

**UNIVERSIDAD NACIONAL DEL ALTIPLANO
ESCUELA DE POST GRADO
MAESTRÍA EN INFORMÁTICA**



**“SISTEMA DE CONSULTA DE NOTAS EN MÓVILES CON JAVA MICRO
EDITION DE LOS ESTUDIANTES DE LA ESCUELA PROFESIONAL DE
INGENIERÍA DE SISTEMAS DE LA UNIVERSIDAD NACIONAL
DEL ALTIPLANO - PUNO 2010”**

TESIS

PRESENTADA POR :

ZULEMA LILIAN MAMANI HUACANI

PARA OPTAR EL GRADO ACADÉMICO DE:
**MAGÍSTER SCIENTIAE EN INFORMÁTICA
MENCIÓN: INFORMÁTICA EDUCATIVA**



PUNO - PERÚ

2012

UNIVERSIDAD NACIONAL DEL TILAND • PUNO
BIBLIOTECA GENERAL
AREA DE TESIS
Fecha Ingreso: 21 OCT 2014
No. 100722

UNIVERSIDAD NACIONAL DEL ALTIPLANO
ESCUELA DE POST GRADO
MAESTRÍA EN INFORMÁTICA



TESIS

**“SISTEMA DE CONSULTA DE NOTAS EN MÓVILES CON JAVA MICRO
EDITION DE LOS ESTUDIANTES DE LA ESCUELA PROFESIONAL DE
INGENIERÍA DE SISTEMAS DE LA UNIVERSIDAD NACIONAL DEL
ALTIPLANO - PUNO 2010”**

PRESENTADA POR:

ZULEMA LILIAN MAMANI HUACANI

PARA OPTAR EL GRADO ACADÉMICO DE:

MAGÍSTER SCIENTIAE EN INFORMÁTICA

MENCIÓN: INFORMÁTICA EDUCATIVA

PUNO – PERÚ

2012

UNIVERSIDAD NACIONAL DEL ALTIPLANO
ESCUELA DE POST GRADO
MAESTRÍA EN INFORMÁTICA

**“SISTEMA DE CONSULTA DE NOTAS EN MÓVILES CON JAVA MICRO
EDITION DE LOS ESTUDIANTES DE LA ESCUELA PROFESIONAL DE
INGENIERÍA DE SISTEMAS DE LA UNIVERSIDAD NACIONAL DEL
ALTIPLANO - PUNO 2010”**

TESIS PRESENTADA POR:

ZULEMA LILIAN MAMANI HUACANI

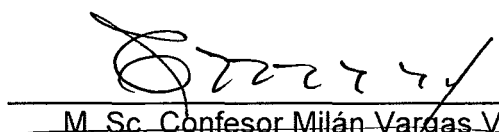
PARA OPTAR EL GRADO ACADÉMICO DE:

MAGÍSTER SCIENTIAE EN INFORMÁTICA

MENCIÓN: INFORMÁTICA EDUCATIVA.

APROBADO POR EL JURADO REVISOR:

PRESIDENTE

: 
M. Sc. Confesor Milán Vargas Valverde

PRIMER MIEMBRO

: 
M. Sc. Edgar Eloy Carpio Vargas

SEGUNDO MIEMBRO

: 
M. Sc. Samuel Pérez Quispe

ASESOR DE TESIS

: 
M. Sc. Edelfré Flores Velásquez

DEDICATORIA

Mi tesis la dedico con todo mi amor y cariño a Dios que me brinda la oportunidad de darme una familia maravillosa.

A mis Padres Mario y Berna por su amor incondicional, su sacrificio y su apoyo en todo momento desde el inicio de mis estudios de Maestría y por creer en mí.

A mi hermano Mario Wilmer y su esposa Jakeline gracias por apoyarme siempre.

A mis hijos: Klemems Neyder, Kazandra Neyla y Katherine Nardy a quienes quiero ofrecerles este trabajo por todas las horas que tuve que quitarles para poder estudiar y porque son la motivación de ser cada día mejor.

AGRADECIMIENTOS

A Dios por la vida y por haber permitido obtener cada uno de mis logros en la vida.

A la Universidad Nacional del Altiplano, la Escuela de Post Grado y en especial a la Maestría en Informática.

A los docentes, quienes compartieron sus conocimientos conmigo y se involucraron en mi formación haciendo posible la conclusión de esta tesis.

A los miembros de jurado, al M. Sc. Confesor Milán Vargas Valverde, al M. Sc. Edgar Eloy Carpio Vargas y al M. Sc. Samuel Pérez Quispe por sus valiosas contribuciones que hicieron al trabajo final y por su comprensión.

A mi asesor de tesis M. Sc. Edelfré Flores Velásquez, por la su asesoría, sus conocimientos y recomendaciones respecto a esta investigación.

A mi hermano y cuñada que siempre me desearon lo mejor.

A mis padres, quienes sin escatimar esfuerzo se han sacrificado para mi formación y educación y a mis adorados hijos por su comprensión

ÍNDICE

RESUMEN

ABSTRACT

Introducción

CAPITULO I

PROBLEMA DE INVESTIGACION

1. Problema de investigación	1
1.1. Planteamiento y Definición del problema	1
1.2. Formulación del problema	3
1.3. Objetivos de la investigación	4
1.3.1. Objetivo General	4
1.3.2. Objetivos Específicos	4
1.4. Hipótesis de la investigación	4

CAPITULO II

MARCO TEORICO

2.1. Antecedentes de la investigación	5
2.2. Marco Teórico	12
2.2.1. Comunicaciones Móviles	12
2.2.2. Telefonía Móvil	13
2.2.3. Acceso a Internet	14
2.2.4. Protocolo HTTP	14
2.2.5. Java	16
2.2.6. Java Micro Edition (JME)	18
2.2.6.1. Máquina Virtual JME	20
2.2.6.2. Configuraciones	21
2.2.6.3. Perfiles	24

2.2.7. Servlets	28
2.2.8. Midlet	28
2.2.8.1. El Gestor de Aplicaciones AMS	29
2.2.8.2. Ciclo de vida de un Midlet	29
2.2.8.3. ADO : Data Transfer Objet	31
2.2.9. Base de Datos	32
2.2.10. Servidor de Aplicaciones	36
2.2.11. Emulador para dispositivos inalámbricos	38
CAPITULO III	
METODOLOGIA	
3.1. Localización	39
3.2. Determinación de la muestra	39
3.2.1. Población de estudiantes	39
3.2.2. Determinación del tamaño de muestra	40
3.3.Opercionalización de variables	42
3.4.Métodos	43
3.4.1. Tipo de Investigación	43
3.4.2. Método de recopilación de datos	44
3.4.3. Técnicas de medición de datos	45
3.4.4. Metodología de desarrollo de Software: ICONIX	46
3.4.4.1. Análisis de requisitos	47
3.4.4.2. Análisis y Diseño preliminar	49
3.4.4.3. Diseño	49
3.4.4.4. Implementación	50
3.4.5. Evaluación de Software del sistema de consulta	50

CAPITULO IV

RESULTADOS Y DISCUSION

4.1. Analisis del sistema de consulta de notas utilizando JME	51
4.1.1. Análisis del sistema actual	53
4.1.2. Análisis del sistema propuesto	56
4.1.3. Análisis de requerimientos del sistema de consulta de notas	60
4.1.3.1. Modelo de dominio	62
4.1.3.2. Prototipo rápido	64
4.1.3.3. Diagrama de casos de uso	65
4.2. Diseño de desarrollo del sistema de consulta de notas en móviles con JME	66
4.2.1. Esquema General del Sistema de consulta de notas	66
4.2.2. Funcionamiento del Sistema de consulta de notas	68
4.2.3. Arquitectura de Comunicación	70
4.2.4. Capa Servidor	71
4.2.5. Capa de atención al cliente	74
4.2.6. Capa cliente	76
4.3. Implementación del sistema de consulta de notas en móviles	77
4.3.1. Software de desarrollo del sistema de consulta de notas	77
4.3.2. Diagrama de clases del sistema de consulta de notas	78
4.3.3. Análisis de Casos de uso del sistema de consulta de notas	79
4.3.4. Comportamiento del sistema de consulta de notas	82
4.3.5. Diseño de base de datos del sistema de consulta de notas	85
4.3.6. Diseño de interfaz – mapa de navegación del sistema	90
4.3.7. Implementación del sistema de consultas de notas	92
4.4. Evaluación del sistema de Consulta de notas	115

Conclusiones	127
Recomendaciones	129
Bibliografía	131
Anexos	

INDICE DE FIGURAS

	Pág.
Figura 1 Componentes de la arquitectura Sugorrate	3
Figura 2 Arquitectura de la plataforma Java 2 de Sun	18
Figura 3. El universo de Java Micro Edition (JME)	20
Figura 4. Entorno de ejecución de Java Micro Edition (JME)	21
Figura 5. Pre-verificación de clases CDLC/KVM de JME	24
Figura 6. Arquitectura del entorno de ejecución JME	25
Figura 7. Ciclo de vida de un Midlet	31
Figura8. Arquitectura del funcionamiento de un servidor de aplicaciones	36
Figura 9. Estudiantes con acceso vía web para seleccionar el enlace consultar notas al acceder al sistema académico de la Universidad Nacional del Altiplano - Puno	53
Figura 10. Formulario de identificación de usuario vía web para acceder a información de notas de estudiantes de la UNA – Puno	54
Figura 11. Docente con acceso vía web para seleccionar enlace y formularios para el registro de notas de estudiantes de la Universidad Nacional del Altiplano - Puno	55
Figura 12. Arquitectura de sistema de consulta de notas propuesto	57
Figura 13. Sistema de ventas AnywERP	58
Figura 14. Modelo de dominio del sistema de consulta de notas	63
Figura 15. Prototipos del identificación e información del sistema de consultas de notas	64

Figura 16. Diagramas casos de uso del sistema de consulta de notas	65
Figura 17. Modelo de 3 capas de la arquitectura Cliente - Servidor	66
Figura 18. Diagrama de bloques del sistema de consulta de notas	68
Figura 19. Topología Cliente / Servidor Synframework	69
Figura 20. Estructura de comunicación del sistema de consulta de notas	70
Figura 21. Subcapa de la capa servidor	72
Figura 22. Modelo de la arquitectura Cliente -Servidor	72
Figura 23. Diagrama de clases del sistema de consulta de notas	78
Figura 24. Diagrama de caso de uso del sistema de consulta de notas	79
Figura 25. Diagrama de actores del sistema de consulta de notas	80
Figura 26. Diagrama de secuencia caso de uso autenticar	83
Figura 27. Diagrama de secuencia caso de uso consultar notas	84
Figura 28. Diagrama de secuencia caso de uso ver horario	84
Figura 29. Diagrama Entidad-Relación del sistema de consulta de notas	89
Figura 30. Mapa de navegación de docente en el sistema	90
Figura 31. Mapa de navegación de estudiante en el sistema	91
Figura 32. Diagrama de clases implementado en Java del sistema de consulta de notas	93
Figura 33. Formulario de ingreso al sistema de consulta de notas	100
Figura 34. Pantalla - Identificación de datos del estudiante	101

Figura 35. Pantalla - Horarios del estudiante	103
Figura 36. Pantalla - Cursos matriculados del estudiante	104
Figura 37. Pantalla - Notas y promedio de curso del estudiante	106
Figura 38. Formulario de ingreso al sistema – categoría docente	108
Figura 39. Pantalla - Identificación de datos del docente	109
Figura 40. Pantalla – Horario y carga académica del docente	110
Figura 41. Pantalla - Lista de cursos de carga académica docente	111
Figura 42. Lista de estudiantes de curso seleccionado	112
Figura 43. Formulario de ingreso de notas del estudiante	113
Figura 44. Porcentaje de estudiantes refieren que la interfaz del sistema de consulta de notas es amigable	115
Figura 45. Porcentaje de estudiantes refieren que el sistema de consulta de notas es legible	116
Figura 46. Porcentaje de estudiantes refieren eficiencia del sistema de consulta de notas	117
Figura 47. Porcentaje de estudiantes refieren satisfacción con las tareas que realiza el sistema de consulta de notas	118
Figura 48. Porcentaje de estudiantes refieren reversibilidad de acciones del sistema de consulta de notas	119
Figura 49. Porcentaje de estudiantes refieren sobre la navegación en el sistema de consulta de notas	119

Figura 50. Porcentaje de estudiantes refieren rapidez de acceso a información al sistema de consulta de notas	121
Figura 51. Referencia de uso del sistema de consulta de notas evita costos de desplazamiento	122
Figura 52. Acceso al sistema de consulta de notas desde cualquier lugar	122
Figura 53. Costos en porcentaje para el acceso al sistema de consulta de notas	123
MANUAL DE USUARIO	152
Figura 54. Menú de celular Nokia para seleccionar aplicaciones del celular	155
Figura 55. Seleccionar aplicación juegos de celular	155
Figura 56. Seleccionar aplicación MobileNotas del celular	155
Figura 57. Pantalla de Apertura de aplicación MobileNotas	156
Figura 58. Pantalla - Ingreso de usuario al sistema	157
Figura 59. Pantalla Ingreso de contraseña al sistema	157
Figura 60. Pantalla – Menú de selección categoría Usuario	157
Figura 61. Pantalla - Selección categoría Docente	158
Figura 62. Pantalla - Selección categoría Estudiante	158
Figura 63. Pantalla de Acceso a la aplicación MobileNotas	158
Figura 64. Pantalla - Datos del docente	159
Figura 65. Pantalla - Datos del estudiante	159

Figura 66. Pantalla - Horario del Docente	160
Figura 67. Pantalla - Horario del Estudiante	160
Figura 68. Pantalla - Carga académica docente	161
Figura 69. Pantalla - Cursos del estudiante	161
Figura 70. Pantalla - Notas del estudiante	161
Figura 71. Pantalla - Lista de estudiantes	162
Figura 72. Pantalla - Formulario para ingresar notas del estudiante	162

ÍNDICE DE CUADROS

Cuadro 1. Librerías de configuración incluidas en CLDC	22
Cuadro 2. Librerías de Perfil MIDP	27
Cuadro 3. Distribución de estudiantes matriculados en la Escuela Profesional de Ingeniería de Sistemas de la Universidad Nacional del Altiplano – Puno.	40
Cuadro 4. Estudiantes matriculados en los Semestres VII, IX y X de la Escuela Profesional de Ingeniería de Sistemas en el año académico 2010 seleccionados para la muestra	41
Cuadro 5. Operacionalización de variables	42
Cuadro 6. Técnicas e instrumentos de recolección de datos	45
Cuadro 7. Descripción de los actores del sistema de consulta de notas	81
Cuadro 8. Descripción de los casos de uso del sistema de consulta de notas	82
Cuadro 9. Descripción de la Tablas Escuela Profesional	86
Cuadro 10. Descripción de la Tablas Curso	86
Cuadro 11. Descripción de la Tablas Usuario	86
Cuadro 12. Descripción de la Tablas CursoMatriculados	87
Cuadro 13. Descripción de la Tablas CargaAcademica	88

RESUMEN

El presente trabajo de tesis **“Sistema de consulta de Notas en móviles con Java Micro Edition de los Estudiantes de la escuela profesional de Ingeniería de Sistemas de la Universidad Nacional del Altiplano - Puno 2010”**; investigación realizada en la escuela profesional de Ingeniería de sistemas de la Universidad Nacional del Altiplano el 2010; cuya finalidad es permitir a los estudiantes emplear las tecnologías de información y comunicación para acceder a información de sus notas sobre las asignaturas en las que se encuentran matriculados y a los docentes ingresar las notas de curso asignado de acuerdo a la carga académica. El objetivo de la presente investigación ha sido desarrollar una aplicación utilizando la tecnología Java Micro Edition (JME) para la consulta de notas de los estudiantes desde un dispositivo móvil (teléfono celular) a través de internet, así como el ingreso de notas de los docentes. El presente trabajo de investigación se enmarca en la ingeniería de software y se empleo la metodología orientada a objetos ICONIX para el desarrollo del software del sistema de consulta de notas que se realiza desde el teléfono móvil con servicio de internet y soporte java. Los resultados de las pruebas respecto a la interacción usuario-sistema demuestran que el 54% lo consideran amigable, el 61% regularmente eficiente y el 61% de alto rendimiento respecto a la satisfacción respecto a las tareas que realiza el sistema. En conclusión tanto la universidad y otros entornos educativos pueden beneficiarse en el acceso a información académica de manera remota y brindar un servicio a los estudiantes respecto al espacio-tiempo; permitiendo un ahorro en costos y desplazamientos.

Palabras clave: Consulta de notas, TIC, dispositivos móviles, tecnología JME.

ABSTRACT

The present thesis titled: “Mark Consulting System in Mobile Devices with Java Micro Edition for Students at the Professional School of System Engineering at National University of the Altiplano, Puno – 2010”, shows the results of research applied in this Professional School, with the aim of allowing students to apply information and communication technologies to get access to information about the marking and assessment of their courses in which they were enrolled; as well as teachers who could publish the marks of assigned courses according to their academic load. The objective of the present research has been to develop an application using the Java Micro Edition Technology (JME) in order to know the marks of students from a mobile device (mobile phone) through the Internet, as well as the loading of marks by teachers. The present research work is framed between the software engineering and the use of methodology oriented to ICONIX objects used for software development of mark consulting system from a mobile phone using java support. The results in relation to the interaction user-system have shown that 54% of the users consider the system friendly, 61% regularly efficient and 61% with high performance in relation to the tasks of the system. In conclusion the university and other educational contexts could get benefit from providing the service to the students in relation to the space – time, allowing saving costs and movements.

Key words: mark consulting, ICT, mobile devices, technology, JME.

INTRODUCCION

Actualmente un estudiante de cualquier escuela profesional de la Universidad Nacional del Altiplano cuando desea consultar las calificaciones obtenidas después de haber rendido su examen, teórico, práctico y exposición de trabajos entre otros tipos de evaluación tiene que esperar hasta la finalización del semestre para saber su nota final y las notas no le son entregadas a través de un módulo del sistema académico, sino es consultado al docente. Por otra parte el ingreso de notas por parte de los docentes de acuerdo a la carga académica en las diferentes escuelas profesionales de la Universidad Nacional del Altiplano también se realiza al finalizar el semestre académico.

El presente trabajo de investigación, plantea una alternativa de la consulta de notas haciendo uso de tecnologías de información y comunicación, con el desarrollo de una aplicación empleando dispositivos móviles e internet, que permita al estudiante de la escuela profesional de Ingeniería de sistemas de la Universidad Nacional del Altiplano consultar información académica de su interés como : calificaciones, horarios e información relacionada a las asignaturas en las que se encuentra matriculado a través del teléfono celular con conexión a internet. El desarrollo de la presente investigación fue motivado por la posibilidad de obtener experiencias y conocimientos vinculados con entornos de trabajo del ambiente universitario y su relación con el medio mediante la web. La web proporciona una infraestructura de comunicaciones que permite el acceso a información y servicios desde cualquier computadora conectada, incluyendo los dispositivos móviles como los teléfonos celulares.

Hoy en día, en el mercado los dispositivos móviles tienen conexión a internet, puesto que los costos de este servicio se ha reducido en estos últimos años y

se pueden ejecutar aplicaciones web. Las aplicaciones móviles web pueden ser desarrolladas para gestionar información por los usuarios en cualquier parte del mundo a través de sus teléfonos celulares con conectividad a internet, donde los usuarios que utilizan los dispositivos móviles para navegar por internet se incrementan día a día, y las actividades académicas no están ajenas a esto, pudiéndose brindar un servicio tanto a docente como a estudiantes. Con un Sistema de consulta de notas mediante dispositivos móviles se soluciona esa necesidad de acceso a la información, en cualquier lugar y cuando se necesite, debido a que la información es muy importante respecto al acceso, almacenamiento, organización y la forma de cómo se puede brindar a los estudiantes independientemente de la localización geográfica; planteando nueva tendencia de acceso y transmisión de información.

El presente trabajo se encuentra estructurado en cuatro capítulos; el primer capítulo se refiere al planteamiento del problema. En el segundo capítulo se presenta el marco teórico; donde se encuentra todos los conceptos relacionados al desarrollo de aplicaciones móviles de la presente investigación, en el capítulo tercero se desarrolla la metodología de desarrollo de software; en el capítulo cuarto se presenta los resultados del sistema; y la discusión en referencia a la investigación realizada.

CAPITULO I

EL PROBLEMA DE INVESTIGACION

1.1. PLANTEAMIENTO Y DEFINICIÓN DEL PROBLEMA

Con el avance de las tecnologías de información y comunicación (TIC) se han producido cambios en la educación y en diferentes áreas del conocimiento, dentro de estas tecnologías emergentes cabe destacar los dispositivos móviles, los cuales tienen diferentes ventajas como: portabilidad, movilidad, acceso a información en tiempo real (a través de internet), medio de comunicación, y permite el desarrollo de aplicaciones acorde a las necesidades del usuario. En estos conceptos se sustentará la presente investigación; que permitirá mostrar la capacidad de los dispositivos móviles como un medio de integración y acceso a los sistemas informáticos.

Si un estudiante de la escuela profesional de Ingeniería de sistemas, requiere información de las notas; es decir, calificaciones de las asignaturas en las que se encuentra matriculado, puede acceder al portal de la Universidad Nacional del Altiplano Puno, pero la necesidad de informarse en un momento determinado puede surgir y no siempre tener un computador conectado a internet en ese instante; entonces esto implica un tiempo de espera para acceder a la información académica en un instante de tiempo determinado

causando insatisfacción y un tiempo de espera, entonces surge la necesidad de contar con un sistema de consultas a través de dispositivos móviles con conexión a internet, convirtiéndose en otro medio de acceso a información académica para los estudiantes, y utilizar aplicaciones móviles para gestionar información.

La presente consiste en la implementación de un sistema de consulta de notas utilizando tecnologías JME (Java Micro Edition) que permita el acceso a una base de datos remota que contiene información sobre el rendimiento académico (notas obtenidas en asignaturas cursadas en un semestre académico determinado); que satisface la necesidad de informarse sin necesidad de realizar desplazamientos de un lugar a otro que le brinde la información que requiere.

El presente sistema en su desarrollo hace uso de tecnología Java, específicamente JME; que permite proporcionar herramientas de desarrollo para aplicaciones en dispositivos móviles, el cual se ha enfocado en los emuladores de dispositivos móviles cuya arquitectura se observa en la figura 1; para mostrar la aplicación al usuario; siendo de interés ofrecer el servicio de consultas a los estudiantes de la escuela profesional de Ingeniería de sistemas, a través de un software de aplicación para móviles y acceder de manera remota; permitiéndoles un ahorro en costos y desplazamientos; que implica el acceso a información en cualquier momento y lugar. Siendo necesario resaltar que se sentará las bases para las próximas investigaciones a realizarse.

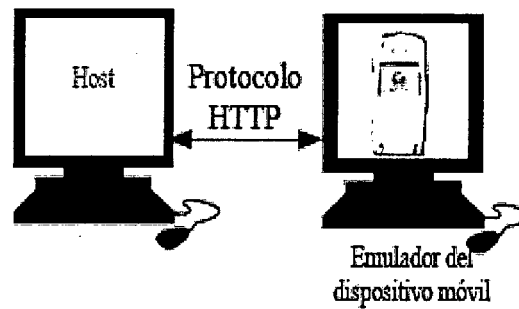


FIGURA 1
COMPONENTES DE LA ARQUITECTURA SURROGATE

Fuente: (Sánchez, 2007)

1.2. FORMULACIÓN DEL PROBLEMA

¿EN QUE MEDIDA EL SISTEMA DE CONSULTA DE NOTAS EN MOVILES CON JAVA MICRO EDITION MEJORA EL ACCESO A LA INFORMACION DE LOS ESTUDIANTES DE LA ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS DE LA UNIVERSIDAD NACIONAL DEL ALTIPLANO - PUNO 2010?

En tal sentido, el presente trabajo de investigación pretende el desarrollo de nuevas formas de interacción (consultas) por medio de dispositivos móviles, que sea accesible e innovadora empleando tecnologías JME (Java Micro Edition); y protocolos de comunicación necesarios para su respectiva aplicación.

1.3.1. OBJETIVOS DE LA INVESTIGACIÓN

1.3.1. Objetivo general

Desarrollar un sistema de consulta de notas en móviles con Java Micro Edition de los estudiantes de la Escuela Profesional de Ingeniería de Sistemas de la Universidad Nacional del Altiplano - Puno 2010.

1.3.2. Objetivos específicos

- Analizar y Diseñar la arquitectura Java Micro Edition que permita la transferencia de información empleando un emulador.
- Implementar del sistema de consultas aplicado en tecnologías móviles.
- Evaluar el sistema de consulta de notas.

1.4. HIPÓTESIS DE LA INVESTIGACIÓN

El desarrollo de un sistema de consulta de notas en móviles con Java Micro Edition mejorará en tiempo y costos el acceso a información de los estudiantes de la Escuela Profesional de Ingeniería de Sistemas de la Universidad Nacional del Altiplano Puno 2010.

CAPITULO II

MARCO TEÓRICO

2.1. ANTECEDENTES DE LA INVESTIGACIÓN

Dentro de los trabajos de investigación relacionados a sistemas móviles o programas de aplicación para móviles relacionados al presente trabajo se tienen los siguientes:

2.1.1. Proceso de consultas móviles adaptables a diferentes tipos de negocios basados en Telefonía Celular (METAMORF)

Este proyecto de investigación, presenta el desarrollo de un proceso centrado en una solución móvil con una arquitectura orientada a servicios, con capacidad de adaptación a diferentes contextos de negocio; brinda un sistema que a través de consumos del servicio web por conexiones TCP/IP puras permite la realización de consultas en ambientes móviles. La metodología de desarrollo aplicada se basó en RUP (Rational Unified Process), así mismo por las características y las necesidades del proyecto se propusieron y analizaron tres procesos que se componen de tres dispositivos un equipo celular con MIDP (Mobile Information Device Profile), un servidor proveedor para el

almacenamiento de configuración de los dispositivos móviles y un servidor cliente residente en la entidad que da acceso al sistema de información que va ha ser consultado (Cruz, Garzon, Montes y et. al., 2008).

2.1.2. Plataforma de desarrollo de aplicaciones multiagente sobre dispositivos móviles con JME

Este proyecto presenta una descripción de la arquitectura del contenedor para sistemas multiagente BESA (Behavior - oriented, Event driven and Social based Agent Framework), orientado al desarrollo de aplicaciones que involucren teléfonos inteligentes (Smartphone's). El desarrollo de esta implementación se basa en las características de programación ofrecidas por la arquitectura JME, debido a que la mayoría de las empresas que elaboran teléfonos inteligentes lo utilizan como estándar de facto, pretendiendo así ampliar el campo de acción e investigación para aplicaciones basadas en sistemas multi Agentes y crear software con dispositivos de larga difusión.

El desarrollo utilizando JME permite ampliar el uso a diferentes plataformas de hardware y software en un entorno restrictivo con la instalación de los siguientes componentes: Configuración para Dispositivos con Limitada Conectividad (CLDC), que incluye la máquina virtual (KVM) y el perfil para dispositivos con información multimedia (MIDP); MIDP permite entre otras ventajas conectividad inalámbrica con diferentes tecnologías como Wi-Fi, Bluetooth e IRDA, principalmente manejo de un entorno gráfico de alto nivel, persistencia (RMS) y administración del ciclo de vida de la aplicación.

BESA/ME incluye un conjunto especial de componentes a nivel de la arquitectura para sistemas multiagente ASMA que permite incluir en el sistema de dispositivos móviles con soporte JME, principalmente teléfonos y agendas digitales. (Garzón & Gonzales, 2009).

2.1.3. Sistema de Información Turística de Bocayá basado en tecnología WAP utilizando parámetros de localización.

En este resultado de investigación se aborda la programación de dispositivos móviles y el turismo en el departamento de Bocayá, usando dispositivos celulares que tengan conexión WAP y la capacidad de iniciar una conexión bluetooth para el envío de parámetros de localización. Dentro de las conclusiones de esta investigación se plantea que la correcta integración de tecnologías WAP, Bluetooth, Web y GPS da como resultado aplicaciones robustas y versátiles; y al ser la telefonía móvil un recurso importante en la actualidad permite a los desarrolladores explorar y explotar nuevos campos de acción (Callejas, Meléndez y et. al., 2010)

2.1.4. Campus Móvil.net: La primera red Socialuniversitaria vía dispositivos móviles de Ibero América. Un estudio de caso

El proyecto de la Spin-Off Campus Móvil.net es el resultado de un proceso de investigación que finaliza en el desarrollo de una red social vía dispositivos móviles para la comunidad universitaria iberoamericana, con acceso exclusivo a través de una cuenta de correo electrónico asignada por la universidad a sus miembros. Se trata de un campus virtual no oficial, basado en el uso de

dispositivos móviles. CampusMóvil.net aspira a cubrir necesidades no resueltas por parte de la comunidad universitaria relacionadas con la falta de servicio ubicuo y con acceso a internet para múltiples funciones de uso diferencial al que se produce desde los ordenadores personales dentro del campus y las propias aulas.

CampusMóvil.net se trata de un tipo de aplicación pensada para el consumo sobre dispositivos móviles, con complementariedad en usos de escritorio y no a la inversa. Su condición de red cerrada universitaria y con énfasis en la navegación vía dispositivos móviles es su estrategia de diferenciación en el mercado (Kuklinski, 2009).

2.1.5. Integración de una aplicación móvil a una Intranet. Caso: Toma de asistencia estudiantil

Este estudio tuvo como objetivos facilitar al docente y al Jefe de prácticas una herramienta que permita un seguimiento actualizado y en tiempo real de la asistencia a clases del alumnado, permitir al docente la creación, eliminación y modificación de sesiones como también permitirle realizar anotaciones específicas sobre alumnos en particular durante el desarrollo de las clases. Para este fin se desarrollaron dos sistemas uno para un sistema web y otro para dispositivos móviles. En ambas versiones de aplicación se permite la verificación y modificación de la asistencia de los alumnos asignados al docente como también hacer comentarios individuales por sesión y alumno y calificar su participación.

Adicionalmente el docente puede crear sesiones en caso sean necesarias. Por medio de la aplicación web, podrán consultar su asistencia, así como ver un listado de la asistencia de sus compañeros de clase, porcentajes de asistencia e inasistencia. Los docentes tienen también acceso a esta información con la diferencia que además podrá ver la asistencia particular de cada alumno (García, 2005).

2.1.6. Aplicaciones Web para celulares

El Objetivo General del proyecto fue proporcionar un modulo WAP al Sistema GIU Guaraní, donde SIU-Guaraní es un sistema de información para gestión de alumnos de las universidades, que registra y administra todas las actividades académicas de la institución, desde que el alumno se inscribe como aspirante hasta que obtiene su título. Fue concebido para administrar la gestión de alumnos en forma segura, con la finalidad de obtener información consistente para los niveles operativos y directivos. El sistema brinda servicios para alumnos, docentes, usuarios administrativos y autoridades, ya que pueden explorar los datos y obtener información como soporte para toma de decisiones.

Una conclusión relevante: El prototipo implementado, justamente consiste en la creación de una nueva Interfaz para un sistema existente, el cual funciona en la actualidad con una interfaz cliente/servidor y otra interfaz Web para ejecutar a través de Internet. Reutiliza la lógica de negocio y la programación de la aplicación (De la Riva, 2007)

2.1.7. Sistema ubicuo mediante telefonía móvil para el acceso a información académica

En los últimos años, hemos asistido a la revolución de la telefonía móvil, la cual ha entrado de lleno en la sociedad y de la que hoy por hoy no se puede prescindir. A la par, el fenómeno Internet se ha extendido masivamente, convirtiéndose en la actualidad en un elemento de uso casi imprescindible en la vida cotidiana. A la unión de las dos tecnologías anteriores es a lo que se denomina "Internet móvil". La potencia de los últimos dispositivos, los hacen adecuados para la ejecución de aplicaciones, que hasta hace poco sólo podían ser ejecutadas desde un ordenador conectado a la red. La Universidad, y los entornos educativos en general, pueden verse beneficiados de este hecho, por lo que proponemos el desarrollo de un sistema ubicuo que permita a los alumnos acceder a información de interés desde cualquier lugar con ayuda únicamente de un teléfono móvil (Peñalver, Botella y et. al., 2005).

2.1.8. Acceso a bases de datos multiplataforma a través de Teléfonos celulares

Los dispositivos móviles y particularmente los teléfonos celulares, hoy en día no son un lujo sino una necesidad. Prácticamente cada integrante de una familia ya dispone de un teléfono celular con buenas capacidades graficas y de computo. Es por esto que las empresas están trabajando para brindarles a sus clientes nuevos servicios para este tipo dispositivos. El desarrollo de las nuevas tecnologías de la información y las comunicaciones (NTICS) en los últimos años impulsan la implantación de sistemas distribuidos que puedan ser

accedidos a través de los teléfonos celulares. Cuando se habla de tecnología se refiere a GSM, GPRS, WAP que permiten que un teléfono celular pueda mantener conexiones de datos y poder consultar cualquier tipo de información que se encuentre en internet, o interactuar con el servidor web o aplicación web de la empresa. Como por ejemplo un banco. 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 dispositivos móviles se han convertido en una plataforma rica para el consumo de contenidos digitales. Con la nueva aparición de los teléfonos de tercera generación 3G se están desarrollando una enorme cantidad de servicios. Entre ellos Internet de banda ancha en el móvil; participar en juegos con otras personas; recibir noticias y previsiones del clima; así como disfrutar de servicios de audio y video. (Soto, 2008).

2.1.9. PROMETEO: Plataforma integrada de desarrollo de aplicaciones para dispositivos móviles con JME

Las plataformas de desarrollo facilitan la implementación de aplicaciones teniendo como principal objetivo la disminución del tiempo de salida al mercado de los servicios desarrollados, mediante el ocultamiento de detalles de implementación a los desarrolladores, ofreciendo una funcionalidad completa dentro de un ambiente integrado de creación y ejecución de aplicaciones. El presente trabajo expone una plataforma integrada de desarrollo de aplicaciones para dispositivos móviles con JME, como solución a la inexistencia de un conjunto de componentes software que interactuando entre sí puedan ser utilizados específica y exclusivamente en el desarrollo de aplicaciones móviles,

utilizando tecnología JME y, que a su vez, puedan ser personalizados dependiendo del dominio de aplicación. Esta plataforma, proporciona herramientas para la creación de los archivos de despliegue necesarios para instalar y ejecutar una aplicación JME, así como generadores de código para crear rápidamente interfaces de usuario, clases genéricas, conexiones HTTP y almacenes persistentes de datos (en el dispositivo móvil).

