

UNIVERSIDAD NACIONAL DE INGENIERÍA

FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA



APLICACIÓN WAP PARA EJECUCION DE OBRAS PARA LUZ DEL SUR

INFORME DE SUFICIENCIA

PARA OPTAR EL TÍTULO PROFESIONAL DE:

INGENIERO ELECTRÓNICO

PRESENTADO POR:

CHRISTIAN ALFREDO VILA ORMEÑO

**PROMOCIÓN
2003-I**

**LIMA – PERÚ
2006**

**APLICACION WAP PARA EJECUCION DE OBRAS
PARA LUZ DEL SUR**

*Dedico este trabajo a:
Mi abuelo, que demostró que nada se
consigue sin sacrificio, que lo mas
valioso en la vida de cada uno es la
familia y que aunque él ya no se encuentre
en este mundo fue mi imagen a imitar y
mi ejemplo a seguir en la vida.*

**GRACIAS ABUELO.
CARLOS ORMEÑO VERA**

S U M A R I O

Se ha pretendido hacer un informe que recopile todo lo relacionado con la tecnología WAP.

Lo que se esta desarrollando son programas aplicados a la información en línea vía Nextel, Luz del Sur tiene un sistema operativo llamado SPO (Sistema de Proceso de Obras) que es una herramienta para controlar:

- Fecha de inicio de la OT
- Fecha de instalación del medidor
- Fecha de término de OT
- Pre - Liquidación
- Liquidación
- Facturación
- Pago

Durante todas estas etapas la información es actualizada mediante el ingreso de datos al sistema, el usuario tiene que estar actualizándolo mediante el ingreso de datos desde una PC y esto normalmente toma un determinado tiempo entre lo realizado y la actualización, es aquí donde el desarrollo entra a formar parte del ingreso de datos de manera inmediata en línea, este ingreso se hace desde el equipo Nextel.

INDICE

PROLOGO

CAPÍTULO I: INTRODUCCION Y DESCRIPCION DEL SISTEMA WAP

| | | |
|--------|--|----|
| 1.1 | Definición de WAP | 02 |
| 1.2 | WAP 2.0 | 03 |
| 1.3 | Componentes de la Arquitectura WAP | 04 |
| 1.3.1. | Capa de Aplicación (WAE) | 05 |
| 1.3.2. | Capa de Sesión (WSP) | 06 |
| 1.3.3. | Capa de Transacciones (WTP) | 06 |
| 1.3.4. | Capa de Seguridad (WTLS) | 07 |
| 1.3.5. | Capa de Transporte (WDP) | 07 |
| 1.4 | Como Realizar una Aplicación WAP | 08 |
| 1.5 | Ejemplo de una Red WAP | 11 |
| 1.6 | WAP, la unión de Internet y el celular | 12 |

CAPÍTULO II: INTRODUCCION AL LENGUAJE WML

| | | |
|--------|-------------------------------------|----|
| 2.1 | Definición WML | 15 |
| 2.2 | Características WML | 16 |
| 2.3 | Definición de Carta y Baraja | 16 |
| 2.4 | Conjunto de juego de caracteres WML | 19 |
| 2.5 | Etiquetas WML | 20 |
| 2.5.1. | La etiqueta Card | 20 |
| 2.5.2. | La etiqueta template | 22 |
| 2.5.3. | La etiqueta accés | 23 |
| 2.5.4. | La etiqueta do | 24 |
| 2.5.5. | La etiqueta go | 26 |
| 2.5.6. | La etiqueta img | 27 |
| 2.5.7 | La etiqueta input | 29 |
| 2.5.8 | La etiqueta option | 31 |

| | | |
|--|--|----|
| 2.5.9 | La etiqueta select | 32 |
| 2.5.10 | La etiqueta prev | 34 |
| 2.5.11 | La etiqueta wml | 35 |
| | | |
| CAPÍTULO III: DESCRIPCION DEL PROYECTO Y DE SUS COMPONENTES | | |
| 3.1 | Descripción del Proyecto de Aplicación Wap via Nextel para Luz del Sur | 37 |
| 3.2 | Diagrama de Comunicación | 39 |
| 3.3 | Definición de Nextel | 40 |
| 3.4 | Definición de IDEN | 41 |
| 3.5 | Servidores Utilizados por Luz del Sur | 41 |
| 3.6 | Definición de Linux | 44 |
| 3.7 | Definición de Informix | 45 |
| 3.7.1 | Informix® el Servidor Dinámico (IDS) | 45 |
| 3.7.2 | Plataformas Soportadas | 47 |
| 3.7.3 | Requerimientos Mínimos | 47 |
| 3.7.4 | Integración con otros Productos | 48 |
| 3.8 | Diferencia entre Informix y Oracle | 48 |
| | | |
| CAPÍTULO IV: EVOLUCION DEL PROCESO DE EJECUCION DE OBRAS POR CONTRATISTAS | | |
| 4.1 | Introducción | 50 |
| 4.2 | Proceso de ejecución de obras: Anterior de la obra | 50 |
| 4.3 | Proceso de ejecución de obras: Nuevo | 57 |
| 4.4 | Tabla comparativa de los procesos Anterior, Actual y Nuevo. | 63 |
| | | |
| | CONCLUSIONES | 65 |
| | ANEXO A | 67 |
| | BIBLIOGRAFIA | 75 |

PROLOGO

Este informe consta de cuatro capítulos y un anexo, estructurados de la siguiente manera:

El Capítulo I, referido al la introducción y descripción del sistema WAP, cual es su definición, componentes, arquitectura, ejemplo y utilidades.

