((linea=sr.ReadLine())!=null)
    {
        string[] campos = linea.Split('\t'); //Corta por los tabs

        DropDownList1.Items.Add (new ListItem(campos [0],campos [1]) );
        DropDownList1.DataBind();

        Session["sistema"] = DropDownList1.SelectedValue.ToString();

    }

}

protected void Login1_Authenticate(object sender, AuthenticateEventArgs e)
{
    string sistema = Convert.ToString(Session["sistema"]);

    OdbcConnection conexion = new OdbcConnection(" dsn=" + sistema + ";Uid=notas;Pwd=12345");

    try
    {
        conexion.Open();
        //asigno a la variable comando la consulta
        OdbcCommand consultaDoc = new OdbcCommand("Select * from DBA.tbl_profesor where
profesor_usuario=" + Login1.UserName + " And profesor_password=" + Login1.Password + """,
conexion);
        // declaro la variable de lectura
        OdbcDataReader resultadoDoc;

        resultadoDoc = consultaDoc.ExecuteReader();

        if (resultadoDoc.Read())
        {
            Session["usuario"] = Login1.UserName;
            Session["password"] = Login1.Password;
        }
    }
}
}

```

```

        Response.Redirect("paralelo_materia.aspx");
    }
    else
    {
        System.Web.UI.WebControls.TextBox usuario1 = Login1.FindControl("UserName") as
System.Web.UI.WebControls.TextBox;
        System.Web.UI.WebControls.TextBox password1 = Login1.FindControl("Password") as
System.Web.UI.WebControls.TextBox;

        if (((string.IsNullOrEmpty(usuario1.Text )) && (string.IsNullOrEmpty(password1.Text ))) ||
((string.IsNullOrEmpty(usuario1.Text )) || (string.IsNullOrEmpty(password1.Text ))))
        {
            System.Web.UI.WebControls.Label text = Login1.FindControl("Label1") as
System.Web.UI.WebControls.Label;
            text.Text = " ";
        }
        else
        {
            System.Web.UI.WebControls.Label text = Login1.FindControl("Label1") as
System.Web.UI.WebControls.Label;
            text.Text = "Usuario o password incorrectos";
        }
    }
}
catch (Exception i)
{
    Response.Write("ACCESO DENEGADO : " + i );
}

finally
{
    if (conexion.State == ConnectionState.Open)
    {
        conexion.Close();
        conexion.Dispose();
    }
}
}
</script>

```

Paralelo.ascx

```

<%@ import Namespace="System.Threading" %>
<%@ import Namespace="System.IO" %>
<%@ import Namespace="System.Data" %>
<%@ import Namespace="System.Data.Odbc" %>
<script runat="server">
protected void Page_Load(object sender, EventArgs e)
{
    //variables de session
    string usuario = Convert.ToString(Session["usuario"]);
    string password = Convert.ToString(Session["password"]);
    //conexion
    string sistema = Convert.ToString(Session["sistema"]);
    OdbcConnection conexion = new OdbcConnection(" Dsn=" + sistema + "; Uid=notas;Pwd=12345");
    try
    {
        if (!Page.IsPostBack)
        {
            OdbcDataAdapter adapterNivel = new OdbcDataAdapter("select (pm.np_cod+' '+
m.materia_nombre )as nivel from dba.tbl_pen_ciclo_sisteval pc, dba.tbl_nivel_paralelo np,
dba.tbl_materia m, dba.tbl_profesor p, dba.tbl_prof_mat_np_pen pm where
pc.pen_ciclo_sisteval_cod=pm.pen_det_cod and np.niv_par_cod=pm.np_cod and
m.materia_codigo=pm.mat_cod and p.profesor_cod=pm.pro_cod and pc.estado='a' and
profesor_usuario="" + usuario + "" And profesor_password="" + password + "" ", conexion);
            DataTable datosNivel = new DataTable();
            adapterNivel.Fill(datosNivel);
            List1.DataSource = datosNivel;
            List1.DataTextField = "nivel";
            List1.DataBind();
        }
    }
    catch (Exception i)
    {
        Response.Write("ACCESO DENEGADO : " + i);
    }

    finally
    {
        if (conexion.State == ConnectionState.Open)
        {
            conexion.Close();
            conexion.Dispose();
        }
    }
}
protected void List1_ItemCommand(object sender, ListCommandEventArgs e)
{
    string nivel = e.ListItem.ToString();

    Session["nivel"] = nivel ;

    Response.Redirect ("parcial.aspx");
}
</script>

```

Parcial.ascx

```

<%@ import Namespace="System.Threading" %>
<%@ import Namespace="System.IO" %>
<%@ import Namespace="System.Data" %>
<%@ import Namespace="System.Data.Odbc" %>

<script runat="server">

protected void Page_Load(object sender, EventArgs e)
{
    string usuario = Convert.ToString(Session["usuario"]);
    string password = Convert.ToString(Session["password"]);
    string sistema = Convert.ToString(Session["sistema"]);

    OdbcConnection conexion = new OdbcConnection(" Dsn=" + sistema + "; Uid=notas;Pwd=12345");

    try
    {
        if (!Page.IsPostBack)
        {
            OdbcDataAdapter comando = new OdbcDataAdapter("select ( parcial_desc) as parcial from
dba.tbl_parcial pa, dba.tbl_sisteval_parcial sp, dba.tbl_sisteval si, dba.tbl_pen_ciclo_sisteval pc where
pa.parcial_codigo=sp.parcial_cod_se_par and si.sisteval_codigo=sp.sisteval_cod_se_par and
si.sisteval_codigo=pc.sisteval_cod and pc.estado='A'", conexion);
            DataSet datos = new DataSet();
            comando.Fill(datos);
            List1.DataSource = datos.Tables[0];
            List1.DataTextField = "parcial";
            List1.DataBind();
        }
    }
    catch (Exception i)
    {
        Response.Write("ACCESO DENEGADO : " + i);
    }
    finally
    {
        {
            if (conexion.State == ConnectionState.Open)
            {
                conexion.Close();
                conexion.Dispose();
            }
        }
    }
}

protected void List1_ItemCommand1(object sender, ListCommandEventArgs e)
{
    string parcial = e.ListItem.ToString();
    Session["parcial"] = parcial;
    Response.Redirect("registro_alumno.aspx");
}