Adicionalmente, ofrece la posibilidad de configurar bibliotecas de clases que son incluidas automáticamente cuando se crea el JAR de la aplicación y de integrar fácilmente emuladores de terceros. La implementación de la plataforma propuesta se denomina Prometeo y se desarrolló bajo el paradigma del desarrollo de software basado en componentes con tecnología Java y XML (Vásquez, Caicedo, y et. al., 2006)

2.2. MARCO TEÓRICO

2.2.1. Comunicaciones Móviles

Desde el principio de las telecomunicaciones hay dos opciones fundamentales para llevar a cabo una comunicación: existen con o sin hilos, por cable o por aire. De todas formas, en este caso se trata fundamentalmente de una transmisión vía radio, ya que en detalle se explica la telefónica celular que de alguna forma ingresa a ser parte del mundo de las comunicaciones móviles. En cuanto a las comunicaciones móviles, no aparecen a nivel comercial hasta finales del siglo XX. Después llegó la telefonía móvil digital, las agendas personales, miniordenadores, laptops y un sinnúmero de dispositivos dispuestos a conectarse vía radio con otros dispositivos o redes (Gaptel, 2008).

La integración de los dispositivos móviles, internet y el amplio desarrollo de aplicaciones cliente-servidor, proveen una excelente oportunidad para las organizaciones de incrementar su productividad. Este escenario no puede pasar desapercibido frente a una sociedad de información que día a día demanda mayor cantidad de servicios y exige soluciones en cualquier momento y en cualquier lugar (Vásquez, 2008).

2.2.2. Telefonía Móvil

La telefonía móvil es un servicio de la telecomunicación entre al menos un usuario de localización no determinada (móvil) y situado dentro de un área definida, con otros usuarios fijos o móviles. Los principales elementos que constituyen el sistema de telefonía móvil son: terminales, estaciones base, casetas y centrales de conmutación (Menéndez, 2009).

El teléfono celular es uno de los elementos de las nuevas tecnologías que introducen la posibilidad de elegir entre dos modalidades de lenguaje: el oral y el escrito. El teléfono actual es más compacto y con mayores prestaciones de servicio. Además el avance de la tecnología ha hecho que estos aparatos incorporen otras funciones y tengan otros usos del teléfono fijo tradicional. La evolución tecnológica de las redes y los terminales posibilitaron que los teléfonos móviles dejarán de ser sólo un soporte para las telecomunicaciones, para hablar y escribir mensajes de texto, y se transformaran en receptores multimedia, en una tecnología de acceso a datos e información (Hung, 2010).

2.2.3. Acceso a Internet

Con la aparición de la telefonía móvil digital, fue posible acceder a páginas de internet especialmente diseñadas para móviles, tecnología WAP. Las primeras conexiones se efectuaban mediante una llamada telefónica a un número del operador a través de la cual se transmitían los datos de manera similar a como lo haría un modem de PC. El GPRS permitió acceder a internet a través del protocolo TCP/IP. Mediante el software adecuado es posible acceder, desde un terminal móvil, a servicios como FTP, Telnet, mensajería instantánea, correo electrónico, utilizando los mismos protocolos que un ordenador convencional. La velocidad del GPRS es de 54 kbits/s en condiciones óptimas, y se tarifa en función de la cantidad de información transmitida y recibida (Jerez, 2009)

2.2.4. Protocolo HTTP

El protocolo de transferencia de hipertexto (HiperText Transfer Protocol, HTTP) es un protocolo de nivel de aplicación para sistemas de información distribuidos. HTTP ha sido utilizado en la WWW desde 1990. El protocolo permite el intercambio de información hipertextual (enlaces) en las páginas web. Se trata de un protocolo genérico orientado a objetos que puede usarse para muchas tareas como en servidores de nombres y sistemas distribuidos orientados a objetos, por extensión de comandos o los métodos usados (Sivianez, 2010).

El principio del funcionamiento del protocolo HTTP se basa en el tándem pregunta-respuesta. El cliente genera una pregunta con la forma de una petición HTTP, esta petición contiene por lo menos los datos que permiten identificar el recurso solicitado por el cliente. La petición HTTP simplemente es un bloque de texto que transita del cliente al servidor. Este bloque de texto tiene que tener un formato de texto para ser reconocido por el servidor (Groussard, 2010).

El protocolo HTTP viene siendo ampliamente utilizado desde el nacimiento de la web, en donde es indispensable para la creación de aplicaciones que hacen uso de esta. Con la llegada de las tecnologías móviles pueden darse diversas situaciones en las que la creación de un servidor HTTP en dispositivos limitados pueda resultar de gran utilidad. Un ejemplo, es la compartición de la información almacenada en las agendas electrónicas de los asistentes a una reunión para poder concertar otra reunión en las próximas semanas (Diez, 2003).

La plataforma JME dispone del soporte necesario para la creación de todo tipo de clientes HTTP. Estos clientes HTTP están pensados en un principio para la utilización y comunicación con aplicaciones residentes en PCs, de modo que los dispositivos móviles puedan aprovechar todos los servicios ofrecidos por estas máquinas y tener acceso a todo tipo de información almacenada en ellas (páginas Web, aplicaciones para su descarga, etc.). En este artículo se describe de forma detallada tanto el diseño como el desarrollo de un servidor

HTTP en dispositivos limitados basado en la tecnología JME. Para que de esta forma, sean los propios dispositivos móviles los que proporcionen estos servicios, permitiendo acceder a ellos, tanto a otros dispositivos móviles como a los que no lo son, sin necesidad de un servidor intermedio (Diez, 2003)

El principal objetivo a conseguir es construir un servidor HTTP sobre el perfil JME MIDP, de manera que diferentes aplicaciones residentes en distintos dispositivos puedan comunicarse directamente a través del protocolo HTTP. El servidor residente en el dispositivo móvil sería el encargado de proporcionar los datos solicitados por las diferentes aplicaciones, de modo que no sea necesario un elemento intermedio (Diez, 2003)

2.2.5. JAVA

Java es un lenguaje de programación orientado a objetos desarrollado por Sun Microsystems, una compañía reconocida por sus estaciones de trabajo en UNIX de alta calidad. Fundamentado en C++, se diseñó para ser pequeño, sencillo y portátil a través de plataformas y sistemas operativos, tanto a nivel de código como binario, lo que significa que los programas Java (Applets y aplicaciones), pueden ejecutarse en cualquier computadora que tenga instalada una máquina virtual de java (Flores, 2006).

La intención de los desarrolladores fue conseguir una herramienta independiente del tipo de CPU utilizada, por ello se crearon un código “neutro”, al cual nombraron bytecode, el cual se ejecutaba sobre una “máquina

hipotética o virtual” denominada JVM (Java Virtual Machine). Era la máquina virtual de Java quien interpretaba el código neutro convirtiéndolo a código particular de la CPU utilizada. Java se centra en la creación, manipulación y construcción de objetos. El mundo real está lleno de objetos, todo objeto tiene unas propiedades y un comportamiento, cualquier concepto que se desee implementar en un programa java debe ser encapsulado en una clase (Jaramillo, 2008).

Java es un lenguaje orientado a objetos, esto indica que posee ciertas características que se consideran estándares en los lenguajes OO como son: objetos, clases, subclasses, métodos, herencia simple, encapsulamiento, enlace dinámico. La claridad, eficiencia y mantenibilidad del programa resultante dependerá principalmente de la calidad del diseño de clases. Un buen diseño de clases significa una gran economía en tiempo de desarrollo y mantención. Lamentablemente se necesita mucha habilidad y experiencia para lograr diseños de clases de calidad. Un mal diseño de clases puede llevar a programas OO de peor calidad y de más alto costo que el programa equivalente no OO (Castillo, 2007).

Además el lenguaje Java cuenta con diversos API (Application Program Interface) los cuales contienen clases e interfaces para el desarrollo de programas en Java. Es de esta manera que en la Plataforma Java 2 se introdujeron tres APIs los cuales se detallaran a continuación:

- Java 2 Standar Edition (J2SE): es aplicado para el desarrollo de aplicaciones del lado del cliente o applets.
- Java 2 Enterprise Edition (J2EE): es aplicado para el desarrollo de aplicaciones del lado del servidor como Java Servlets y Java Server Pages.
- Java 2 Micro Edition (J2ME): es aplicado para aplicaciones para dispositivos móviles o teléfonos celulares (Liang, 2007).

La figura 2 nos muestra la arquitectura de la plataforma Java 2.

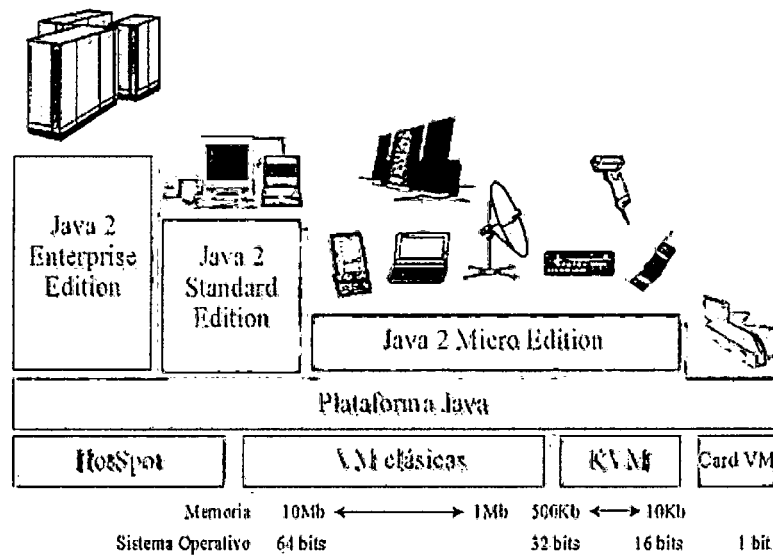


FIGURA 2
ARQUITECTURA DE LA PLATAFORMA JAVA 2 DE SUN

Fuente: Java al tope (Gálvez, 2003)

2.2.6. Java Micro Edition (JME)

La introducción de JME (Java Micro Edition) surgió de la necesidad de definir una plataforma que se use para programar dispositivos electrónicos y sistemas embebidos. La creación de la plataforma J2ME distingue dos categorías:

- Personal, información para dispositivos móviles: que son capaces de comunicar dos personas en red – teléfonos móviles, PDAs y agendas.
- Información para dispositivos con conexión fija compartida: conexión de red sin interrupciones, set-top boxes, internet TV, pantalla plana de internet habilitada, teléfonos de la gama alta de comunicaciones, tarjetas de entretenimiento y sistemas de navegación (Piroumian, 2002).

J2ME está dirigido a desarrollos para dispositivos wireless y dispositivos digitales limitados que necesitan incorporar la funcionalidad multi-plataforma en sus productos. Un beneficio clave de la utilización de J2ME es la compatibilidad con todos los dispositivos habilitados para Java. Motorola, Nokia, Panasonic todos tienen Java-enabled devices. Una aplicación J2ME que procura el equilibrio de procesos entre el dispositivo local y el server-side (Rojas, 2010)

a) Arquitectura de Java Micro Edition (JME)

Como un ambiente completo de ejecución de Java (Java Runtime Environment), JME abarca una máquina virtual, configuraciones, perfiles y un conjunto de paquetes adicionales u opcionales que dependen de la aplicación específica. JME utiliza las configuraciones y perfiles para personalizar el ambiente de ejecución de Java (JRE). La siguiente figura muestra la arquitectura de la tecnología:

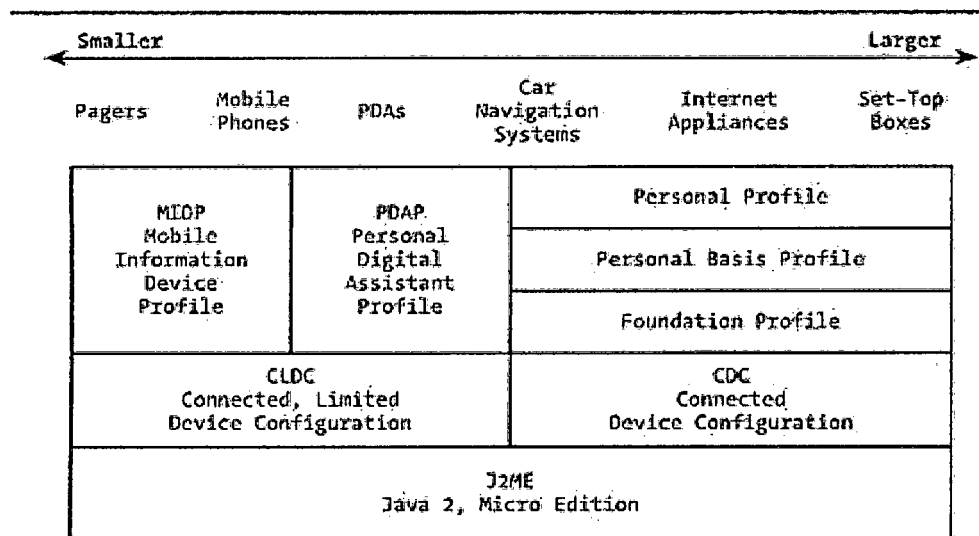


Figura 3

EL UNIVERSO DE JAVA MICRO EDITION (JME)

Fuente: Wireless Java: Developing with J2ME (Knudsen, 2003)

b) Nociones Básicas de JME

Un entorno de ejecución JME se compone de una selección de:

- Máquina Virtual
- Configuración
- Perfil
- Paquetes opcionales

2.2.6.1. Máquina Virtual J2ME

Una máquina virtual de Java (JVM) es un programa encargado de interpretar código intermedio (**bytecode**) de los programas Java precompilados a código máquina ejecutable por la plataforma, efectuar las llamadas pertinentes al sistema operativo subyacente y observar las reglas de seguridad y corrección de código definidas para el lenguaje Java.

De esta forma, la JVM proporciona al programa Java independencia de la plataforma con respecto al hardware y al sistema operativo. Las implementaciones tradicionales de JVM son, en general, muy pesadas en cuanto a memoria ocupada y requerimientos computacionales (Gálvez, 2003).

La arquitectura JME está formada por configuraciones y perfiles. Una configuración incluye una máquina virtual y un conjunto mínimo de clases soportada por un grupo de dispositivos que tienen unas características de memoria y capacidad parecidas. Un perfil tiene asociado un conjunto específico de bibliotecas que completa las necesidades específicas de una familia de dispositivos. (Quintas, 2008)

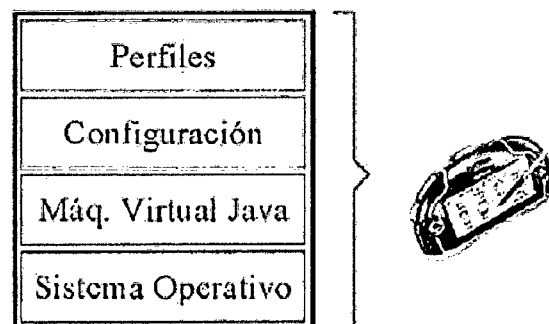


Figura 4
ENTORNO DE EJECUCIÓN DE JAVA MICRO EDITION (JME)

Fuente: Java 2 Micro Edition (Quintas, 2008)

2.2.6.2. Configuraciones

La configuración es un mínimo grupo de APIs, útiles para desarrollar las aplicaciones destinadas a un amplio grupo de dispositivos. Actualmente existen dos configuraciones:

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

CLDC: Connected Limited Device Configuration

Está orientada a dispositivos dotados de conexión y con limitación respecto a capacidad grafica, cómputo y memoria. Los dispositivos que encajan dentro de este grupo son los teléfonos móviles, pagers (buscapersonas), organizadores personales, entre otros. Los dispositivos CLDC deben cumplir con los siguientes requisitos:

- Disponer entre 160Kb y 512Kb de memoria total disponible
- Procesador de 16 a 32 bits con al menos 25 Mhz. De velocidad
- Ofrecer bajo consumo, debido a que estos dispositivos trabajan con suministro de energía limitado, normalmente baterías.
- Tener conexión a algún tipo de red, normalmente sin cable, con conexión intermitente y ancho de banda limitado (unos 9600bps) (Quintas, 2004).

El cuadro 1 nos muestra las librerías incluidas en la CLDC.

CUADRO 1
LIBRERÍAS DE CONFIGURACIÓN INCLUIDAS EN CLDC

Nombre de Paquete CLDC	Descripción
java.io	Clases y paquetes estándar de E/S. Subconjunto de JSE
java.lang	Clases e interfaces de la Máquina Virtual. Subconjunto de JSE
java.lang.util	Clases, interfaces y utilidades estándar. Subconjunto de JSE
javax.microedition.io	Clases e interfaces de conexión genérica CLDC

Fuente: Java a tope: J2ME (Gálvez, 2003)

La KVM (Kilobyte Virtual Machine)

Uno de los objetivos de diseño de la máquina virtual es que puede ser emulado por un programa en una máquina real. Esto permite que casi cualquier equipo sirve como una máquina virtual Java JVM utilizando un emulador. Una implementación de JVM es un programa para su sistema.

A diferencia de los programas de Java Virtual Machine, cosa que se ejecutará en cualquier aplicación depende de las características específicas de su sistema (Engel, 2000)

La KVM es una implementación particular de máquina virtual de Java que cumple las especificaciones CLDC. Un ejemplo, el KVM es una pequeña y portátil Máquina Virtual Java, diseñado para dispositivos pequeños con recursos limitados. El objetivo principal es crear la máquina virtual de java más pequeña posible que mantiene todos los aspectos centrales del lenguaje java, pero se ejecutan en los dispositivos de recursos limitados con sólo unos pocos cientos de kilobytes de memoria total. Este fue diseñado para ser:

- Pequeña, con superficie de memoria estática de 40 a 80 kilobytes
- Alta portabilidad
- Modular y personalizable
- Diseñada para ser completa y rápida posible.

El verificador estándar de clases de Java es demasiado grande para la KVM y el consumo de memoria es excesivo 100 Mb para aplicaciones típicas. Este verificador de clases es el encargado de rechazar las clases no validas para el tiempo de ejecución. (Fox, 2002)

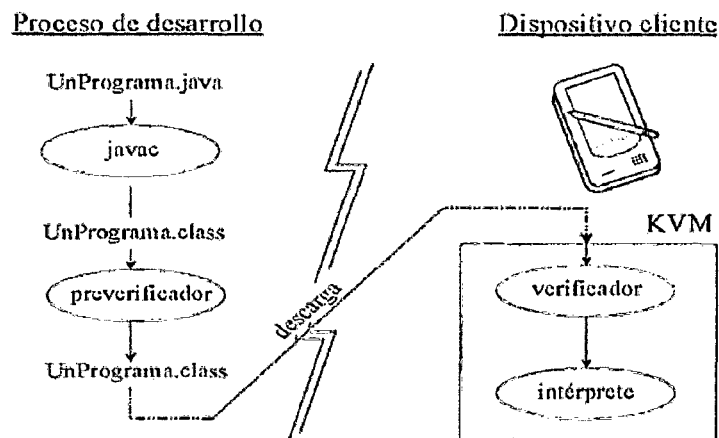


FIGURA 5
PRE VERIFICACIÓN DE CLASES CDLC/KVM DE JME

Fuente: Java a tope: J2ME (Gálvez, 2003)

2.2.6.3. Perfiles

Un perfil define un conjunto de APIs y características comunes para una franja vertical de dispositivos que controlan el ciclo de vida de la aplicación e interfaz de usuario. Las clases de perfil permiten el acceso a funcionalidades específicas de los dispositivos como interfaz gráfica, funcionalidades de red, almacenamiento persistente, etc. Las aplicaciones desarrolladas sobre un determinado perfil van a ser portables a cualquier dispositivo que soporte ese perfil; cabe destacar que un dispositivo puede soportar varios perfiles y que sobre una configuración pueden existir diversos perfiles. Existen perfiles que se construyen sobre la configuración CDC y CLDC, para la configuración CLDC tenemos los siguientes:

- PDA Profile
- Mobile Information Device Profile (MIDP). (Ortega, 2003)

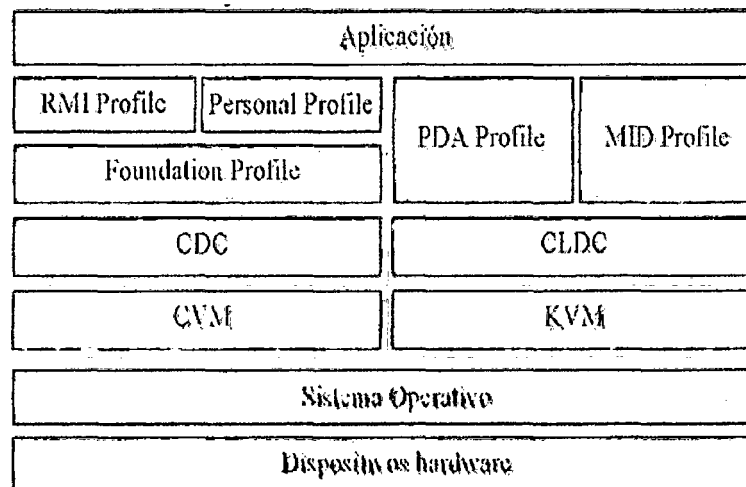


FIGURA 6
ARQUITECTURA DEL ENTORNO DE EJECUCIÓN DE JME

Fuente: Java a tope: J2ME (Gálvez, 2003)

Mobile Information Device Profile (MIDP)

Un perfil MIDP fue desarrollado para dispositivos de información móviles (Mobile Information Device), de acuerdo con la especificación MIDP 2.0. un dispositivo MID debe tener las siguientes características hardware:

- Pantalla de tamaño 96 X 54 pixeles, profundidad de color 1 bit, forma de pixel 1:1.
- Entrada: uno o más mecanismos teclado de 1 ó 2 manos o pantalla táctil.
- Memoria: 128 Kb de memoria no volátil para componentes MIDP, 8 Kb de memoria no volátil para datos persistentes de aplicaciones, 128 kilobyte de memoria volátil en tiempo de ejecución java.
- Comunicación: inalámbrico de dos vías y con ancho de banda limitada (Cardoso, 2007).

En relación al software, un MIDP asume funcionalidades mínimas una vez que el nivel de los dispositivos, hay una gran variedad de arquitecturas de software de sistema. Los requisitos mínimos son los siguientes:

- Un mecanismo para escribir en la memoria no volátil
- Acceso de lectura y escritura en un sistema de comunicación inalámbrico de dispositivos
- Un mecanismo que ofrezca una base de tiempo usado en API de temporizadores.
- Capacidad de escribir gráficos tipo mapa de bits
- Capacidad para capturar entradas del usuario
- Capacidad para gestionar el ciclo de vida de aplicaciones del dispositivo.
(Cardoso, 2007).

Un MIDP ofrece al programador algunas bibliotecas básicas para el desarrollo de aplicaciones, estas bibliotecas cubren las siguientes áreas:

- Bibliotecas nucleares
- Ciclo de vida de aplicación
- Interfaz Grafica
- Almacenamiento persistente
- Comunicación
- Multimedia
- Seguridad (Cardoso, 2007).

CUADRO 2
LIBRERÍAS DEL PERFIL MIDP

Paquetes del MIDP	Descripción
java.lang	Bibliotecas nucleares
java.util	Clases e interfaces de la Máquina Virtual
	Ciclo de vida de aplicación
javax.microedition.midlet	Clases de definición de la aplicación
	Interfaz Grafica
javax.microedition.lcdui	Clases e interfaces para GUIs
	Almacenamiento persistente
javax.microedition.rms	Record Management Storage.
javax.microedition.io	Comunicación
java.io	Clases e interfaces de conexión genérica
Java.pki	Seguridad

Fuente: Java para Tele movéis MIDP 2.0 (Cardoso, 2007)

OTA (Over the Air): Provisioning

Es un proceso en el cual se usa la red inalámbrica como medio para entregar una aplicación. Recibe este nombre debido al hecho de que el proceso de instalación tiene lugar a través del aire. Implementar, encontrar, instalar, actualizar y eliminar una aplicación a través de la red inalámbrica desde el servidor web y hacia dispositivo móvil. Una sesión de protocolo HTTP utiliza en la transferencia de Midlet y asocia archivo JAD (Naing, 2009).

Típicamente, el sistema de almacenamiento OTA se apoya en dos mecanismos de aplicaciones registradas. Con el primer método, el programador puede cargar el archivo de aplicación JAR y JAD como lo especifica el sistema OTA. El sistema de almacenamiento físicamente mantiene estos elementos en su repositorio. Utilizando el segundo, el desarrollador simplemente registra una URL y un archivo JAD que indica ubicaciones donde el administrador puede recuperar la aplicación cuando sea necesario (Piroumian, 2002).

2.2.7. Servlets

Los Servlets son programas de Java que construyen respuestas dinámicas para el cliente, tal como páginas Web. Los servlets reciben y responden a las demandas de los clientes web, normalmente por HTTP. Los servlets son más eficientes que los programas (CGI) porque son cargados de una sola vez en la memoria, y cada demanda es manejada por un hilo de la máquina virtual de Java, no por el sistema operativo. Además los servlets son escalables, dando soporte para una multi aplicación de configuración del servidor. Permiten utilizar datos caché, acceso a información de base de datos, y compartir datos con otro servlets, archivos JSP y (en algunos ambientes) con los bean empresariales (Maidana, 2008).

2.2.8. Midlet

Los Midlet son aplicaciones construidas usando el perfil MIDP. Un midlet pasa por cuatro estados (cargado, activo, pausa, destruido) que son controlados por un gestor de aplicaciones llamado AMS. Una interfaz de un MIDlet se estructura en base a pantallas, donde cada pantalla contiene tareas posibles de ejecutar, ya que en un determinado momento sólo puede mostrarse una única pantalla en el LCD del dispositivo (Bertuzzi, 2007).

Las aplicaciones Java que se ejecutan en dispositivos MIDP (perfil para información de dispositivo móvil). se conoce como Midlet. Un MIDlet consiste en una clase java que deriva de una MIDP clase abstracta definida **javax.microedition.midlet.MIDlet**.

MIDlet se ejecuta dentro de la máquina virtual de Java que proporciona un ciclo de vida definido y controlado a través de métodos de la clase MIDlet que cada MIDlet debe implementar (Topley, 2002).

2.2.8.1 El Gestor de Aplicaciones o AMS (*Application Management System*)

Sistema que se utiliza para gestionar aplicaciones Midlet. Permiten controlar los estados de la aplicación Midlet durante la ejecución, así como el seguimiento de los estados de la aplicación y eliminación de aplicaciones (Hansmann, 2002)

El Gestor de aplicaciones AMS, o software encargado de gestionar Midlet, controla los procesos de instalación, ejecución y eliminación Midlet y cuando un usuario decide ejecutar un midlet se crea una instancia de clase MIDlet que ejecuta el método en él. Esta secuencia de métodos que serán llamados en la subclase Midlet y están definidos por el ciclo de vida del MIDlet. El Midlet como los applets y servlets tiene un pequeño conjunto de estados bien definidos (Knudsen, 2003).

2.2.8.2 Ciclo de vida de un MIDlet

Todos los dispositivos J2ME tienen un gestor de aplicación (AMS), esto forma parte del sistema operativo que se encarga de la ejecución de los Midlets. El AMS crea, inicia, detiene y destruye los Midlets por una serie de razones.

Por ejemplo; si se selecciona el midlet desde el menú del teléfono, el AMS va a crear e iniciar el Midlet. Si recibes una llamada telefónica, el AMS pondrá el Midlet en estado de pausa. Se debe manejar estos eventos en el Midlet con métodos abstractos como: *startApp*, *pauseApp* y *destroyApp*.

Un Midlet pasa por los siguientes estados:

- Cuando el midlet está para ser ejecutado, se crea una instancia, el constructor Midlet se ejecuta y el Midlet está en el estado de pausa.
 - A continuación, el Midlet pasa al estado activo después de que el gestor de aplicaciones llama al método *startApp()*.
 - Mientras el Midlet está activo, el administrador de aplicaciones puede suspender su ejecución llamando al método *pauseApp()*. Esto pone al Midlet en estado de pausa. Un Midlet puede ponerse en estado de pausa llamando al método *notifyPaused()*
 - El administrador de aplicaciones puede terminar la ejecución del Midlet llamando al método *destroyApp()*, punto en el cual el Midlet se destruye y espera. Un midlet puede destruirse llamando al método *notifyDestroy()*.
- (Knudsen, 2003).

A continuación la figura 7 muestra cómo estos estados y métodos se interrelacionan.

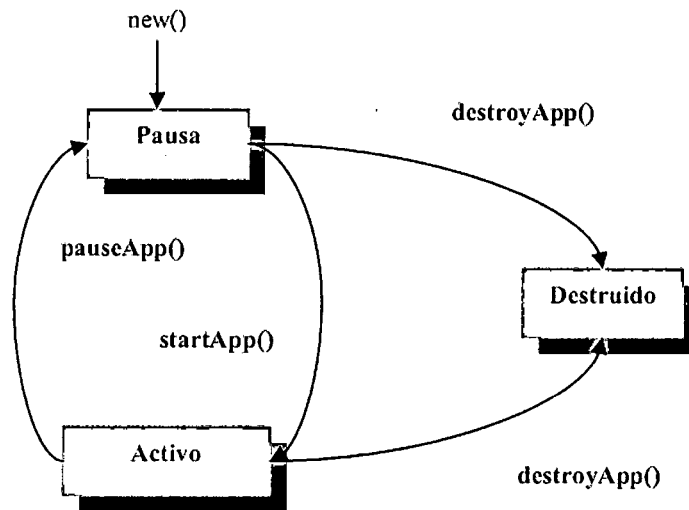


FIGURA 7
CICLO DE VIDA DE UN MIDLET

Fuente: Wireless Game Development in Java with MIDP 2.0 (Barbagallo, 2003)

2.2.8.3 DTO: Data Transfer Objects

Es un patrón de diseño usado para transferir datos, a través de subsistemas o capas de una aplicación para de esta manera mantener bajo acoplamiento entre los subsistemas y/o las capas. Los DTOs son livianos puesto que no contienen ningún comportamiento (funcionalidad) excepto el encapsulamiento (set y get) de sus propios datos. Generalmente los DTO representan a los objetos del dominio de la aplicación, ya que estos son los datos que la aplicación requiere que se muevan entre capas para su presentación, operación sobre ellos y/o recuperación o almacenamiento. También pueden ser objetos que no son parte del dominio pero que requieren transportar información entre capas y que por lo tanto han sido compuestos por conveniencia para ese propósito. Los DTOs representan al mismo patrón de diseño que los Value Objects (VO) o transfer objects que tiene la responsabilidad del transporte de información (Meza, 2008)

2.2.9. Base de Datos

Una base de datos es un conjunto, colección o depósito de datos almacenados en un soporte informático de acceso directo. Los datos deben estar relacionados y estructurados de acuerdo con un modelo capaz de recoger el contenido semántico de los datos almacenados. Dada la importancia que tienen en el mundo real las relaciones entre los datos, es imprescindible que las bases de datos sea capaz de almacenar estas interrelaciones. (Llanos, 2010).

Un sistema de base de datos, es básicamente un sistema computarizado para guardar registros. Es decir, es un sistema computarizado cuya finalidad general es almacenar información y permitir a los usuarios recuperar y actualizar esa información con base en peticiones. La información en cuestión puede ser cualquier cosa que sea de importancia para el individuo y organización; en otras palabras todo lo que sea necesario para auxiliarle en el proceso general de su administración (Date, 2004).

Tal como se ha visto las bases de datos hacen referencia, a una colección específica de datos, aunque es habitual usarlo, de manera errónea, como sinónimo del software que gestiona dicha colección de datos, que se conoce como Sistema Gestor de Base de datos. (Llanos, 2010)

Sistema de Gestión de Base de Datos

Un Sistema Gestión de Bases de Datos –SGBD- (Data Base Management System DBMS) es un software de gestión de base de datos que almacena los datos en forma de tablas. Las características más comunes que proporcionan los gestores de base de datos son las siguientes:

- Recuperación y visualización de los registros, permitiendo la modificación de los mismos al momento.
- Facilidad de actualización (añadir, borrar y modificar registros).-
- posibilidad de mantener datos ordenados por diversos campos distintos mediante indexación.
- Facilidad de obtención de informes y documentos con diversos formatos.
- Compartición de los datos por distintos usuarios de la red, etc. (Gonzales, 2003)

Un SGBD es un programa de ordenador que facilita una serie de herramientas para manejar bases de datos y obtener resultados (información) de ellas. Además de almacenar la información, se le pueden hacer preguntas sobre esos datos, obtener listados impresos, generar pequeños programas de mantenimiento de la BD, o ser utilizado como servidor de datos para programas más complejos realizados en cualquier lenguaje de programación. Además, ofrece otras herramientas más propias de la gestión de BD como sistemas de permisos para autorización de accesos, volcados de seguridad, transferencia de ficheros, recuperación de información dañada e indización. (Gómez, 2002)

Posee herramientas para asegurar:

Independencia de datos: a varios niveles, permitiendo la modificación de las definiciones de datos sin afectar a las aplicaciones o esquemas que no utilizan esos datos.

Integridad de los datos: que los datos sean correctos en todo momento, de acuerdo con las especificaciones o reglas impuestas al sistema

Seguridad de datos: que sólo las personas autorizadas puedan acceder a determinados datos y que sólo puedan efectuar las operaciones para las que han sido autorizados.

Al decir que un SGBD es relacional, estamos hablando de que, como mínimo, sigue todas las reglas y conceptos propuestos por el modelo relacional. El modelo relacional se basa en la teoría de conjuntos y es, por tanto, un modelo con un fundamento matemático. Este modelo maneja una estructura de datos, la relación (concepto matemático que se representa “físicamente” como una tabla), y unos operadores definidos sobre ella. (Gómez, 2002)

Bases de datos Relacionales

Una base de datos relacional es básicamente un conjunto de tablas, similares a las tablas de una hoja de cálculo, formadas por filas (registros) y columnas (campos). Los registros representan cada uno de los objetos descritos en la tabla y los campos los atributos (variables de cualquier tipo) de los objetos.