El Capítulo II, referido al lenguaje WML, definición, conjunto de caracteres, etiquetas.

El Capítulo III, se encarga del proyecto en si, los componentes que utiliza y la descripción de cada uno de ellos.

El Capítulo IV, como se realiza el proceso de la ejecución de la obra, haciendo una comparación entre lo que era la ejecución anterior y lo que será el nuevo proceso con la implementación del proyecto.

CAPITULO I

DEFINICIONES Y ARQUITECTURA WAP

1.1 Definición Wap

WAP son las siglas de *Wireless Application Protocol*, o, dicho de otra manera, se trata de un protocolo para la transmisión de datos por un medio inalámbrico.

Este protocolo de aplicaciones inalámbricas surge de dos tecnologías como son:

- Las comunicaciones Inalámbricas
- Internet

WAP, es un estándar global, no controlado por ninguna compañía en particular. Ericsson, Motorola y Unwires Planet fundaron el WAP FORUM en 1997 con el propósito de definir especificaciones para el desarrollo de aplicaciones en redes de comunicación inalámbricas.

Esta plataforma lo que pretende es ofrecer a los usuarios las mismas funcionalidades que se nos ofrecen en la actualidad en la Web, pero con las ventajas y características propias que ofrece este dispositivo: el teléfono móvil.

Para realizar los documentos que se transmiten vía este protocolo, se utiliza un lenguaje llamado WML, que responde a las siglas de *Wireless Markup Language*, o lenguaje de marcas para inalámbricos.

WAP es una serie de tecnologías que consisten:

- WML, que es el lenguaje de etiquetas.

- WMLScript es un lenguaje de script, lo que vendría a ser JavaScript
- Wireless Telephony Application Interface (WTAI).

Este lenguaje es descendiente del XML (*eXtensible Markup Language*), con unas etiquetas bastante similares a las del HTML. De hecho, si sabes HTML tienes mucho terreno avanzado.

Hoy en la actualidad cualquier usuario desde su PC puede conectarse desde su hogar a Internet y gracias a un proveedor de servicios web, puede acceder de forma sencilla a cualquier tipo de aplicación que se ofrece en la red, por ejemplo: compra de billetes de avión, acceder a boletines informativos, gestión de sus carteras de valores, etc, en fin un gran número de aplicaciones y utilidades, que ahora gracias a la tecnología WAP se hacen asequibles desde tu teléfono móvil, claro que con las ventajas y desventajas que el teléfono aporta como nuevo medio de navegación.

1.2 Wap 2.0

WAP Forum lanzó su primera especificación WAP 1.0 en 1998. WAP 2.0 es la siguiente generación de especificaciones que, como las versiones anteriores, marcó los esfuerzos continuos del Foro WAP para adoptar las más recientes normas y protocolos de Internet. WAP 2.0 también se optimiza para el uso de anchos de banda más altos y la conexión de paquetes, basado en redes inalámbricas. Mientras mejora la utilización y el soporte en las capacidades de las últimas tecnologías de los dispositivos inalámbricos e Internet, WAP 2.0 también proporciona manejo para la compatibilidad en lo que refiere a aplicaciones y servicios que proporcionan las versiones anteriores de WAP.

Los elementos principales de la arquitectura WAP 2.0 son los siguientes:

Protocol Stack Support : Además de WAP stack introducida en WAP 1, WAP 2.0 agrega soportes y servicios en una stack basada en la stack de la Internet estándar, incluyendo el soporte para TCP, TLS y HTTP. Al contener dos stacks, WAP 2.0 proporciona un modelo de conectividad en un amplio rango de redes y portadores inalámbricos.

WAP Application Environment : Normalmente visto como el “WAP Browser” (Navegador de WAP), el ambiente de la aplicación WAP 2.0 ha evolucionado para adoptar los estándares de la Internet Browser Markup Language. Esto ha llevado a la definición del XHTML Mobile Profile (XHTMLMP). XHTMLMP se basa en la plataforma de Extensible HyperText Markup Language (XHTML) desarrollado por el W3C (World Wide Web Consortium) para reemplazar y mejorar el lenguaje HTML actualmente muy usado.