</script>

```

CAPITULO IV

VALIDACIÓN Y VERIFICACIÓN DE LOS RESULTADOS

4.1. Análisis General

En el presente capítulo se muestra los resultados obtenidos en la verificación y validación de los resultados en el uso de la aplicación, que permitieron analizar el nivel de funcionalidad de la aplicación móvil web denominada RMW-PUCE “Registro Móvil Web PUCE”.

4.2. Validaciones

4.3. Conclusiones

- ✓ Cada tecnología proporciona características funcionales, pero basándonos en el desarrollo de aplicaciones móviles las tecnologías que más contribuyen al desarrollo de las mismas son .NET y J2ME, cada una con distinto entorno de desarrollo y lenguaje de programación. La tecnología .NET cuenta con un entorno de desarrollo integrado con un conjunto de librerías y varios lenguajes de programación, permitiendo al programador escoger el lenguaje de programación más apropiado para el desarrollo de aplicaciones.
- ✓ La tecnología .NET permite desarrollar aplicaciones móviles web con ASP.NET que cubren el requerimiento de una aplicación cliente-servidor. Proporciona un conjunto de herramientas móviles que ayudan en la interfaz para que se adapte al dispositivo móvil, autenticación de usuario, acceso a una base de datos, compatible con exploradores.
- ✓ Las aplicaciones nativas al ser instalables requieren que se tome en cuenta el Sistema Operativo del dispositivo, el tamaño de la pantalla y otros recursos propios, a diferencia de las aplicaciones móviles web que necesitan que el dispositivo móvil tenga un browser o navegador web para poder acceder a la aplicación. El uso de la aplicación móvil web hacen que se mantenga actualizado los datos ya que se registran los datos en tiempo real.
- ✓ Se ha desarrollado una aplicación móvil web denominado AMW-EIS que cumple con los propósitos de seguridad, interfaz amigable, útil para que el docente de la Escuela de Ingeniería de Sistemas realice el registro de las calificaciones y la asistencia de los alumnos, mediante el uso de un dispositivo móvil.

- ✓ Para no tener problemas al probar la aplicación publicada en el servidor Apache es necesario instalar el modulo “mod_aspdotnet-2.0.0“ y configurarlo en Apache.
- ✓ Las pruebas de funcionalidad realizadas en los dispositivos notebook, celular Iphone y celular WI-FI TV fueron satisfactorias, el único requerimiento fue tengan incluido un browser o navegador web y WI-FI para la comunicación.

4.4. Recomendaciones

- ✓ Para el desarrollo de aplicaciones se debe elegir un software estable, conocido, no de versión beta porque son de prueba y tienen un corto periodo de uso.
- ✓ Establecer correctamente los requerimientos de la aplicación es fundamental para poder establecer correctamente el tiempo estimado en el desarrollo de la aplicación y que la aplicación sea funcional.
- ✓ Si los datos a visualizar en el dispositivo móvil son muchos, es recomendable cargar los datos en múltiples pantallas y en forma vertical para una mejor presentación y visualización.
- ✓ Orientar al desarrollo de nuevas aplicaciones móviles nativas y web realizando un estudio previo de compatibilidad con el dispositivo.
- ✓ Implementar en la aplicación AMW-EIS las opciones para modificar y consultar los datos de las calificaciones y faltas registradas.

BIBLIOGRAFÍA

- ✓ Platt, David. Así es Microsoft .NET. Carmelo Sánchez Gonzales. 2001. Washington EE.UU. McGRAW-HILL /INTERAMERICANA DE ESPAÑA S.A.U MADRID: Sep.2009
- ✓ Dave, Mercer. Fundamentos de Programación en ASP 3.0. Eder Mauricio Hernandez, 2001. Bogotá D-C, Colombia. McGRAW-HILL INTERAMERICANA, S.A: Sep.2009.
- ✓ MacDonald, Matthew. Manual de Referencia ASP.NET. Enrique M. Alcazar Fernandez. 2002. Madrid. McGRAW-HILL INTERAMERICANA DE ESPAÑA, S.A.U: Sep. 2009.
- ✓ Sierra, Martin, Antonio J. Ajax en J2EE. Alfaomega Grupo Editor, S.A de CV, Mexico. Primera/Alfaomega Grupo Editor, Mexico, Nov. 200. Madrid-España. RA-MA, Sep. 2009

Sitios Web

Aplicaciones Móviles

- ✓ Identificación y Clasificación de Patrones en el Diseño de Aplicaciones Móviles, Nov. 2009, Ing. Darío Yorio, < <http://postgrado.info.unlp.edu.ar/Carrera/Magister/Ingenieria%20de%20Software/Tesis/Yorio.pdf> >
- ✓ Diaweb Móvil: Adaptación del portal Diaweb para el acceso desde dispositivos móviles, Nov.2009, < <http://majuelo.usal.es/diaweb20/archivos/10008835resumenMovil.pdf> >

- ✓ Tipos de dispositivos, Nov.2009, Krasis Consulting S.L < <http://aulas.campusmvp.com> >
- ✓ Tipos de dispositivos, Nov.2009, Juan Manuel Fernández Luna < http://leo.ugr.es/J2ME/INTRO/intro_4.htm >

Tecnología ASP

- ✓ ASP. Recursos y Características, Dic. 2009, < http://tgp0607.awardspace.com/Recursos_ASP.pdf >
- ✓ ASP, Dic.2010, Autor original : Sergio Luján Mora, Laboratorio Multimedia de la Universidad de Alicante, Alicante, España, <[http://gplsi.dlsi.ua.es/~slujan/asp/Introduccion a ASP.htm](http://gplsi.dlsi.ua.es/~slujan/asp/Introduccion_a_ASP.htm)>

Tecnología AJAX

- ✓ AJAX un nuevo acercamiento a las aplicaciones Web, Ene. 2010, monografias.com, < <http://www.monografias.com/trabajos43/ajax/ajax2.shtml#ajax> >
- ✓ Arquitectura de AJAX, Ene. 2010, <<http://faustol.wordpress.com/2007/12/12/arquitectura-ajax/>>
- ✓ Ajax, Ene. 2010, <<http://www.librosweb.es/ajax/capitulo1.html>>

Tecnología J2EE

- ✓ Investigación de la plataforma j2ee y su aplicación práctica, Ene. 2010, Juan Manuel Barrios Núñez, < <http://www.dcc.uchile.cl/~jbarrios/J2EE/MemoriaJ2EE.pdf> >
- ✓ Arquitectura empresarial y software libre, J2EE, Ene. 2010, < http://www.uhu.es/josel_alvarez/NvasTecnProg/recursos/articulo-j2ee.pdf >

Tecnología J2ME

- ✓ Implementación de un Sistema para la administración de servicios web en telefonía móvil a través de la plataforma j2me, Ene. 2010, Andrés Arley Rincón Pacheco, < <http://www.slideshare.net/arpa1986/aplicaciones-moviles-empresariales-con-servicios-web-y-j2me> >
- ✓ Análisis y Selección de Tecnologías, Nov.2010, <http://catarina.udlap.mx/u_dl_a/tales/documentos/lis/rivera_d_g/capitulo3.pdf>
- ✓ Programación de Aplicaciones Móviles, José Calderón, Dic.2010, < <http://www.slideboom.com/presentations/107714/04>>

Tecnología .NET

- ✓ La plataforma .NET, Feb. 2009, < <http://people.es.uchicago.edu/~borja/pubs/revistaeside2002.pdf> >
- ✓ Feb. 2009, nireblog, < <http://ange.nireblog.com>>
- ✓ Ventajas y Desventajas de .Net, Dic.2010, < http://www.qualitrain.com.mx/index2.php?option=com_content&do_pdf=1&id=14>
- ✓ Introducción a la Plataforma .NET, Dic.2010, < <http://img.redusers.com/imagenes/libros/lpcu113/capitulogratis.pdf> >