En el modelo relacional de base de datos, las tablas comparten algún campo entre ellas. La idea básica de las bases de datos relacionales es la existencia de *entidades* (filas en una tabla) caracterizadas por *atributos* (columnas en la tabla). Cada tabla almacena entidades del mismo tipo y entre entidades de distinto tipo se establecen relaciones. (Shekar, 2005)

Las tablas comparten algún campo entre ellas, estos campos compartidos van a servir para establecer relaciones entre las tablas. Los atributos pueden ser de unos pocos tipos simples: números enteros, números reales, cadena de caracteres de longitud variable, etc. Estos tipos simples se denominan *tipos atómicos* y permiten una mayor eficacia en el manejo de la base de datos pero a costa de reducir la flexibilidad a la hora de manejar los elementos complejos del mundo real y dificultar la gestión de datos espaciales, en general suponen un problema para cualquier tipo de datos geométricos. Las relaciones que se establecen entre los diferentes elementos de dos tablas en una base de datos relacional pueden ser de tres tipos distintos:

Relaciones uno a uno, se establecen entre una entidad de una tabla y otra entidad de otra tabla.

Relaciones uno a varios, se establecen entre varias entidades de una tabla y una entidad de otra tabla.

Relaciones varios a varios, se establecen entre varias entidades de cada una de las tablas (Shekar, 2005)

2.2.10. Servidor de Aplicaciones

Un servidor de aplicaciones no es más que un cambio de nombre para algunos servidores Web de nueva generación que permiten construir aplicaciones. Suelen asociarse con servidores de alto rendimiento pensados para dar servicio a sitios Web con grandes necesidades para gestionar movimientos de datos, afluencia de visitas, atención de transacciones hacia bases de datos, etc. Generalmente los fabricantes del sector tienen a disposición del público un servidor Web básico y otro con multitud de extensiones integradas al que llaman servidor de aplicaciones. (López, 2001)

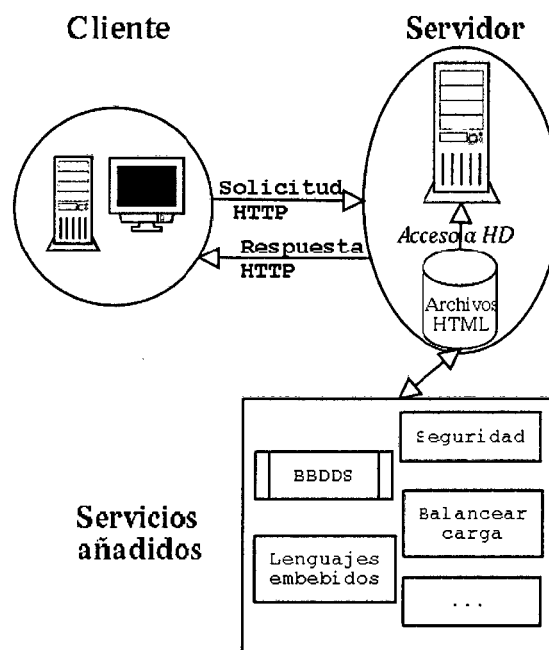


Figura 8

ARQUITECTURA DEL FUNCIONAMIENTO DE UN SERVIDOR DE APLICACIONES

Fuente: Servidores Web (López, 2001)

La arquitectura Servidor de aplicaciones permite disfrutar los beneficios de la computación Cliente / Servidor, incluso cuando las computadoras cliente no son suficientemente poderosas para ejecutar alguna de las aplicaciones Cliente/ Servidor. Este estilo arquitectónico permite que cualquier aplicación resida en una poderosa computadora conocida como servidor de aplicaciones, a ser ejecutada y compartida por muchos clientes menos poderosos y las computadoras cliente tan solamente se encargan de la presentación de la salida de la aplicación (Rob, 2006).

Sistema Gestor de Base de Datos MySQL

MySQL es un sistema de administración de base de datos relacionales rápido, sólido y flexible. Es ideal para crear base de datos con acceso desde páginas web dinámicas, para la creación de sistemas de transacciones on-line o para cualquier otra solución profesional que implique almacenar datos, teniendo la posibilidad de realizar múltiples y rápidas consultas (Cobo, 2005)

MySQL ofrece ventajas frente a otros sistemas gestores de base de datos:

- MySQL utiliza el lenguaje SQL (Structured Query Language) – Lenguaje de consulta estructurado, que es el lenguaje de consulta más usado y estandarizado para acceder a base de datos relacionales.
- Es un sistema Cliente-Servidor, permitiendo trabajar como servidor multiusuario y subprocesamiento múltiple; es decir, cada vez que se establece una conexión con el servidor, el programa servidor crea un subproceso para

manejar la solicitud del cliente, controlando el acceso simultaneo de un gran número de usuarios a los datos y asegurando el acceso sólo a usuarios autorizados (Cobo, 2005).

2.2.11. Emulador para dispositivos inalámbricos

El emulador de J2ME herramienta inalámbrica es como un teléfono móvil celular genérico, que presenta varias características y botones que se encuentran típicamente en dispositivos reales (Tavares, 2005)

Las herramientas de J2ME para dispositivos inalámbricos incluyen diferentes emuladores que se usan para probar aplicaciones. Al hacer clic en el botón de ejecución en estas herramientas J2ME, la aplicación se ejecuta en el emulador seleccionado (Knudsen, 2003).

CAPITULO III

METODOLOGIA

3.1. LOCALIZACIÓN

El presente trabajo de investigación respecto al desarrollo de sistemas de consulta de notas en dispositivos móviles con Java Micro Edition se enmarca en el área de Ingeniería de Software, y la aplicación del presente estudio se realizó en la ciudad de Puno, capital de la Región Puno ubicada al sureste del país que se encuentra a 3827 metros sobre el nivel del mar; y para la prueba del funcionamiento del software se aplicará a estudiantes de la Escuela Profesional de Ingeniería de Sistemas de la Universidad Nacional del Altiplano de esta ciudad.

3.2. DETERMINACIÓN DE LA MUESTRA

3.2.1. Población de estudiantes

La Población considerada para la validez del presente trabajo de investigación son los estudiantes matriculados de la Escuela Profesional de Ingeniería de

Sistemas de la Universidad Nacional del Altiplano, que está constituido por un aproximado de 469 estudiantes matriculados en el semestre académico 2010, los cuales están distribuidos en diferentes semestres académicos cuyos datos mencionados se pueden observar en el cuadro 3.

CUADRO 3

DISTRIBUCION DE ESTUDIANTES MATRICULADOS EN LA ESCUELA PROFESIONAL DE INGENIERIA DE SISTEMAS DE LA UNIVESIDAD NACIONAL DEL ALTIPLANO PUNO 2010

POBLACION		
Semestre		Cantidad de estudiantes matriculados
I	Semestre	119
II	Semestre	84
III	Semestre	55
IV	Semestre	41
V	Semestre	38
VI	Semestre	36
VII	Semestre	24
VIII	Semestre	34
IX	Semestre	20
X	Semestre	18
Total		469

Fuente: Relación de estudiantes matriculados - Oficina de Coordinación Académica de la Escuela Profesional de Ingeniería de Sistemas de la Universidad Nacional del Altiplano – Puno

3.2.2. Determinación del tamaño de muestra

La muestra se determinó para la aplicación del uso del Sistema de Consulta de notas de los estudiantes de la Escuela Profesional de Ingeniería de Sistemas a

través de móviles, donde se consideró una muestra constituida por 59 estudiantes, (ver anexo) los que se encuentran comprendidos en los semestres VII, IX y X de la Escuela Profesional de Ingeniería de Sistemas de la Universidad Nacional del Altiplano - Puno.

Método de Selección de Muestra

La selección de la unidad de muestra que se empleo permitió que cada unidad de muestra seleccionada cumpla un procedimiento según el criterio del investigador, por lo que se selecciono unidades de muestra representativas en razón a la accesibilidad con la que se disponía, puesto que los estudiantes que colaboraron al presente trabajo de investigación estuvo conformado por una cantidad de estudiantes determinada en el tamaño muestra con 59 estudiantes matriculados. Los estudiantes considerados para la muestra se encuentran matriculados en el año académico 2010 y corresponde a estudiantes matriculados en los semestres VII, IX y X de la Escuela Profesional de Ingeniería de Sistemas de la UNA – Puno, datos pueden observarse en el cuadro 4.

CUADRO 4

ESTUDIANTES MATRICULADOS EN LOS SEMESTRES VII, IX y X DE LA ESCUELA PROFESIONAL DE INGENIERIA DE SISTEMAS EN EL AÑO ACADEMICO 2010 SELECCIONADOS PARA LA MUESTRA

ESCUELA	SEMESTRE	N° DE ALUMNOS	PORCENTAJE
INGENIERIA DE SISTEMAS	VII	21	36%
INGENIERIA DE SISTEMAS	IX	20	34%
INGENIERIA DE SISTEMAS	X	18	30%
TOTAL		59	100%

Fuente: Elaboración propia

3.3. OPERACIONALIZACIÓN DE VARIABLES

Las variables que se encuentran en el presente trabajo de investigación son:

CUADRO 5
OPERACIONALIZACIÓN DE VARIABLES

VARIABLES	DIMENSION	INDICADOR	INDICE
Variable Independiente			
Sistema de consulta de notas móviles JME	Consultas de los estudiantes docentes a través de tecnologías móviles.	El Sistema de consulta permite al usuario funcionalidad: - Amigable - Legible - Satisfacción - Reversibilidad - Interfaz grafica	Muy alto Alto Medio Bajo Muy bajo
Variable Dependiente			
Mejora el acceso a información de los estudiantes	La información que provee el sistema es adecuada a las necesidades del usuario.	Rapidez	Muy alto Alto Medio Bajo Muy bajo
		Costos de desplazamiento	Alto Regular Bajo
		Acceso a información desde cualquier lugar	Alto Regular Bajo
		Costos de otras formas de acceso a información	Alto Regular Bajo

Fuente: Elaboración propia

3.4. MÉTODOS

3.4.1. Tipo de investigación

El presente trabajo de investigación se caracteriza por ser una investigación de tipo experimental. El cual está orientado a la población estudiantil que empleará el Sistema de Consultas de notas en móviles con JME (Java Micro Edition) con los alumnos seleccionados en la muestra.

En la investigación se adopta el diseño cuasi experimental que: “Controla algunas aunque no todas, las fuentes que amenazan la validez, se emplean en situaciones en las cuales es difícil o casi es imposible el control experimental riguroso. Una de estas situaciones es precisamente el ámbito en el que se desarrolla la educación y el fenómeno social en general”. (Sánchez, Carlessi, et. al., 2009).

Para la presente investigación se aplicará post prueba, con un solo grupo; como se observa en el siguiente cuadro.

Grupo	Tratamiento	Acciones
Experimental	X	P.S.

Donde:

P.S. : Evaluación de Sistema de Consulta

X : Con tratamiento de estudio: Uso del sistema

3.4.2. Métodos de recopilación de datos

Técnicas e Instrumentos de Recolección de datos

Técnicas.- Para la recolección de datos se consideró lo siguiente:

Encuesta: Después del desarrollo del Sistema se realizó encuestas a los estudiantes sobre su opinión acerca del uso del Sistema de consulta de notas en móviles con JME para obtener información sobre su rendimiento académico (notas) en la escuela profesional de Ingeniería de Sistemas de la Universidad Nacional del Altiplano - Puno 2010 para que realicen consultas de calificaciones o notas de sus asignaturas (ver anexo).

Instrumento de Recolección de Datos

Cuestionario: En esta investigación se utilizó el cuestionario como instrumento para recolectar información, el cual fue diseñado en forma sencilla y clara, de tal manera que sea comprendida con facilidad; así mismo cabe indicar que contiene preguntas cerradas. El cuestionario que se realizó a los estudiantes considerados como unidad de muestra (ver anexo).

Registro de notas: Donde se encuentran las notas y/o calificaciones de los estudiantes (rendimiento académico). Esto con la finalidad de comprobar la utilidad del sistema de consulta de notas en móviles. Se empleó la información de los promedios finales; con la finalidad de almacenar dicha información en la base de datos correspondiente.

CUADRO 6
TECNICAS E INSTRUMENTOS DE RECOLECCION DE DATOS

Nº	Técnicas	Instrumentos	Dirigido a:
1	ENCUESTA	Cuestionario	Estudiantes
2	OBSERVACION	Registros y Actas de Evaluación	Estudiantes

Fuente: Elaboración propia

Procedimiento de Recolección de datos

Para la recolección de datos se realizó lo siguiente:

- Se coordinó con la Dirección de Estudios de la Escuela Profesional de Ingeniería de sistemas; con la finalidad de obtener autorización correspondiente para la ejecución de la presente investigación.
- Se coordinó con los docentes responsables de asignaturas para obtener las calificaciones obtenidas por los estudiantes de la escuela profesional de Ingeniería de Sistemas de la Universidad Nacional del Altiplano-Puno.

3.4.3. Técnicas de medición de datos

Tabulación De Datos

El uso del Sistema de Consulta por parte de los estudiantes permitió observar resultados, respecto al empleo del sistema de consultas; para la medición de los datos se utilizó las siguientes técnicas:

Estadística descriptiva, que permitió analizar y describir a la muestra de individuos de la población estudiantil de la Escuela Profesional de Ingeniería de Sistemas.

- a. Selección de los elementos de muestra para ser estudiados; para que el acceso a la información sea factible.
- b. Encuesta, obtención del valor de cada individuo en los caracteres seleccionados.
- c. Elaboración de tablas de frecuencias
- d. Representación gráfica de los resultados (elaboración de gráficas estadísticas); Se analizarán los cuadros, destacando los más relevantes para comprobar y evaluar los resultados desde el punto de vista del usuario final. Los resultados obtenidos del tratamiento de datos se describirán.

3.4.4. Metodología de desarrollo de software: ICONIX

El presente trabajo de investigación: Sistema de consulta de notas en móviles con JME (Java Micro Edition) de los estudiantes de la escuela profesional de Ingeniería de Sistemas de la Universidad Nacional del Altiplano - Puno 2010; empleó la metodología de desarrollo de software ICONIX; considerada como una metodología intermedio entre XP y RUP, siendo el primero útil para Software's pequeños y, el segundo, para software's industriales; por tanto, ICONIX es una mezcla entre la agilidad de XP y la robustez de RUP.

La Metodología ICONIX no es tan burocrática como la metodología RUP; es decir, no genera tanta documentación, a pesar de ser un proceso simple como XP, realiza análisis, diseño y destaca un proceso de software, que una también con el lenguaje de modelado UML.

Las principales fases de ICONIX son:

1. Análisis de requerimientos
2. Análisis y diseño preliminar
3. Diseño detallado
4. Implementación

3.4.4.1. Análisis de Requisitos

Se determina los requisitos del sistema; es decir, se identifican los objetos del mundo real, las relaciones de agregación y generalización entre ellos. En esta etapa se captura información sobre como desearían los usuarios finales el sistema propuesto.

Modelo de dominio

Se utiliza el diagrama de clases de alto nivel definido como modelo de dominio. Luego con los requisitos se construirá el diagrama de clases, que representa las agrupaciones funcionales del Sistema.

Prototipo rápido

Presentar un prototipo rápido de las interfaces del sistema, diagramas de navegación de tal manera que se pueda comprender el sistema propuesto. La rapidez con la que se genera el sistema es para que los usuarios puedan evaluar la aplicación en la mayor brevedad posible. En esta evaluación se debe capturar lo que agrada o desagrada al usuario respecto al sistema.

El diseño de prototipos es una técnica de ingeniería, para desarrollar modelos a escala o simulados de un producto o de sus componentes. Cuando se aplica al desarrollo de sistemas de información el desarrollo de prototipos implica la creación de un modelo o modelos operativos de trabajo de un sistema.

Los prototipos de pantallas proporcionan información acerca de la reacción de los usuarios; hacia la cantidad de información presentada sobre la pantalla de visualización.

Modelos de caso de uso

Se identifica los actores de casos de uso del sistema, mostrando los actores involucrados; de tal manera que se representa en un Modelado de Casos de Uso.

Caso de Uso: es una manera específica de utilizar el sistema, es la imagen de la funcionalidad del sistema, desencadenada en respuesta a la estimulación externa.

Los casos de uso permiten definir como se desea interactuar con el sistema y permite describir que se debe de realizar para obtener el resultado que se espera.

3.4.4.2. Análisis y Diseño preliminar

En esta espata se realizó lo siguiente:

Descripción de casos de uso

Se describe los casos de uso, como un flujo principal de acciones que puede contener flujos alternativos y de excepción; la principal sugerencia de ICONIX en esto es que no se debe perder mucho tiempo en la descripción textual.

Diagrama de robustez

Se realiza un diagrama de robustez, donde se ilustra gráficamente las interacciones entre los objetos participantes en los casos de uso. Este diagrama permite analizar el texto narrativo de cada caso de uso e identificar un conjunto inicial de objetos participantes de cada caso de uso. El análisis de robustez ayuda a identificar los objetos que participan en cada caso de uso, estos objetos forman parte del diagrama y se pueden clasificar en: objetos de interfaz, objetos entidad y objetos de control.

3.4.4.3. Diseño

Se diseña el diagrama de secuencia, que forma parte del modelo dinámico del sistema, y muestra el comportamiento del sistema propuesto a través de las interacciones, permitiendo identificar los cursos alternos de cada caso de uso, así como identificar los mensajes entre los diferentes objetos. Es necesario verificar si el diseño satisface todo los requisitos identificados.

3.4.4.4. Implementación

Se escribe y genera el código de la aplicación del sistema, pudiendo reutilizar código de componentes de otras aplicaciones. Luego se realiza las pruebas con datos y observando los resultados, integrando a los usuarios para verificar los resultados.

La intervención de los usuarios es parte fundamental de estas metodologías ágiles de desarrollo software, para la verificación de requisitos, prototipo rápido y las pruebas del sistema.

3.4.5. Evaluación de software del sistema de consulta

Prueba de caja negra

Se empleo el método de prueba de caja negra, que se centra en los requerimientos funcionales del software. Es decir, la prueba de la caja negra permite al ingeniero del software derivar conjuntos de condiciones de entrada que ejerciten completamente todos los requerimientos funcionales de un programa. Su enfoque es complementario que intenta descubrir diferentes tipos de errores que los métodos de la caja blanca. La prueba de la caja negra intenta encontrar errores de las siguientes categorías:

- Funciones incorrectas o ausentes
- Errores de interfaz
- Errores en estructura de datos o en acceso a base de datos externas
- Errores de rendimiento
- Errores de inicialización y de terminación

CAPITULO IV

RESULTADOS Y DISCUSIÓN

4.1. ANÁLISIS DEL SISTEMA DE CONSULTA EN MOVILES USANDO JME

Debido a que la Universidad Nacional del Altiplano actualmente brinda a los estudiantes un sistema académico disponible en: <http://www.unap.edu.pe> que les permite acceder a la consulta de notas vía web para obtener información sobre su rendimiento académico, y considerando que en un momento determinado se desea informar sobre sus notas y no se tenga el servicio de internet en ese momento implica tiempo y costos de desplazamiento a una cabina de internet para dicha consulta; por lo que se decidió desarrollar un sistema de consulta de notas en móviles con Java Micro Edition para mejorar en tiempo y costos el acceso a información.

Según el proyecto aplicaciones web para celulares, en otra de sus conclusiones indica que: hoy en día el desarrollo de aplicaciones para celulares consiste principalmente en adaptar aplicaciones existentes con cambios que permitan satisfacer los requerimientos de los dispositivos inalámbricos (De la Riva, 2007).

En el desarrollo del sistema de consulta de notas, se empleó la metodología ICONIX, que no es tan burocrática como la metodología RUP y es sencilla como la metodología XP (Extreme Programming), metodología práctica que permite realizar un análisis y diseño del proceso de software considerando las siguientes fases de desarrollo: análisis de requerimientos, análisis y diseño preliminar, diseño detallado e implementación, con soporte de modelado UML.

Las etapas metodológicas a llevar a cabo para el desarrollo de sistemas de información, se establecen de la siguiente manera: análisis, diseño, codificación, prueba y mantenimiento (Pressman, 2004)

Por otra parte, las metodologías tradicionales generalmente imponen un proceso de desarrollo demasiado pesado y burocrático (Fowler, 2001) lo que impide un desarrollo ágil y rápido para este tipo de aplicaciones.

Una aplicación web, al igual que una aplicación para dispositivos móviles tiene un ciclo de desarrollo como cualquier otro producto software en el que se pueden encontrar descritas las fases del UML: concepción, elaboración, construcción, transición. (García, 2005).

Las metodologías de desarrollo de software plantean diferentes fases, y al seleccionar la metodología ICONIX se procedió a detallar cada una de las fases correspondientes, pero previamente se realizó un análisis del sistema académico que actualmente utilizan los estudiantes de la universidad Nacional del Altiplano.

4.1.1. Análisis del Sistema Actual

El sistema académico actual para los estudiantes incluye los siguientes aspectos:

La figura 9 muestra la interfaz cuando los estudiantes acceden vía web al sistema académico de la Universidad Nacional del Altiplano y desean conocer sus notas (rendimiento académico) luego de haber presentado sus evaluaciones de los cursos en los que se encuentran matriculados.

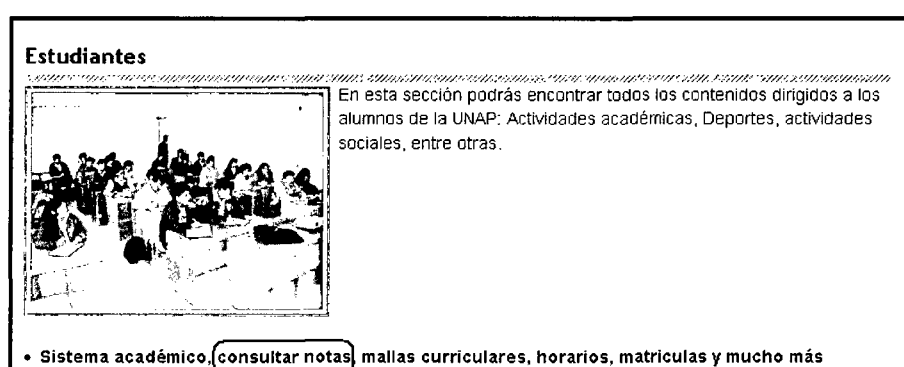


FIGURA 9

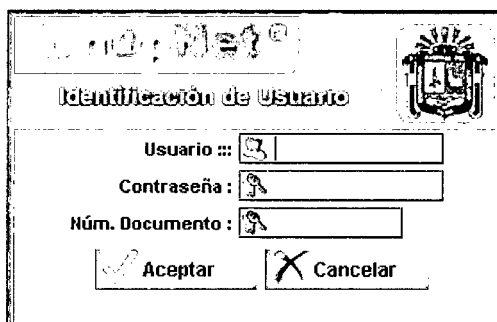
ESTUDIANTES CON ACCESO VIA WEB PARA SELECCIONAR EL ENLACE CONSULTAR NOTAS AL ACCEDER AL SISTEMA ACADEMICO DE LA UNIVERSIDAD NACIONAL DEL ALTIPLANO - PUNO

Fuente: www.unap.edu.pe / (2011)

La figura 9 muestra los diferentes enlaces como son: sistema académico, consultar notas, mallas curriculares, horarios, matriculas y mucho más, enlaces a los cuales el estudiante puede acceder en la siguiente dirección: <http://www.unap.edu.pe>, donde puede seleccionar uno de los enlaces para consultar información de su interés.

Es previsible que la evolución de las tecnologías y del software impulsen la adopción de aplicación de navegación web totalmente abiertas para dispositivos móviles (Kuklinski, 2009).

La figura 10 muestra un formulario de acceso que permite la identificación de los estudiantes en el sistema académico de la Universidad Nacional del Altiplano vía web, formulario donde ingresarán datos necesarios y obligatorios para obtener información sobre sus notas (rendimiento académico), sólo de interés personal.



The image shows a web browser window displaying a login form. At the top left, there is a logo that says 'una.net'. To the right is the university's crest. The title of the form is 'Identificación de Usuario'. Below the title, there are three input fields: 'Usuario', 'Contraseña', and 'Núm. Documento'. Each field has a small icon to its left. At the bottom of the form, there are two buttons: 'Aceptar' with a checkmark icon and 'Cancelar' with an 'X' icon.

FIGURA 10
FORMULARIO DE IDENTIFICACION DE USUARIO VÍA WEB PARA
ACCEDER A INFORMACIÓN DE NOTAS DE ESTUDIANTES DE LA
UNA - PUNO

Fuente: www.unap.edu.pe / (2011)

Al ingresar al portal de la universidad, y seleccionar consultar notas en la figura 10 aparece el formulario para ingresar los siguientes: usuario, contraseña, número de documento y luego aceptar; entonces el sistema permitirá identificar al usuario y acceder a la información requerida por el estudiante. Observándose que el estudiante tiene que estar matriculado y tener un código y contraseña de acceso. La información contenida es referente a los cursos en los que se encuentra matriculado, siendo necesario resaltar que el sistema académico de la Universidad Nacional del Altiplano es flexible, entonces cada estudiante lleva diferentes cursos, lo cual hace que la información del sistema actual sea personalizada a cada estudiante. Entonces cada estudiante interactúa y realiza las consultas requeridas.

La figura 11 muestra el sistema académico de la UNA - Puno para los docentes, quienes pueden realizar el registro de notas en el sistema ingresando las notas obtenidas por los estudiantes matriculados en los cursos que ellos imparten y para los docentes también se despliega el formulario de identificación de usuario.

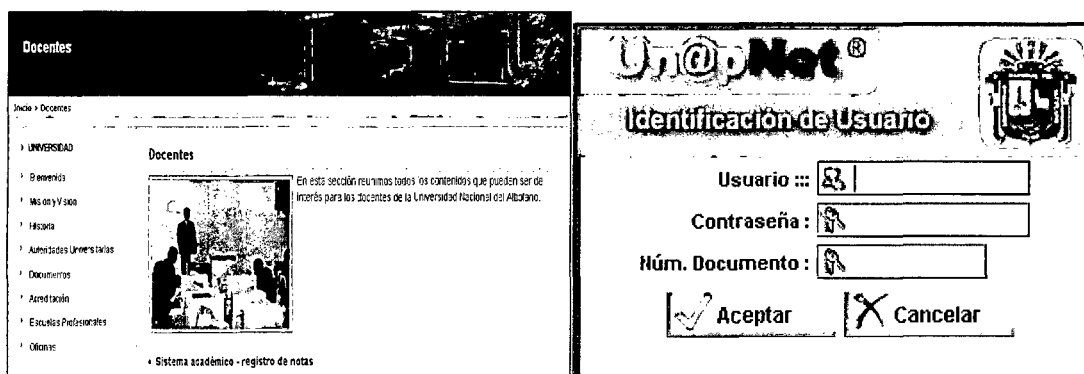


FIGURA 11

DOCENTE CON ACCESO VIA WEB PARA SELECCIONAR ENLACE Y FORMULARIO PARA EL REGISTRO DE NOTAS DE ESTUDIANTES DE LA UNIVERSIDAD NACIONAL DEL ALTIPLANO – PUNO

Fuente: www.unap.edu.pe / (2011)

La Figura 11 contiene en primer lugar el enlace que permitirá al docente seleccionar el enlace registro de notas, luego de ello presentará un formulario para ingresar los datos de usuario, contraseña, numero de documento y aceptar lo ingresado; para posteriormente seleccionar el curso, estudiante y el registro de notas respectivo.

En las figuras 9, 10 y 11 se observa la interfaz y formularios de acceso al sistemas de información académico vía tecnología web, acordes a las necesidades de docentes y estudiantes; donde ambos requieren un usuario y contraseña de acceso para ingresar notas o acceder a información académica respectivamente, entonces se mencionará algunos servicios que brinda este sistema:

- Ingreso de notas de los estudiantes al sistema de información académico de la Universidad Nacional del Altiplano
- Proceso de búsqueda de información en el sistema y constatación de notas por parte del estudiante
- Registro de notas almacenado en la base de datos, etc.

La web proporciona una infraestructura de comunicaciones que permite el acceso a multitud de información y servicios desde casi cualquier ordenador conectado, incluyendo teléfonos móviles como PDAs (Peñalver, 2005).

4.1.2. Análisis del Sistema Propuesto

El sistema propuesto permitirá brindar a los estudiantes un sistema sencillo y rápido, así como dar un valor agregado a los servicios académicos de la Universidad Nacional del Altiplano mediante el uso de dispositivos móviles existentes. El sistema propuesto plantea que los estudiantes puedan acceder a información de sus notas mediante un teléfono móvil, con este sistema los estudiantes pueden ahorrar tiempo; puesto que no siempre en todo momento se puede contar con una computadora que tenga acceso a internet. Los requisitos que debe cumplir el sistema para el estudiante, son los siguientes:

- Ingreso al sistema de consulta de notas (SCN)
- Selección del curso sobre el que desea conocer información
- Presentación de información por teléfono móvil
- Constatación de notas por parte del estudiante

La Figura 12 indica la arquitectura del sistema de consulta de notas propuesto en dispositivos móviles con Java Micro Edition (JME), modelo que permitirá identificar componentes del sistema y como se visualizará la información en el dispositivo móvil al extraer información de la base de datos.

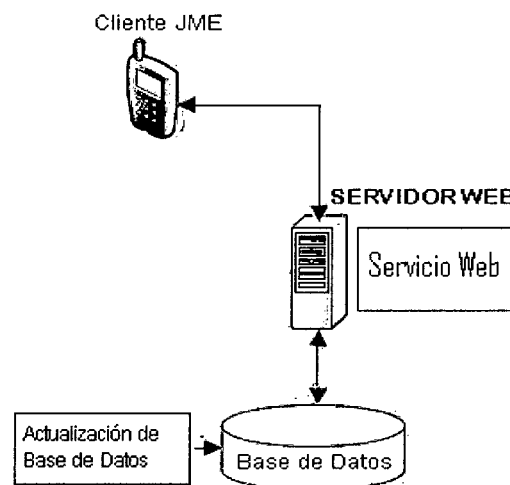


FIGURA 12
ARQUITECTURA: SISTEMA DE CONSULTA DE NOTAS PROPUESTO

Fuente: (Bertuzzi, 2007)

La Figura 12 de la arquitectura del sistema de consultas propuesto complementa servicios al sistema vía web, donde se observa la arquitectura cliente/ servidor; considerándose como cliente JME el dispositivo móvil quien solicitará información y el servidor web procesa la información para lo cual accederá a la base de datos para obtener la información y dar respuesta al cliente JME. Esta arquitectura no considera otros servicios como puede ser Bluetooth, MMS, SMS y otros; puesto que se basa en enviar una solicitud y esperar la respuesta del servidor.

El modelo de la arquitectura propuesta es similar al trabajo de diseño e implementación de un servicio de localización y visualización de mapas utilizando JME para dispositivos móviles y herramientas de libre distribución menciona que: para la descarga de mapas el servicio de Bluetooth sirve de puente entre el cliente y el servidor web. (Bertuzzi, 2007).

El sistema de consulta de notas propuesto emplea un servidor donde se tiene la instalación del servlet y el administrador de la base de datos, entonces si el docente ingresa notas permitirá actualizar la base de datos y los estudiantes realizan las consultas de notas; en ambos casos el sistema permitirá ahorro de tiempo, mejorar el tiempo de espera en un momento determinado con un sistema que les permita acceder a información en cualquier momento y lugar innovando el sistema académico actual de la universidad.

La Figura 13 presenta los pedidos a través de un dispositivo móvil (celular) al sistema de ventas AnywERP.

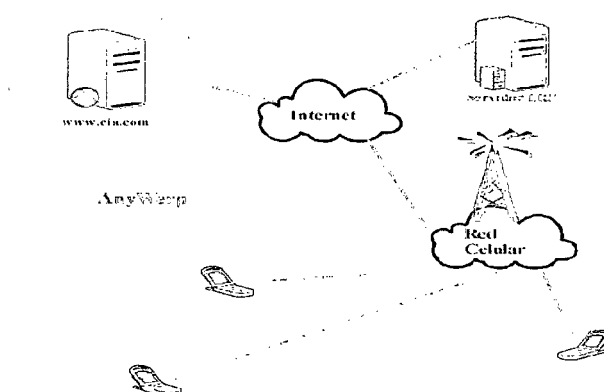


FIGURA 13

SISTEMA DE VENTAS AnywERP

Fuente: <http://colombia.acambiode.com/producto>

La figura 13 permite observar que varios usuarios pueden acceder al servidor a través de sus dispositivos móviles, empleando el servicio de internet conjuntamente con las redes de telefonía celular.

El sistema AnywERP sugiere una temática distinta, pero se destaca el acceso a información; que permitió proponer el sistema similar pero en este caso para la consulta de notas, como alternativa de solución orientada a la gestión académica para registrar y brindar información sobre las notas de los estudiantes de la escuela profesional de ingeniería de sistemas y posteriormente ampliar el sistema para otras escuelas profesionales de la Universidad Nacional del Altiplano; información consultada a través de dispositivos móviles utilizando el servicio de internet.

Las soluciones móviles están mostrando sus beneficios para la mejora de gestión en las empresas en la mejora de la productividad, en la creación de nuevos servicios. Los dispositivos móviles se han convertido en una plataforma rica para el consumo de contenidos digitales (Soto, 2008).

Campusmovil.net es una aplicación que plantea desarrollar un tipo de plataforma inexistente en el ecosistema académico y que puede funcionar como un excelente complemento de los campus virtual tradicionales (Kuklinski, 2009).

El desarrollo del proyecto se realizó considerando las fases de la metodología ICONIX y considerando que se cuenta con el portal universitario con diferente información sobre la universidad, se desarrolló el sistema académico de consulta de notas para dispositivos móviles.

4.1.3. Análisis de requerimientos del sistema de consulta de notas

Sobre la base del análisis del sistema de consultas de notas en la Universidad Nacional del Altiplano - Puno, e identificadas las ventajas del sistema propuesto como una nueva alternativa para docentes y estudiantes, entonces se identifica la necesidad de desarrollar un software para dispositivos móviles, donde este sistema académico debe ser capaz de:

- Acceder al sistema con nombre de la persona y clave de acceso
- Mostrar datos de la persona
- Consultar información sobre los cursos que tiene como carga académica el docente
- Registrar las notas obtenidas por los estudiantes para que el docente actualice los datos del rendimiento académico de estudiantes
- Mostrar información de horarios para los docentes sobre su carga académica
- Mostrar información de los cursos en los que encuentra matriculado el estudiante
- Mostrar información de calificaciones a los estudiantes

Utilización de recursos

Teniendo en cuenta que el sistema de consulta de notas es una aplicación web para móviles, se requiere un servidor, que deberá cumplir con ciertas especificaciones mínimas, mencionadas a continuación:

- Procesador: se debe contar con un procesador Intel Core de 133 MHz
- Sistemas Operativos: se debe contar con Windows XP o Windows 7.
- Memoria del servidor Web y Base de Datos: como mínimo se requiere 128 MB de RAM pero se recomienda 256 MB o superior.
- Espacio de disco duro: se requiere 210 GB de espacio para la aplicación Web
- Base de datos: Motor de base de datos MySQL.