Servicios adicionales y Capacidades : Las especificaciones de WAP tuvieron artículos que no eran parte del “WAP Stack” ni del “WAP Browser”, pero ayudaron a enriquecer las especificaciones de WAP. Con WAP 2.0, hay un aumento considerable en el número de características disponibles para diseñadores, operadores y usuarios.

1.3 Componentes de la Arquitectura WAP

La arquitectura WAP está pensada para proporcionar un entorno escalable y extensible para el desarrollo de aplicaciones para dispositivos de comunicación móvil. Para ello, se define una estructura en capas, en la cual cada capa es accesible por la capa superior así como por otros servicios y aplicaciones a través de un conjunto de interfaces muy bien definidos y especificados. Las capas de la arquitectura WAP se recogen en el siguiente diagrama:



Fig. 1.1: Capas de la arquitectura WAP

Tabla 1.1: Descripción de las Capas WAP

| Identificación | Componente |
|-------------------------|---|
| WAE | Wireless Application Environment |
| WSP | Wireless Session Protocol |
| WTP | Wireless Transaction Protocol |
| WTLS | Wireless Transport Layer Security |
| WDP | Wireless Datagram Protocol |
| SMS, GPRS, CSD, USSD | Métodos de acceso al proveedor de servicios |

A continuación se pasará a describir cada una de las capas:

1.3.1 Capa de Aplicación (WAE)

Contiene las herramientas que usan los desarrolladores de contenido inalámbrico. Ahí se incluye WML y WMLScript, donde este último opera de forma similar a como lo hace JavaScript.

Es un entorno de aplicación de propósito general, basado en la combinación del World Wide Web y tecnologías de Comunicaciones Móviles.

Este entorno incluye un micro navegador, que posee las siguientes funcionalidades:

1. El lenguaje WML, del cual ya hemos hablado anteriormente.
2. El lenguaje WMLS, similar al JavaScript.
3. WTA (Wireless Telephony Applications), es un entorno para aplicaciones u servicios de telefonía.
4. WTAI (Wireless Telephony Application Interface), es una interfaz utilizada en los terminales móviles para operaciones locales de control de llamadas (recepción, iniciación y terminación) y acceso a listines telefónicos.

Una serie de formatos de contenido, que son un conjunto de datos definidos, entre los que se encuentran: imágenes, información de calendario.

1.3.2 Capa de Sesión (WSP)

Determina si la sesión entre el dispositivo y la red será orientada a la conexión o sin conexión, o sea, si el dispositivo necesitará o no de establecer comunicación una y otra vez hacia la red, después de iniciada la sesión.

Este protocolo proporciona a la Capa de Aplicación (WAE) interfaz con dos servicios de sesión:

1. Un servicio orientado a conexión que funciona por encima de la Capa de Transacciones (WTP).
2. Un servicio no orientado a conexión que funciona por encima de la Capa de Transporte (WTP), y que proporciona servicio de datagramas seguro o no seguro.

Esta capa proporciona las siguientes funcionalidades:

1. Establecimiento y liberación de conexiones entre cliente y servidor.,
2. Intercambio de información entre cliente y servidor.
3. Negociación de las características del protocolo.
4. Suspensión y reanudación de la sesión.

1.3.3 Capa de Transacciones (WTP)

Actúa como un controlador de tráfico, manteniendo los datos en flujo constante. También define cómo clasificar cada petición de transacción, si es posible hacerla en dos sentidos o en uno solo.

Este protocolo funciona por encima de un servicio de datagramas ya sean seguros como no seguros, y proporciona las siguientes funcionalidades:

Proporciona los servicios necesarios para soportar las transacciones, estos servicios pueden ser de tres clases:

1. Peticiones inseguras de un solo camino
2. Peticiones seguras de un solo camino
3. Transacciones seguras de dos caminos

También proporciona seguridad en las transacciones.

1.3.4 Capa de Seguridad (WTLS)

Se encarga de las capacidades de seguridad, revisa la integridad de los datos, así como el encriptamiento, y realiza autenticación de clientes y servidores.

La Capa Inalámbrica de Seguridad de transporte (WTLS) es un protocolo basado en el estándar SSL, utilizado en el entorno Web, para la seguridad en la transferencia de datos, esta capa proporciona a las capas de nivel superior de WAP una interfaz de servicio de transporte seguro, que lo resguarde de una interfaz de transporte inferior.

Las funcionalidades de esta capa son las siguientes:

1. **Integridad de los datos:** Se asegura que la información intercambiada entre el terminal y el servidor de aplicaciones, no haya sido modificada.
2. **Privacidad de los datos:** Se asegura que la información intercambiada entre el terminal y el servidor de aplicaciones, no pueda ser captada ni entendida por elementos externos a la comunicación.
3. **Autenticación:** Se ofrecen servicios para determinar la autenticidad del terminal y del servidor de aplicaciones.

También puede ser utilizado para el establecimiento de una comunicación segura entre terminales.