Información Software

- ✓ <http://pcd.ucuenca.edu.ec/pcd.xml>, Ciudad Digital, Abr. 2010
- ✓ http://es.wikipedia.org/wiki/Microsoft_Visual_Studio, Abr. 2010
- ✓ http://es.wikipedia.org/wiki/Microsoft_Visio, Abr. 2010
- ✓ http://es.wikipedia.org/wiki/Microsoft_Project, Abr. 2010
- ✓ http://es.wikipedia.org/wiki/Rational_Software, Abr. 2010

GLOSARIO DE TÉRMINOS

A.

ActiveX: Active X(control). Componente que se puede insertar en una página Web para proporcionar una funcionalidad que no está directamente disponible en HTML, como secuencias de animación. Los controles ActiveX se pueden implementar en diferentes lenguajes de programación.

Pequeños programas que permiten mostrar páginas web dinámicas en el PC y que suplen las limitaciones que, al respecto, tiene el lenguaje HTML. Los controles ActiveX tienen que descargarse al disco duro del ordenador para que los documentos que los utilizan puedan visualizarse.

Ancho de Banda: El ancho de banda es la máxima cantidad de datos que pueden pasar por un camino de comunicación en un momento dado, normalmente medido en segundos. Cuanto mayor sea el ancho de banda, más datos podrán circular por ella al segundo.

Característica de la línea telefónica que determina la cantidad de conexiones simultáneas que se pueden establecer entre los usuarios y el servidor. Cuando mayor sea el ancho de banda de la línea que ofrece un servidor, más usuarios podrán conectarse a la vez, y más rápida será la conexión.

Apache: Servidor web de código abierto. Su desarrollo comenzó en febrero de 1995, por Rob McCool, en una tentativa de mejorar el servidor existente en el NCSA. La primera versión apareció en enero de 1996, el Apache 1.0. Hacia el 2000, el servidor Web Apache era el más extendido en el mundo. El nombre «Apache» es un acrónimo de «a

patchy server» -un servidor de remiendos-, es decir un servidor construido con código preexistente y piezas y parches de código. Es la auténtica «kill app» del software libre en el ámbito de los servidores y el ejemplo de software libre de mayor éxito, por delante incluso del kernel Linux. Desde hace años, más del 60% de los servidores web de Internet emplean Apache.

Applet: Aplicaciones Java que se ejecutan en el contexto o que componen otro programa como por ejemplo una página Web.

Asíncrono: Que no tiene un intervalo de tiempo constante entre cada evento.

B.

Base de Datos: Conjunto de ficheros dedicados a guardar información relacionada entre sí, con referencias entre ellos de manera que se complementen con el principio de no duplicidad de datos.

C.

Cookies: Cookie (galleta) Cuando se visita una página Web, es posible recibir una Cookie. Este es el nombre que se da a un pequeño archivo de texto, que queda almacenado en el disco duro del ordenador. Este archivo sirve para identificar al usuario cuando se conecta de nuevo a dicha página Web.

D.

Diagrama Entidad-Relación: Denominado por sus siglas como: E-R; este modelo representa a la realidad a través de entidades, que son objetos que existen y que se distinguen de otros por sus características.

Dispositivo: Mecanismo para la obtención de un resultado automático.

Driver: Programa informático que permite al sistema operativo interactuar con un periférico, haciendo una abstracción del hardware y proporcionando una interfaz para usarlo. Se puede esquematizar como un manual de instrucciones que le indica cómo debe controlar y comunicarse con un dispositivo en particular. Por tanto, es una pieza esencial, sin la cual no se podría usar el hardware.

F.

Framework: Representa una arquitectura de software definida en la cual otro proyecto de software puede ser organizado y desarrollado.

H.

Hacker: Gente apasionada por la seguridad informática. Esto concierne principalmente a entradas remotas no autorizadas por medio de redes de comunicación como Internet

I.

Interface: Conexión que permite la comunicación entre dos o más dispositivos.

Internet: Red mundial de acceso a diversa información, que funciona a través de cables, vía satélite o microonda

Interoperabilidad: Habilidad que tiene un sistema o producto para trabajar con otros sistemas o productos sin un esfuerzo especial por parte del usuario.

J.

JavaScript: JavaScript no es un lenguaje de programación propiamente dicho. Es un lenguaje script u orientado a documento, como pueden ser los lenguajes de macros que tienen muchos procesadores de texto. Nunca podrás hacer un programa con JavaScript, tan sólo podrás mejorar tu página Web con algunas cosas sencillas (revisión de formularios, efectos en la barra de estado, etc...) y, ahora, no tan sencillas (animaciones usando HTML dinámico, por ejemplo).

M.

Middleware: Es un software de conectividad que ofrece un conjunto de servicios que hacen posible el funcionamiento de aplicaciones distribuidas sobre plataformas heterogéneas.

N.

Navegador Web: (Browser, explorador, navegador web). Aplicación que sirve para acceder a la WWW (todas las páginas web) y "navegar" por ella a través de los enlaces. Generalmente estos programas no sólo traen la utilidad de navegar por la WWW, sino que pueden también administrar correo, grupos de noticias, ingresar al servicio de FTP, etc.

Actualmente los navegadores más populares son Internet Explorer, Netscape, Opera y Firefox.

Notebook: (También llamada laptop o computadora portátil). Pequeña computadora personal con un peso entre 1 y 6 kilogramos, pensada para que pueda ser fácilmente transportable.

O.

Online: Estado de un usuario cuando está conectado a internet . En telecomunicaciones, on-line (en línea) se utiliza para designar a una computadora que está contactada al sistema, está operativa, está encendida o accede a internet.

Offline: Estado de un usuario cuando no está conectado a internet . En telecomunicaciones, off-line (fuera de línea) se utiliza para designar a una computadora que no está contactada al sistema, no está operativa, está apagada o no accede a internet.

P.

Performance: Desempeño con respecto al rendimiento de una computadora, un dispositivo, un sistema operativo, un programa o una conexión a una red.

Procedimiento: Conjunto de instrucciones, controles, etc. Que hacen posible la resolución de una cuestión específica.

Plug-in: En computación , **un plug-in** (o plugin) es un conjunto de componentes de software que agrega capacidades específicas a una mayor aplicación de software.

S.