Restricciones de Diseño

El producto deberá ser desarrollado bajo un esquema de tres capas:

- Capa de Presentación; deberá desarrollarse usando la tecnología web para el acceso a la información.
- Capa de Lógica de Negocio
- Capa de Servicio de Datos; deberá ser desarrollada usando MySQL.

Interfaz del Sistema

El sistema debe ser diseñado de tal manera que el usuario pueda ser autodidacta, se requiere del uso de un estándar de interfaz de manera que las

operaciones comunes en varias pantallas se encuentren en la misma ubicación y la distribución sea similar para que el usuario sienta homogeneidad en la navegación. Esta característica permitirá que el usuario se familiarice con la interfaz rápidamente debido a que el entorno de trabajo es similar a los utilizados por el usuario con los celulares.

Tiempo de respuesta de la aplicación

El tiempo de respuesta de una aplicación es vital para el usuario. La espera de procesamiento rápido de información que permita realizar su labor de la mejor manera.

4.1.3.1. Modelo de dominio

El análisis de requerimientos nos brinda el primer modelo de dominio; es decir, la definición del modelo estático del sistema académico para dispositivos móviles que se presenta en la figura 14, donde se definió las clases del modelo de dominio, para lo cual se elaboro un listado de los posibles objetos o clases de dominio:

- Persona
- Docente
- Estudiante
- Cursos
- Carga academica

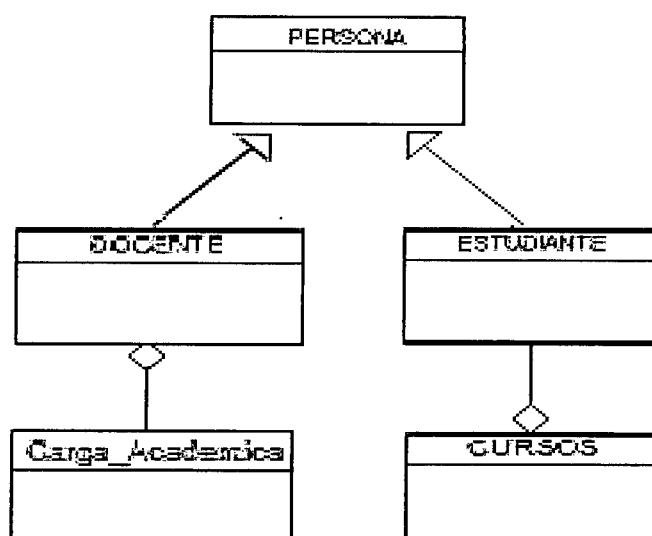


FIGURA 14
MODELO DE DOMINIO DEL SISTEMA DE CONSULTAS DE NOTAS

Fuente: Elaboración propia

La figura 14 muestra la clase persona que presenta relaciones de generalización con las clases docente y estudiante; donde la clase docente establece una asociación con carga académica y la clase estudiantes establece otra asociación con cursos.

El modelo de dominio permitió definir las clases que inicialmente se modeló en el sistema de consulta de notas; no definiéndose clases de asociación, posteriormente el diagrama de clases se reformulara para su implementación.

El modelo de dominio es la tarea de descubrir los objetos (las clases), estos representan cosas y conceptos. Dentro del proceso de ICONIX, el modelo de dominio activado involucra fuera de los requisitos de los datos construir un modelo estático del dominio del problema pertinente al sistema propuesto (Cerna y Crisologo, 2008).

4.1.3.2. Prototipo rápido

El prototipo es un software provisional que permite ver al usuario el contenido y apariencia del futuro software, donde los prototipos rápidos permitirán verificar que cumplan con los requisitos funcionales planteados. La figura 15 identifica la interfaz de los prototipos iniciales del sistema de consulta de notas propuesto:

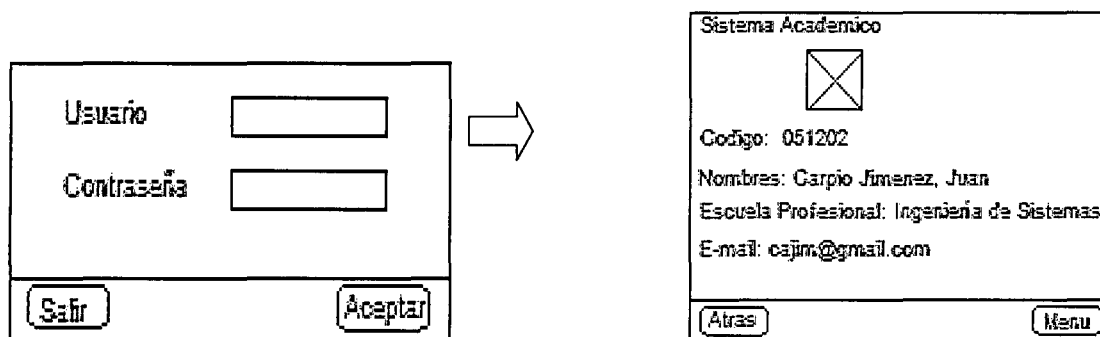


FIGURA 15
PROTOTIPOS DE IDENTIFICACION E INFORMACION DEL USUARIO DEL SISTEMA DE CONSULTAS DE NOTAS

Fuente: Elaboración propia

La figura 15 sugiere el prototipo de identificación e información del usuario del sistema de consulta de notas; considerando para la identificación el usuario y la contraseña similar al sistema académico vía web y al acceder al sistema se muestra información solicitada por el usuario del sistema.

El modelo de prototipos comienza con la recolección de requisitos, el desarrollador y el cliente definen los objetivos globales para el software, originándose un diseño rápido que se centra en una representación de esos aspectos software que sean visibles para el usuario/cliente. De este diseño surge la construcción de un prototipo y este es evaluado por el cliente/ usuario (Pressman, 2004)

4.1.3.3. Diagrama de casos de uso

Los diagramas de use case del sistema de consulta de notas en dispositivos móviles, que se presenta en la Figura 16 permitió identificar los actores y los procesos del sistema;

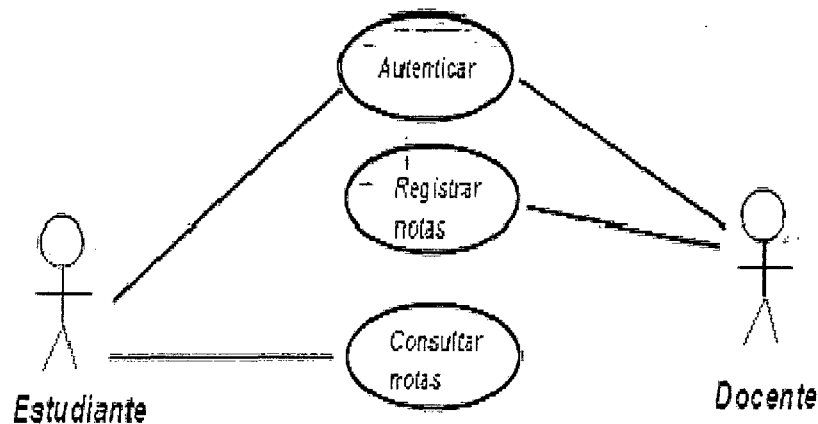


FIGURA 16

DIAGRAMA DE CASOS DE USO DEL SISTEMA DE CONSULTAS DE NOTAS

Fuente: Elaboración propia

El diagramas de use case inicial de la figura 16 identifica los casos autenticar donde ambos actores se relacionan con el caso autenticar, luego el caso registrar notas relación específica para el actor docente y caso consultar notas se relaciona con el actor estudiante. El diagrama de casos de uso permite definir la interacción de los usuarios con el sistema.

Un aspecto importante de iconix es que un requisito se distingue explícitamente de un caso de uso. En este sentido un caso de uso describe un comportamiento; un requisito describe una regla para el comportamiento. Además un caso de uso uno o más requisitos funcionales (San Martín, 2009).

4.2. DISEÑO DEL SISTEMA DE CONSULTA DE NOTAS EN MOVILES CON JME (JAVA MICRO EDITION)

4.2.1. Esquema general del sistema de consulta de notas

La presente de investigación plantea el diseño del sistema de consulta de notas con el uso de la arquitectura Cliente/ Servidor en capas, para lo cual la división de la aplicación en capas que se denomina arquitectura de tres capas:

Capa de Presentación (*Dispositivo móvil*): Esta capa servirá para como interfaz gráfica que facilite al usuario el uso del sistema.

Capa de Negocio: Esta capa servirá para centralizar la lógica de negocio (modelo).

Capa de datos: ésta capa servirá para almacenar los datos (base de datos)

La figura 17 presenta el modelo de tres capas de la arquitectura cliente/servidor donde se distingue las capas de: presentación, negocio y datos.

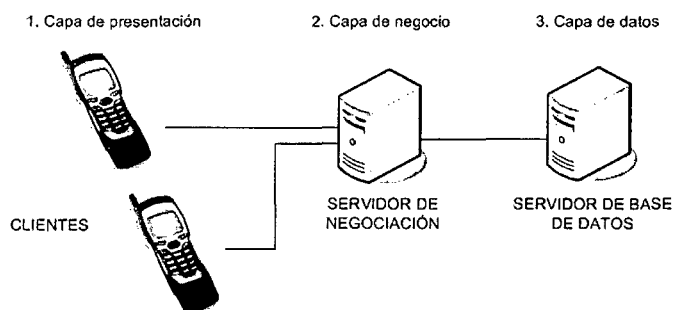


FIGURA 17
MODELO DE 3 CAPAS DE LA ARQUITECTURA CLIENTE- SERVIDOR

Fuente: <http://upload.wikimedia.org>

La figura 17 distingue en el modelo de tres capas de la arquitectura cliente/servidor las funciones de la capa presentación que presenta la interfaz de usuario por medio de una aplicación cliente, la capa denominada negocio realiza los cálculos definidos por la aplicación servidor y la capa de datos que sirve para el almacenamiento de datos requeridos por la capa negocio.

En esta arquitectura la presentación, el procesamiento de la aplicación y la gestión de los datos son procesos lógicamente separados que se ejecutan sobre procesadores diferentes (Somerville, 2005).

El sistema de consulta de notas en dispositivos móviles tiene dos componentes principales:

- Cliente móvil
- Servidor

El cliente móvil: Se implemento como una aplicación JME que se ejecutará inicialmente en un emulador de dispositivo móvil, para luego instalarlo en el dispositivo móvil (celular) que realizará la solicitud del usuario (docente y/o estudiante) del rendimiento académico de los estudiantes de los cursos en los que se matriculo en el semestre académico.

El servidor: Atiende las peticiones de lectura y escritura; es decir, cambio de valor de las variables del sistema de consultas por parte del cliente, para lo cual realiza transacciones de consulta.

La figura 18 distingue los componentes del sistema de consulta de notas y la interacción entre ellos en el diagrama de bloques del sistema de consulta de notas.

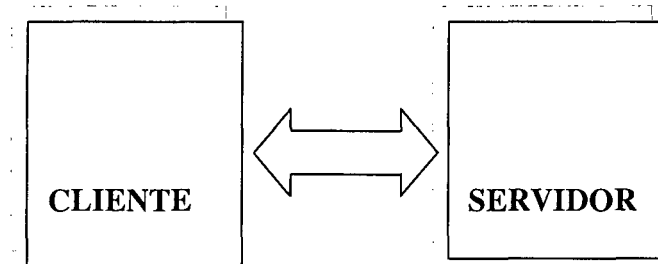


FIGURA 18
DIAGRAMA DE BLOQUES DEL SISTEMA DE CONSULTA DE NOTAS

Fuente: Elaboración propia

La figura 18 evidencia la arquitectura cliente servidor de tres capas que físicamente se relaciona entre dos componentes: cliente y servidor del sistema de consulta de notas; donde el servidor integra el servidor de aplicaciones y el servidor de las bases de datos.

4.2.2. Funcionamiento del sistema de consulta de notas

Las tareas de comunicación de la aplicación móvil, generarán peticiones HTTP respectivas (lectura/escritura), que serán enviadas al proveedor GSM, a través de la red GPRS, estas serán encaminadas a través de Internet al servidor de aplicaciones Web, Apache Tomcat, vía TCP/IP. El servidor de aplicaciones java (servlets) contiene valores en una tabla Mysql en un computador (servidor).

La figura 19 muestra la topología cliente / servidor syncframework y los flujos que se dan cuando existe comunicación entre los componentes.

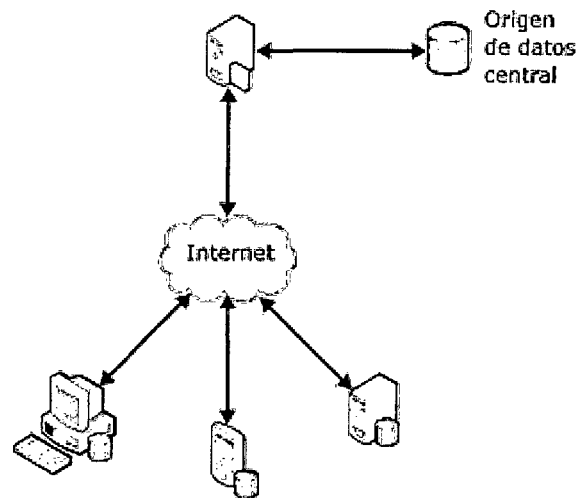


FIGURA 19
TOPOLOGIA CLIENTE/ SERVIDOR SYNCFRAMEWORK

Fuente: <http://msdn.microsoft.com/en-us/library>

En la figura 19 se observa que un cliente (ordenador o dispositivo móvil) solicita una petición al servidor; donde la topología cliente / servidor syncframework permite distinguir que el servidor web y un servidor base de datos donde se originan los datos solicitados por el cliente que se enviara a la capa presentación mostrando la información al cliente empleando la red internet.

Ahora nos encontramos con que la arquitectura se basa en la red internet. En lo que se refiere hacia la evolución a internet computing de todas las arquitecturas lo que más de acuerdo está con internet es la arquitectura de tres capas (De Pablo, 2004).

4.2.3. Arquitectura de comunicación

Para desarrollar la tarea de comunicación de datos desde el cliente móvil hacia el servidor, se han identificado dos elementos cliente y servidor dentro de la estructura de comunicación del sistema. Estos poseen una comunicación cliente-servidor independiente de la implementación física o de red, como la figura 20 muestra a continuación:

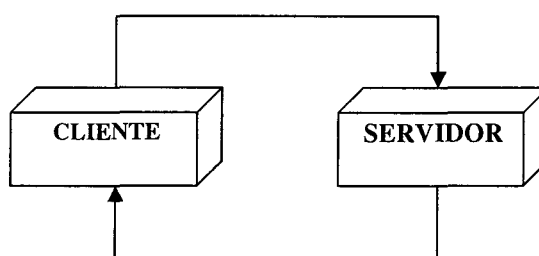


FIGURA 20
ESTRUCTURA DE COMUNICACIÓN DEL SISTEMA DE CONSULTA DE NOTAS
Fuente: Elaboración propia

La figura 20 de la estructura de comunicación del sistema de consulta de notas distingue la vista del diagrama de despliegue del nodo cliente y nodo servidor; así como las relaciones existentes.

Esta distribución física permite identificar un computador para el servidor y los dispositivos móviles para el cliente. Entonces se detalló las especificaciones necesaria para estos componentes; pero adecuados para el desarrollo de las aplicaciones con dispositivos móviles.

4.2.4. Capa servidor

Esta capa recibe peticiones de lectura y/o escritura de la capa cliente, que se realizan sobre una tabla MySQL, y así mismo ésta capa lee y escribe desde la misma tabla MySQL, que se encuentra implementado en un computador personal con las siguientes características:

- Procesador Core 2 Duo de 2.0 GHZ
- Memoria RAM de 2 GB
- Monitor VGA
- Sistema Operativo: Windows XP SP2
- Adaptador de red
- Acceso directo a Internet.
- Servidor HTTP Apache
- Servidor de aplicaciones Apache Tomcat
- Sistema manejador de base de datos MySQL

La tarea fundamental de esta capa es procesar las peticiones de lectura y escritura de la capa cliente, que accederán a la base datos que contiene información solicitada por la capa cliente.

La capa servidor posee una capa que realiza la tarea independiente de encargarse de atender peticiones de la capa cliente denominada SCCP que se muestra en la figura 21. A este nivel se precisa de una base de datos como nexo de comunicación.

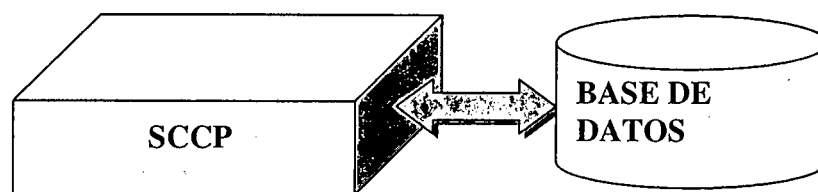


FIGURA 21
SUBCAPA DE LA CAPA SERVIDOR

Fuente: Elaboración propia

La figura 21 identifica una subcapa servidor que se interrelaciona con el gestor de la base de datos, que permite la lectura/ escritura de los datos respecto a la solicitud del cliente.

Subcapa de comunicación

Esta subcapa tiene la tarea de establecer la comunicación de datos con la fuente de datos de lectura y/o escritura que es una tabla Mysql, también denominada destino de datos. En esta capa se ha implementado una aplicación Java, que debe establecer conexión con el sistema de base de datos MySQL. Para lo cual emplea el modelo de la arquitectura cliente-servidor que se observa en la figura 22:

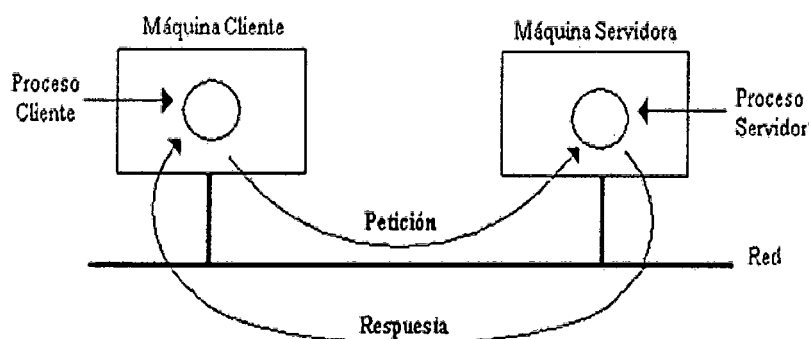


FIGURA 22
MODELO ARQUITECTURA CLIENTE / SERVIDOR

Fuente: Elaboración propia

La figura 22 que describe los componentes físicos máquina cliente y máquina servidor y las relaciones respecto a la solicitud del cliente y la respuesta del servidor en la red; permitiendo comprender que se tiene que dar los procesos en ambos componentes.

Un problema potencial en el diseño de arquitecturas de tres capas, es que la separación de la interfaz de usuario, la lógica de gestión de procesamiento y la lógica de datos no siempre es obvia. Algunas lógicas de gestión de procesamiento pueden aparecer en las tres capas (Gonzales, 2000)

En esta capa se implementó la aplicación Java que debe establecer conexión con el sistema de base de datos MySQL. El servidor MysqI se desarrolló en el entorno de programación para Java NetBeans, y los requisitos para desarrollar esta aplicación fueron:

- JVM, plataforma de soporte de aplicaciones Java.
- Acceso (local o remoto) a una Base de datos MySQL.

Las principales ventajas del servidor MySQL:

- Bajo Costo, implementado con Software Libre (no se pagan licencias por uso o desarrollo)
- Portabilidad, capacidad de funcionar en diferentes sistemas operativos sin realizar cambios sustanciales.

- La fuente de datos útiles para el usuario: Tabla Mysql puede estar contenida en el propio computador, servidor local o remoto. Siempre que se tenga acceso este (Internet).
- Es fuente para aplicaciones de tipo Web Service.

Funcionamiento básico del Servidor Mysql

1. Establece comunicación con el gestor de base de datos MySQL
2. Lee valores de la tabla MySQL
3. Fin

En la arquitectura cliente/servidor, todo el procesamiento de la base de datos queda consolidado en una sola computadora, y esta consolidación permite un alto grado de integridad de los datos. Al procesar las solicitudes a la base de datos en el servidor, si las restricciones han quedado definidas en el servidor, se pueden aplicar consistentemente (Gonzales, 2000).

4.2.5. Capa de atención al cliente

La implementación se realizó en un computador servidor y la tarea fundamental de esta subcapa consiste en atender las peticiones (HTTP) de la capa cliente, peticiones de lectura y escritura desde y hacia la tabla que se utiliza en la capa SCCP.

En esta capa se ha desarrollado aplicaciones de servidor, o también denominados Servlets, que serán contenidas en un servidor de aplicación, que es requisito tener instalado en el computador servidor. El servlet atenderá y realizará las peticiones de lectura de datos, los servlets se han desarrollado en el entorno de programación de java NetBeans, para el desarrollo y ejecución de estas aplicaciones se precisa:

- La instalación del servidor de aplicaciones Apache Tomcat, configurar y habilitar el puerto 8080 para la escucha de peticiones clientes a través del internet.
- La instalación del JVM (Java Virtual Machine).

Cuando se invoca una aplicación que ejecuta un ciclo de lectura de los valores de la tabla Mysql, construye una trama y luego la envía al cliente, para que este interprete la trama y muestre los valores al usuario.

Los servlets son más eficientes que los programas (CGI) porque son cargados de una sola vez en la memoria, y cada demanda es manejada por un hilo de la máquina virtual de Java, no por el sistema operativo (Maidana, 2008).

Dada las características de hardware limitada de los celulares, existe una versión de Java optimizada para uso en móviles, denominada Java Micro Edition (Alvarez, 2008). La nueva plataforma para la programación de aplicaciones Java en dispositivos limitados (Almenares, 2003).

4.2.6. Capa Cliente

La comunicación de datos y la estructura planteada hace factible implementar aplicaciones móviles. La capa cliente se ha implementado desarrollando una aplicación móvil JME (Java Micro Edition) que es ejecutado en un emulador de dispositivo móvil, con soporte Java. Al desarrollar la aplicación móvil, se ha considerado las especificaciones Java que soporta un dispositivo móvil:

- MIDP 2.0
- CLDC 1.0

La aplicación fue desarrollada en el entorno de programación NetBeans cuya aplicación móvil realiza la tarea fundamental de mostrar información académica a los estudiantes respecto a la solicitud requerida al sistema. La aplicación móvil que se implementa en esta capa cuya ejecución se desarrolla en el emulador móvil realiza las siguientes funciones:

- Mostrar en las pantallas información del estudiante, así como también información sobre de las asignaturas en las que se encuentra matriculado.
- Desempeño ante los eventos del teclado

Un midlet es una aplicación Java diseñada para ser ejecutada en un dispositivo móvil, más específicamente un Midlet tiene como las clases principales el perfil MIDP y la configuración CLDC (Gomez, 2005).

4.3. IMPLEMENTACION DEL SISTEMA DE CONSULTAS DE NOTAS EN MOVILES CON JME

4.3.1. Software de desarrollo del sistema de consulta de notas

Para el desarrollo del presente sistema se empleó lo siguiente:

- Sistema Operativo: Windows XP
- Documentación: Microsoft Office 2007 o Microsoft Office 2010
- Modelado de Sistema con UML: Rational Rose
- Programación: NetBeans
- Contenedor de Servlet: TomCat
- Java Platform Micro Edition Software Development Kit
- Gestor de Base de Datos: Mysql

Requerimientos para el Funcionamiento de Prototipos

Hardware Mínimo

- Computadora: Core Duo 2.33 Ghz, con 1,98 Gb de Memoria
- Disco Duro de 140 Gb.
- Monitor, con una resolución mínima de 1360 por 768 pixeles.
- Tarjeta de Video de 128 Mb de Memoria.

Plataformas de Ejecución: Windows XP ó Windows 7

4.3.2. Diagrama de clases del sistema de consulta de notas

El diseño de clases representa el modelo estático del sistema, para ello se emplea esquemas lógicos de bases de datos. A continuación la figura 23 muestra el diagrama de clases para la aplicación en JME, que se utilizarán para el funcionamiento del sistema.

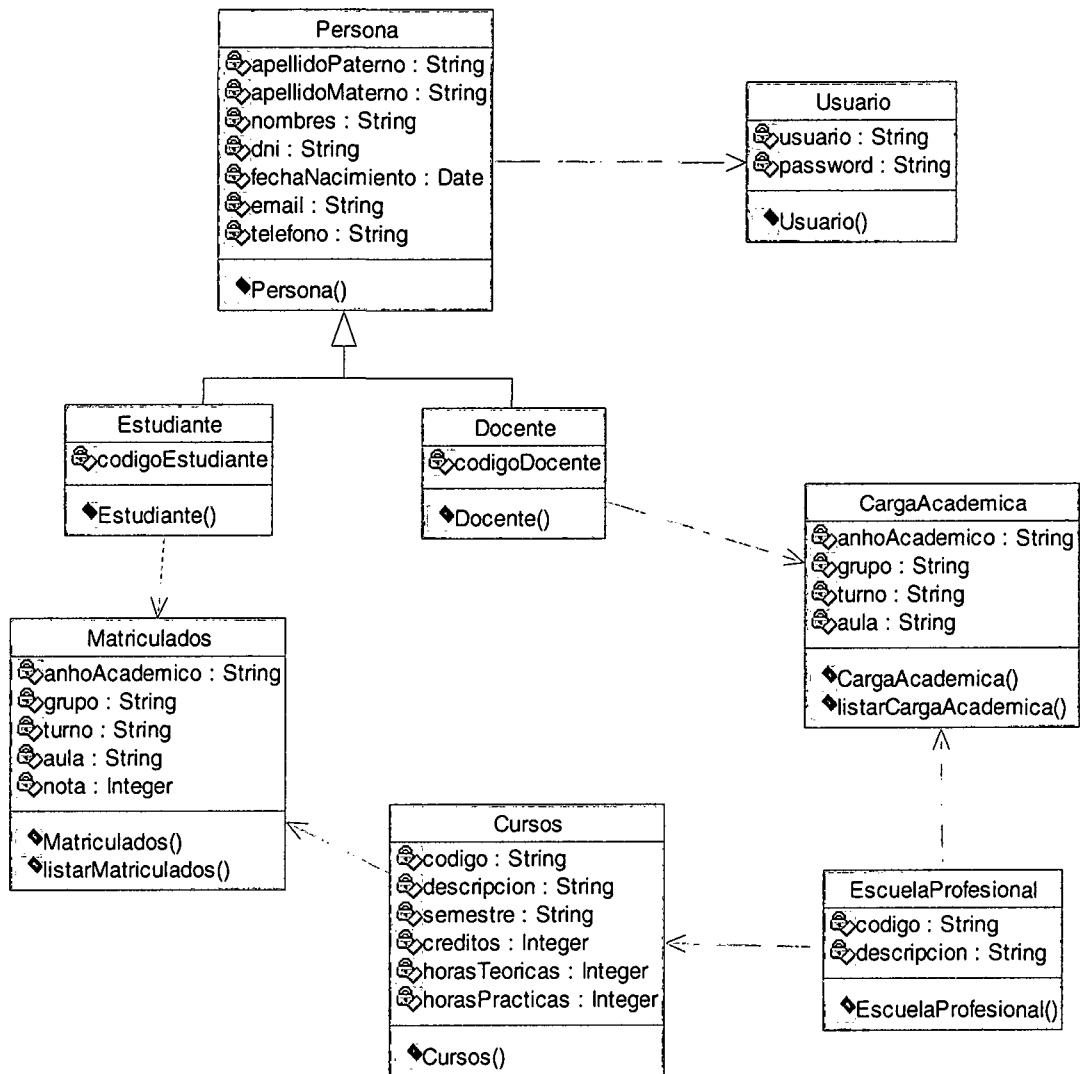


FIGURA 23
DIAGRAMA DE CLASES DEL SISTEMA DE CONSULTA DE NOTAS

Fuente: Elaboración propia

La figura 23 del diagrama de clases del sistema de consulta de notas, permitió abstraer del modelo de dominio las siguientes clases: persona, docente, estudiante, cursos, cargaacademica y se agregaron las clases usuarios, matriculados y escuela profesional; donde estos últimos permitieron actualizar el diagrama de clases.

El objetivo principal del diagrama de clases es mostrar la relación de las clases principales con otras clases adicionales que sirven de apoyo para el correcto funcionamiento de la aplicación (Espinoza, 2009).

4.3.3. Análisis de Casos de Uso del sistema de consulta de notas

La funcionalidad del sistema de consulta de notas ha sido representada a través de casos de uso, que permite una descripción general acorde a los requerimientos funcionales del sistema propuesto. A continuación en la figura 24 se muestra los casos de uso más importantes en el sistema.

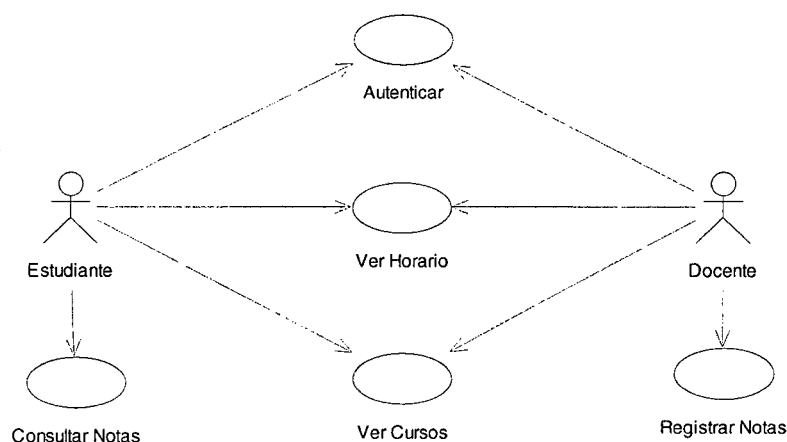


FIGURA 24
DIAGRAMA DE CASOS DE USO DEL SISTEMA DE CONSULTA DE NOTAS

Fuente: Elaboración propia

En el diagrama de casos de uso del sistema permite determinar los actores del sistema: docente y estudiante, que se define como usuario del sistema que se observa en la figura 25:

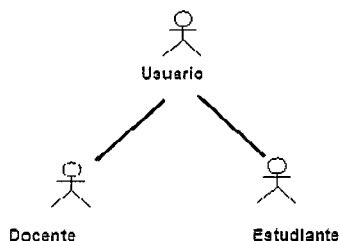


FIGURA 25
DIAGRAMA DE ACTORES DEL SISTEMA DE CONSULTA DE NOTAS

Fuente: Elaboración propia

En las figuras 24 y 25 se identifican los casos de uso y actores del sistema de consulta de notas de los cuales se tiene como casos de uso generales: autenticar, ver horario y ver cursos relacionados con los actores docente y estudiante. El diagrama de casos de uso del sistema permite realizar un modelo del funcionamiento del sistema de consulta de notas para los dispositivos móviles; permitiendo que el usuario comprenda las tareas que realizará el sistema.

Es importante resaltar que los diagramas de casos de uso no están pensados para representar el diseño y no puede describir los elementos internos de un sistema. Los diagramas de casos de uso sirven para facilitar la comunicación con los futuros usuarios del sistema, y con el cliente, y resultan especialmente útiles para determinar las características necesarias que tendrá el sistema permiten determinar las características necesarias del sistema (Ramírez, 2002).

El cuadro 7 presenta la descripción de los actores del sistema de consulta de notas en dispositivos móviles:

CUADRO 7
DESCRIPCION DE ACTORES DEL SISTEMA DE CONSULTA DE NOTAS

Actores	Descripción	Casos de uso con los que se relaciona
Estudiante	Éste actor representa a la persona física que interactuará con el sistema. Las funciones que realizará este actor consisten en elegir las opciones que se aplicarán, y acceder a información académica.	Autenticar Ver Horarios Ver Cursos Consultar Notas
Docente	Éste actor representa a la persona física que interactuará con el sistema. Las funciones que realizará este actor consisten en registrar información de interés para el estudiante actualizando la información académica.	Autenticar Ver Horarios Ver Cursos Registrar notas

Fuente: Elaboración propia

Casos de Uso

Muestra los procesos que se emplean en el sistema actual y los cuales deben ser automatizados para la aplicación móvil y estos son:

El cuadro 8 presenta la descripción de los casos de uso del sistema de consulta de notas:

CUADRO 8
DESCRIPCION DE CASOS DE USO DEL SISTEMA DE CONSULTA DE
NOTAS

Casos de Uso	Descripción	Actores con los que se relaciona
Autenticar	El usuario del sistema que puede ser de tipo Docente o estudiante ingresa su nombre de usuario que es código de alumno y/o docente según el usuario que este ingresando al sistema luego coloca su contraseña o clave.	Estudiante Docente
Ver Horario	El usuario del sistema puede verificar su horario de clases en caso de ser de la categoría estudiante, y su horario de trabajo en caso de ser de la categoría Docente.	Estudiante Docente
Ver cursos	El usuario del sistema puede acceder al menú y seleccionar cursos, entonces el sistema le permitirá elegir entre los cursos que dicta en caso de ser Docente y elegir el curso que desea informarse en el caso de ser estudiante	Estudiante Docente
Consultar notas	El estudiante dentro del sistema revisa la información de la calificación sobre la asignatura que desea información.	Estudiante
Registrar Notas	El docente dentro del Sistema tiene la relación de estudiantes en los cursos que dicta, entonces puede ingresar las notas correspondientes a cada estudiante.	Docente

Fuente: Elaboración propia

4.3.4. Comportamiento del sistema de consulta de notas

En esta sección se muestran los eventos que realizan los actores sobre el sistema y la interacción entre los objetos del sistema, para lo cual se usa una técnica denominada diagrama de secuencia.