1.3.5 Capa de Transporte (WDP)

Trabaja en conjunto con la capa de transporte de la red, esto facilita la adaptación de WAP a una variedad de ambientes, ya que todas las necesidades

de cambio de información se mantienen en este nivel. Los transportes de red son las diversas tecnologías disponibles para el intercambio de datos digitales

El Protocolo Inalámbrico de Datagramas (WDP) proporciona las siguientes funcionalidades:

- Proporciona un servicio fiable a los protocolos de las capas superiores de WAP.
- Permite la comunicación de forma transparente sobre los protocolos portadores: CDMA, SMS, GSM.
- Una vez que la información llega al cliente WAP, se pasa a un navegador con funciones básicas, incluido en el software del dispositivo móvil.

1.4 Como realizar una aplicación WAP

Para realizar una aplicación WAP, tiene que quedar claro que es lo mismo que hacer una página Web, aunque con alguna variación.

Se ha dividido esta sección en dos partes:

1. Desarrollo de una página en Web.

Para el desarrollo de una página Web, lo primero de todo es estudiar un poco del lenguaje de desarrollo, en este caso, el lenguaje es el HTML para las páginas estáticas, y para las páginas dinámicas el Asp, Peral, etc.

Para empezar ya con la programación de la página, solo se precisa de un editor para escribir el código HTML de nuestra página; para el tema WAP, todavía no existe ningún programa de este tipo, y aún estamos sometidos a la pura y dura programación en WML.

Una vez que tenemos ya nuestra página codificada, para poder ver como quedaría si realmente estuviera en un servidor, tenemos los browser o navegadores: Microsoft Internet Explorer, Netscape Communicator, que nos permiten ver nuestra página aunque este ubicada en nuestro disco duro; lo que no nos permiten es ejecutar las

páginas dinámicas, y entonces, si que habría que ubicarla en el servidor para poder ejecutarla.

Cuando ya este lista, y hayamos hecho las pruebas pertinentes de cómo queda con nuestro navegador, entonces podemos llevar la página Web a un servidor, y una vez ubicada, cualquiera que posea un ordenador con acceso a Internet con el navegador que este utilizando, podrá acceder a nuestra página.

11. **Desarrollo de una página para WAP.**

Para el desarrollo de una pagina WAP, lo primero es familiarizarse con el lenguaje de desarrollo, veíamos que para las páginas Web, usábamos el HTML, asp, JavaScript, en este caso el lenguaje de desarrollo es el WML para las páginas estáticas, y el WMLS para las páginas dinámicas.

A continuación se precisa cualquier editor de textos, para escribir el código de nuestra pagina WAP, aunque es recomendable, como primer paso, utilizar un developer toolkit, es decir, un paquete de herramientas para desarrolladores de aplicaciones WAP, en la actualidad tenemos varios tipos: Nokia Wap Toolkit, UP.SDK, Kit de Ericsson. Estas herramientas para desarrollar aplicaciones WAP, son muy útiles, ya que contienen editores, compiladores y ejemplos de aplicaciones, además estos paquetes, contienen también el agente que interpreta este lenguaje, de modo que además de servir como editores, también nos sirven de navegadores.

Se han seleccionado los kits de desarrollo mas completos, que contienen, los editores, compiladores, y ejemplos para el programador, y que además sirven de navegadores, estos kits se encuentran en la carpeta de 'Kits de desarrollo'.

Una vez que tenemos ya nuestra página codificada, podemos ver como quedaría si realmente estuviera en un servidor, para ello tenemos los browser o navegadores que vienen incluidos en los 'kits de desarrollo'. Estos navegadores adoptan el aspecto de un teléfono móvil y simulan como quedaría nuestra aplicación si realmente estuviera en un servidor y estuviéramos accediendo desde nuestro teléfono.

Cuando ya esta lista nuestra aplicación WAP, entonces si aún no estaba, se llevará a un servidor, una vez hecho esto, existen dos posibilidades para acceder a la aplicación:

- 1ª Que se acceda desde un teléfono móvil con la tecnología WAP; cada teléfono móvil posee su propia configuración para poder insertar la dirección de la pagina WAP.
- 2ª Cualquier persona que no posea un teléfono WAP, puede acceder desde su PC, gracias a los navegadores o browser que comentábamos en el punto anterior.

El único problema que nos podemos encontrar al desarrollar una aplicación para WAP, es el tema de los servidores, a continuación se han plasmado las diferentes preguntas que pueden surgir y se han dado respuesta a cada una de ellas:

¿Hace falta un servidor especial que soporte las páginas WAP?

En principio, basta con un servidor normal que sirve páginas de Internet, lo único que tendrás que hacer es configurar los MIME Types.

¿Qué son los MIME Types y como se configuran ?

Al acceder a una página web usando un navegador de un PC, el servidor le dice a tu navegador qué clase de documento está sirviendo, función que se realiza usando los caracteres MIME.

Si no configuras los MIME Types característicos del WAP, el servidor no podrá decir al navegador nada sobre el tipo de documento.