Script: Grupo de lenguajes de programación que son típicamente interpretados y pueden ser tipeados directamente desde el teclado.

Los scripts son un conjunto de instrucciones generalmente almacenadas en un archivo de texto que deben ser interpretados línea a línea en tiempo real para su ejecución, se distinguen de los programas, pues deben ser convertidos a un archivo binario ejecutable para correrlos.

Los scripts pueden estar embebidos en otro lenguaje para aumentar las funcionalidades de este, como es el caso los scripts PHP o Javascript en código HTML.

Servlet: Tecnología Java que consta de un mecanismo consistente para la extensión de funcionalidades de un Servidor Web.

Síncrono: Que tiene un intervalo de tiempo constante entre cada evento. Son procesos síncronos los que dependen de un acontecimiento externo que los dispara.

Sincronización: Sincronizar se refiere a que dos o más elementos, eventos u operaciones sean programadas para que ocurran en un momento predefinido de tiempo o lugar.

En informática, sincronizar hace referencia a la coordinación de procesos que se ejecutan simultáneamente para completar una tarea, con el fin de obtener un orden de ejecución correcto y evitar así estados inesperados.

W.

Web: Es la parte de Internet a la que se accede a través del protocolo HTTP y en consecuencia gracias a exploradores normalmente gráficos como Netscape y Internet Explorer.

ABREVIATURAS UTILIZADAS

A.

ADO: Activex Data Object

AJAX: Asynchronous JavaScript and XML. (JavaScript asíncrono y XML)

API: Application Programming Interface. (Interfaz de Programación de Aplicaciones.)

ASP: Active Server Pages. (Páginas Activas en el Servidor)

AWT: Abstract Window Toolkit. (Herramienta para Ventanas Abstracta)

C.

CDC: Configuración de Dispositivos con Conexión.

CISC: Complex Instruction Set Computer. (Computadoras con Conjuntos de Instrucciones Complejas.)

CLDC: Configuración de Dispositivos Limitados con Conexión.

CLR: Common Language Runtime. (Entorno de Ejecución)

CPU: Central Processing Unit (Unidad de Proceso Central)

CVM: Compact Virtual Machine

D.

DOM: Document Object Model (Modelo de Objeto de Documento)

DNS: Domain Name System. (Sistema de Nombre de Dominio)

E.

EIS: Enterprise Information System

EJB: Enterprise JavaBeans. (Componentes Empresariales de Java.)

F.

FTP: File Transfer Protocol. (Protocolo de Transferencia de Archivos.)

H.

HTTPS: Hypertext Transfer Protocol Secure. (Protocolo de Transferencia de Hipertexto Seguro.)

HTML: HyperText Markup Language (Lenguaje de Marcado de Hipertexto)

I.

IDEs: Integrated Development Environment. (Entorno de Desarrollo Integrado)

IIS: Internet Information Server

J.

JAF: JavaBeans Activation Framework

JavaIDL: Java Interface Definition Language

JAXP: API Java para el Procesamiento de XML.

JCP: Comunidad de Procesos Java.

JDBC: Conectividad de Bases de Datos Java.

JDK: Java Development Kit. (Que incluye, el lenguaje java, herramientas de desarrollo, APIs, JRE)

JIT: Just In Time. (Método Justo a Tiempo)

J2EE: Java 2 Enterprise Edition. (Java 2 Edición Empresarial.)

J2ME: Java 2 Edición Micro.

JMS: Java Message Service

JNDI: Java Naming and Directory Interface

JNI: Interfaz Nativa de Java

JRE: Java Runtime Environment. (Que incluye librerías principales, plugins, JVM.)

J2SE: Java 2 Standard Edition. (Java 2 Edición Estándar.)

JSP: Java Server Pages. (Tecnología basada en Java que permite la generación de contenido dinámico para páginas Web.)

JVM: Java Virtual Machine. (El compilador de java genera bytecodes, los cuales son interpretados por una máquina virtual, esta máquina virtual procesa los bytecodes sobre una arquitectura específica.)

JVMCI: Interfaz de Depuración de la Máquina Virtual

M.

MIDP: Mobile Information Device Profile. (Perfil de Información del Dispositivo Móvil.)

MIDlets: Aplicación J2ME desarrollada bajo la especificación MIDP.

MMAPI: Mobile Media API (API de Contenido Multimedia Móvil).

MSIL: Microsoft Intermediate Language. (Lenguaje Intermedio de Microsoft)

MVC: Modelo Vista Controlador.

O.

ODBC: Open DataBase Connectivity. (Estándar de acceso a Bases de datos)

P.

PDA: Personal Digital Assistant. (Asistente Digital Personal)

PHP: Hypertext Pre-processor. (Inicialmente PHP Tools, o, Personal Home Page Tools).

POO: Programación Orientada a Objetos

PWS: Personal Web Server

R.

RAM: Random Access Memory (Memoria de Acceso Aleatorio)

RISC: Computadoras con Conjuntos de Instrucciones Reducida.

RMI: Remote Method Invocation (Invocación de Métodos Remotos)

RMI-IIOP: Remote Method Invocation – Internet Inter ORB

RMS: Record Management Storage. (Manejo de Almacenamiento de Registros.)

RPC: Llamado a Procedimientos Remotos.

ROM: Read-only Memory (Memoria de sólo Lectura)

S.

SAX: API Simple para XML.

SDK: Kit de Desarrollo Estándar.

SIP: Session Initiation de Protocol. (Protocolo de Iniciación de Sesión)

SMS: Short Message Service

SMTP: Protocolo Simple de Transferencia de Correo.

SO: Sistema Operativo

SOAP: Protocolo de Acceso Simple a Objetos.

SQL: Structure Query Language. (Lenguaje estructurado de Consulta)

U.

URL: Uniform Resource Locator. (Localizador de Recurso Uniforme)

W.

WAP: Wireless Application Protocol. (Protocolo de Aplicaciones Inalámbricas)

WCF: Windows Communication Foundation.

WMA: Wireless Messaging API (API de Mensajería Inalámbrica).

WSA: Web Services API (Especificación de Servicios Web en J2ME)

W3C: Consorcio World Wide Web

X.

XHTML-MP: Hypertextual Computer Language Standard. (Hipertextual Lenguaje Informático Estandar diseñado específicamente para teléfonos móviles).

XML: eXtensible Markup Language (Lenguaje de Marcas Extensible)

XSLT: Extensible Stylesheet Language Transformation (Lenguaje de Transformación basado en Hojas de Estilo).

ANEXOS

AMW-EIS

APLICACIÓN MÓVIL WEB

“REGISTRO DE CALIFICACIONES Y ASISTENCIA DE LOS ALUMNOS DE LA ESCUELA DE INGENIERÍA DE SISTEMAS DE LA P.U.C.E.S.A APLICADA AL AÑO ACADÉMICO 2010”



CONTENIDO

INTRODUCCIÓN	160
REQUERIMIENTOS DE HARDWARE Y SOFTWARE	160
USO DEL SISTEMA	160
1. Ingresar al Sistema de Registro de Calificaciones y Asistencia	160
2. Opciones del Sistema	162
2.1. Cambiar Password	163
2.2. Seleccionar el Paralelo-Materia	164
2.2.1 Seleccionar el Parcial	164
2.2.2 Registro de Notas y Faltas	165
CONSIDERACIONES	166

ÍNDICE FIGURAS

Figura 1. Dirección Web	160
Figura 2. Pantalla de Presentación de la aplicación	161