Diagrama de Secuencia

El diagrama de secuencia es una técnica que muestra la secuencia de mensajes entre objetos durante un escenario concreto, cada objeto viene dado por una barra vertical, el tiempo transcurre de arriba hacia abajo, cuando existe demora en la atención se puede indicar usando una línea oblicua.

La interacción entre los objetos del sistema, se muestra en las figuras 26, 27 y 28 que corresponden al diagrama de secuencia del sistema de consulta de notas. Los diagramas de secuencia son los siguientes:

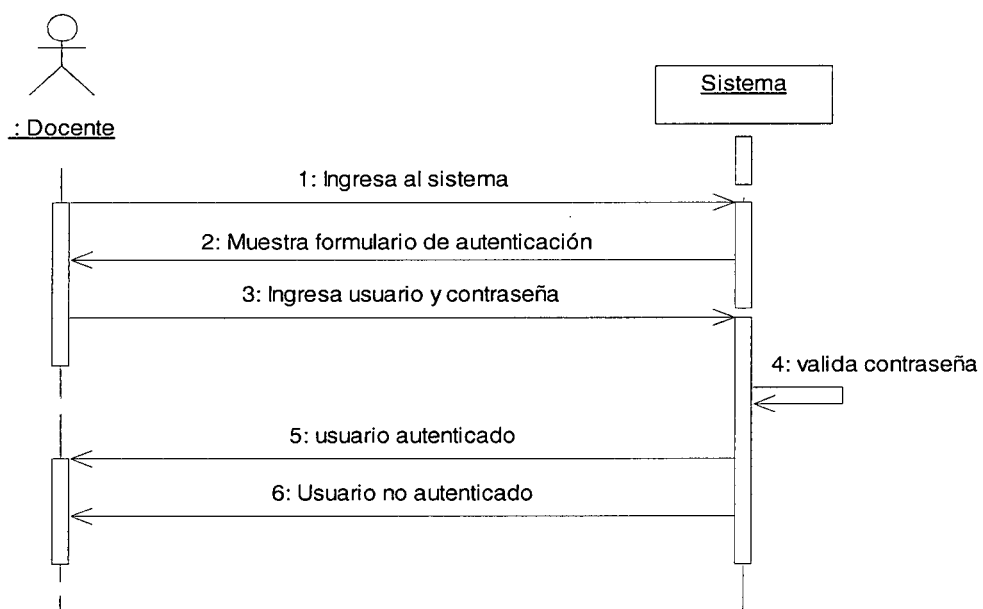


FIGURA 26

DIAGRAMA DE SECUENCIA DE CASO DE USO AUTENTICAR

Fuente: Elaboración propia

El diagrama de secuencia de la figura 26, el usuario (docente / estudiante) inicia el sistema, iniciada la aplicación este muestra un formulario permite ingresar un código de usuario y contraseña para autenticarse previa validación.

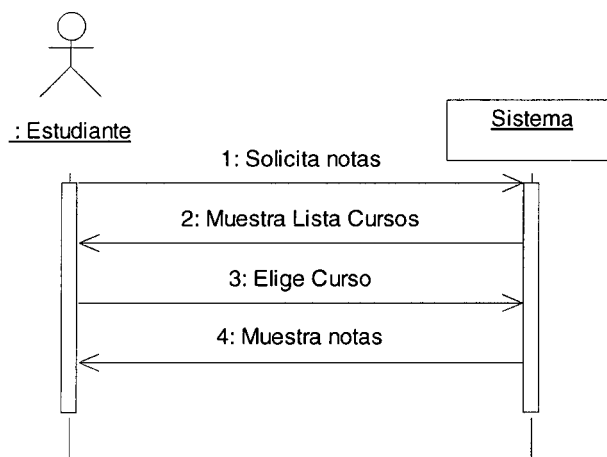


FIGURA 27
DIAGRAMA DE SECUENCIA DE CASO DE USO CONSULTAR NOTAS

Fuente: Elaboración propia

El diagrama de secuencia del caso de uso consultar notas que se observa en la figura 27, el usuario (estudiante) solicita notas, el sistema muestra los cursos en los que está matriculado el estudiante, y el estudiante elige el curso que desea verificar notas y enseguida el sistema muestra las notas del curso elegido.

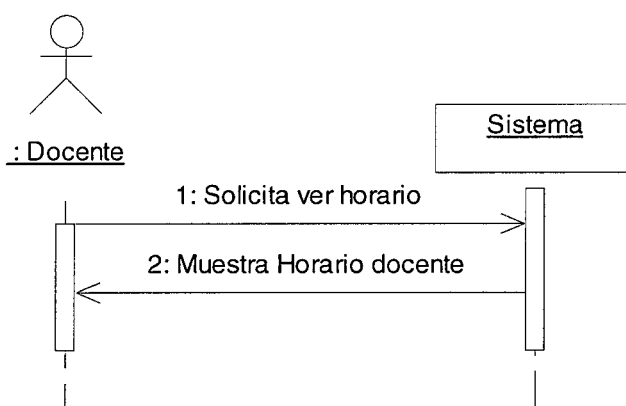


FIGURA 28
DIAGRAMA DE SECUENCIA DE CASO DE USO VER HORARIO

Fuente: Elaboración propia

El diagrama de secuencia del caso de uso ver horario de la figura 28, el usuario (docente o estudiante) solicita ver su horario, entonces el sistema muestra el horario del docente o del estudiante dependiendo la categoría de usuario del sistema.

Se ha presentado los diagramas de secuencia de los tres casos de uso que tiene el sistema de consulta de notas. La cantidad de diagramas de secuencia depende de la aplicación a desarrollar; puesto que si se cuenta con mayor cantidad de casos de usos y actores en la aplicación que se desee desarrollar, esto se plasmara en el diagrama de casos de uso incrementando las tareas que realice el sistema y los diagramas de secuencia respectivamente.

Los diagramas de secuencia constituyen una de las herramientas más efectivas para modelar la interacción entre los objetos de un sistema (Espinoza, 2009)

4.3.5. Diseño de base de datos del sistema de consulta de notas

El sistema de requiere de un medio de almacenamiento de datos, que proveerá los datos solicitados por los usuarios que emplean la aplicación que brinda el servicio de consulta de notas, para ello se realizó el diseño de base de datos respectivo.

El diseño de la base de datos realizado para el sistema de consulta de notas para dispositivos móviles, con la descripción de la tabla y los campos definidos en la misma, que a continuación los cuadros 9, 10, 11, 12 y 13 presentan:

CUADRO 9
DESCRIPCION DE LA TABLA ESCUELA PROFESIONAL

EscuelaProfesional					
Descripción		Almacena datos de la Escuela Profesional			
Nombre	Tipo Columna	Opción Nulo	Descripción	PK	FK
idEscuelaProfesional	tinyint(2)	unsigned NOT NULL	Identificador de la tabla de Escuela Profesional	Si	No
Nombres	Varchar(150)	NotNull	Indica el nombre de la Escuela Profesional	No	No

Fuente: Elaboración propia

CUADRO 10
DESCRIPCION DE LA TABLA CURSO

Cursos					
Descripción		Almacena datos de los cursos			
Nombre	Tipo Columna	Opción Nulo	Descripción	PK	FK
idEscuelaProfesional	tinyint(2)	Unsigned NOT NULL	Identificador de la tabla de Escuela Profesional	Si	Si
codigoCurso	Char(6)	NotNull	Indica el código del curso	Si	No
nombre	Varchar(50)	NotNull	Indica el nombre del curso	No	No
Créditos	int(2)	Null	Indica los créditos del curso	No	No
semestre	Varchar(5)	nonnull	Indica el semestre en el que se dicta el curso	No	No
horasTeoricas	int(2)	Null	Indica las horas teóricas del curso	No	No
horasPracticas	int(2)	Null	Indica las horas prácticas del curso	No	No
numeroHoras	int(2)	Null	Indica el número total de horas por semana	No	No

Fuente: Elaboración propia

CUADRO 11
DESCRIPCION DE LA TABLA USUARIOS

Usuarios					
Descripción		Almacena datos de las personas, tanto estudiantes como docentes			
Nombre	Tipo Columna	Opción Nulo	Descripción	PK	FK
Código	Varchar(15)	NotNull	Es el código que los identifica tanto al docente como al estudiante	Si	No
Usertype	Varchar(15)	NotNull	Indica el tipo de usuario que representa, puede tomar valores como: 'estudiante' y 'docente'	Si	No
idEscuelaProfesional	tinyint(2)	unsigned NOT NULL	Identificador de la tabla de Escuela Profesional	No	Si

paterno	Varchar(25)	Null	Indica el apellido paterno de la persona	No	No
materno	Varchar(25)	Null	Indica el apellido materno de la persona	No	No
nombres	Varchar(25)	NotNull	Indica nombres de persona	No	No
Dni	Varchar(15)	Null	Indica el número de documento nacional de identidad de la persona	No	No
fechaNacimiento	Date	Null	Indica la fecha de nacimiento de la persona	No	No
email	Varchar(50)	NotNull	Indica el correo electrónico de la persona	No	No
Teléfono	Varchar(50)	Null	Indica el teléfono de la persona	No	No
Passwd	Varchar(50)	Null	Indica la contraseña de la persona, está encriptada	No	No

Fuente: Elaboración propia

CUADRO 12

DESCRIPCION DE LA TABLA CURSOSMATICULADOS

CursosMatriculados					
Descripción		Almacena datos de los cursos matriculados por los estudiantes			
Nombre	Tipo Columna	Opción Nulo	Descripción	PK	FK
idCursosMatriculados	int(10)	unsigned NOT NULL	Identificador de la table de cursos matriculados	Si	No
idEscuelaProfesional	tinyint(2)	unsigned NOT NULL	Identificador de la tabla de Escuela Profesional	No	Si
codigoMatricula	Varchar(15)	NotNull	Es el código que los identifica al estudiante	No	Sí
codigoCurso	Char(6)	NotNull	Indica el código del curso	No	No
anhoAcademico	char(6)	NotNull	Indica el año académico en que se realiza un determinado curso	No	No
grupo	Varchar(5)	Null	Indica el grupo al que está matriculado el estudiante	No	No
nota1	int(3)	Unsigned Null	Indica la nota del Primer Examen del curso	No	No
nota2	int(3)	Unsigned Null	Indica la nota del Segundo Examen del curso	No	No
nota3	int(3)	Unsigned Null	Indica la nota de los Trabajos Encargados	No	No
nota4	int(3)	Unsigned Null	Indica la nota de las Exposiciones	No	No

Fuente: Elaboración propia

CUADRO 13
DESCRIPCION DE LA TABLA CARGAACADEMICA

CargaAcademica					
Descripción	Almacena datos de los cursos asignados al docente para el dictado de cursos				
Nombre	Tipo Columna	Opción Nulo	Descripción	PK	FK
idEscuelaProfesional	tinyint(2)	unsigned NOT NULL	Identificador de la tabla de Escuela Profesional	Si	Si
codigoDocente	Varchar(15)	NotNull	Es el código que los identifica al Docente	Si	Sí
codigoCurso	Char(6)	NotNull	Indica el código del curso	No	No
anhoAcademico	char(6)	NotNull	Indica el año académico en que se realiza un determinado curso	No	no
grupo	Varchar(5)	Null	Indica el grupo al que está matriculado el estudiante	No	No
turno	Varchar(5)	Null	Indica el turno en el que se desarrolla el curso	No	No
turno	Char(1)	Null	Indica el turno en el que se desarrolla el curso	No	No
aula	Varchar(100)	Null	Indica el aula en el que se desarrolla el curso	No	No

Fuente: Elaboración propia

Los cuadros 9, 10, 11, 12 y 13 describen las tablas diseñadas para la base de datos del sistema de consulta de notas; cada una de las tablas incluye una descripción de la tabla donde se indica que datos se almacenan en la tabla, el nombre se refiere a la definición de los campos de las tablas, el tipo de dato, la descripción de los campos definidos y si tienen clave principal o foránea.

La representación de los datos en los dispositivos físicos del ordenador no es una tabla, sino registros estructurados, de una manera más o menos eficiente. La tabla, por tanto una abstracción de su representación física, la cual es desconocida por el usuario final (Gomez y et. al, 2006)

Diagrama Entidad- Relación del sistema de consulta de notas

El modelo relacional o conocido también modelo entidad – relación (E/R) es un modelo de datos de los SGBD actuales. En la figura 29 se muestra el diagrama entidad – relación con las tablas y las relaciones del modelo relacional que conforman el sistema.

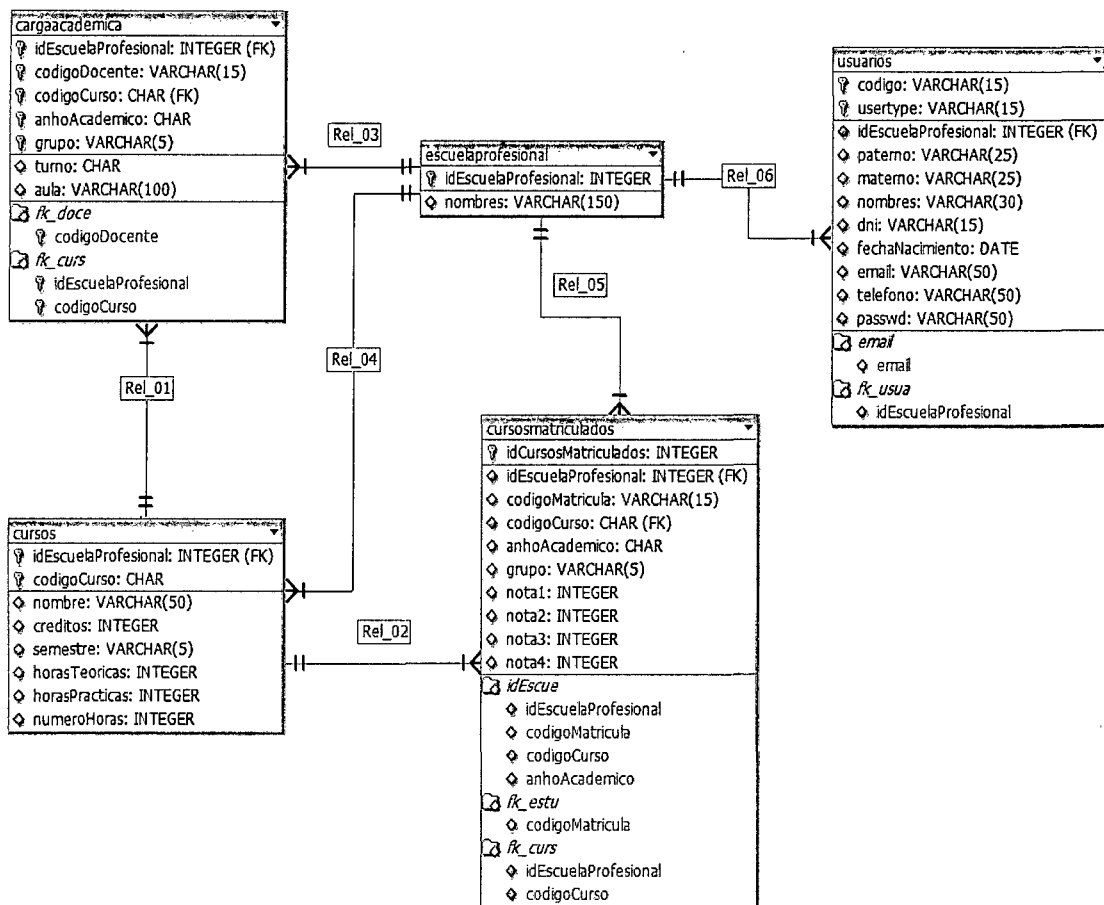


FIGURA 29

DIAGRAMA DE ENTIDAD – RELACION DEL SISTEMA DE CONSULTA DE NOTAS

Fuente: Elaboración propia

La figura 29 incluye las tablas, los campos de las tablas, los métodos y las relaciones existentes en el modelo de base de datos relacional del sistema de consulta de notas que se administrara en el servidor.

Un SGBDR (sistema gestor de bases de datos relacional) almacena la información en tablas organizadas lógicamente que se enlazan definiendo relaciones y contienen datos (Lamarca, 2011)

4.3.6. Diseño de Interfaz - Mapas de Navegación

El mapa de navegación muestra como manipula la persona (docente/ estudiante) el sistema de consulta de notas, donde la representación jerárquica muestra el numero de niveles necesarios para realizar una tarea deseada.

La figura 30 del mapa de navegación según el actor docente, tal como se observa:

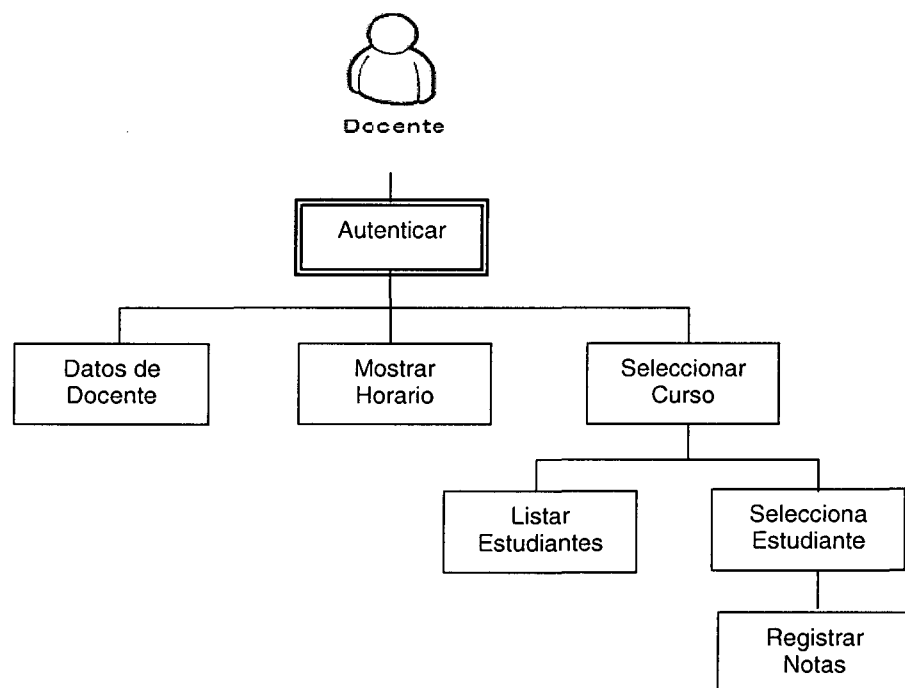


FIGURA 30
MAPA DE NAVEGACIÓN DE DOCENTE EN EL SISTEMA

Fuente: Elaboración propia

En la figura 30 del mapa de navegación del docente en el sistema se observa que este actor puede realizar más acciones en el sistema que el actor estudiante; la navegación en el sistema implica más tiempo en el docente.

La figura 31 del mapa de navegación del estudiante en el sistema se presenta de la siguiente manera:

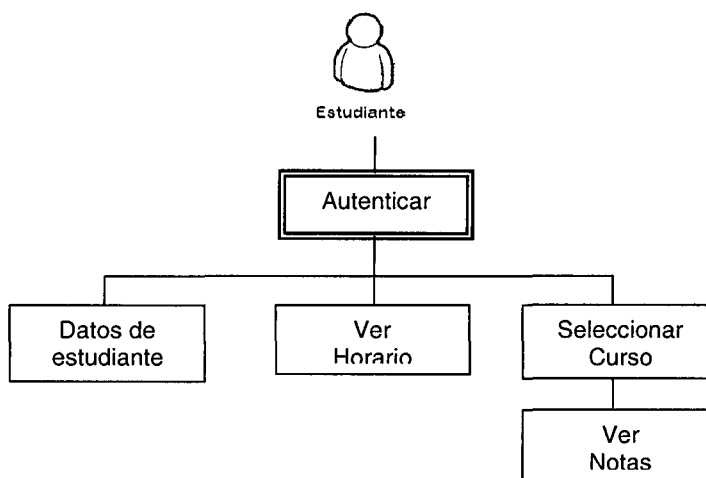


FIGURA 31
MAPA DE NAVEGACIÓN DE ESTUDIANTE EN EL SISTEMA

Fuente: Elaboración propia

El mapa de navegación del estudiante distingue menos acciones para el actor estudiante; puesto que el estudiante tan sólo podrá visualizar sus notas y horario de sesiones.

Estos diagramas permitirán ver el orden en el que se presentará de las pantallas con los contenidos de información para los usuarios del sistema.

4.3.7. Implementación del Sistema de Consulta de Notas

El análisis de requerimientos, el análisis y diseño del sistema académico, ha considerado como entidades principales a: estudiantes, docentes y cursos; así como las relaciones entre estas entidades como las matriculas a los cursos, evaluaciones de cursos matriculados, horarios, y carga académica. Este modelado lógico de la base de datos ha permitido administrar los datos utilizando la aplicación en JME (servlet) en los dispositivos móviles (midlet) para que el estudiante tenga acceso a las notas; esto a su vez permite la implementación de una interfaz para el dispositivo móvil mediante el uso de las tecnologías estándar.

En la implementación del modelo de las bases de datos del sistema de consulta de notas en móviles con Java Micro Edition se empleo software libre como DBMS MySQL; el cual se tiene instalado para la conexión a la base de datos del servidor.

En la implementación del sistema de consulta de notas académico se ha utilizado el lenguaje de programación Java, puesto que java se caracteriza por su seguridad, robustez y portabilidad; entonces luego de haber diseñado la base de datos, ha sido necesario la creación de los modelos de negocio de la aplicación móvil para el sistema académico, tales como la aplicación MobileNotas que contiene las clases necesarias para permitir realizar el registro y la consulta de notas, que la figura 32 muestra a continuación:

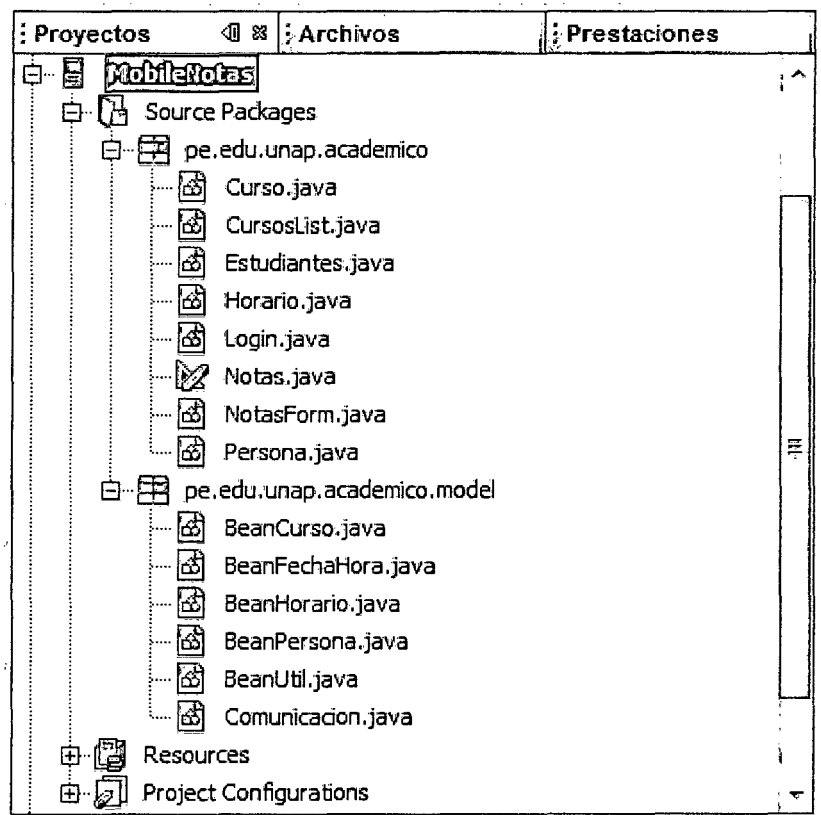


FIGURA 32

**DIAGRAMA DE CLASES DEL SISTEMA IMPLEMENTADO EN JAVA
DEL SISTEMA DE CONSULTA DE NOTAS**

Fuente: Elaboración propia

La figura 32 del diagrama de clases implementado en Neatbeans para el modulo del sistema académico como la aplicación MobileNotas, contiene las clases necesarias para el funcionamiento de la aplicación.

El objetivo es realizar la aplicación móvil en J2ME, con la cual el alumno pueda consultar sus notas desde cualquier lugar sin necesidad de dirigirse al establecimiento educativo, como así también el desarrollo de la aplicación de administración en Java, mediante la cual los usuarios autorizados (por ej. profesores, adscriptos, etc.) puedan agregar, editar y eliminar tanto usuarios como alumnos (Maidana, 2008).

MIDLETS

Localización de la Aplicación

El descubrimiento de una aplicación es el proceso por el cual un usuario a través de su dispositivo localiza un MIDlet. El usuario debe ser capaz de ver la descripción del MIDlet a través de un enlace que, una vez seleccionado, inicializa la instalación del MIDlet. Si éste enlace se refiere a un archivo JAR, el archivo y su URL son enviados al AMS del dispositivo para empezar el proceso de instalación. Si el enlace se refiere a un archivo JAD se realizan los siguientes pasos:

1. El descriptor de la aplicación (archivo JAD) y su URL son transferidos al AMS para empezar la instalación. Este descriptor es usado por el AMS para determinar si el MIDlet asociado puede ser instalado y ejecutado satisfactoriamente, en el siguiente fragmento de código se puede ver el enlace URL para la descarga de una aplicación.

```
<html>
<head>
  <title>MobileNotas</title>
</head>
<body>
  <a href="http://www.nspasac.com/MobileNotas.jad">MobileNotas.jad</
a>
</body>
</html>
```

2. Este archivo JAD debe ser convertido al formato Unicode antes de ser usado. Los atributos del JAD deben ser comprensibles, acorde con la sintaxis de la especificación MIDP, y todos los atributos requeridos por la especificación MIDP deben estar presentes en el JAD, en el siguiente

fragmento de código se muestra la descripción de una aplicación JME en formato JAD generado por el compilador.

```
MIDlet-1: Notas, /unap.png, pe.edu.unap.academico.Notas  
MIDlet-Jar-Size: 168308  
MIDlet-Jar-URL: MobileNotas.jar  
MIDlet-Name: MobileNotas  
MIDlet-Vendor: Zulema  
MIDlet-Version: 1.0  
MicroEdition-Configuration: CLDC-1.1  
MicroEdition-Profile: MIDP-2.0
```

3. El usuario debería de tener la oportunidad de confirmar que desea instalar el MIDlet. Asimismo debería de ser informado si se intenta instalar una versión anterior del MIDlet o si la versión es la misma que ya está instalada. Si existen problemas de memoria con la ejecución del MIDlet se intentarían solucionar liberando componentes de memoria para dejar espacio suficiente.

Instalación de MIDlets

La instalación de la aplicación es el proceso por el cual el MIDlet es descargado al dispositivo y puede ser utilizado por el usuario. Cuando existan múltiples MIDlets en la aplicación que deseamos descargar, el usuario debe ser avisado de que existen más de uno.

Durante la instalación, el usuario debe ser informado del progreso de ésta y se le debe de dar la oportunidad de cancelarla. La interrupción de la instalación debe dejar al dispositivo con el mismo estado que cuando se inició ésta. Veamos cuáles son los pasos que el AMS sigue para la instalación de un MIDlet:

1. Si el JAD fue lo primero que descargó el AMS, el MIDlet debe tener exactamente la misma URL especificada en el descriptor.
2. Si el servidor responde a la petición del MIDlet con un código 401 (Unauthorized) o un 407 (Proxy Authentication Required), el dispositivo debe enviar al servidor las correspondientes credenciales.
3. El MIDlet y las cabeceras recibidas deben ser chequeadas para verificar que el MIDlet descargado puede ser instalado en el dispositivo. El usuario debe ser avisado de los siguientes problemas durante la instalación:
 - Si no existe suficiente memoria para almacenar el MIDlet, el dispositivo debe retornar el Código de Estado (Status Code) 901.
 - Si el JAR no está disponible en la URL del JAD, el dispositivo debe retornar el Código 907 - 903
 - Si el JAR recibido no coincide con el descrito por el JAD, el dispositivo debe retornar el Código 904.
 - Si el archivo manifiesto cualquier otro no puede ser extraído del JAR, o existe algún error al extraerlo, el dispositivo debe retornar el Código 907.
 - Si los atributos "MIDlet-Name", "MIDlet-Version" y "MIDlet Vendor" del archivo JAD, no coinciden con los extraídos del archivo manifest del JAR, el dispositivo debe retornar el Código 905.
 - Si la aplicación falla en la autenticación, el dispositivo debe retornar el código 909.
 - Si falla por otro motivo distinto del anterior, debe retornar el Código 911.

- Si los servicios de conexión se pierden durante la instalación, se debe retornar el Código es posible.

La instalación se da por completa cuando el MIDlet esté a nuestra disposición en el dispositivo, o no haya ocurrido un error irre recuperable. Los archivos descritos anteriormente deben tener la siguiente forma para que se instale correctamente.

```
/MobileNotas/bin>dir  
25/02/2011 04:14 p.m. 151 MobileNotas.html  
25/02/2011 11:13 p.m. 365 MobileNotas.jad  
25/02/2011 11:56 p.m. 91,249 MobileNotas.jar  
25/02/2011 11:13 p.m. 323 MANIFEST.MF
```

En este caso, los atributos comunes tienen el mismo valor tanto en el archivo de manifiesto como en el archivo JAD. Para que el suite sea portable, es preciso que el archivo JAR esté codificado usando la codificación iso-8859-1, ya que se precisa que todas las implementaciones de MIDP soporten esta codificación (Gomez, 2006).

Actualización de MIDlets

La actualización se realiza cuando instalamos un MIDlet sobre un dispositivo que ya contenía una versión anterior de éste. El dispositivo debe ser capaz de informar al usuario cual es la versión de la aplicación que tiene instalada.

Cuando comienza la actualización, el dispositivo debe informar si la versión que va a instalar es más nueva, más antigua o la misma de la ya instalada y debe

obtener verificación por parte del usuario antes de continuar con el proceso. En cualquier caso, un MIDlet que no posea firma no debe de reemplazar de ninguna manera a otro que sí la tenga.

Ejecución de MIDlets

Cuando un usuario comienza a ejecutar un MIDlet, el dispositivo debe invocar a las clases CLDC y MIDP requeridas por la especificación MIDP. Si existen varios MIDlets presentes, la interfaz de usuario debe permitir al usuario seleccionar el MIDlet que desea ejecutar.

Eliminación de MIDlets

Los dispositivos deben permitir al usuario eliminar MIDlets. Antes de eliminar una aplicación el usuario debe dar su confirmación. El dispositivo debería avisar al usuario si ocurriese alguna circunstancia especial durante la eliminación del MIDlet. Por ejemplo, el MIDlet a borrar podría contener a otros MIDlets, y el usuario debería de ser alertado ya que todos ellos quedarían eliminados.

El gestor de aplicaciones es el software encargado de gestionar los midlets. Este software reside en el dispositivo y es el que nos permite ejecutar, pausar y destruir nuestras aplicaciones J2ME. Las principales funciones que realiza un gestor de aplicaciones son la carga, ejecución, actualización y desinstalación de un midlet (Navarro, 2005)

Pantallas del sistema de Consulta de Notas

La Interfaz del sistema de consulta de notas se desarrollo en base a los requerimientos del sistema, cuenta con formularios que forman parte del sistema de consulta de notas para acceder al sistema académico donde se encuentra información sobre el rendimiento académico de los estudiantes de la Escuela Profesional de Ingeniería de Sistemas de la Universidad Nacional del Altiplano - Puno.

El sistema permitirá el acceso al sistema, tanto a los docentes como a los estudiantes, y las pantallas que se muestran a continuación podrán ser manejadas dependiendo del usuario que acceda al sistema.


Pantallas en Móvil (Celular)

Iniciando la aplicación teléfono celular (móvil), se muestra una pantalla y se despliega una forma de acceso tanto para docente y estudiante, si el usuario es un estudiante ingresa los datos necesarios en este formulario para acceder al sistema de consultas de la siguiente manera:

- Escribir el nombre de usuario (que corresponde al código del estudiante)
- Escribir la contraseña
- Seleccionar categoría: Estudiante
- Presionar la opción que indica Aceptar para ingresar al sistema

Pantallas de Interacción del Estudiante

La pantalla de la figura 33. Formulario de ingreso al sistema de consulta de notas muestra como acceder al sistema con la categoría estudiante:



The image shows a mobile application interface for a 'Sistema Académico' (Academic System). The screen is titled 'Sistema Académico' and features a logo at the top. Below the title, there is a user login form with the following fields:

- Usuario:** A text input field containing the value '051201'.
- Contraseña:** A password input field with masked characters (dots).
- Categoría:** A dropdown menu currently set to 'Estudiante'.

At the bottom of the form, there are two buttons: 'Salir' (Exit) on the left and 'Aceptar' (Accept) on the right. Below the form is a standard mobile keypad with numeric keys (1-9, *, #) and function keys like 'SHIFT' and 'SPACE'.

FIGURA 33

FORMULARIO DE INGRESO AL SISTEMA DE CONSULTA DE NOTAS

Fuente: Elaboración propia

La interfaz gráfica está diseñada para permitir el acceso solamente a usuarios que se encuentran registrados y validados en la base de datos, para implementar el sistema de consulta de notas se realizó esta interfaz y clases de java que verificará si el usuario está o no registrado, permitiendo elegir una categoría del usuario para acceder al sistema de consulta.

En el sistema académico vía web se considera como dato adicional para autenticar al usuario el número de documento de identidad; dato que no se ha considerado en el diseño de pantalla del sistema de consulta de notas.

El usuario, contraseña y número identificador del equipo son los datos que se utilizan para la autenticación en la base de datos a la que se conecta la aplicación servidor (Espinoza, 2009).

En lo referente al proyecto de integración de una aplicación móvil a una intranet de toma de asistencia estudiantil sólo se considero el ingreso de usuario y contraseña (García, 2005); debido a que en este caso sólo cuenta con un usuario que accede al sistema.

La figura 34. Identificación de datos del estudiante muestra información sobre datos del estudiante.