MIME type Extensión

Para el código WML text/vnd.wap.wml .wml

Para las imágenes wml (extensión wbmp) image/vnd.wap.wbmp .wbmp

Para el WML Script text/vnd.wap.wmlscript .wmls

Para el WML compilado text/vnd.wap.wmlc .wmlc

Para el WML Script compilado text/vnd.wap.wmlscriptc .wmlsc

1.5 Ejemplo de una Red WAP

Para seguir con el estudio vamos a ver un ejemplo de una red WAP:

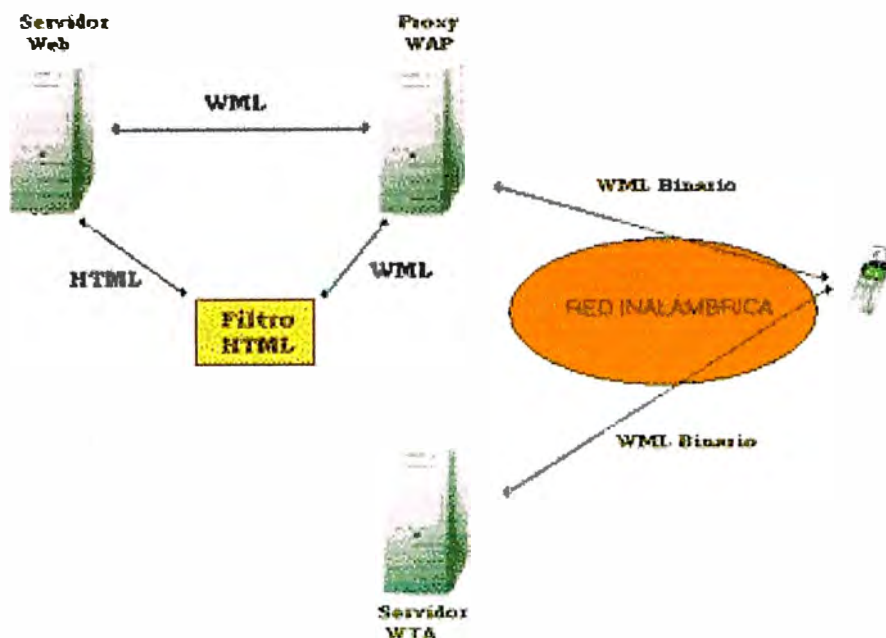


Fig. 1.2: Ejemplo de una red WAP

En este ejemplo, se muestra como el cliente WAP se comunica con 2 servidores a través de la red, con el Proxy WAP y con el Servidor WTA.

El Proxy WAP se encarga de dos labores:

- Traducir la petición WAP, escrita en WML, a una petición WWW, permitiendo así que el cliente WAP pueda realizar peticiones al servidor Web.
- Codificar las respuestas del servidor a un formato binario de modo que sea entendible por el cliente WAP.

El Servidor WTA (Wireless Telephony Application) puede responder las peticiones WAP del cliente directamente; de este modo, permite ofrecer acceso WAP a determinadas características de la infraestructura de comunicaciones del operador de red.

Por otro lado tenemos el Servidor Web, que se comunica con el Proxy WAP de dos posibles modos:

- Si el servidor Web proporciona un contenido WAP, como por ejemplo WML o WMLS, entonces no se necesita ningún filtro HTML.
- Si el servidor Web proporciona un servicio WWW, como HTML, entonces se usa un filtro HTML para traducir el contenido WWW en uno WAP. Como se ve en el ejemplo 1, el filtro HTML, puede encargarse de traducir una respuesta HTML en una WML, y devolvérsela al Proxy WAP.

1.6 WAP, la unión de Internet y el celular

Algunos usuarios dependen tanto de su correo electrónico, información en páginas WWW o servicios relacionados con Internet, que la necesidad de acceso inalámbrico, en cualquier lugar y en cualquier momento, va creciendo constantemente.

La existencia del Internet inalámbrico seguro, eficiente y sin depender de una computadora (para no confundirlo con el acceso a redes locales inalámbricas) está relacionada con la proliferación de la telefonía celular digital y los Servicios de Comunicación Personal (PCS), por lo que se ha desarrollado el Protocolo de Aplicación Inalámbrica, conocido como WAP. Este protocolo está diseñado para funcionar en una diversidad de servicios celulares ya existentes, apoyándose en los siguientes estándares:

- Servicio de Mensajes (SMS).
- Circuito de Datos de Alta Velocidad (CSD).
- Servicio General de Paquetes por Radio (GPRS).
- Servicio Suplementario de Datos no Estructurados (USSD).

WAP emplea un lenguaje específico para presentar la información, llamado WML (Wireless Markup Language) cuyo origen es el XML (eXtensible Markup Language). A diferencia de HTML (Hypertext Markup Language), WML es un metalenguaje, lo que implica que además de usar etiquetas predefinidas se pueden crear componentes propios. En materia de protocolos de comunicación, WAP permite el uso de estándares,

como UDP (User Datagram Protocol) e IP (Internet Protocol), así como el propio XML.