Figura 3. Inicio de Sesión	161
Figura 4. Pantalla de Ingreso de datos	162
Figura 5. Menú Niveles	163
Figura 6. Cambiar Password	163
Figura 8. Menú Parcial	164
Figura 9. Pantalla para el Registro de notas y faltas	165

INTRODUCCIÓN

El presente documento es una guía que explica sobre el funcionamiento de la aplicación móvil web y tiene como objetivo otorgar soporte a los usuarios, proporcionando información necesaria para el uso de la página web a través del uso de un dispositivo móvil.

REQUERIMIENTOS DE HARDWARE Y SOFTWARE

- Dispositivo Móvil (celular, PDA, notebook, etc).
- Navegador de Internet (Internet Explorer)
- Conexión a Internet WI-FI

USO DEL SISTEMA

El Manual está organizado de acuerdo a la secuencia de ingreso a las pantallas de la aplicación móvil web, la cual se detalla a continuación cada una:

1. Ingresar al Sistema de Registro de Calificaciones y Asistencia

Abra el navegador y en la barra de direcciones escriba <http://localhost/SampleASP/default.aspx> y presione Enter.

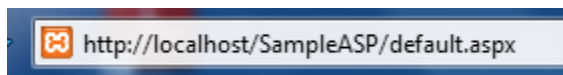


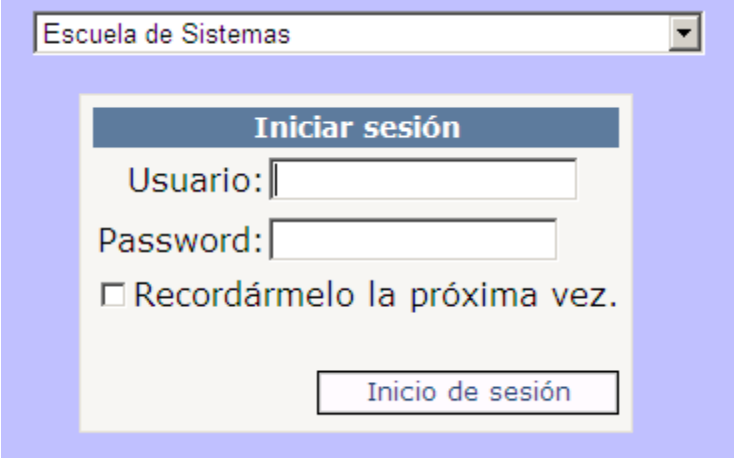
Figura 1. Dirección Web

Al visualizar el sitio Web de inicio, seleccionar la opción Ir.



Figura 2. Pantalla de Presentación de la aplicación

Para ingresar al sistema de registro, el docente primero debe seleccionar la Escuela (Escuela de Sistemas).



Escuela de Sistemas


Iniciar sesión

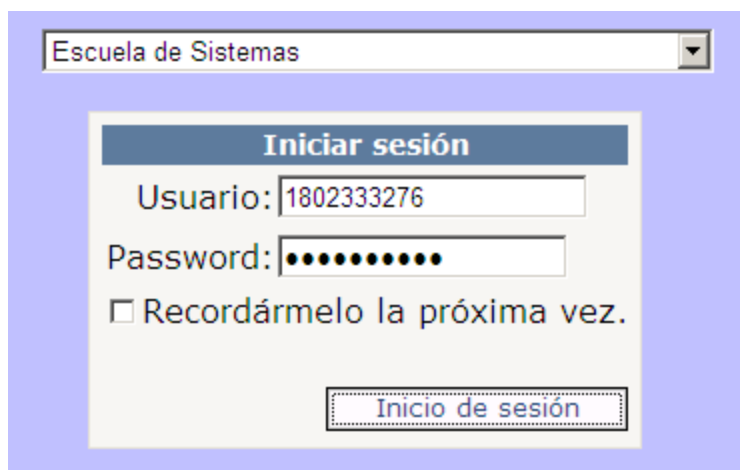
Usuario:

Password:

Recordármelo la próxima vez.

Figura 3. Inicio de Sesión

Posteriormente se debe identificar con su nombre de **Usuario** y **Password**. Ingresar la información solicitada y presionar sobre el botón .



Escuela de Sistemas

Iniciar sesión

Usuario: 1802333276

Password: ●●●●●●●●

Recordármelo la próxima vez.




Figura 4. Pantalla de Ingreso de datos

La primera vez que se va a iniciar la sesión es necesario que el nombre de usuario y el password hayan sido otorgados previamente por el administrador, persona encargada de agregar usuarios al sistema.

2. Opciones del Sistema

Una vez ingresado al sistema se desplegará un menú.

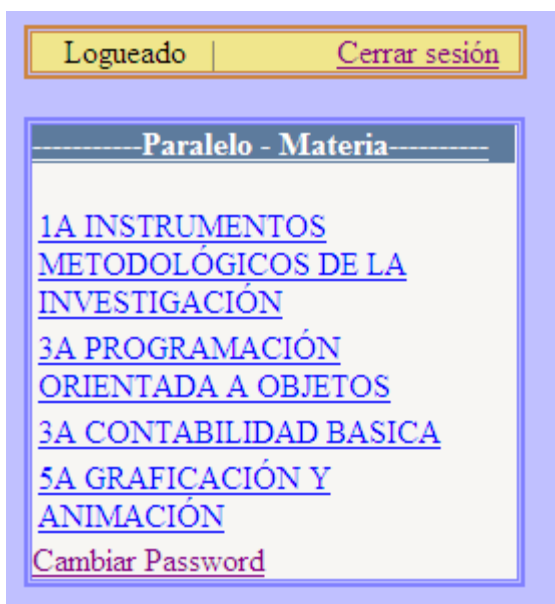


Figura 5. Menú Niveles

2.1. Cambiar Password

La opción de **cambio de password**, lo que permite al docente restablecer una nueva contraseña.

Logeado | Cerrar sesión

Cambiar Password

Nuevo password:

Confirmar password:

[Paralelo Materia](#)

Figura 6. Cambiar Password

2.2. Seleccionar el Paralelo-Materia

Seleccionar el nivel de alumnos del cual vaya a realizar el registro de notas y faltas.

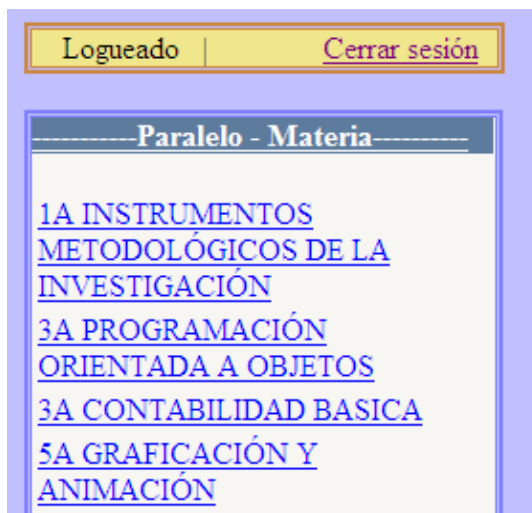


Figura 7. Menú de Selección de Nivel

2.2.1 Seleccionar el Parcial

Una vez seleccionado el paralelo-materia se accede a una ventana, donde se debe seleccionar el parcial.

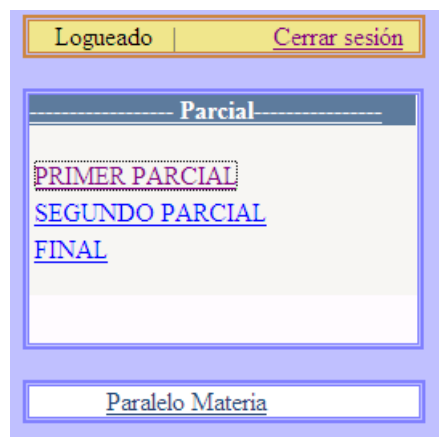


Figura 8. Menú Parcial

2.2.2 Registro de Notas y Faltas

Al haber seleccionado el nivel-materia y posteriormente el parcial se desplegará la ventana que presentara el siguiente formulario para realizar el registro del número de faltas y la nota de cada alumno.



Logueado | [Cerrar sesión](#)

3A CONTABILIDAD BASICA---PRIMER PARCIAL

REGISTRO ALUMNO

Apellido: ALTAMIRANO
IBARRA

Nombre: GABRIEL
EDUARDO

Nota:

Falta:

[Paralelo Materia](#) [Parcial](#)

Figura 9. Pantalla para el Registro de notas y faltas

Los campos que intervienen en esta pantalla se especifican a continuación:


Apellido: Apellido del alumno.

Nombre: Nombre del alumno.

Nota: Nota del alumno

Falta: Número de faltas del alumno.

Ingresada toda la información solicitada se debe presionar en el botón

, el cual despliega un mensaje si se guardo o no correctamente la información ingresada.

CONSIDERACIONES

1. Cada docente es responsable de su clave. Él cual puede cambiarla cuando lo desee.
2. Los campos para el registro no deben ir en blanco.
3. Al concluir con el registro, presionar Cerrar Sesión.

APLICACIÓN MÓVIL WEB

“REGISTRO DE CALIFICACIONES Y ASISTENCIA DE LOS
ALUMNOS DE LA ESCUELA DE INGENIERIA DE SISTEMAS
DE LA P.U.C.E.S.A APLICADA AL AÑO ACADEMICO 2010”



MANUAL DE INSTALACIÓN

TABLA DE CONTENIDO

1.	<i>INSTALACIÓN DEL SERVIDOR WEB APACHE</i>	170
2.	<i>INSTALACIÓN DEL MODULO DE ASP.NET 2 PARA APACHE 2.2</i>	176
2.1.	CONFIGURACIÓN DEL SERVIDOR WEB APACHE PARA EJECUTAR PÁGINAS ASP.NET	180

TABLA DE FIGURAS

Figura 1. Programa de instalación xampp-win32-1.6.2-installer.exe	170
Figura 2. Pantalla para elegir el idioma del programa	170
Figura 3. Recomendaciones previo a la Instalación	171
Figura 4. Ubicación física del programa	171
Figura 5. Pantalla para elegir el Servicio Apache	172
Figura 6. Progreso de instalación	172
Figura 7. Proceso de Configuración	173
Figura 8. Fin de la Instalación	173
Figura 9. Iniciándose el servicio de Apache 2.2	174
Figura 10. Completado el servicio de Intalación	174
Figura 11. Instalación completada	175
Figura 12. Panel de Control de XAMPP	175
Figura 13. Aplicación .exe	176
Figura 14. Pantalla inicial del proceso de instalación	176
Figura 15. Términos de la Licencia	177
Figura 16. Selección del directorio físico	177
Figura 17. Inicio de la Instalación	178
Figura 18. Proceso de Instalación	178
Figura 19. Finalización de la instalación del módulo MOD_ASPDOTNET para Apache	179
Figura 20. Ubicación física de la aplicación	181
Figura. 21. Panel de Control	182
Figura 22. Pantalla de Inicio de la Aplicación	182

1. INSTALACIÓN DEL SERVIDOR WEB APACHE

Para la instalación del servidor apache se debe seguir los siguientes pasos:

1. Ejecutamos el instalador XAMPP al dar doble clic.

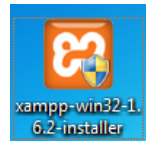


Figura 1. Programa de instalación xampp-win32-1.6.2-installer.exe

Se iniciará el asistente de instalación.

2. El asistente iniciará la pantalla para la selección del idioma.

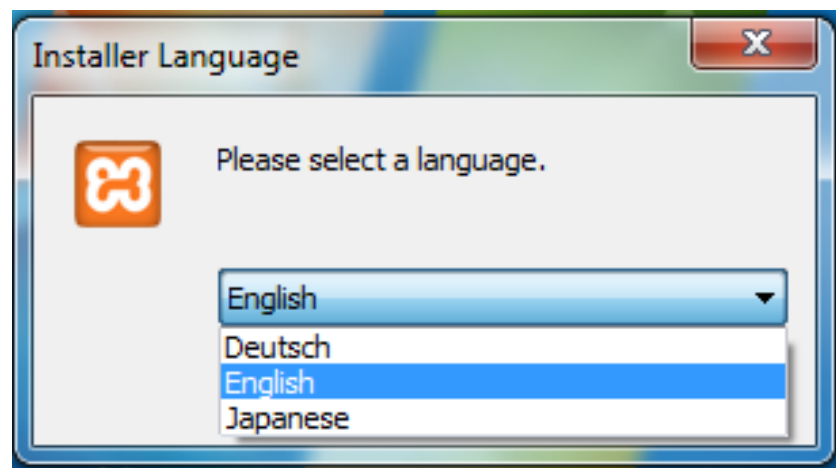


Figura 2. Pantalla para elegir el idioma del programa

3. Recomendaciones previas a la instalación.



Figura 3. Recomendaciones previo a la Instalación

4. Seleccionar la ubicación en la cual se desea instalar, clic en Next

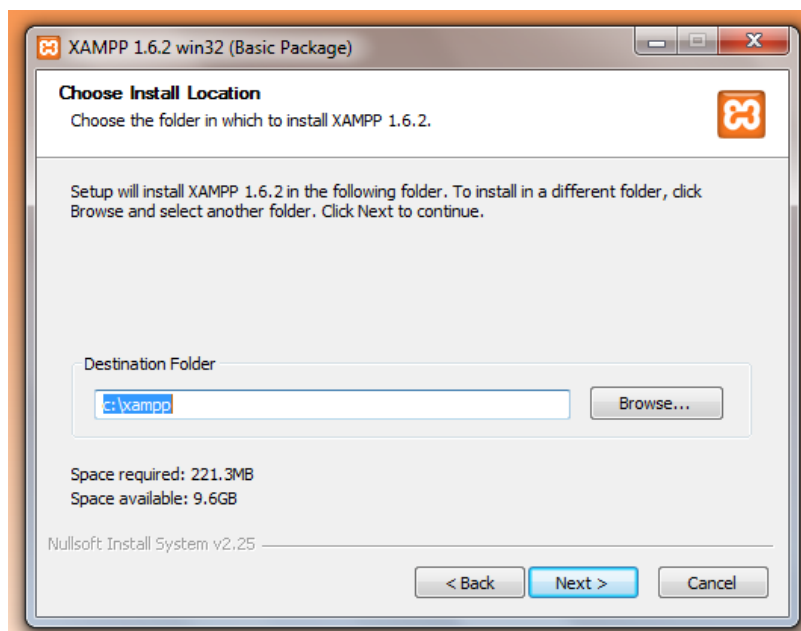


Figura 4. Ubicación física del programa

5. Elegir Install Apache as service, click en Install.

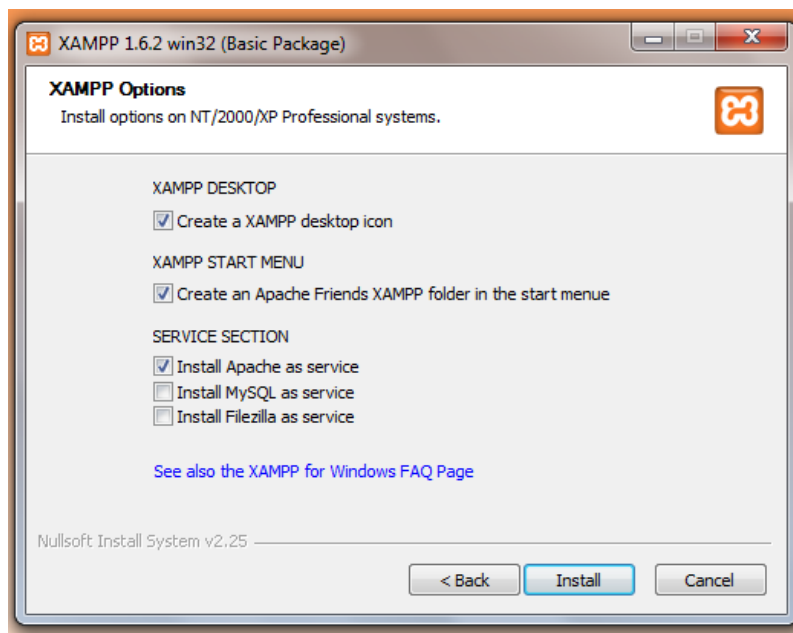


Figura 5. Pantalla para elegir el Servicio Apache

Se realiza la instalación de los componentes del Servidor Apache.

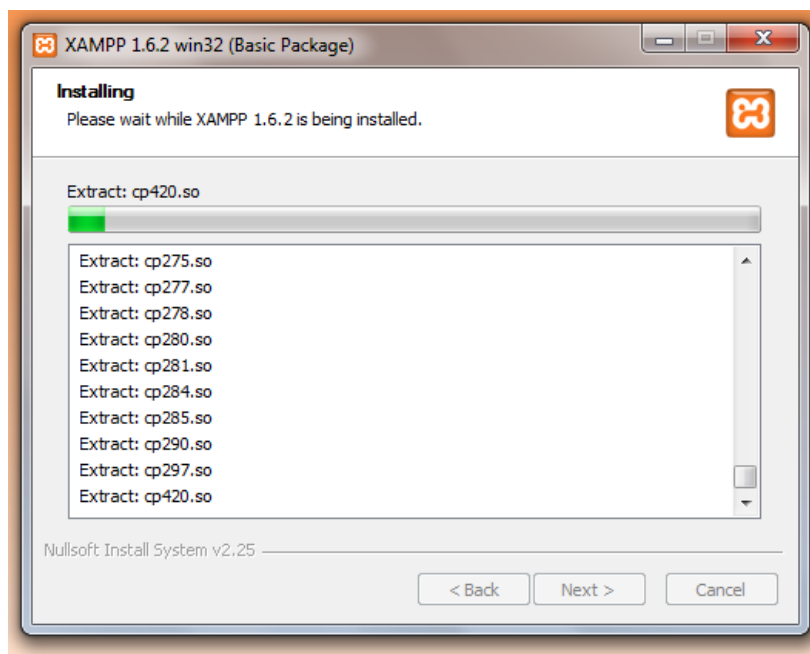
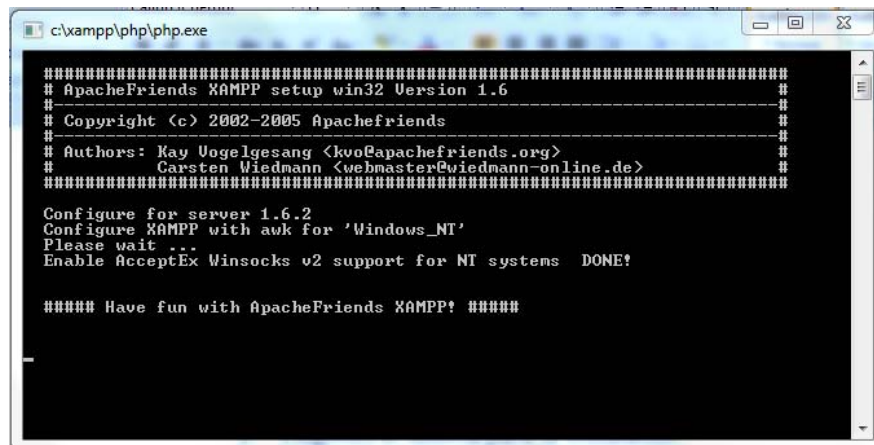


Figura 6. Progreso de instalación

6. Proceso de configuración