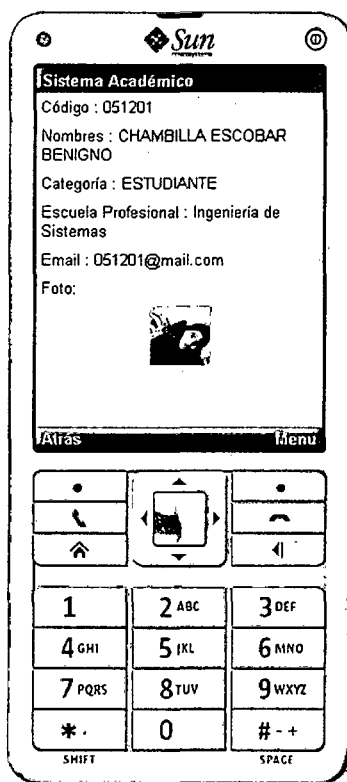


FIGURA 34

PANTALLA - IDENTIFICACION DE DATOS DEL ESTUDIANTE

Fuente: Elaboración propia

En la pantalla de identificación de datos del estudiante, se accede a una página que mostrará la siguiente información:

- Código del estudiante
- Apellidos y Nombres del estudiante
- Categoría
- Escuela Profesional
- Email (Correo electrónico)
- Foto (Archivo de imagen)

La información mostrada en la pantalla de datos del estudiante, son datos básicos que se debe almacenar el sistema de consulta de nota para mostrar al usuario y éste verifique que ha iniciado la aplicación correctamente. En todas las aplicaciones donde se solicita iniciar una sesión, es necesario contar con los datos generales del usuario.

Sin embargo en la pantalla que se presenta después de la autenticación en el proyecto aplicaciones web para celulares presenta directamente una pantalla con enlaces a exámenes, cursadas, parciales y certificados (García, 2007).

El sistema de consultar notas el estudiante presenta otra forma de navegación para lo cual tiene que presionar el botón de menú de opciones, la interfaz de las pantallas presenta una navegación diferente al estudiante donde una de ellas es similar al manejo del dispositivo móvil y la otra es similar a la navegación web.

El estudiante al haber revisado sus datos personales puede interactuar con el sistema de consulta de notas seleccionando del menú una opción y elegir entre ver su horario o acceder a los cursos en los que se encuentra matriculado presionando la opción que desea seleccionar.

La Figura 35 muestra la pantalla horarios del estudiante, donde el estudiante selecciono del Menú - Horario, y tiene acceso a información sobre el horario de clase de los cursos en los que se encuentra matriculado que se visualiza en una tabla.

Hora	Curso	Mater.	Mies.	Aula
07-08			SIS341	
08-09			SIS341	
09-10		IN0706		
10-11		IN0706		
11-12	IN0706	SIS341		IN0706
12-13	IN0706	SIS341		IN0706
15-16	SO0902			SO0902
16-17	SO0902			SO0902
17-18	AO0903		AO0903	AO0903
18-19	AO0903		AO0903	

SO0902: Diseño y Administración Avanzada
 SIS341: Algoritmos y Estructuras de Datos
 AO0903: Diagnóstico de Organizaciones
 IN0706: Taller de Sistemas de Información

FIGURA 35
PANTALLA - HORARIOS DEL ESTUDIANTE

Fuente: Elaboración propia

Los horarios de los cursos asignados al estudiante, es una tarea que se agrega en la aplicación que ayudara en la planificación de tiempo a los estudiantes, puesto que está brindando información necesaria para el estudiante.

Se observa que la tabla del horario no se ajusta a la pantalla; siendo necesario considerar que si se reduce el tamaño de fuente del texto no sería visible la información que se desea brindar; para ello es necesario agregar las barras de desplazamiento.

Dentro de las consideraciones de diseño de la interfaz móvil para un portal educativo, respecto a agilizar la interacción con el usuario (Díaz, 2007).

Si el estudiante selecciona del Menú la otra opción Cursos, se mostrará la lista de los cursos en los que se encuentra matriculado en el semestre, entonces el estudiante podrá seleccionar el curso sobre el cual desea conocer información de su rendimiento académico. La figura 36 permite observar lo siguiente:

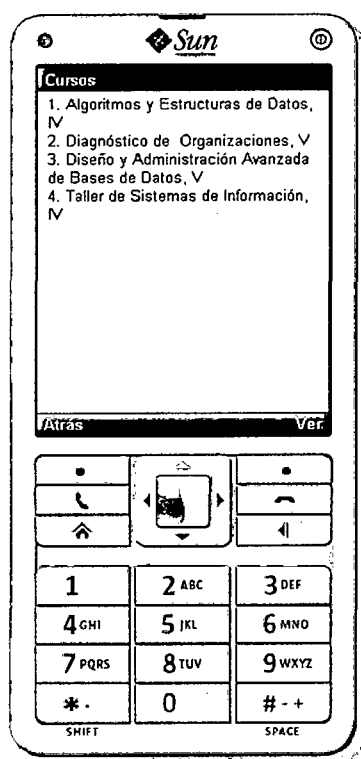


FIGURA 36

PANTALLA - CURSOS MATRICULADOS DEL ESTUDIANTE

Fuente: Elaboración propia

Los estudiantes se matriculan en diferentes cursos, puesto que la Universidad Nacional del Altiplano se encuentra con un sistema flexible; pero se considero el listado de cursos en la pantalla acorde a la matricula del estudiante.

Otras aplicaciones debido a la capacidad de la pantalla del celular abrevian los nombre por ejemplo prefieren denominarlo materia1 en lugar de realizar el listado con el nombre completo. El principio de familiaridad del usuario sugiere que los usuarios no deben ser obligados a adaptarse a una interfaz sólo porque sea conveniente para implementarla (Somerville, 2005).

La Figura 37 Pantalla de notas y promedio del curso seleccionado, muestra información del curso que el estudiante selecciono, al elegir la opción ver y no la opción salir; que le permite visualizar información del curso seleccionado. Esta pantalla muestra detalles del curso, que contiene los siguientes datos:

- Código de la asignatura
- Nombre de la Asignatura
- Semestre
- Horas : Teóricas , practicas y total de Horas
- Año académico , grupo
- Nota de Primer Examen
- Nota del Segundo Examen
- Trabajos Encargados
- Exposiciones
- Promedio

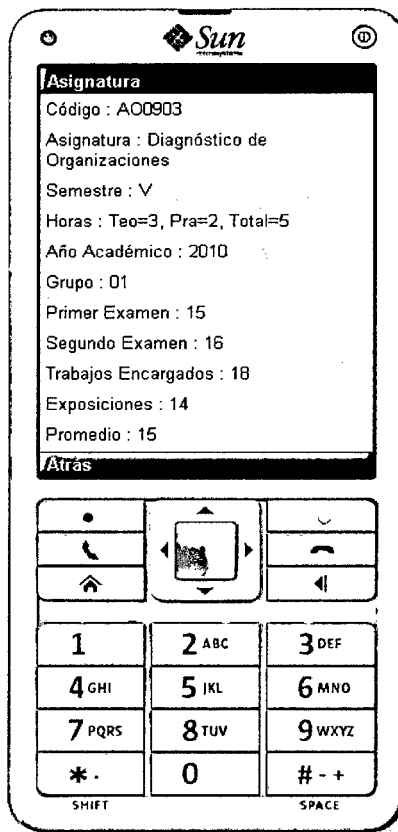


FIGURA 37

PANTALLA NOTAS Y PROMEDIO DE CURSO DEL ESTUDIANTE

Fuente: Elaboración propia

Esta pantalla permite observar en forma detallada información del rendimiento académico del estudiante; donde especifica las calificaciones obtenidas del estudiante sobre sus pruebas escritas, trabajos encargados y exposiciones, que obtiene el estudiante de la Escuela Profesional de Ingeniería de Sistemas de UNAP- Puno, cuyos datos son ingresados por el docente del curso. En este sistema de consulta de notas es acorde a la función promedio y los criterios de evaluación de los docentes; quienes tienen su sistema de evaluación no se considera en la presente aplicación.

Pantallas de Interacción del Docente

El docente se considera como otra categoría del sistema de consulta de notas utilizando Java Micro Edition a través de tecnologías móviles para obtener información del rendimiento académico de los estudiantes de la Escuela profesional de Ingeniería de Sistemas de la Universidad Nacional del Altiplano-Puno.

El docente utilizará el formulario que utilizó el estudiante para acceder al sistema, el único dato que variará será la categoría; que se tiene que realizar de la siguiente manera:

- Escribir el nombre de usuario (corresponde al código del docente)
- Escribir la contraseña
- Seleccionar categoría: Docente
- Presionar la opción que indica Aceptar para ingresar al sistema

En esta pantalla la figura 38 Formulario de ingreso al sistema de consulta de notas muestra como acceder al sistema con la categoría docente:

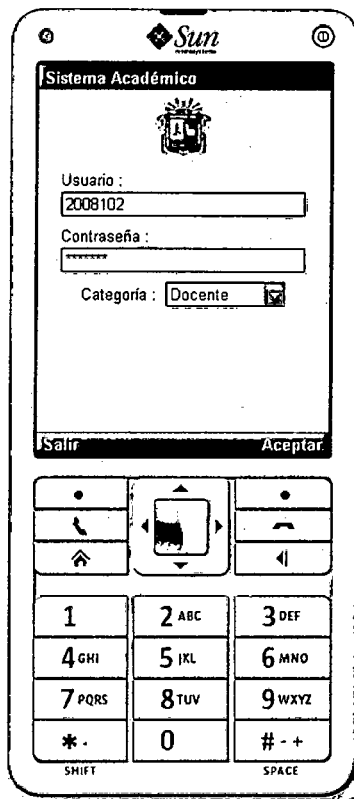


FIGURA 38

FORMULARIO DE INGRESO AL SISTEMA - CATEGORIA DOCENTE

Fuente: Elaboración propia

Los docentes también pueden iniciar la aplicación al igual que los estudiantes; pero es necesario resaltar que las tareas que realizan en la aplicación varía según la categoría de usuario que ingresa al sistema. Cabe destacar que el sistema de consulta de notas considera los niveles de seguridad, donde cada usuario dependiendo de la categoría puede realizar de manera restrictiva determinadas tareas en la aplicación, por ejemplo el estudiante no puede ingresar notas, esto sólo lo realiza el docente.

Respecto a facilitar la interacción con el usuario la entrada de datos dentro del portal se encuentra restringida a completar un formulario con datos básicos (Díaz, 2007).

La figura 39 Identificación de datos - docente. Al ingresar al sistema de consulta de notas se accede a una pantalla que mostrará la siguiente información:

- Código del docente
- Apellidos y Nombres del estudiante
- Categoría
- Escuela Profesional
- Email (Correo electrónico)
- Foto (Archivo de imagen)

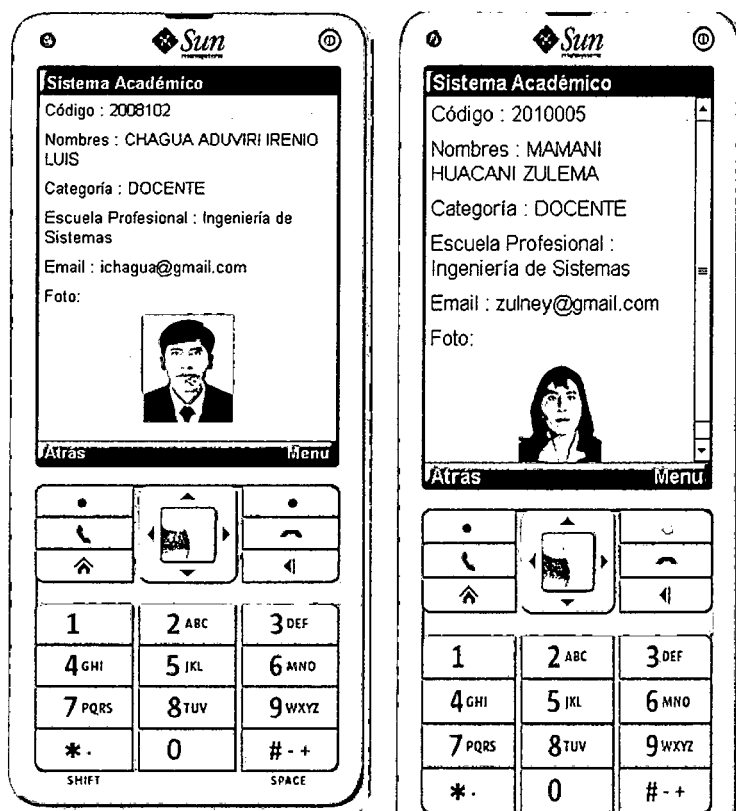


FIGURA 39

PANTALLA IDENTIFICACIÓN DE DATOS DEL DOCENTE

Fuente: Elaboración propia

El sistema de consulta de notas tiene almacenado datos necesarios de los docentes en la base de datos, otras aplicaciones sólo muestran los nombres y apellidos de los usuarios, pero los datos adicionales del usuario se encuentra en la pagina denominada perfil; considerando que la aplicación es para dispositivos móviles se estructura una navegación con el menor número de pantallas; puesto que esta aplicación está orientada a dispositivos móviles con capacidades limitadas.

El docente al verificar sus datos personales puede seleccionar del menú dos opciones, el horario de los cursos según su carga académica o acceder a los cursos que dicta. La Figura 40 Pantalla horario carga académica docente muestra la selección de la opción horario del menú.



FIGURA 40

PANTALLA HORARIO Y CARGA ACADÉMICA DEL DOCENTE

Fuente: Elaboración propia

En el menú, al seleccionar la opción cursos, la pantalla muestra la lista de cursos asignados según su carga académica, que le permite seleccionar un curso de la lista para ingresar las notas al sistema de consulta de notas, como se observa en la figura 41. Pantalla lista de cursos – carga académica docente.

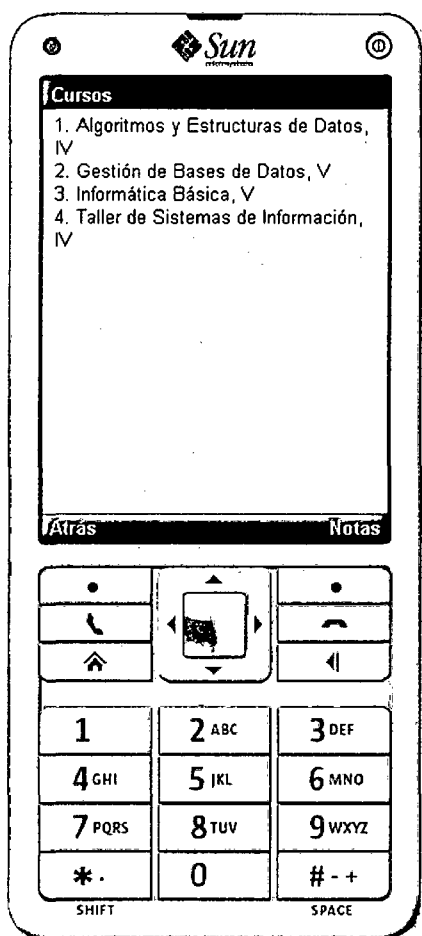


FIGURA 41

PANTALLA LISTA DE CURSOS DE CARGA ACADÉMICA DOCENTE

Fuente: Elaboración propia

El docente tiene la carga académica en diferentes semestres, y los cursos son diferentes para cada docente; si el docente dictara cursos adicionales, el sistema no incluye cursos no asignados en la carga académica.

Al seleccionar la opción curso del menú, se mostrará una pantalla con la lista de estudiantes matriculados en el curso, permitiéndole al docente ingresar las notas del estudiante que selecciono, como se puede observar en la figura 42

Lista de estudiantes – curso seleccionado.

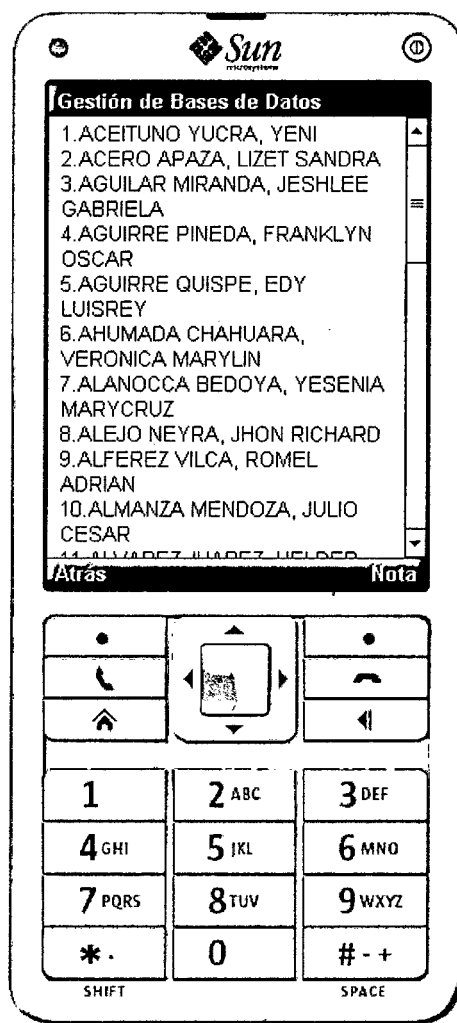


FIGURA 42

LISTA DE ESTUDIANTES DE CURSO SELECCIONADO

Fuente: Elaboración propia

Como se pudo observar el sistema de consulta de notas, permite al docente ingresar notas en diferentes semestres y cursos acorde a la carga académica. En caso que el docente no realizo el ingreso de notas de un estudiante; entonces el estudiante no podrá visualizar sus notas.

En la figura 43 Formulario de ingreso de notas de estudiante, se permitirá al docente ingresar las notas del estudiante que selecciono para ingresar notas, previa selección del curso. En este formulario se tienen los siguientes criterios de evaluación para ingresar datos:

- Primer Examen
- Segundo Examen
- Trabajos
- Exposiciones

The image shows a mobile phone screen with a form titled 'Nota' for a student named 'ACERO APAZA, LIZET SANDRA'. The form contains the following data:

Criterio	Valor
Primer Examen	14
Segundo Examen	16
Trabajos Encargados	15
Exposiciones	171 123

Navigation buttons 'Atras' and 'Menu' are located at the bottom of the screen. The phone's keypad is visible below the screen.

FIGURA 43

FORMULARIO DE INGRESO DE NOTAS DE ESTUDIANTE

Fuente: Elaboración propia

Las figuras 39, 40, 41 y 42 permiten mostrar los datos del docente, la carga académica, su horario de trabajo donde el manejo es similar al usuario estudiante y muestra la lista de estudiantes matriculados en los cursos asignados como carga académica para registrar notas. La figura 43 presenta el formulario para registrar notas del estudiante; que se mencionaron anteriormente, en algunos casos los docentes no consideran estos criterios que pueden agregarse o en caso contrario reducir los criterios mencionados.

Como se puede observar el diseño del modelo de datos; sirve de base para considerar la información que se presentará al usuario donde la forma de presentación se basa en la interfaz de usuario; pero en este caso para dispositivos móviles.

El diseño de la interfaz de usuario es la categoría de diseño que establece un medio de comunicación entre el hombre y la máquina. La solución para una buena interfaz, es construir un proyecto multimedia que contenga gran poder de navegación, brindando acceso al contenido y tareas para todos los usuarios de todos los niveles, junto a un sistema de ayuda que ofrezca una orientación (Pressman, 2004).

Un diseño cuidadoso de la interfaz de usuario es parte fundamental del diseño del proceso general del software. Un buen diseño de la interfaz de usuario es crítico para la confiabilidad del sistema (Somerville, 2005).

4.4. EVALUACION DEL SISTEMA DE CONSULTA DE NOTAS

La evaluación del sistema de consulta de notas se ha realizado, para ello los estudiantes utilizaron el sistema mencionado. Luego se procedió a realizar encuestas a los estudiantes sobre la funcionalidad del sistema, evaluación que permitirá determinar la concordancia con las especificaciones de diseño y el cumplimiento de funciones requeridas. Las pruebas se realizaron con los usuarios, estudiantes de la Escuela Profesional de Ingeniería de sistemas.

La evaluación del sistema de consultas de notas permitirá conocer la calidad del software desarrollado considerando los siguientes factores:

Amigable: Se refiere a la interacción con el sistema, es decir que la aplicación tenga facilidad para usarse, sin necesidad de consultar la ayuda o una ayuda en línea sobre la aplicación.

La figura 44 muestra el porcentaje de estudiantes que refieren que el sistema de consultas respecto a si es amigable:

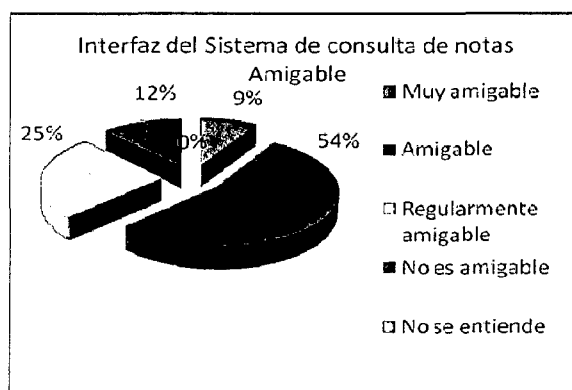


FIGURA 44

PORCENTAJE DE ESTUDIANTES REFIEREN QUE LA INTERFAZ DEL SISTEMA DE CONSULTA DE NOTAS ES AMIGABLE

Fuente: Elaboración propia

En la figura 44, el 54% de los estudiantes consideraron que la interfaz del sistema de consulta de notas es regularmente amigable, observacion que se realiza en comparación a los dispositivos moviles actuales que son mas amigables; siendo necesario indicar que el presente sistema de consulta de notas esta dirigido a dispositivos móviles con capacidades de hardware mínima y de bajo costo.

Legibilidad: En esta parte se evalúa color de texto, fondo y contrastes, así como el tamaño de fuente; los cuales tienen que ser adecuado para ser legibles a cualquier usuario.

La figura 45 muestra el porcentaje de estudiantes que refieren que el sistema de consultas respecto a si es legible:

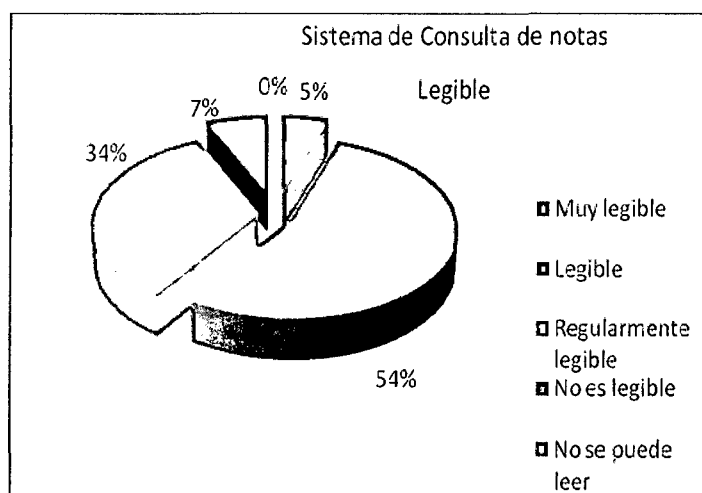


FIGURA 45
PORCENTAJE DE ESTUDIANTES REFIEREN QUE EL SISTEMA DE CONSULTA DE NOTAS ES LEGIBLE

Fuente: Elaboración propia

La figura 45 describe presentación de la interfaz del sistema es muy importante respecto a la legibilidad del sistema, y el 54% de los estudiantes indican que es muy legible la interfaz del sistema, y 5% indica baja legibilidad, donde aspectos en el diseño de pantalla son necesarios mejorar porque son importantes para la interacción con el usuario.

Eficiencia: Se mide cuando las tareas que se realizan en el sistema pueden llevarse a cabo rápida y fácilmente.

La figura 46 muestra el porcentaje de estudiantes que refieren que el sistema de consultas respecto a si es eficiente

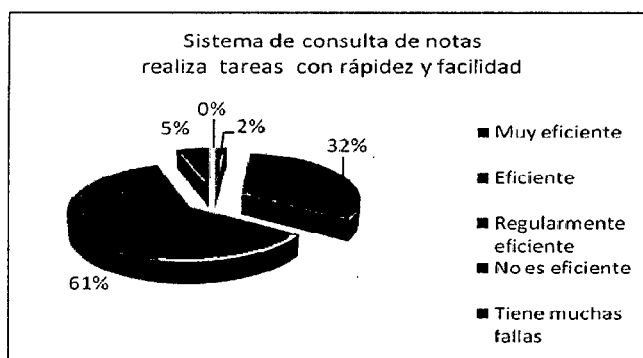


FIGURA 46

PORCENTAJE DE ESTUDIANTES REFIEREN EFICIENCIA DEL SISTEMA DE CONSULTA DE NOTAS

Fuente: Elaboración propia

En la figura 46, indica que el sistema de consulta de notas muestra la eficiencia respecto a la facilidad y rapidez de manejo del sistema de consultade notas, donde el 61% de los estudiantes indican que es regularmente eficiente y un 32% eficiente; donde más del 90% consideran la eficiencia del sistema. La velocidad de acceso a datos del dispositivo móvil a través de internet es dependiente del servicio de telefonía móvil.

Satisfacción: El usuario da conformidad a las tareas que realiza el sistema.

La figura 47 muestra el porcentaje de estudiantes que refieren que el sistema de consultas respecto a la satisfacción con tareas del sistema:

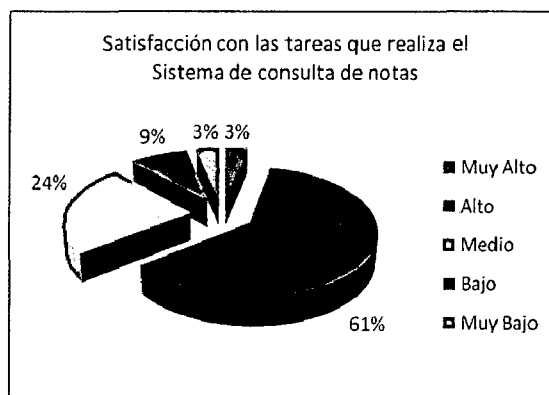


FIGURA 47

PORCENTAJE DE ESTUDIANTES REFIEREN SATISFACCION CON LAS TAREAS QUE REALIZA EL SISTEMA DE CONSULTA DE NOTAS

Fuente: Elaboración propia

La figura 47 visualiza que las tareas que realiza el sistema de consulta de notas, se encuentran acordes a los requisitos planteados previos al desarrollo del sistema, incluso agregando tareas; 61% de los estudiantes refieren satisfacción alta y muy alta 3%. El sistema de consulta de notas permite a los estudiantes visualizar horarios y notas de cursos; siendo necesario para ampliar las tareas que los estudiantes sugieran necesarias en el sistema.

Reversibilidad: Se refiere a la capacidad del sistema para permitir deshacer acciones realizadas.

La figura 48 muestra el porcentaje de estudiantes que refieren que el sistema de consultas respecto a su reversibilidad:

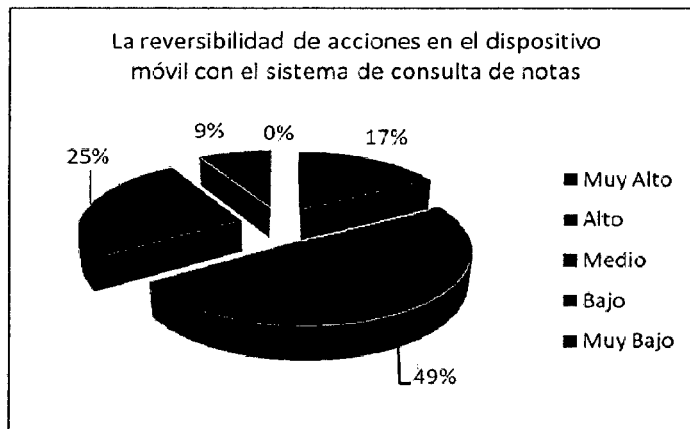


FIGURA 48

PORCENTAJE DE ESTUDIANTES REFIEREN REVERSIBILIDAD DE ACCIONES EN EL SISTEMA DE CONSULTA DE NOTAS

Fuente: Elaboración propia

La figura 48 muestra que las acciones no deseadas o deshacer acciones erróneas realizadas por el estudiante en el manejo del sistema de consulta de notas implica la reversibilidad, donde un 49% de estudiantes refieren reversibilidad media del sistema y 17% reversibilidad alta. En el manejo de dispositivos móviles hay acciones involuntarias, que implican mayor tiempo de acceso a información; y la reversibilidad permite deshacer dichas acciones.

Interfaz Grafica: Se refiere a la navegación en el sistema que incluye imágenes, colores y posición de elementos del sistema.

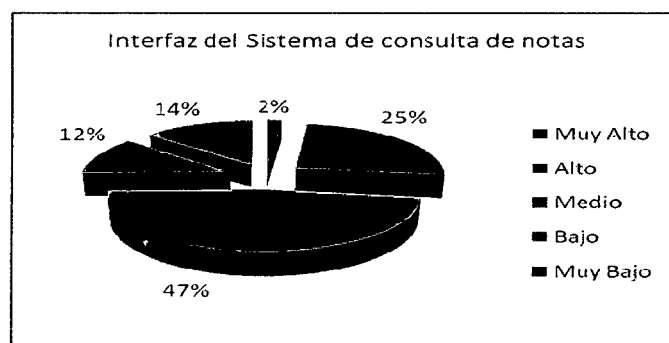


FIGURA 49

PORCENTAJE DE ESTUDIANTES REFIEREN SOBRE LA NAVEGACIÓN EN EL SISTEMA DE CONSULTA DE NOTAS

Fuente: Elaboración propia

La figura 49 indica que la navegación es dependiente del diseño de la interfaz de pantallas que interactúa con el usuario; en este caso el usuario – estudiante, empleado el sistema de consulta de notas un 47% de estudiantes indican navegación regular y 2% muy alta. Entonces, al acceder el usuario al sistema presenta menú de opciones que consideran debe mejorarse el modelo de navegación respecto a los eventos que el usuario realiza.

Las figuras 44 al 49 muestran la evaluación sobre la funcionalidad del sistema, como también determino la calidad del software con algunos atributos y se detectó las deficiencias del sistema de acuerdo al uso del sistema por parte de los estudiantes; usuarios finales de la aplicación. Destacando que la metodología ICONIX considera al usuario como la base del desarrollo de una aplicación software.

Respecto a determinar la calidad del software, el modelo McCall se focaliza en el producto final, identificando atributos claves desde el punto de vista del usuario, estos atributos se denominan factores de calidad y son normalmente atributos externos (Fillotrani, 2007).

Para determinar si el sistema de información mejora en tiempo y costos el acceso a información, se considero los siguientes criterios en la encuesta a los estudiantes:

Rapidez: Se refiere al acceso rápido a la información del rendimiento académico del estudiante:

La figura 50 presenta el porcentaje de estudiantes que refieren la rapidez de acceso al sistema

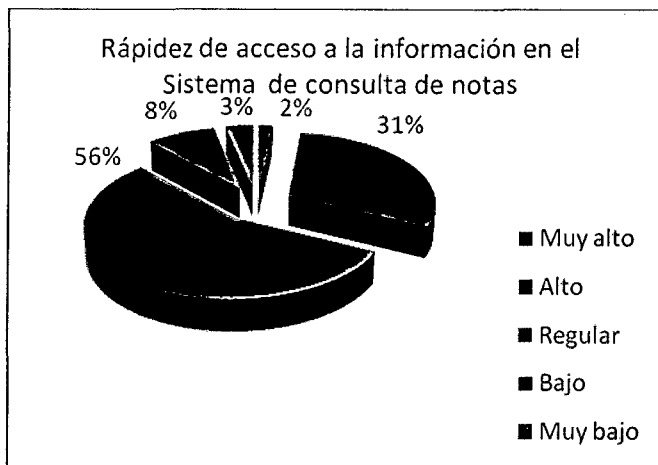


FIGURA 50

PORCENTAJE DE ESTUDIANTES REFIEREN RAPIDEZ DE ACCESO A INFORMACIÓN DEL SISTEMA DE CONSULTA DE NOTAS

En la figura 50, el usuario emplea el sistema de consulta de notas y verifica el acceso al sistema de consulta de notas; indicando en un 56% regular la rapidez de acceso, 31% alta, 8% baja rapidez, 3% muy bajo y 2% muy alta. Es necesario reiterar que la velocidad de acceso al sistema depende del tipo de servicio que disponga el usuario en su servicio de telefonía móvil, que es un factor a considerar respecto a la accesibilidad.

Evita costo de desplazamiento: Se refiere a la comparación que se realiza frente al empleo de la página web, o realizar sus consultas en las oficinas que le brinde información acerca de sus notas.

La figura 51 presenta el porcentaje de estudiantes que refieren si el sistema les evita costos en desplazamiento:

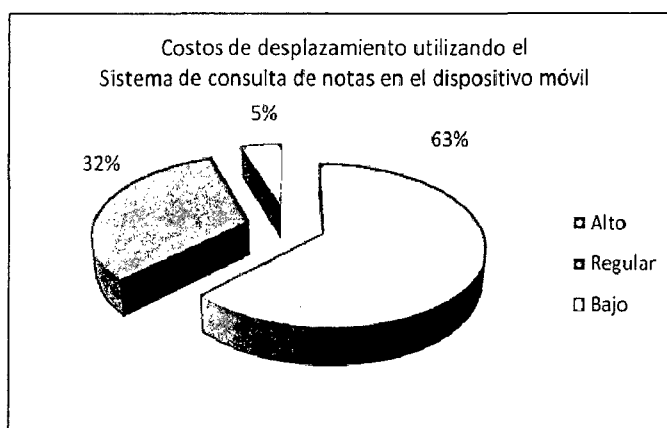


FIGURA 51

REFERENCIA DE USO DEL SISTEMA DE CONSULTA DE NOTAS EVITA COSTOS DE DESPLAZAMIENTO

Fuente: Elaboración propia

De la figura 51, se observa que el 63% de estudiantes indican que el sistema de consulta de notas le evita costos de desplazamientos, 32% que es regular y 5% no se evita estos costos. Es necesario mencionar que los estudiantes cuando se apersona a la oficina de coordinación académica o ingresan a cabinas de internet para obtener información de su rendimiento académico, esto implicara un costo mayor porque repercute en el factor tiempo.

Acceso a la información desde cualquier lugar: Se refiere a consultar información desde el lugar que se encuentre el estudiante.