¿Porqué no usar TCP/IP y HTML en los servicios de Internet a celulares, en lugar de fabricar una nueva norma como lo es WAP? Existen tres razones fundamentales:

- 1. La velocidad de transferencia.** La mayoría de los asistentes personales y teléfonos celulares tienen anchos de banda de 14.4 Kbps o menos, lo cual ni siquiera es comparable con la velocidad que ofrecen los módem caseros, los cables módem o las conexiones DSL. Por lo anterior, el contenido para dispositivos inalámbricos en Internet, generalmente sólo ofrece información basada en texto, para evitar el pesado tráfico de archivos gráficos y las limitaciones que algunos modelos de celulares y PDAs (Personal Digital Assistant) tienen para el despliegue de gráficos.
- 2. El tamaño y la legibilidad de la información.** El pequeño tamaño de una pantalla de LCD de los teléfonos celulares o de las PDAs implica otro reto. Muchas páginas WWW están diseñadas con una resolución de 800 x 600 pixeles, lo cual está bien para una computadora de escritorio o una portátil, pero esa misma página no podría visualizarse en una pantalla de teléfono celular, que a lo más llegan a 150 x 150 pixeles, o bien 320 x 240 en los dispositivos portátiles como Pocket PC® y Palm®.
- 3. La forma de navegación en el contenido.** Seleccionar ligas, imágenes o menús en una computadora con la asistencia de un ratón es tarea fácil, pero no trivial, programar esas funciones con las teclas de desplazamiento del teléfono celular.

WAP considera cada una de esas limitaciones y establece la forma de comunicación con un dispositivo inalámbrico convencional, pero eso sí, digital.

Esto es lo que ocurre cuando se accede a un sitio WWW por medio de WAP:

1. Se activa el dispositivo y se ejecuta el navegador de WWW incluido.
2. El teléfono establece una conexión por radiofrecuencia, solicitando el servicio.
3. Se establece una conexión con el proveedor.
4. Se selecciona el sitio WWW que se desea ver.
5. Se envía una solicitud a un servidor especial, llamado gateway, usando WAP.

6. El gateway recupera la información, vía el Protocolo de Hipertexto (HTTP) desde el sitio WWW.
7. El gateway codifica los datos HTTP como WML.
8. Los datos en WML se envían al dispositivo portátil.
9. Se visualiza la versión inalámbrica de la página que se ha seleccionado.

Lo anterior sucede para un servidor WWW que no posee, específicamente, una sección WAP, que se caracteriza por tener pocas o nulas imágenes y presenta menús simplificados. El proceso de conversión emplea demasiado tiempo, por lo que muchos usuarios pueden desesperarse y terminar la conexión. Aun con conexiones y traducciones rápidas, la información no puede ser lo suficientemente clara como para que el usuario navegue libremente entre las opciones de la página. Por ello, el crecimiento de los servicios WAP demanda que los administradores de sitios WWW construyan su propia versión WML, reduciendo las cargas de trabajo e inconsistencias que pudieran surgir en los gateway localizados en las instalaciones del proveedor de telefonía celular.

WAP se estructura en diversas capas, cada una de ellas con funciones muy concretas y delimitadas, lo que optimiza tanto la velocidad como la calidad de acceso.

CAPITULO II

INTRODUCCIÓN AL LENGUAJE WML

2.1 Definición WML

El lenguaje WML (*Wireless Markup Language*) esta basado en el lenguaje XML (*Extensible Markup Language*) y fue desarrollado para especificación de contenidos visualizables desde un terminal WAP (un terminal dotado con un micro-navegador WAP).

El alcance de este dispositivo tiene capacidad limitada, o son diseñados para propósitos especiales. Un teléfono móvil típicamente tiene un teclado con pocas funciones específicas. Dispositivos más sofisticados pueden tener funciones para programar software, pero no tienen por ejemplo un mouse.

Las páginas WML podrán ser ubicadas en cualquier servidor *web* convencional, siempre que éste, haya sido correctamente configurado para servirlas.

El WML está diseñado para trabajar con dispositivos inalámbricos pequeños que poseen cuatro características:

- Pantalla pequeña de baja resolución. Por ejemplo, la mayoría de los teléfonos móviles pueden sólo mostrar unas pocas líneas de texto, y cada una de ellas puede contener solamente de 8 a 12 caracteres.
- Los aparatos de entrada tienen una capacidad limitada, o están diseñados para un propósito determinado. Un teléfono móvil tiene comúnmente teclas numéricas y un reducido número de teclas adicionales con funciones específicas.
- Los recursos computacionales están limitados por un procesador de baja potencia, una memoria reducida y una potencia restringida.
- La red ofrece un reducido ancho de banda y una alta latencia. No son infrecuentes los aparatos con conexiones de red de 300 bit/s a 10 Kbit/s y latencia "*round-trip*" de entre 5 y 10 segundos.