```
c:\xampp\php\php.exe
#####
# ApacheFriends XAMPP setup win32 Version 1.6
# Copyright (c) 2002-2005 Apachefriends
# Authors: Kay Vogelgesang <kvo@apachefriends.org>
#           Carsten Wiedmann <wehmaster@wiedmann-online.de>
#####
Configure for server 1.6.2
Configure XAMPP with awk for 'Windows_NT'
Please wait ...
Enable AcceptEx Winsocks v2 support for NT systems  DONE!

##### Have fun with ApacheFriends XAMPP! #####
```

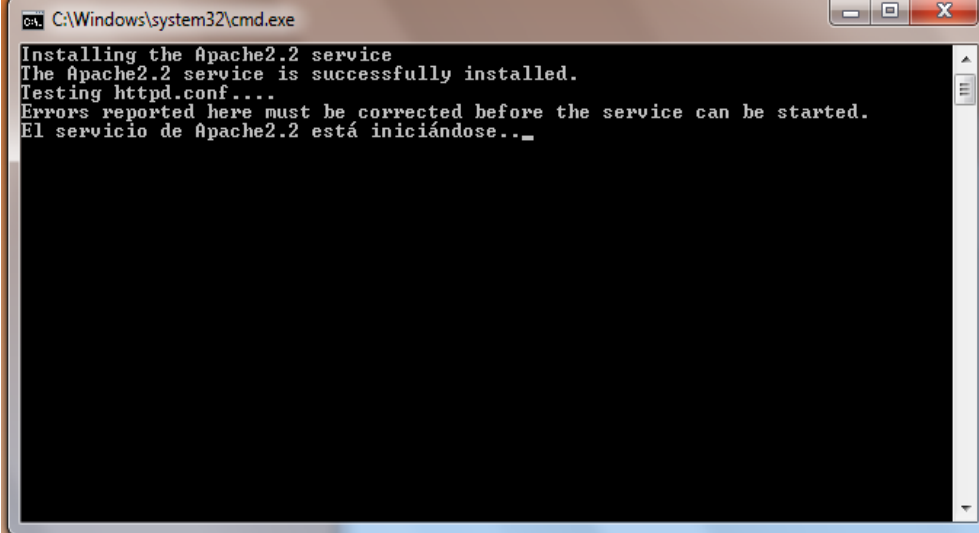
Figura 7. Proceso de Configuración

7. Terminada la instalación seleccionar Finish.



Figura 8. Fin de la Instalación

8. Reporte de inicialización del Servidor Apache2.2