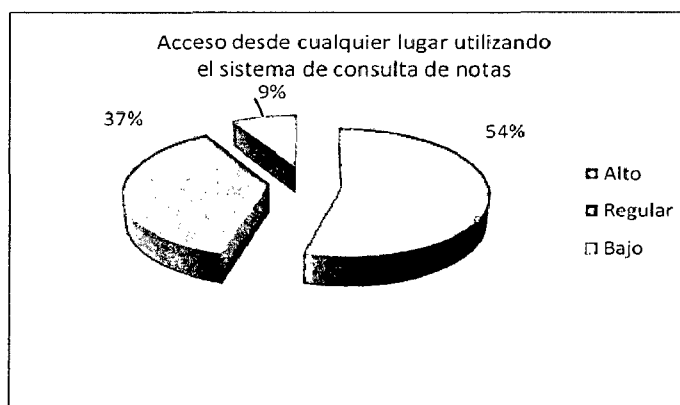


FIGURA 52

ACCESO AL SISTEMA DE CONSULTA DE NOTAS DESDE CUALQUIER LUGAR

Fuente: Elaboración propia

En la figura 52, se observa que el 63% de estudiantes indican que el sistema de consulta de notas le evita costos de desplazamientos, 32% indica que es regular y 5% indica que es bajo evitar estos costos. Es necesario mencionar que los estudiantes cuando se apersona a la oficina de coordinación académica o ingresan a cabinas de internet para obtener información de su rendimiento académico, esto implicara un costo mayor porque repercute en el factor tiempo.

Costos de de acceso: Se refiere a los costos adicionales al servicio de telefonía que implica el servicio de internet.

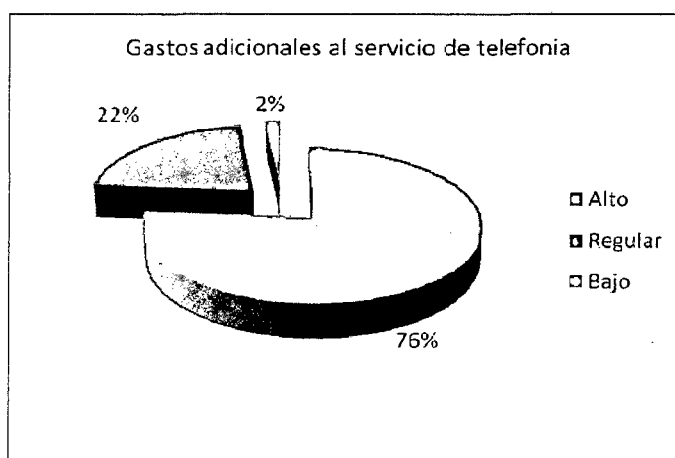


FIGURA 53

COSTOS EN PORCENTAJE PARA ACCESO AL SISTEMA DE CONSULTA DE NOTAS

Fuente: Elaboración propia

Las figura 53, muestra que los estudiantes en un 76% refieren que los costos de servicio de acceso son altos, en un 22% consideran que el costo es regular y un 2% consideran los costos bajos. Los costos dependen del servicio de telefonía que brinde el servicio de internet de acceso a datos y sus costos respectivos. A este punto es necesario mencionar que estos costos son cada vez mas accesibles a los usuarios del sistema.

La evaluación de la aplicación de ingreso y consulta de notas se ha realizado con los indicadores respecto a la funcionalidad del sistema, la cual está acorde a las especificaciones de diseño, para ello es necesario verificar que el sistema lleve a cabo las funciones requeridas. Los resultados de la prueba han sido los siguientes:

Amigable, que se refiere a la interacción con el sistema sin necesidad de consultar ayuda o una ayuda en línea (54%), regularmente amigable (25%) y no es amigable (12%).

Legibilidad, en esta parte se evalúa el texto, fondo y su contraste; así como el tamaño de la fuente; los cuales tienen que ser adecuados para ser legibles por cualquier usuario, teniendo estos resultados: Muy legible (5%), legible (54%), regularmente legible (34%) y no es legible (7%).

Eficiencia, se mide cuando las tareas que se realizan en el sistema pueden llevarse a cabo rápida y fácilmente, teniendo estos resultados: muy eficiente (2%), eficiente (32%), regularmente eficiente (61%), no es eficiente (5%).

Satisfacción, el usuario da conformidad a las tareas que realiza el sistema, teniendo estos resultados: muy alto (3%), alto (61%), bajo (9%) y muy bajo (3%).

Interfaz gráfica, que refiere la navegación en el sistema donde se incluye imágenes, colores y posición de elementos del sistema, se obtuvo los siguientes resultados: navegación alta (25%), regular (47%), baja (12%), muy baja (14%) y muy alta (2%).

Respecto a las pruebas utilizadas para verificar si el sistema de consultas mejora en tiempo y costos respecto a los sistemas vía web respecto al acceso a información de consulta de notas se obtuvo lo siguiente:

La rapidez de acceso al sistema de consulta de notas del estudiante; se obtuvo los resultados: regular (56%), alta (31%), baja rapidez (8%). Así mismo los estudiantes refieren que el uso del sistema evita costos de desplazamiento (63%), de modo regular (32%) y no evita costos de desplazamiento (5%). Si se compara los gastos que realiza el estudiantes al utilizar otros medios de información, ellos consideran en un (76%) que costos de servicio de internet son altos, regular (22%) y bajo (2%).

Entonces, los estudiantes indican que usando el sistema de consulta de notas les permite rapidez de acceso a información y les evita desplazamientos, puesto que les permite acceder desde el lugar donde se encuentran; entonces de los datos obtenidos se acepta la hipótesis porque se confirma que el sistema mejora en tiempo y costos el acceso a información de notas del estudiante.

Campusmovil.net cubre necesidades como: captar el punto de inspiración del entorno académico, aprovechar el tiempo sin necesidad de computación (transporte público, tiempo entre clase y clase, bibliotecas, espacios públicos fuera del campus) para seguir interactuando con la comunidad universitaria tanto a nivel de servicios como de gestión de conocimientos son usos académicos, generar retazos de información para ser recuperados y reutilizados en otros entornos (Kuklinski, 2009).

Sin embargo el proceso de desarrollo de software móvil tiene criterios de diseño muy particulares con respecto a otros tipos de software haciendo difícil competir todas las etapas y documentación por lo cual es conveniente buscar metodologías y procesos específicos para el desarrollo de software móvil (Cruz, Garzon y et.al, 2008).

CONCLUSIONES

- Con la presente investigación se ha desarrollado una aplicación utilizando la tecnología Java Micro Edition (JME) que es una especificación de un subconjunto de la plataforma Java para la consulta de notas de los estudiantes desde un dispositivo móvil (teléfono celular) a través de internet; así como el ingreso de notas para los docentes. Otro de los aspectos es que todo dispositivo móvil que aparece en el mercado tiene una conexión a internet y el costo de acceso a este servicio se ha reducido en el transcurso de los últimos años.

- En el desarrollo de la aplicación se ha utilizado la metodología ICONIX; el cual es un proceso intermedio entre las metodologías XP y RUP. La arquitectura utilizada para el desarrollo de la aplicación ha sido del patrón de diseño MVC y la arquitectura de 3 capas con la lógica de presentación, de negocio y de datos.

- La evaluación de la aplicación de ingreso y consulta de notas se ha realizado con los indicadores respecto a la funcionalidad y los resultados han sido que la mayoría de los estudiantes encuestados han respondido que es amigable, legible, eficiente y de alto grado de satisfacción, permitiendo reducción de costos de desplazamiento y ahorro de tiempo a los estudiantes.

RECOMENDACIONES.

- Se sugiere el desarrollo de aplicaciones relacionadas al empleo de tecnologías móviles, y motivar el desarrollo de aplicaciones orientadas al área educativa que permitan a los estudiantes beneficiarse de estas tecnologías ya sea para acceder a información académica de su interés.
- El empleo de las tecnología móviles orientadas a la educación se pueden dar orientados hacia otro tipo de aplicaciones como pueden ser para que los docentes puedan administrar información de sus estudiantes de manera remota; y también realizar aplicaciones relacionadas al proceso de enseñanza-aprendizaje como es la nueva tendencia m-learning.
- La metodología ICONIX empleada en el presente trabajo de investigación es útil debido a que permite lograr un desarrollo en base a prototipos, beneficiándose del empleo de modelado UML.

- Aplicar el uso de estas tecnologías no solo al campo de la educación, sino lograr automatizar el manejo de la información en diversas áreas del conocimiento que puedan realizarse y beneficiarse de las TICs.

BIBLIOGRAFIA

- BARBAGALLO, Ralph (2004) *Wireless Game Development in Java with MIDP 2.0*. Edi. Wordware Publishing. Texas - Estados Unidos de América
- COBO, Ángel; GÓMEZ, Patricia; PÉREZ, Daniel y ROCHA, Rocío (2005). *PHP y MySQL: Tecnologías para el desarrollo de aplicaciones Web.* : Edi. Díaz de Santos. España
- DATE C. J. (2004) *Introducción a los Sistemas de Base de Datos*. Edi. Prentice Hall. 7ma. Edición. México.
- DE PABLO, Carmen; LOPEZ H., José J.; Martín R., Santiago y Medina, Sonia (2004). *Informática y comunicaciones en la empresa*. Edi. ESIC. Madrid. España.
- FOWLER, Martín y KENDALL, Scoth (2001). *UML Gota a Gota*. (1999). Edi. Addison Wesley. 1ra. Edición. Estados Unidos de América.
- GÁLVEZ R., Sergio y ORTEGA D., Lucas (2003). *Java a Tope: Java 2 Micro Edition*. Edi. Universitaria Málaga. Madrid.

- HANSMANN, Uwe; NICKOUS, Martin S.; SCHACK, Thomas; SCHNEIDER, Achim y SELIGER, Frank (2002). *Smart Card Application Development using Java*. Edi. Springer-Verlag Berlin Herderbeg. 1ra. Edición. New York - Estados Unidos de América.
- KNUDSEN, Jhonatan (2003). *Wireless Java: Developing with J2ME*. Edi. Springer – Vernag. 2da. Edición. New York - Estados Unidos de América.
- LAMARCA, María J. (2011). *Hipertexto: El Nuevo Concepto de documento en cultura de la imagen*. Edi. Complutense. 1ra. Edición. Madrid - España.
- LAUDON, Kenneth C. y LAUDON, Jane P. (2004). *Essential of Management Information System*. Edi. Prentice Hall. 6ta. Edición. México.
- LLANOS Ferraris, Diego R. (2010). *Fundamentos de Informática y programación en C*. Edi. Paraninfo. 1ra. Edición. España.
- NAING, Ko Ko; SRINIVASAN, Sathya; DAVIS, Chad y UMAPATHY, Sivasundaram (2009). *SCMAD: Exam Guide- Exam CX-310 -110*. Edi. McGraw-Hill. New Delhi. India.
- PIROUMIAN, Vartan (2002). *Wireless J2ME Platform Programming*. Edi. Prentice Hall. California-Estados Unidos de América.
- PRESSMAN, Roger (2004). *Ingeniería del Software: Un Enfoque Práctico*. Editorial McGraw Hill. 2da. Edición. México.
- RAMÍREZ, Luis E. (2002). *Aplicando UML*. Edi. Macro. 1ra. Edición. Lima – Perú.
- ROB, Peter & CORONEL, Carlos (2006). *Sistemas de Base de Datos: Diseño, implementación y administración*. Edi. Thomson. México.

- SÁNCHEZ M., Paul (2007) *Control Remoto de Factores Ambientales*. Tesis Universidad Nacional de San Agustín. Arequipa-Perú.
- SHEKHAR, S. y CHAWLA, S. (2002) *Spatial Databases: A Tour*. Edi. Prentice Hall. New Jersey-Estados Unidos de América.
- SING, Li y KNUDSEN J. (2005). *Beginning J2ME from novice to professional*. Edi. Apress. 3ra. Edición. Estados Unidos de América.
- SOMERVILLE, Ian (2005). *Ingeniería del software*. Edi. Perason Education. 7ma. Edición. Madrid. España.
- TAVARES, Erico (2005). *Programação JAVA para Wireless: Aprenda a desenvolver sistemas em J2ME*. Edi. Digereti Comunicação y Tecnología. Sao Paulo – Brasil.
- TOPLEY, Kim (2002). *J2ME in a nutshell. A Desktop Quick References*. Edi. O'Reilly & Associates. 1ra. Edición. Estados Unidos de América.

WEBGRAFIA:

- BERTUZZI, Renzo C. y SALAZAR F., Juan P. (2007). *Diseño e implementación de un servicio de localización y visualización de mapas utilizando J2ME para dispositivos móviles y herramientas de libre distribución*. Publicación Síntesis Tecnológica de Universidad Austral de Chile, 3,2 (40-41). Recuperado de: <http://mingaonline.uach.cl/pdf/sintec/v3n2/art01.pdf>
- CALLEJAS C., Mauro; MELÉNDEZ A., Liliana F.; CORTES R., Andrea P. (2010). *Desarrollo de aplicaciones móviles enfocadas al turismo Departamento*

de Bocayá. Revista de la Universidad Católica del Norte. Medellín. Colombia, 29 (166-178). Recuperado de:

<http://www.revistavirtual.ucn.edu.co/index.php/RevistaUCN/article>

- CERNA L., Albert y CRISOLOGO, Martin (2008). *Sistema Web para la gestión de venta de equipos de cómputo para la empresa PROCOM utilizando la metodología ICONIX*. Tesis de la Universidad Privada Antenor de Orrego, Trujillo. Perú. Recuperado de: <http://es.scribd.com/doc/58752746/Plan-de-Tesis-Original-Utilizando-Iconix-y-Joomla>

- CRUZ G., Daniel F; GARZON G., Julian E.; MONTES H. Diana y et.al. (2008). *Proceso de Consultas Móviles adaptables a diferentes tipos de negocios basados en telefonía celular (METAMORF)*. Revista de la biblioteca digital de la Universidad ICESI - Colombia, 6 (170-180). Recuperado de: <http://hdl.handle.net/10906/1732>.

- DE LA RIVA, Germán D. (2007). *Aplicaciones Web Para Celulares*. Tesis de Facultad de Informática - Universidad Nacional de Plata. Argentina. Recuperado de: <http://sedici.unlp.edu.ar/handle/10915/3972>

- DÍAZ M., David y HERNANDEZ S., Antonio (2002). *Introducción a JME, Programación orientada a objetos*. Universidad de Salamanca. España. Recuperado de: <http://www.youblisher.com/p/282027-Introduccion-a-J2ME-David-Diaz>

- DIAZ, J.; Harari, I. y Amadeo, A. (). *Consideraciones de diseño de una interfaz móvil para un portal educativo*. Facultad de Informática. Universidad de la Plata. Recuperado de:

[http://www.linti.unlp.edu.ar/uploads/docs/consideraciones de diseno de la interfaz movil para un portal educativo.PDF](http://www.linti.unlp.edu.ar/uploads/docs/consideraciones_de_diseno_de_la_interfaz_movil_para_un_portal_educativo.PDF)

- DIEZ, Guillermo; SANCHO, Andino; GARCÍA Rioja Rosa Má y CAMPO Vázquez Celeste (2003). *Desarrollo de un Servidor HTTP para Dispositivos Móviles en JME*. Departamento de Ingeniería Telemática. Universidad Carlos III de Madrid. España. Recuperado de: <http://www.it.uc3m.es/celeste/papers/ServidorHTTP.pdf>

- ESPINOSA F.; SOTO A.; NOVILLO F.; Salamea M. (2009). *Pago Electrónico a través de Tecnologías Móviles*. Facultad de Ingeniería en Electricidad y Computación. Escuela Superior Politécnica del Litoral (ESPOL), Guayaquil. Ecuador. Recuperado de:

http://www.cib.espol.edu.ec/Digipath/REVISTA_CICYT/Articulo/146.pdf

- FILLOTRANI, Pablo (2007). *Calidad en el desarrollo de software*. Universidad Nacional del Sur. Recuperado de: <http://www.cs.uns.edu.ar>

- GARCÍA S., Alberto (2004). *Programación de juegos para móviles con J2ME*. Recuperado de: <http://www.emagister.com/curso-programacion-juegos-moviles-j2me/desarrollo-midlets>

- GÓMEZ O., Manuel. (2006). *Tutorial: Herramientas de programación Java de dispositivos móviles*. Universidad de Granada. España. Recuperado de: <http://leo.ugr.es/J2ME/TOOLS/index.html>

- GÓMEZ Ballester, Eva; MARTÍNEZ B., Patricio y MOREDA P., Paola (2002). *Base de Datos*. Universidad de Alicante. España. Recuperado de: <http://www.dlsi.ua.es/asignaturas/bd>

- GONZALES M., Oscar (2000). *Arquitectura de sistemas de bases de datos*. Universidad Castilla de la Mancha. Recuperado de url:
<http://www.itescam.edu.mx/principal/sylabus/fpdb/recursos/r88431.PDF>
- KUKLINSKI (2009). *Campus móvil. La primera red social universitaria vía dispositivos móviles de Iberoamérica. Un estudio de caso*. Revista Iberoamericana de Educación a distancia, 12 [21-32]. Recuperado de:
<http://www.utpl.edu.ec/ried/images/pdfs/vol12N2/campusmovil.pdf>
- LÓPEZ F., José M. (2001). *Arquitectura de funcionamiento de un servidor de aplicaciones*. Publicación electrónica disponible en url:
<http://trevinca.ei.uvigo.es/~txapi/espanol/proyecto/superior/memoria/node21.html#serv:aplicaciones>
- MEZA G., David A. (2008). *Sistema Generador de aplicaciones Multicapa – CODEV*. Tesis electrónica de la Universidad de Chile. Recuperado de:
http://www.tesis.uchile.cl/tesis/uchile/2008/meza_dg/sources/meza_dg.pdf
- MEZA M., Jorge I. *Midlets con J2ME*. Recuperado de:
<http://www.slideshare.net/jimezam/midlets-con-j2me>
- MONTROYA M., Juan D.; TORRES, Saulo y VILLA, Paula A. (2010). *Primeros pasos para programar aplicaciones móviles en Java*. Revista Scienta y Technica. Universidad Tecnológica de Pereira, 1, 44(173-177). Recuperado de:
revistas.utp.edu.co/index.php/revistaciencia/article/download/
- NAVARRO M. Javier; SAN PABLO H., Javier. (2005). *Introducción JME*. Universidad de Salamanca, España. Recuperado de:
<http://zarza.usal.es/~fgarcia/docencia/poo/04-05/Trabajos/J2ME.pdf>

- PEÑALVER B., Antonio; BOTELLA, Federico; MARTÍNEZ B., Oscar, FERNÁNDEZ C., Antonio y GONZÁLEZ, Pascual (2005). *Sistema Ubicuo Mediante Telefonía Móvil Para el acceso a Información Académica*. Universidad ICESI. España. Recuperado de:

http://www.umh.es/otri/Servicios_Universidad/9_Publicacion/1_Memoria_investigacion/2005/cio.pdf.

- PÉREZ, Jorge (). *Comunicaciones Móviles e Inalámbricas*, Grupo de análisis y prospectiva del sector de las telecomunicaciones. Edita Red.es. Madrid. España. Recuperado de:

<http://observatorio.red.es/estudios/documentos/gaptelmoviles>

- PINILLA A. y C. Diana. (2006). *Transmisión Inalámbrica de Datos a través de Teléfonos Celulares*. Universidad Pontificia Bolivariana - Colombia Recuperado de:

http://eav.upb.edu.co/banco/sites/default/files/files/Tesistransmisioninalambrica_datos.pdf

- PRIETO, Manuel J. (2002). *Introducción a J2ME*. Recuperado de: <http://es.scribd.com/doc/7136526/Manual-Programacion-Java-Curso-J2ME>

- RINCÓN B, José. *Conociendo tu entorno de programación Java*. California. Estados Unidos de América. Recuperado de:

<http://www.slideshare.net/Stephenson/conociendo-tu-entorno-de-programacion-java>

- SALAS C., Nohely (2010). *Manual para crear una aplicación J2ME*. Universidad los Ángeles de Chimbote, Perú. Recuperado de:

<http://es.scribd.com/doc/30575785/MANUAL-PARA-CREAR-UNAAPLICACIÓN-JAVA-J2ME>

- SÁNCHEZ, Jorge. (2004). *Manual breve para el manejo del entorno de programación de código abierto Netbeans para la programación java.*

Recuperado de: www.jorgesanchez.net

- SAN MARTIN Oliva. *Metodología ICONIX.* Argentina Recuperado de:

<http://www.portalhuarpe.com.ar/Seminario09/archivos/MetodologiaICONIX.pdf>

- SOTO, Sergio A. (2008). *Acceso a Bases de Datos Multiplataforma a través de Teléfonos Celulares,* Facultad de Ciencias Exactas, Naturales y Agrimensura Universidad Nacional del Nordeste, Corrientes Argentina.

Recuperado de:

- <http://exa.unne.edu.ar/depar/areas/informatica/SistemasOperativos>

- VÁSQUEZ M., Gabriel A; CAICEDO R., Oscar M. y TREJO N., Euler A. *PROMETEO: Plataforma Integrada de Desarrollo de Aplicaciones para Dispositivos Móviles con JME.* Departamento de Electrónica Instrumentación y Control de la Universidad del Cauca. Recuperado de:

<http://atenea.unicauca.edu.co/~omcaicedo/prometeo.pdf>

- VIERA Balanta, Victor. (2010). *Computación Móvil, Principios y Técnicas.* Asociación Colombiana de Ingenieros de Sistemas. Bogota. Colombia.

Recuperado de:

<http://www.acis.org.co/archivosAcis/ComputacionMovilTEcnicasyPrincipios.pdf>

ANEXOS

DETERMINACION DEL TAMAÑO DE LA MUESTRA

Los 469 estudiantes que se encuentran matriculados en el Escuela Profesional de Ingeniería de Sistemas de la Universidad Nacional del Altiplano de la ciudad de Puno. Para hallar el tamaño de la muestra se determino de la siguiente manera:

Población	: N = 469
Nivel de confianza al 95%	: Z = 1.96
Estimado de la población	: p = 50% y q = 50%
Error muestral	: e = 12%

$$n = \frac{Z^2 \times p \times q \times N}{(N-1) \times e^2 + Z^2 \times p \times q}$$
$$n = \frac{(1.96)^2 \times 0.5 \times 0.5 \times 469}{(469-1) \times (0.12)^2 + (1.96)^2 \times 0.5 \times 0.5} \quad n = 59$$

El tamaño de la muestra es de 59 estudiantes de la Escuela Profesional de Ingeniería de Sistemas de la Universidad Nacional del Altiplano – Puno 2010.

ENCUESTA

Edad: _____

Sexo:

M

Instrucciones: *El propósito fundamental de esta encuesta es recolectar información acerca de la creencia u opinión que usted tiene sobre el sistema de consulta a través de tecnologías móviles.*

Evaluación de la información del Sistema de Consulta:

1. *¿Puede ud. usar el sistema de manera sencilla sin necesidad de ayuda?*

- a) *Muy amigable* b) *Amigable* c) *Regularmente amigable* d) *No es amigable*
e) *No se entiende*

2. *¿Se puede leer los datos mostrados en el sistema de consulta de notas?*

- a) *Muy legible* b) *Legible* c) *Regularmente legible* d) *No es legible*
e) *No se puede leer*

3. *¿El sistema de consulta de notas realiza tareas manera rápida y fácil?*

- a) *Muy eficiente* b) *Eficiente* c) *Regularmente eficiente* d) *No es eficiente*
e) *Tiene muchas fallas*

4. *¿Esta ud. conforme con las tareas que realiza el sistema de consulta de notas a través del dispositivo móvil?*

- a) *Muy alto* b) *Alto* c) *Medio* d) *Bajo* e) *Muy bajo*

5. *¿Puede ud. deshacer acciones realizadas el sistema de consulta de notas a través del dispositivo móvil?*

- a) *Muy alto* b) *Alto* c) *Medio* d) *Bajo* e) *Muy bajo*

6. *¿Puede ud. usar el sistema de consulta de notas a través del dispositivo móvil de manera sencilla?*

- a) *Muy alto* b) *Alto* c) *Medio* d) *Bajo* e) *Muy bajo*

7. *El sistema de consulta accede rápido a las tarea de acceso a la información de su rendimiento académico*

a) *Muy alto* b) *Alto* c) *Regular* d) *Bajo* e) *Muy bajo*

8. *El sistema le evita costos de desplazamientos desde su casa a la Universidad:*

a) *Alto* b) *Regular* c) *Bajo*

9. *Puede Ud. acceder Ud. desde cualquier lugar, teniendo el celular con el Sistema de consulta.*

a) *Alto* b) *Regular* c) *Bajo*

10. *Es costoso en comparación a otros sistemas de información via web*

a) *Alto* b) *Regular* c) *Bajo*

CODIGO FUENTE DE LA APLICACIÓN DEL SISTEMA DE CONSULTAS DE NOTAS

En el siguiente código se muestra la estructura del MIDlet creado para la aplicación móvil.

```
packagepe.edu.unap.academico;

import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.util.Hashtable;
import java.util.Vector;
import javax.microedition.midlet.*;
import javax.microedition.lcdui.*;
import pe.edu.unap.academico.model.BeanCurso;
import pe.edu.unap.academico.model.BeanHorario;
import pe.edu.unap.academico.model.BeanPersona;
import pe.edu.unap.academico.model.Comunicacion;

public class Notas extends MIDlet implements ItemStateListener
{
    public static final String file_schedule = "/horario.xml";

    private boolean midletPaused = false;
    private Vector vCursos;
    private Login loginForm;
    private Personapersona;
    private CursosList cursosList;
    private Curso curso;
    private Estudiantes estudiantes;
    private Horario horario;
    private Document docSchedule;

    public Notas() {
        vCursos = new Vector();
        loginForm = new Login(this);
        docSchedule = new Document();
    }

    private void initialize() {

    }

    public void startMIDlet() {
        Display display = getDisplay();
        display.setCurrent(loginForm);
        beginParse();
    }

    public void resumeMIDlet() {

    }
}
```

```

public void switchDisplayable(Alert alert, Displayable nextDisplayable)
{
    Display display = getDisplay();
    if (alert == null) {
        display.setCurrent(nextDisplayable);
    } else {
        display.setCurrent(alert, nextDisplayable);
    }
}

public Display getDisplay () {
    return Display.getDisplay(this);
}

public void exitMIDlet() {
    switchDisplayable (null, null);
    destroyApp(true);
    notifyDestroyed();
}

public void startApp() {
    if (midletPaused) {
        resumeMIDlet ();
    } else {
        initialize ();
        startMIDlet ();
    }
}

midletPaused = false;
}

public void pauseApp() {
    midletPaused = true;
}

public void destroyApp(boolean unconditional) {
}

public void itemStateChanged(Item item)
{
}
}

```

Listado de Horario

En la Aplicación se ha incluido también la lista de las asignaturas en los días que le toca asistir y en las horas, esto también se muestra en formato XML.

```

<?xmlversion="1.0"encoding="ISO-8859-1"?><horario>
  <diaInicio="07:00"hFinal="08:00"id="2"idCurso="PB0405"
  mnemonico="Met. Num."sDia="Lunes"/>
  <diaInicio="08:00"hFinal="09:00"id="2"idCurso="PB0405"
  mnemonico="Met. Num."sDia="Lunes"/>
  <diaInicio="09:00"hFinal="10:00"id="2"idCurso="SO090"
  mnemonico="S. I. Dist."sDia="Lunes"/>
  <diaInicio="10:00"hFinal="11:00"id="2"idCurso="SO0903"
  mnemonico="S. I. Dist."sDia="Lunes"/>

```

<diaInicio="11:00"hFinal="12:00"id="2"idCurso="IN0706"
 mnemonico="Tall. S. I."sdia="Lunes"/>
 <diaInicio="12:00"hFinal="13:00"id="2"idCurso="IN0706"
 mnemonico="Tall. S. I."sdia="Lunes"/>
 <diaInicio="13:00"hFinal="14:00"id="2"idCurso="SIS201"
 mnemonico="Inf. Bas."sdia="Lunes"/>
 <diaInicio="14:00"hFinal="15:00"id="2"idCurso="SIS201"
 mnemonico="Inf. Bas."sdia="Lunes"/>
 <diaInicio="15:00"hFinal="16:00"id="2"idCurso="SO0902"
 mnemonico="D. A A BD"sdia="Lunes"/>
 <diaInicio="16:00"hFinal="17:00"id="2"idCurso="SO0902"
 mnemonico="D. A A BD"sdia="Lunes"/>
 <diaInicio="17:00"hFinal="18:00"id="2"idCurso="AO0903"
 mnemonico="Diag. Org."sdia="Lunes"/>
 <diaInicio="18:00"hFinal="19:00"id="2"idCurso="AO0903"
 mnemonico="Diag. Org."sdia="Lunes"/>
 <diaInicio="07:00"hFinal="08:00"id="3"idCurso="IN0205"
 mnemonico="C. P. B. N."sdia="Martes"/>
 <diaInicio="08:00"hFinal="09:00"id="3"idCurso="IN0205"
 mnemonico="C. P. B. N."sdia="Martes"/>
 <diaInicio="09:00"hFinal="10:00"id="3"idCurso="IN0706"
 mnemonico="Tall. S. I."sdia="Martes"/>
 <diaInicio="10:00"hFinal="11:00"id="3"idCurso="IN0706"
 mnemonico="Tall. S. I."sdia="Martes"/>
 <diaInicio="11:00"hFinal="12:00"id="3"idCurso="SIS341"
 mnemonico="A. E. Dat."sdia="Martes"/>
 <diaInicio="12:00"hFinal="13:00"id="3"idCurso="SIS341"
 mnemonico="A. E. Dat."sdia="Martes"/>
 <diaInicio="13:00"hFinal="14:00"id="3"idCurso="PE0801"
 mnemonico="G. B. Dat."sdia="Martes"/>
 <diaInicio="14:00"hFinal="15:00"id="3"idCurso="PE0801"
 mnemonico="G. B. Dat."sdia="Martes"/>
 <diaInicio="15:00"hFinal="16:00"id="3"idCurso="TO1003"
 mnemonico="Com. El."sdia="Martes"/>
 <diaInicio="16:00"hFinal="17:00"id="3"idCurso="TO1003"
 mnemonico="Com. El."sdia="Martes"/>
 <diaInicio="07:00"hFinal="08:00"id="4"idCurso="SIS341"
 mnemonico="A. E. Dat."sdia="Miercoles"/>
 <diaInicio="08:00"hFinal="09:00"id="4"idCurso="SIS341"
 mnemonico="A. E. Dat."sdia="Miercoles"/>
 <diaInicio="09:00"hFinal="10:00"id="4"idCurso="PB0405"
 mnemonico="Met. Num."sdia="Miercoles"/>
 <diaInicio="10:00"hFinal="11:00"id="4"idCurso="PB0405"
 mnemonico="Met. Num."sdia="Miercoles"/>
 <diaInicio="11:00"hFinal="12:00"id="4"idCurso="IN0205"
 mnemonico="C. P. B. N."sdia="Miercoles"/>
 <diaInicio="12:00"hFinal="13:00"id="4"idCurso="SO0903"
 mnemonico="S. I. Dist."sdia="Miercoles"/>
 <diaInicio="13:00"hFinal="14:00"id="4"idCurso="SIS201"
 mnemonico="Inf. Bas."sdia="Miercoles"/>
 <diaInicio="14:00"hFinal="15:00"id="4"idCurso="SIS201"
 mnemonico="Inf. Bas."sdia="Miercoles"/>
 <diaInicio="15:00"hFinal="16:00"id="4"idCurso="TO1003"
 mnemonico="Com. El."sdia="Miercoles"/>
 <diaInicio="16:00"hFinal="17:00"id="4"idCurso="TO1003"
 mnemonico="Com. El."sdia="Miercoles"/>
 <diaInicio="17:00"hFinal="18:00"id="4"idCurso="AO0903"
 mnemonico="Diag. Org."sdia="Miercoles"/>
 <diaInicio="18:00"hFinal="19:00"id="4"idCurso="AO0903"
 mnemonico="Diag. Org."sdia="Miercoles"/>
 <diaInicio="07:00"hFinal="08:00"id="5"idCurso="SO0903"

```

mnemonico="S. I. Dist."sdia="Jueves"/>
<diaInicio="08:00"hFinal="09:00"id="5"idCurso="SO0903"
mnemonico="S. I. Dist."sdia="Jueves"/>
<diaInicio="09:00"hFinal="10:00"id="5"idCurso="IN0205"
mnemonico="C. P. B. N."sdia="Jueves"/>
<diaInicio="10:00"hFinal="11:00"id="5"idCurso="IN0205"
mnemonico="C. P. B. N."sdia="Jueves"/>
<diaInicio="11:00"hFinal="12:00"id="5"idCurso="IN0706"
mnemonico="Tall. S. I."sdia="Jueves"/>
<diaInicio="12:00"hFinal="13:00"id="5"idCurso="IN0706"
mnemonico="Tall. S. I."sdia="Jueves"/>
<diaInicio="13:00"hFinal="14:00"id="5"idCurso="SIS201"
mnemonico="Inf. Bas."sdia="Jueves"/>
<diaInicio="14:00"hFinal="15:00"id="5"idCurso="PE0801"
mnemonico="G. B. Dat."sdia="Jueves"/>
<diaInicio="15:00"hFinal="16:00"id="5"idCurso="SO0902"
mnemonico="D. A A BD"sdia="Jueves"/>
<diaInicio="16:00"hFinal="17:00"id="5"idCurso="SO0902"
mnemonico="D. A A BD"sdia="Jueves"/>
<diaInicio="17:00"hFinal="18:00"id="5"idCurso="AO0903"
mnemonico="Diag. Org."sdia="Jueves"/>
<diaInicio="07:00"hFinal="08:00"id="6"idCurso="FG1006"
mnemonico="S. I. Geo."sdia="Viernes"/>
<diaInicio="08:00"hFinal="09:00"id="6"idCurso="FG1006"
mnemonico="S. I. Geo."sdia="Viernes"/>
<diaInicio="09:00"hFinal="10:00"id="6"idCurso="SIS341"
mnemonico="A. E. Dat."sdia="Viernes"/>
<diaInicio="10:00"hFinal="11:00"id="6"idCurso="SIS341"
mnemonico="A. E. Dat."sdia="Viernes"/>
<diaInicio="13:00"hFinal="14:00"id="6"idCurso="PE0801"
mnemonico="G. B. Dat."sdia="Viernes"/>
<diaInicio="14:00"hFinal="15:00"id="6"idCurso="PE0801"
mnemonico="G. B. Dat."sdia="Viernes"/>
<diaInicio="15:00"hFinal="16:00"id="6"idCurso="TO1003"
mnemonico="Com. El."sdia="Viernes"/>
<diaInicio="16:00"hFinal="17:00"id="6"idCurso="TO1003"
mnemonico="Com. El."sdia="Viernes"/>
</horario>