2.2 Características WML

Las características del lenguaje WML pueden agruparse en estas áreas principales:

- El WML ofrece un soporte de texto e imagen y tiene una amplia variedad de formatos y comandos.
- Las cartas WML se agrupan en barajas. Una baraja WML es similar a una página HTML identificada por un URL (*Uniform Resource Locator*, localizador de recursos uniforme) y es la unidad básica de transmisión de contenidos.
- El WML ofrece soporte para gestión de navegación entre cartas y barajas, e incluye comandos para su manejo. Estos pueden usarse para navegar o ejecutar "scripts". El WML también provee de conexiones de anclaje similares a las usadas en el HTML versión 4.
- Se pueden establecer parámetros para todas las barajas de WML usando un modelo establecido. Se pueden usar variables en lugar de cadenas y sustituirse en el tiempo de ejecución. Esta forma de establecer parámetros permite que los recursos de la red sean usados de forma eficiente.
- WML es un lenguaje de marcas similar al HTML. WML es compatible con XML 1.0. Las páginas WML son las llamadas barajas ya que están compuestas por cartas, un navegador WAP, solo puede mostrar una carta al mismo tiempo.

Para poder editar los documentos WML nos sirve cualquier editor de texto plano (como el vi, el notepad, etc). No obstante, existen emuladores de teléfonos WAP que nos permiten editar y visualizar el resultado final de lo que estamos haciendo.

2.3 Definición de carta y baraja

Para empezar a trabajar, en primer lugar debemos conocer el significado de los dos siguientes conceptos, bajo el contexto del WML: *carta* y *baraja*.

Dado que los terminales móviles pueden mostrar muy poca información en la pantalla, para intentar minimizar el tráfico de peticiones de documentos al servidor, lo que se hace es enviar un documento completo en el que hay varias "pantallas".

Cada una de estas pantallas se conoce como *carta*, y el documento completo, al ser una colección de cartas, se conoce como *baraja*.

La siguiente figura ilustra el significado de carta y baraja:

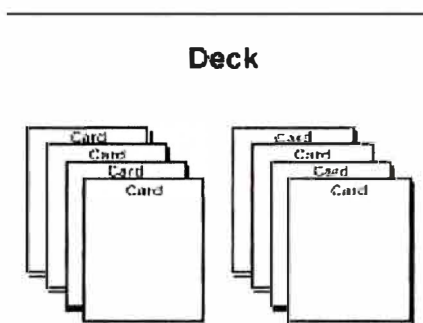


Fig. 2.1. Significado de carta y baraja

Con estas ideas, veamos las primeras etiquetas que debemos conocer para crear documentos WML. En primer lugar, debemos comenzar todos los documentos WML con una cabecera que indique de que se trata, efectivamente, de un documento WML.

Abriremos la baraja escribiendo `<wml>`, y la cerraremos con `</wml>`. No debe ir más texto al cerrar la baraja.

Para escribir texto dentro de una carta, es obligatorio ponerlo dentro de la directiva pareada `<p> ... </p>`, no podemos dejarnos texto fuera.

Vamos a explicar detenidamente un ejemplo:

```
<?xml version="1.0"?>
```

Indica que es un documento XML de versión 1.0 por lo tanto cumple todas las restricciones y reglas de los documentos XML.

```
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
```

```
"http://www.wapforum.org/DTD/wml_1.1.xml">
```

Indica el tipo de documento XML y donde localizar la especificación del tipo de documento.

<wml>

Indica que comienza un página WML.

<card id="carta1" title="Welcome">

Indica que comienza una tarjeta que tiene como identificador "carta1" y cuyo título es "Welcome"

<p>

Indica que comienza un párrafo de texto. A diferencia de HTML esta etiqueta es obligatoria si queremos escribir texto, además debe cerrarse con la correspondiente

</p>

Hello World!

Este es el texto que aparecerá en nuestro navegador.

</p>

Como señalamos anteriormente, con esta etiqueta indicamos que el párrafo ha terminado y no vamos a escribir más texto dentro de este párrafo.

</card>

Fin de la tarjeta.

</wml>

Fin de la página WML

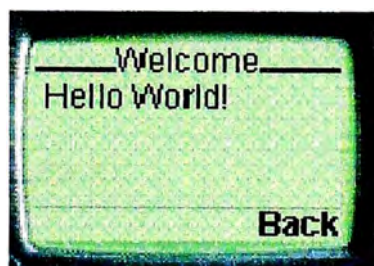


Fig. 2.2. Pantalla que muestra el resultado del ejemplo

2.4 Conjunto de juego de caracteres WML

En WML, un juego de caracteres de documento es el juego o conjunto de todos los caracteres que un tipo de documento puede contener, por ejemplo la letra "T" y un número entero fijo identificando esa letra.

El juego de caracteres de documento para XML y WML es el juego de caracteres universal de ISO/IEC-10646. Actualmente, este sets de caracteres es idéntico al Unicode.