```
C:\Windows\system32\cmd.exe
Installing the Apache2.2 service
The Apache2.2 service is successfully installed.
Testing httpd.conf...
Errors reported here must be corrected before the service can be started.
El servicio de Apache2.2 está iniciándose.._
```

Figura 9. Iniciándose el servicio de Apache 2.2

9. El servicio de instalación a finalizado, clic en Aceptar.

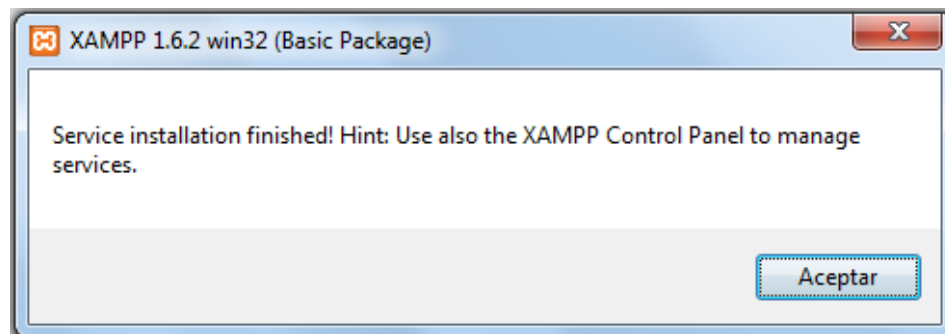


Figura 10. Completado el servicio de Instalación

10. Acceder al Panel de Control de la Aplicación XAMPP, para comprobar el estado actual del servidor Apache, clic en Sí.

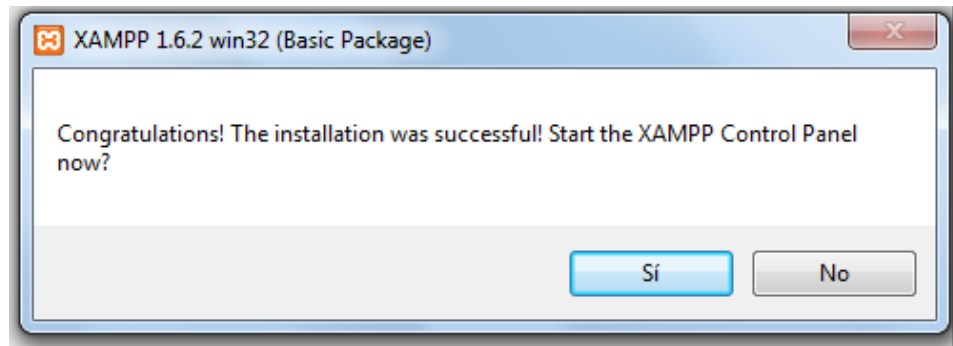


Figura 11. Instalación completada

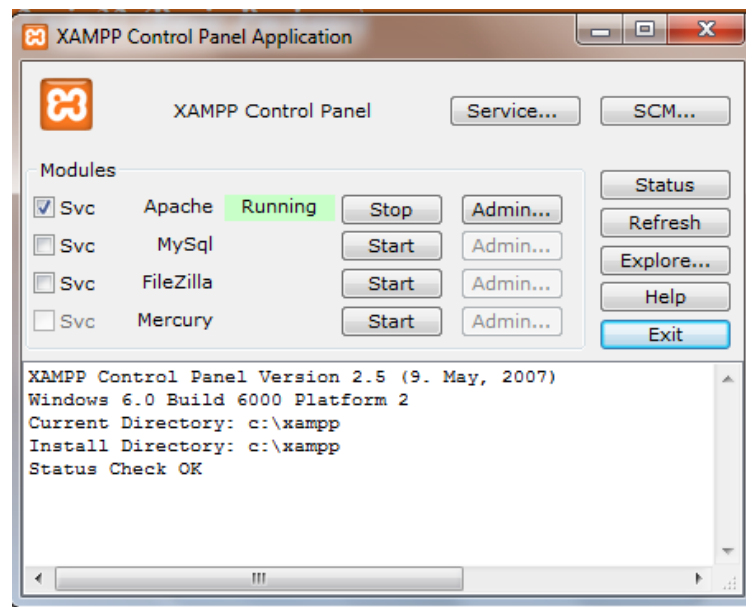


Figura 12. Panel de Control de XAMPP

Actualmente el servidor Apache está levantado.

2. INSTALACIÓN DEL MODULO DE ASP.NET 2 PARA APACHE 2.2

Para que las paginas elaboradas en ASP.NET 2 se ejecuten utilizando el servidor Web Apache, es necesario instalar el modulo mod_aspdotnet 2.2. Los siguientes pasos se deben seguir para su instalación y configuración:

1. Inicializar la instalación al dar doble clic sobre el instalador mod_aspdotnet 2.2

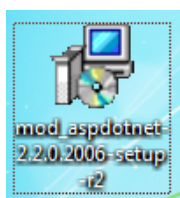


Figura 13. Aplicación .exe

2. Se inicializa la intalacion, clic en Next para continuar.

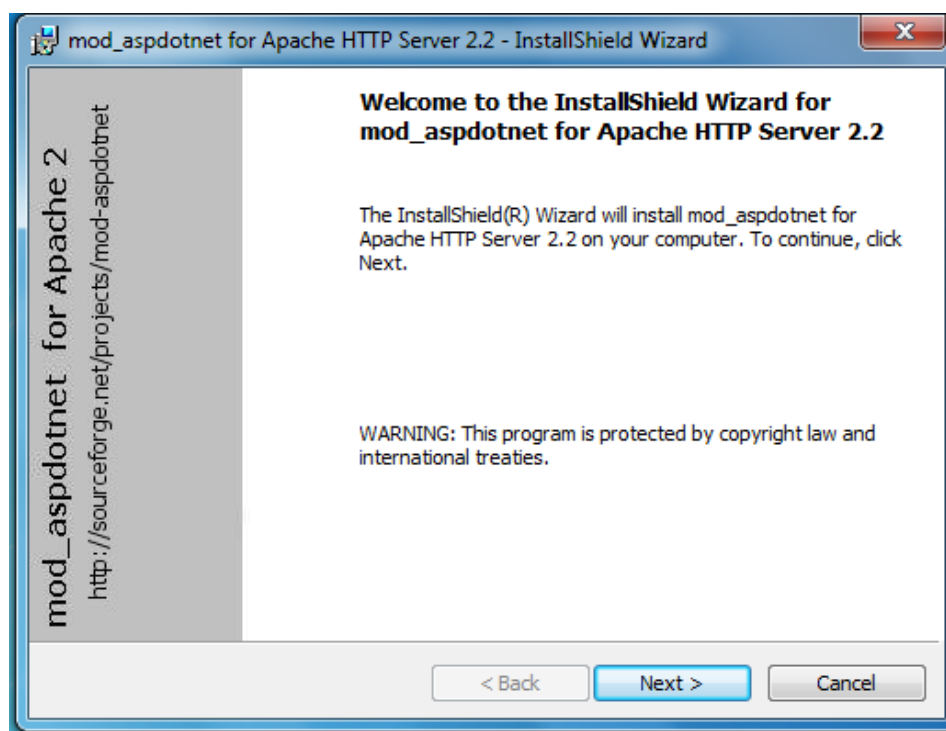


Figura 14. Pantalla inicial del proceso de instalación

3. Elegir la opción Aceptar los términos de la licencia, clic en Next.

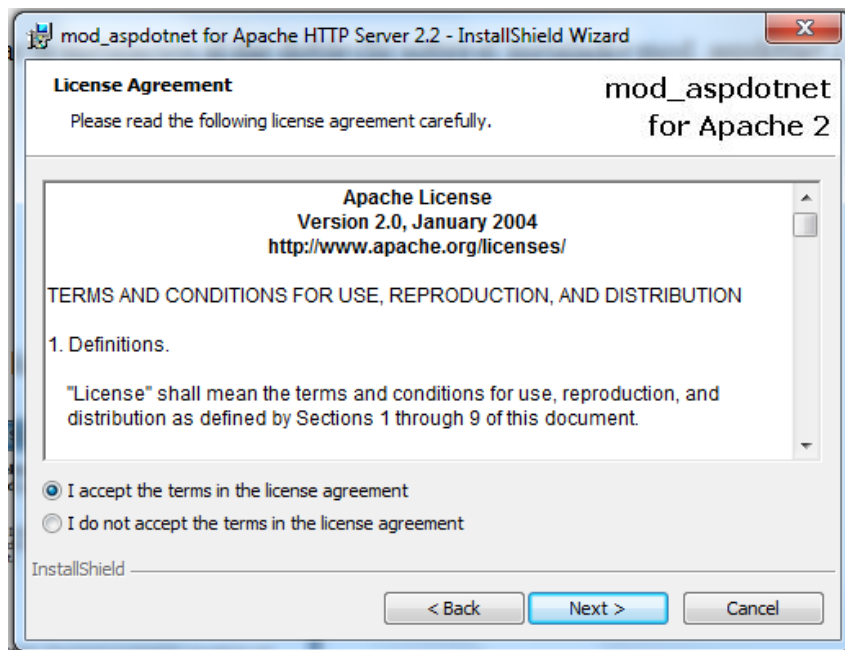


Figura 15. Términos de la Licencia

4. Seleccionar el directorio por defecto, clic en Next

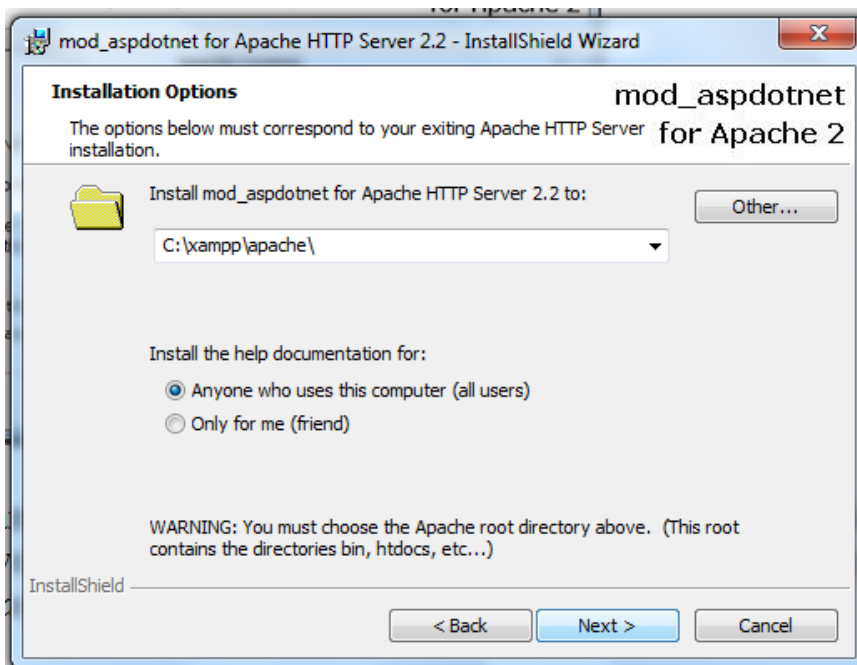


Figura 16. Selección del directorio físico

5. Seleccionar Install para que el asistente empiece la instalación

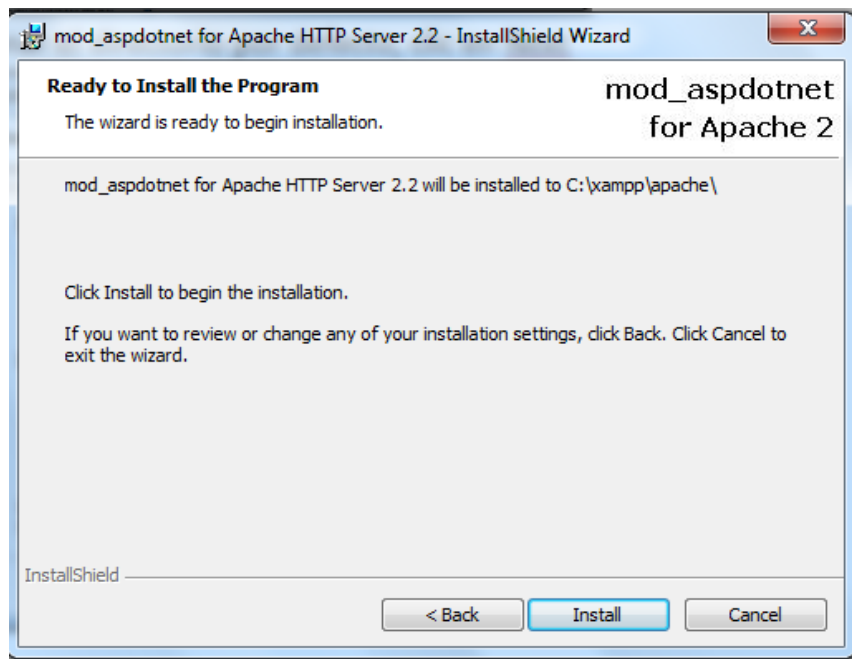


Figura 17. Inicio de la Instalación

6. El asistente empezara el proceso de instalación

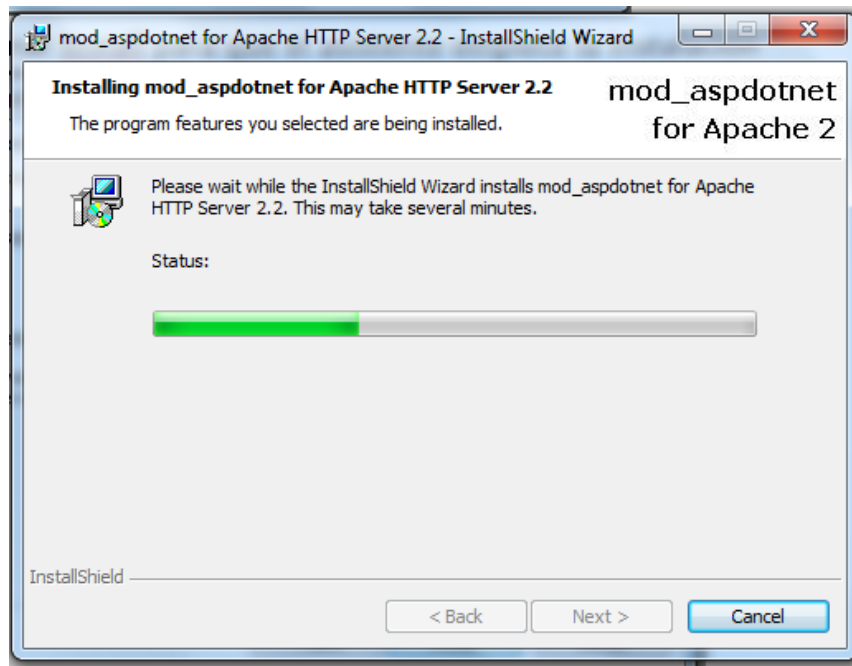


Figura 18. Proceso de Instalación

7. Al concluir la instalación dar clic en Finish.

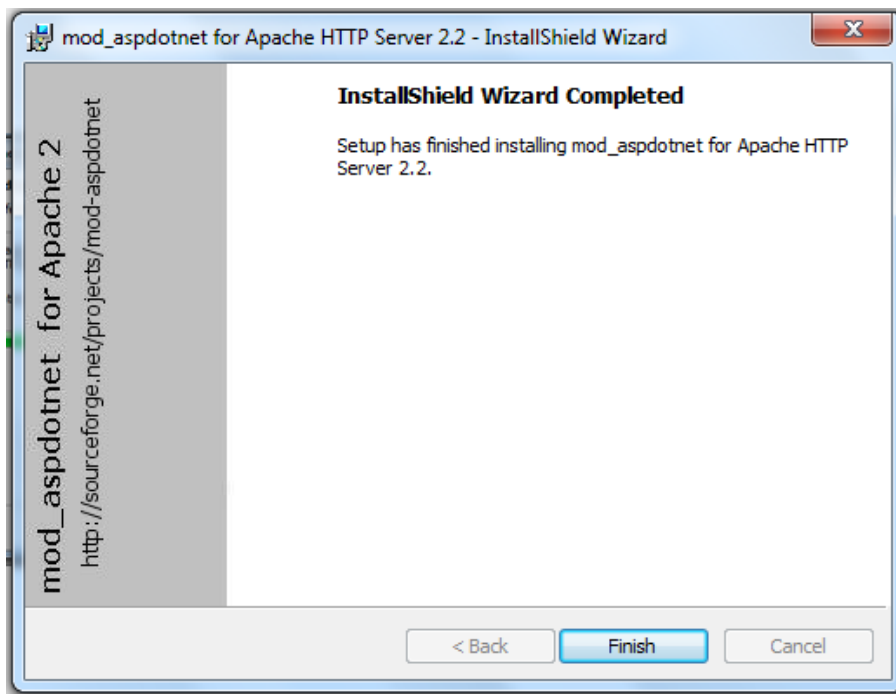


Figura 19. Finalización de la instalación del módulo MOD_ASPDOTNET para Apache

2.1. CONFIGURACIÓN DEL SERVIDOR WEB APACHE PARA EJECUTAR PÁGINAS ASP.NET

1. Añadir al final de C:\xampp\apache\conf\httpd.conf