```

Operaciones en la Aplicación

En la aplicación se ha generado las siguientes operaciones para la interacción de cada una de las opciones diseñadas.

```

public void loadPersona(String usuario, String passwd, String
categoria)
{
Comunicacion comunicacion = new Comunicacion();
comunicacion.conectar("persona", usuario, passwd, categoria);
comunicacion.run();
BeanPersona beanPersona = comunicacion.getBeanPersona();
if( !beanPersona.getCodigo().equals("") ) {
persona = new Persona(this, beanPersona);
getDisplay().setCurrent(persona);
} else {

```

```

StringItem message = newStringItem("", "Usuario ó Contraseña No
Válida");
loginForm = newLogin(this);
loginForm.setMessage(message);
getDisplay().setCurrent(loginForm);
}
}

publicvoid loadCursos(String codigo, String categoria)
{
Comunicacion comunicacion = newComunicacion();
    comunicacion.conectar("cursos", codigo, "", categoria);
    comunicacion.run();
vCursos = comunicacion.getvCursos();
    codigo = comunicacion.getCodigo();
int numCursos = vCursos.size();
String[] cursos = new String[numCursos];
for(int i = 0; i < numCursos; i++)
{
BeanCurso beanCurso = (BeanCurso)vCursos.elementAt(i);
cursos[i] = (i+1)+" "+beanCurso.getNombre()+"",
"+beanCurso.getSemestre();
}
cursosList = newCursosList(this, cursos, codigo, categoria);
getDisplay().setCurrent(cursosList);
}

publicvoid viewCurso(String codigo, int opcion, Vector vCursos,
String categoria)
{
this.vCursos = vCursos;
int nuCursos = vCursos.size();
if(nuCursos > opcion) // Encasodetener elementos
{
BeanCurso beanCurso = (BeanCurso)vCursos.elementAt(opcion);
Comunicacion comunicacion = newComunicacion();
    comunicacion.conectar("curso",
beanCurso.getCodigoCurso(), codigo, categoria);
    comunicacion.run();
    beanCurso = comunicacion.getBeanCurso();
curso = newCurso(this, beanCurso, codigo, categoria);
getDisplay().setCurrent(curso);
}
}

publicvoid loadEstudiantes(int opcion, Vector vCursos, String
categoria)
{
this.vCursos = vCursos;
int numCursos = vCursos.size();
if(numCursos > opcion) // Encasodetener elementos
{
BeanCurso beanCurso = (BeanCurso)vCursos.elementAt(opcion);
    beanCurso.setOpcion(opcion);
Comunicacion comunicacion = newComunicacion();
    comunicacion.conectar("estudiantes",
beanCurso.getCodigoCurso(), "", categoria);
    comunicacion.run();
    Vector vEstudiantes = comunicacion.getvEstudiantes();
estudiantes = newEstudiantes(this, beanCurso, vEstudiantes,
categoria);
}
}

```

```

    getDisplay().setCurrent(estudiantes);
    }
}

public void setNota(BeanCurso beanCurso, Vector vEstudiantes, int
opcion)
{
    int numEstudiantes = vEstudiantes.size();
    if(numEstudiantes > opcion) // Encasodetener elementos
    {
        BeanPersona beanPersona =
        (BeanPersona)vEstudiantes.elementAt(opcion);
        NotasForm notasForm = new NotasForm(this, beanCurso, beanPersona);
        getDisplay().setCurrent(notasForm);
    }
}

public void saveNota(BeanCurso beanCurso, BeanPersona beanPersona)
{
    Comunicacion comunicacion = new Comunicacion();
    comunicacion.setBeanCurso(beanCurso);
    comunicacion.setBeanPersona(beanPersona);
    comunicacion.conectar("savenotas", "", "", "");
    comunicacion.run();
    getDisplay().setCurrent(estudiantes);
}

public void loadHorario(String codigo, String categoria)
{
    Comunicacion comunicacion = new Comunicacion();
    comunicacion.conectar("cursos", codigo, "", categoria);
    comunicacion.run();
    vCursos = comunicacion.getvCursos();
    codigo = comunicacion.getCodigo();
    int numCursos = vCursos.size();
    Hashtable htCursos = new Hashtable();
    for(int i = 0; i < numCursos; i++)
    {
        BeanCurso beanCurso = (BeanCurso)vCursos.elementAt(i);
        htCursos.put(beanCurso.getCodigoCurso(), beanCurso);
    }
    Vector vHorario = getHorario();
    horario = new Horario(this, htCursos, vHorario);
    getDisplay().setCurrent(horario);
}

```

Asimismo, se ha creado una clase que hace la petición de datos hacia el servidor donde se encuentra la base de datos.

```

package pe.edu.unap.academico.modelo;

import java.io.*;
import java.util.Vector;
import javax.microedition.io.Connector;
import javax.microedition.io.HttpConnection;

public class Comunicacion implements Runnable
{

```

```

private Thread      thread;
private HttpConnection connection;
private String      type;
private String      codigo;
private String      valor;
private String      categoria;
private BeanPerson beanPersona;
private BeanCurso beanCurso;
private Vector vCursos;
private Vector vEstudiantes;
private static String url =
"http://notas.sl8.eatj.com/academico/";

public Comunicacion()
{
}

public void conectar(String type, String codigo, String valor,
String categoria)
{
this.type = type;
this.codigo = codigo;
this.valor = valor;
this.categoria = categoria;
thread = new Thread(this);
thread.start();
this.vEstudiantes = null;
}

public void run()
{
if (type.equals("persona"))
{
if (beanPersona == null) {
loadPersona();
}
} elseif (type.equals("cursos"))
{
if (vCursos == null) {
loadCursos();
}
} elseif (type.equals("curso"))
{
if (beanCurso == null) {
loadCurso();
}
} elseif (type.equals("estudiantes"))
{
if (vEstudiantes == null) {
loadEstudiantes();
}
} elseif (type.equals("savenotas"))
{
if (vEstudiantes == null) {
saveNotas();
}
}
}

private void loadPersona()
{
beanPersona = new BeanPersona();
}

```



```

try {
    String urlweb =
url+"notas?a=autenticar&username="+codigo+"&passwd="+valor+"&use
rtype="+categoria;
connection = (HttpConnection)Connector.open(urlweb);
connection.setRequestProperty("Content-Language", "es-ES");
connection.setRequestProperty("User-Agent", "Profile/MIDP-2.0
Configuration/CLDC-1.0");
connection.setRequestProperty("Connection", "close");
connection.setRequestMethod(HttpConnection.GET);
    InputStream inputStream =
connection.openInputStream();
    procesarRespuestaPersona(connection, inputStream);
    } catch(Exception e) {
        e.printStackTrace();
    }
}

private void procesarRespuestaPersona(HttpConnection http,
InputStream inputStream) throws IOException
{
    if(http.getResponseCode() == HttpConnection.HTTP_OK)
    {
        byte datos[] = null;
        int len = (int)http.getLength();
        if(len != -1)
        {
            datos = new byte[len];
            inputStream.read(datos, 0, datos.length);
        } else {
            ByteArrayOutputStream baOutputStream = new
ByteArrayOutputStream();
            int ch;
            while( (ch = inputStream.read()) != -1 )
            {
                baOutputStream.write(ch);
            }
            datos = baOutputStream.toByteArray();
            baOutputStream.close();
        }
        beanPersona.changeFromByteArray(datos);
    } else {
        System.out.println( http.getResponseMessage() );
    }
}
}
}

```

Operaciones en el Servidor

Operaciones del servlet que se ejecuta en el lado servidor

```

package pe.edu.unap.academico.servlet;

import java.io.*;
import java.util.*;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;

```

```

import javax.servlet.http.*;
import pe.edu.unap.academico.bean.*;
import pe.edu.unap.academico.model.*;

@WebServlet(name="ServletNotas", urlPatterns={"/notas"})
public class ServletNotas extends HttpServlet {

    protected void processRequest(HttpServletRequest request,
    HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("application/octet-stream");
        response.setStatus(response.SC_OK);
        String accion = request.getParameter("a");
        String usertype = request.getParameter("usertype");
        if(usertype == null || usertype.equals("")) {
            usertype = "estudiante";
        }
        BeanPersona beanPersona = new BeanPersona();
        Acceso acceso = new Acceso();
        byte[] datos = null;
        if(accion.equals("autenticar"))
        {
            String usuario = request.getParameter("username");
            String passwd = request.getParameter("passwd");

            HashMap hmAcceso = new HashMap();
            hmAcceso = (HashMap)acceso.firmar(usuario, passwd,
usertype);
            String autorizado = (String)hmAcceso.get("message");
            if(autorizado != null &&
autorizado.equals("autenticado"))
            {
                beanPersona =
(BeanPersona)hmAcceso.get("beanPersona");
                datos = beanPersona.changeToByteArray(autorizado);
            } else {
                datos = beanPersona.changeToByteArray();
            }
        } else if(accion.equals("cursos"))
        {
            String codigo = request.getParameter("codigo");
            beanPersona = acceso.findPersona(codigo, usertype);
            List listCursos = acceso.getCursos(beanPersona);
            try {
                ByteArrayOutputStream baOutputStream = new
ByteArrayOutputStream();
                DataOutputStream dataOutputStream = new
DataOutputStream(baOutputStream);
                int numCursos = listCursos.size();
                dataOutputStream.writeInt(numCursos);
                for(int i=0; i<numCursos; i++)
                {
                    BeanCurso beanCurso =
(BeanCurso)listCursos.get(i);
                    String curso =
beanCurso.getCodigoCurso()+":"+beanCurso.getNombre()+":"+beanCurso
.getSemestre();
                    dataOutputStream.writeUTF(curso);
                }
                dataOutputStream.flush();
                datos = baOutputStream.toByteArray();
            }
        }
    }
}

```

```

        baOutputStream.close();
        dataOutputStream.close();
    } catch (Exception e) {
        e.printStackTrace();
    }
} else if(accion.equals("estudiantes"))
{
    String codigo = request.getParameter("codigo");
    List listEstudiantes = acceso.getEstudiantes(codigo,
usertype);
    try {
        ByteArrayOutputStream baOutputStream = new
ByteArrayOutputStream();
        DataOutputStream dataOutputStream = new
DataOutputStream(baOutputStream);
        int numEstudiantes = listEstudiantes.size();
        dataOutputStream.writeInt(numEstudiantes);
        for(int i=0; i<numEstudiantes; i++)
        {
            BeanPersona persona =
(BeanPersona)listEstudiantes.get(i);
            String nombres =
persona.getCodigo()+" "+persona.getNombres();
            dataOutputStream.writeUTF(nombres);
        }
        dataOutputStream.flush();
        datos = baOutputStream.toByteArray();

        baOutputStream.close();
        dataOutputStream.close();
    } catch (Exception e) {
        e.printStackTrace();
    }
}

if(datos != null)
{
    response.setContentLength(datos.length);
    OutputStream outputStream =
response.getOutputStream();
    outputStream.write(datos);
    outputStream.close();
}

@Override
protected void doGet(HttpServletRequest request,
HttpServletResponse response)
throws ServletException, IOException {
    processRequest(request, response);
}

@Override
protected void doPost(HttpServletRequest request,
HttpServletResponse response)
throws ServletException, IOException {
    processRequest(request, response);
}

@Override
public String getServletInfo() {    return "Short
description";    }}

```

**Universidad Nacional del
Altiplano**

ESCUELA DE POST GRADO

MAESTRIA EN INFORMATICA

AYUDA DEL SISTEMA



C.U. Puno Abril del 2011

El Sistema de Consulta de Notas SCN, es un software basado en tecnologías móviles que tiene como finalidad brindar a los estudiantes universitarios de la Universidad Nacional del Altiplano el servicio de poder acceder a consultar las notas de las asignaturas en las que se encuentran matriculados utilizando tecnologías móviles. El beneficio que ofrece este software es poder acceder desde cualquier lugar y en el momento que el estudiante desee a la información de su rendimiento académico en la universidad.

El presente software requiere ser instalado en los dispositivos móviles, donde los archivos se pueden descargar, el que se encuentra en la dirección URL: <http://notas.s18.eatj.com> en los archivos de extensión .jar y .jad, los cuales tienen que ser instalados en el dispositivo móvil (celular).

REQUERIMIENTOS MINIMOS DE SISTEMA

Para tener acceso al Sistema de consulta de notas en móviles, se debe contar con un teléfono celular que cuente con las siguientes características:

- Soporte JAVA
- Perfiles : MIDP 1.0
- Configuración: CLDC 2.0
- Factibilidad de driver del equipo celular
- Tener activado el servicio de Internet ofrecido por las empresas de telefonía celular

Previamente al empleo del Sistema, instalar el Midlet en el teléfono celular

INSTALACION DEL SISTEMA ACADEMICO DE CONSULTA DE NOTAS (SCN)

Para instalar la aplicación se debe tener en el celular los archivos con extensión .jar y .jad, los cuales tienen que ser instalados en el dispositivo móvil (celular). Ambos tipos de archivos se refieren a aplicaciones JAVA, los .jad son descompiladores, algunos celulares, hasta lo generan automáticamente, el archivo más importante es el que tiene la extensión .jar y es el que necesitas tener.

Ahora lo que vas a hacer es conectar tu celular al PC, puede ser con el cable que viene con el teléfono. Si se tiene el sistema operativo Windows, te reconoce el teléfono como una unidad de disco. Luego abres la unidad que reconoce el celular, por ejemplo: "Nokia"; allí encontrarás varias carpetas; debes abrir la carpeta "Applications". Luego abrir la carpeta de tu PC donde tienes los archivos que descargaste y vas a pasar esos archivos .jar y .jad a la carpeta "Applications" de tu celular. Cuando veas que ya los pasaste a la carpeta de tu celular lo que haces ahora es desconectar tu celular del ordenador. Pero debes tener en cuenta los pasos para "desenchufar" un dispositivo USB: "quitar hardware con seguridad", luego "extracción

segura...."). Ahora teniendo el celular desconectado del PC, vas a hacer lo siguiente con tu celular: y ubicas la aplicación donde aparece con su nombre que bajaste, en este caso saldra "MobileNotas"; seleccionas la opcion de "selec". Ahi veras al telefono instalando la aplicación. Este mismo proceso lo puedes hacer para instalar otras aplicaciones. Luego buscas en juegos o en aplicaciones lo que recién instalaste.

SISTEMA ACADEMICO DE CONSULTA DE NOTAS (SCN)

El Sistema Academico de consulta de Notas tiene dos perspectivas de uso:

- Docente
- Estudiante

USUARIO : DOCENTE - ESTUDIANTE

Esta ayuda que permite conocer el sistema de consulta de notas en moviles, las opciones y la forma de usar el sistema a los estudiantes de la escuela profesional de Ingeniería de Sistema de la Universidad Nacional del Altiplano Puno.

El Sistema permite al docente y estudiante de la Escuela Profesional de Ingeniería de Sistemas a realizar lo siguiente:

Docente: Ingresar al Sistema
Horario de Asignaturas que dicta (Carga Academica)
Seleccionar asignatura
Ingresar notas de los estudiantes

Estudiante: Ingresar al Sistema
Horario de Asignaturas matriculado
Seleccionar asignatura
Consultar las notas de las asignaturas

ACCESO A LA APLICACIÓN EN EL CELULAR

Los celulares cuentan con menu de opciones, entonces para ejemplificar el uso del sistema de consulta de notas se ha instalado la aplicación en un celular Nokia, entonces para acceder al sistema seleccionar el menu Aplicaciones como se muestra en la siguiente figura:

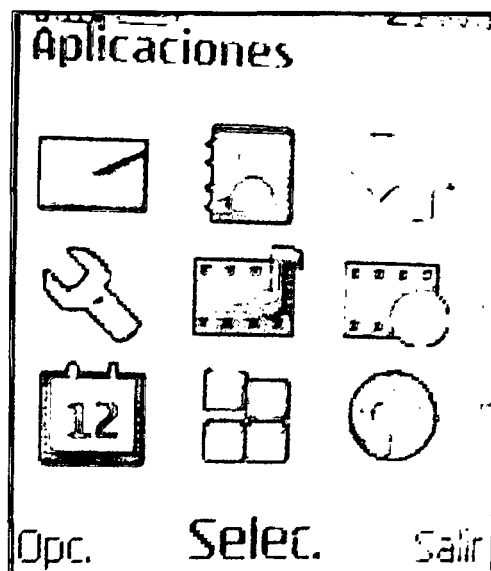


FIGURA 54

MENU DE CELULAR NOKIA PARA SELECCIONAR APLICACIONES DEL CELULAR

Luego de seleccionar aplicaciones, acceder a la opción Juegos del celular y presionar abrir, encontrará la aplicación MobileNotas con el logo de la Universidad Nacional del Altiplano Puno con una llave, luego presionar la opción Abrir .

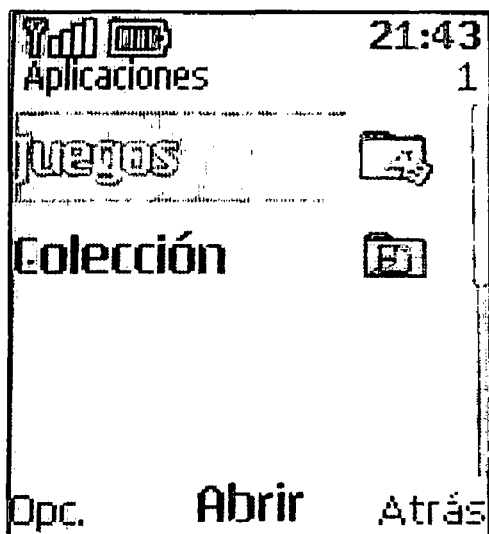


FIGURA 55

SELECCIÓNAR APLICACIÓN JUEGOS DEL CELULAR

Fuente: Elaboración propia



FIGURA 56

SELECCIONAR APLICACIÓN MOBILENOTAS EN EL CELULAR

Fuente: Elaboración propia

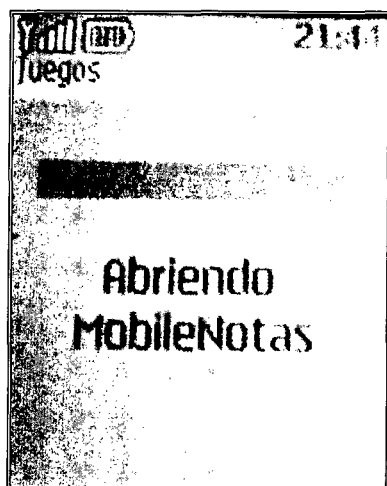


FIGURA 57
PANTALLA APERTURA DE APLICACIÓN MOBILENOTAS

Fuente: Elaboración propia

INGRESO AL SISTEMA DE CONSULTA DE NOTAS

Una vez asignado el login (Usuario) y su clave (Contraseña) de acceso al sistema, en la primera pantalla del sistema presenta opciones para loguarse en el sistema de Consulta de notas denominado Sistema Academico en el celular, aparece un formulario de ingreso, donde se debe ingresar lo siguiente:

- El usuario (codigo del estudiante)
- Contraseña
- Seleccionar la categoria : Docente / Estudiante

La primera opción permite al usuario (Docente/estudiante) que verifique si se encuentra registrado en la base datos del sistema de consulta de notas donde ingresa con el codigo que le corresponde como docente / estudiante y asi mismo ingresar la contraseña asignada que le permite loguarse al docente /estudiante, y seguidamente selecciona la categoría de usuario que presenta sólo dos opciones Docente y Estudiante.

En caso de que estos datos ingresados estuvieran errados envia un mensaje de error; en caso contrario puede acceder a la siguiente pantalla que pregunta: ¿Permitir acceso a la red?, entonces se tiene que presionar la opcion SI para acceder al menu del sistema.

El Usuario escribe el código, la contraseña y selecciona la categoría de usuario, sea Docente/Estudiante de la universidad . En el caso de ser estudiante ingresa su código de matrícula y si el usuario es docente tiene que ingresar el código asignado como docente

**INGRESO AL SISTEMA DE CONSULTA DE NOTAS
DOCENTE / ESTUDIANTE**

Ingreso de código



FIGURA 58

PANTALLA - INGRESO DE USUARIO AL SISTEMA

Fuente: Elaboración propia

Ingreso de contraseña

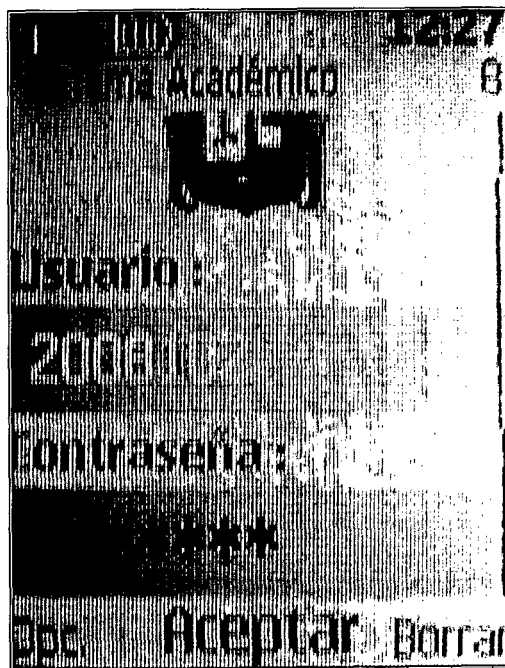
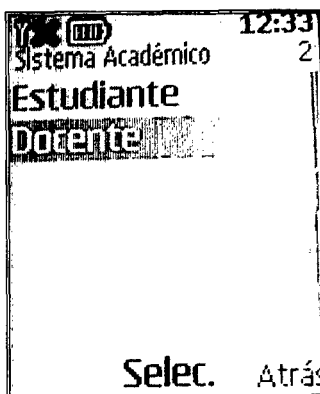


FIGURA 59

PANTALLA - INGRESO DE CONTRASEÑA AL SISTEMA

Fuente: Elaboración propia

SELECCIÓN DE CATEGORIA EN EL SISTEMA DE CONSULTA DE NOTAS



DOCENTE

ESTUDIANTE

FIGURA 60

PANTALLA – MENU DE SELECCION CATEGORÍA DE USUARIO

Fuente: Elaboración propia

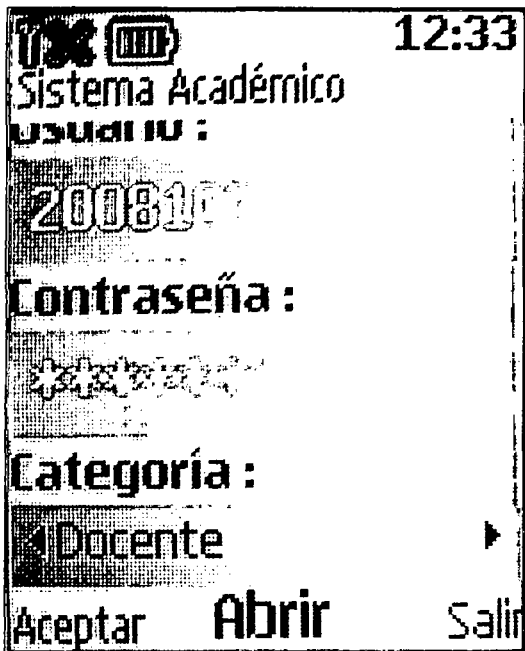


FIGURA 61

PANTALLA SELECCIÓN DE CATEGORIA:
DOCENTE

Fuente: Elaboración propia

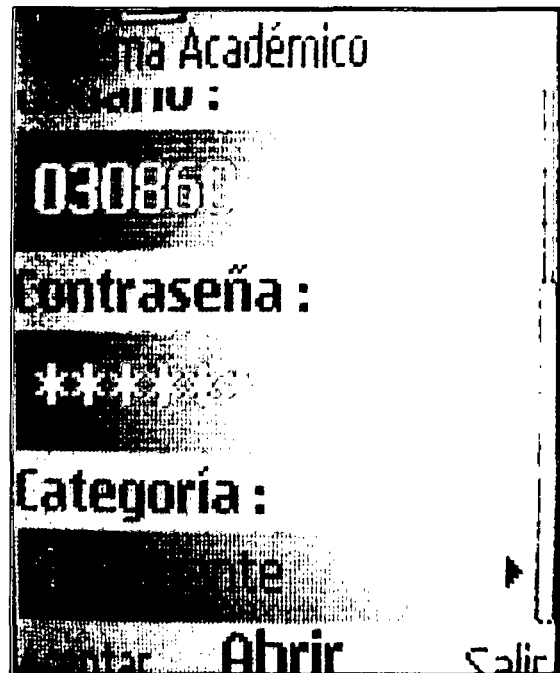


FIGURA 62

PANTALLA DE SELECCIÓN DE CATEGORIA
ESTUDIANTE

Fuente: Elaboración propia

Enseguida presionar **Abrir** , y saldra un mensaje con la pregunta: ¿Permitir acceso a la red? , para poder usar la aplicación que tiene la información en la base de datos de un servidor, seleccionar la opcion SI para el acceso a la aplicación.

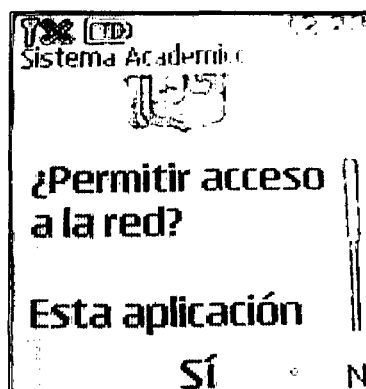


FIGURA 63

PANTALLA DE ACCESO A LA APLICACIÓN MOBILENOTAS

Fuente: Elaboración propia

Si el usuario ingresa mal sus datos aparece nuevamente el formulario de ingreso de datos, para volver a ingresar los datos. Pero si realiza una autenticación exitosa, se puede observar los datos del usuario (Docente / Estudiante) y se tendra menu con opciones

DATOS DEL USUARIO DEL SISTEMA

DOCENTE

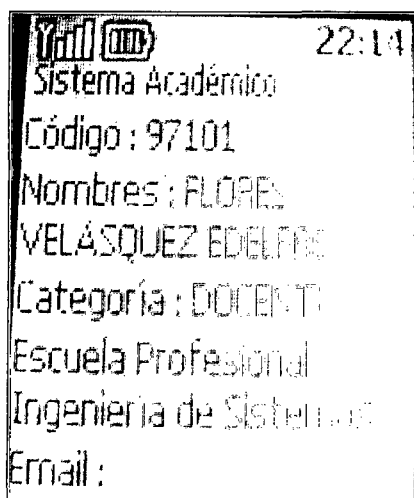


FIGURA 64

PANTALLA-DATOS DEL DOCENTE

Fuente: Elaboración propia

ESTUDIANTE

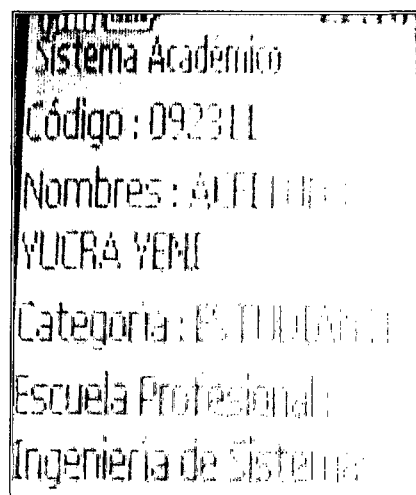


FIGURA 65

PANTALLA-DATOS DEL ESTUDIANTE

Fuente: Elaboración propia

HORARIO DE LAS ASIGNATURAS

Observando el celular y dependiendo de la información que desee consultar el usuario (Docente / Estudiante) se tiene dos opciones Cursos y Horario, entonces si el usuario selecciona **Horario** se puede considerar lo siguiente:

- Si el usuario es **docente** le muestra el horario asignado de los cursos que se le dio en la carga academica.
- Si el usuario es **estudiante** y selecciona la opcion horario le mostrará el Horario de las asignaturas en las cuales se encuentra matriculado.

HORARIO DEL USUARIO DEL SISTEMA

DOCENTE

ESTUDIANTE

Hora	Lun	Dia
16-17		
17-18		
18-19		

50002: Diseño y Elaboración de...
10003: Diagramación de...
10103: Modelos de...

22:14

FIGURA 66
PANTALLA - HORARIO DEL DOCENTE
Fuente: Elaboración propia

Hora	Lun	Dia
07-08	PE0405	R02
08-09	PE0405	R02
09-10	SO0903	
10-11	SO0903	
11-12		
12-13		

22:10

FIGURA 67
PANTALLA - HORARIO DEL ESTUDIANTE
Fuente: Elaboración propia

Debido a la limitación de espacio en los dispositivos móviles, la información mensajes no se muestran en su totalidad, sino que se emplea los botones de desplazamiento que permite visualizar para ver el texto restante.

SELECCIONAR Y CONSULTAR NOTAS DE LAS ASIGNATURAS

El menu del sistema tiene la opción Cursos, en donde puede observar las asignaturas o cursos y dependiendo de la categoría puede realizar lo siguiente:

- Si es Docente puede ver los cursos que dicta, seleccionar uno de ellos para poder ingresar notas.
- Si es estudiante y desea consultar las notas de una asignatura, selecciona cursos entonces se obtendrá la lista de cursos en las que el alumno se encuentra matriculado, luego selecciona el curso sobre el cual desea conocer su nota elige la opción Ver para acceder a las notas obtenidas en el curso seleccionado.

CURSOS EN EL SISTEMA DE CONSULTA DE NOTAS

DOCENTE: Se muestra la carga academica

ESTUDIANTE: Se muestra los cursos en los que se encuentra matriculado el estudiante

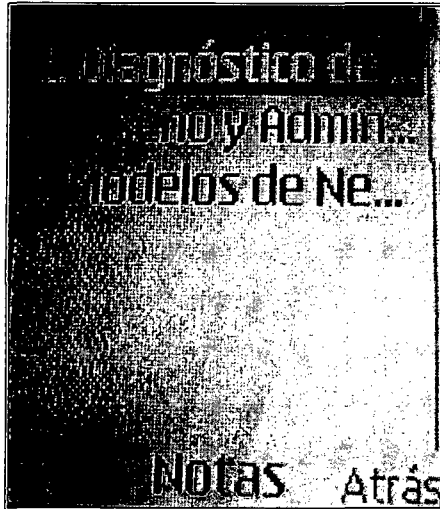


FIGURA 68

PANTALLA - CARGA ACADEMICA DOCENTE

Fuente: Elaboración propia

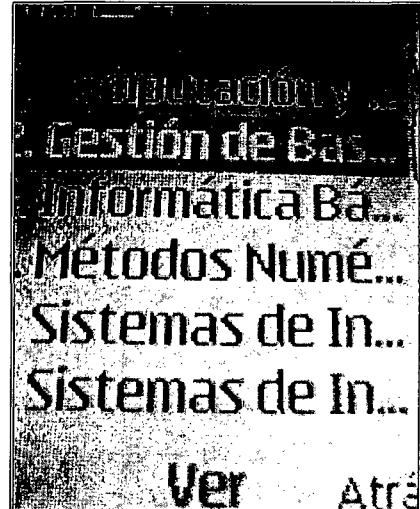


FIGURA 69

PANTALLA - CURSOS DEL ESTUDIANTE

Fuente: Elaboración propia

A partir de esta opción Cursos, varía para cada uno de los usuarios, puesto que si el usuario es Estudiante seleccionando la opción **Ver** puede obtener las notas parciales del curso seleccionado, tal como se muestra en la siguiente figura:

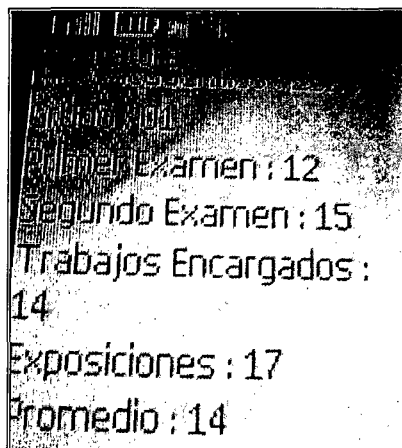


FIGURA 70

PANTALLA - NOTAS DEL ESTUDIANTE

Fuente: Elaboración propia

En caso de ser el usuario Docente, si desea registrar las notas de los estudiantes se aparecerá la lista de estudiantes matriculados en la asignatura que selecciono al seleccionar la opción **notas**, teniendo la lista de estudiantes seleccionar del que desea registrar las notas parciales:

REGISTRO DE NOTAS DE ESTUDIANTES DE LA E.P. INGENIERIA DE SISTEMAS

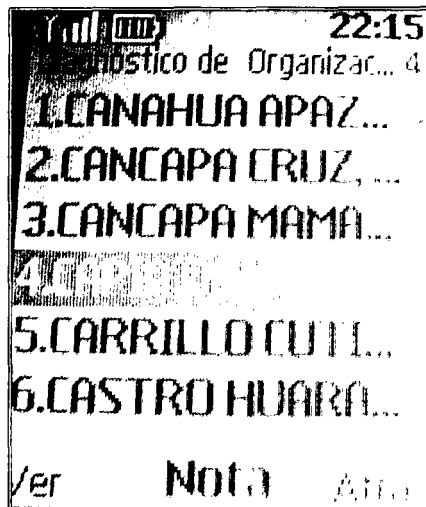


FIGURA 71
PANTALLA - LISTA DE ESTUDIANTES
Fuente: Elaboración propia

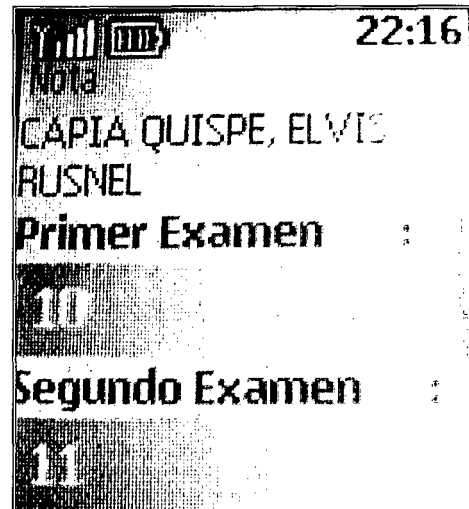


FIGURA 72
PANTALLA – FORMULARIO PARA INGRESAR NOTAS DEL ESTUDIANTE
Fuente: Elaboración propia