```
#asp.net
LoadModule aspdotnet_module "modules/mod_aspdotnet.so"

AddHandler asp.net asax ascx ashx asmx aspx axd config cs csproj licx rem resources resx soap vb vbproj
        vsdisco webinfo

<IfModule mod_aspdotnet.cpp>
    # Mount the ASP.NET /asp application
    AspNetMount /SampleASP "c:/xampp/htdocs/SampleASP"
    #/SampleASP is the alias name for asp.net to execute
    #"c:/xampp/htdocs/SampleASP" is the actual execution of files/folders in that location

    # Map all requests for /asp to the application files
    Alias /SampleASP "c:/xampp/htdocs/SampleASP"
    #maps /SampleASP request to "c:/xampp/htdocs/SampleASP"
    #now to get to the /SampleASP type http://localhost/SampleASP
    #It'll redirect http://localhost/SampleASP to "c:/xampp/htdocs/SampleASP"

    # Allow asp.net scripts to be executed in the /SampleASP example
    <Directory "c:/xampp/htdocs/SampleASP">
        Options FollowSymlinks ExecCGI
        Order allow,deny
        Allow from all
        DirectoryIndex index.htm index.aspx
        #default the index page to .htm and .aspx
    </Directory>

    # For all virtual ASP.NET webs, we need the aspnet_client files
```

to serve the client-side helper scripts.

```
AliasMatch /aspnet_client/system_web/(\d+)_(\d+)_(\d+)_(\d+)/(.*)
    "C:/Windows/Microsoft.NET/Framework/v$1.$2.$3/ASP.NETClientFiles/$4"
<Directory "C:/Windows/Microsoft.NET/Framework/v*/ASP.NETClientFiles">
    Options FollowSymlinks
    Order allow,deny
    Allow from all
</Directory>
</IfModule>
#asp.net
```

2. Ubicar la aplicación móvil Web “SampleASP” en la siguiente dirección:

C:\xampp\htdocs

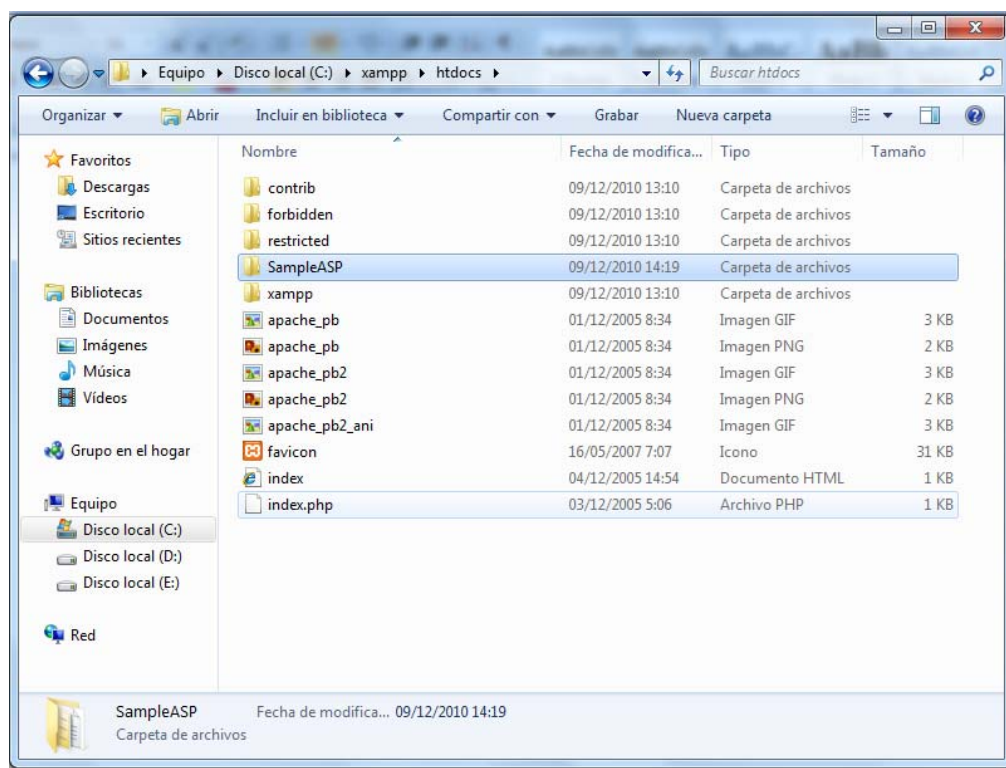


Figura 20. Ubicación física de la aplicación

- Reiniciar el servidor Apache.

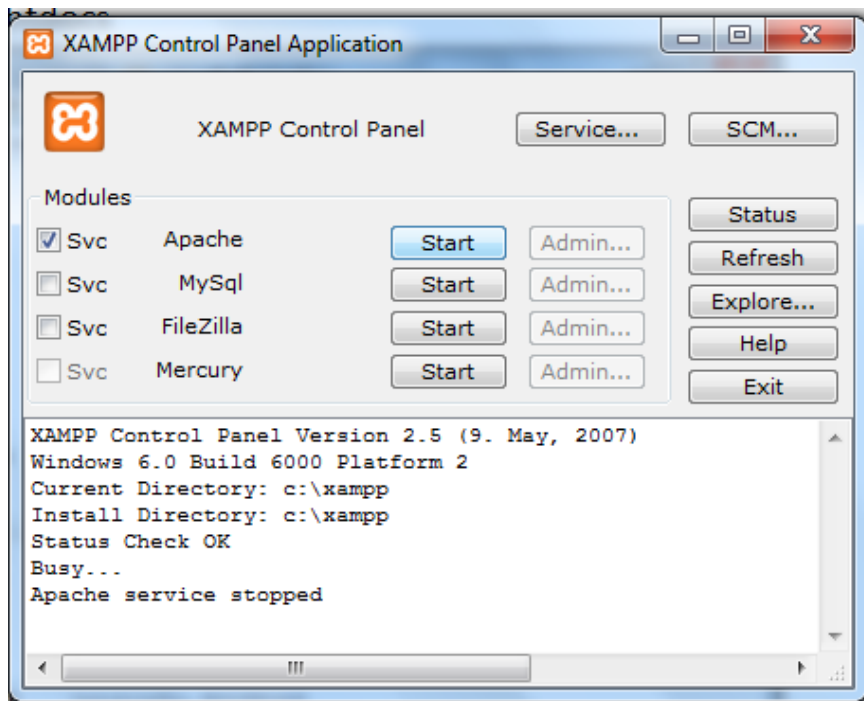


Figura. 21. Panel de Control

- Iniciar Internet Explorer, y digitar la dirección:
<http://localhost/SampleASP/default.aspx> desde el servidor



Figura 22. Pantalla de Inicio de la Aplicación