El protocolo WAP soporta los siguientes conjuntos de caracteres del subconjunto Unicode:

- **UTF-8**

El formato de transformación UCS 8 es usado como un codificador que transmite el juego de caracteres internacional. UTF-8 es un codificador de archivos seguro que evita usar bytes que tengan una especial relevancia durante el análisis de cadenas de rutas. El UTF-8 es un codificador de 8-bit de los caracteres de UCS. Algunas de las utilidades de UTF-8 son:

- Es compatible con el código ASCII de 7-bit, de modo que no afecta a programas que dan un significado especial a varios caracteres de ASCII.
- Es inmune a errores de sincronización; sus reglas de codificación permiten una identificación más fácil.
- Posee espacio suficiente para soportar un gran número de juegos de caracteres.

- **ISO-8859-1 o ISO Latin-1**

El juego de caracteres ISO-8859-1 es una extensión del juego de caracteres ASCII y puede usarse para representar todas las lenguas de Europa Occidental.

También conocido como ISOLatin-1, el ISO-8859-1 es muy similar al juego de caracteres ANSI usado en Windows, aunque los dos no son idénticos.

- **UCS-2**

El USC-2 es el codificador 2-byte (de 16 bits) de el UCS (Universal Multiple-Octet Coded Character Set, El juego de caracteres codificados de octeto múltiple universal") definido en ISO 10646. Los valores de código de caracteres de UCS-2 son idénticos a aquellos del modelo de codificación de caracteres Unicode publicado por el Unicord Consortium.

2.5. Etiquetas WML

2.5.1 La etiqueta card

La etiqueta *card* es un contenedor para texto y elementos de entrada, lo suficientemente flexible para permitir la presentación en una amplia variedad de dispositivos, con una gran variedad de características de entrada y de presentación. El elemento card indica la disposición general y los campos de entrada requeridos, mientras da una considerable libertad para la disposición instrumentalizada y los esquemas de entrada del terminal del usuario.

- **Sintaxis**

Los atributos del elemento card se explican en la siguiente tabla:

| Atributo | Explicación |
|---------------------------------|---|
| <code>title=vdata</code> | Este atributo especifica información adicional sobre la carta. |
| <code>newcontext=boolean</code> | Si se coloca este atributo como true (verdadero), el estado actual del browser es |

reinicializado cuando se ejecute esta carta, realizando las operaciones siguientes:

- Borra el estado del historial de navegación
- Resetea el estado específico de implementación a un valor conocido.

`ordered=boolean`

Este atributo da una indicación al terminal de sobre como se organiza el contenido card.

ordered="true" La carta se organiza como una secuencia lineal de campos.

ordered="false" Esta carta es una colección de elementos de campo sin un orden natural.

`onenterforward=href`

El suceso *onenterforward* ocurre cuando el usuario navega dentro de una carta usando una tarea *go*.

`onenterbackward=href`

El suceso *onenterbackward* ocurre cuando el usuario navega en una carta usando una tarea *prev*.

`ontimer=href`

El suceso *ontimer* ocurre cuando un timer (contador de tiempo) expira.

Ejemplo

Lo que aparece a continuación es un ejemplo de una etiqueta card encerrado en una baraja de WML. Esta carta contiene texto, el cual se muestra en la pantalla del terminal del usuario.

```
<wml>
<template>
<do type="accept" name="exit" label="EXIT">
<prev/>
</do>
```

```

</template>
<card id="card 1" title="Welcome" newcontext="true">
<p>
Hello World!
</p>
</card>
</wml>

```

Esta baraja genera el siguiente interfaz con el usuario en su terminal:



Fig. 2.3. Navegación de una card dentro de una baraja.

2.5.2 La etiqueta *template*

La etiqueta *template* (patrón) declara un patrón para las cartas de la baraja. Las acciones especificadas en la etiqueta *template* (por ejemplo, *do* o *onevent*) se aplican a todas las cartas de la baraja. Especificar un suceso en la etiqueta *template* es equivalente a especificarlo en cada carta. De todos modos, un elemento *card* puede supeditar las acciones especificadas en la etiqueta *template*.

- Sintaxis

Los atributos de la etiqueta *template* se explican en la siguiente tabla:

| Atributo | Explicación |
|-----------------------------------|--|
| <code>onenterforward=href</code> | Especifica un suceso intrínseco que instruye al terminal del usuario a ir al URL especificado cuando el usuario navega hacia delante a esta carta. |
| <code>onenterbackward=href</code> | Especifica un suceso intrínseco que instruye al terminal a ir al URL especificado cuando el usuario navega hacia atrás a esta carta, por ejemplo, usando una tarea <code>prev</code> . |
| <code>ontimer=href</code> | Especifica un suceso intrínseco que instruye al terminal a ir al URL especificado después de que el timer (contador de tiempo) ha expirado. |

Ejemplo

El siguiente elemento `template` incluye un elemento `do` indicando la navegación a la carta anterior de la baraja.

```
<template>
<do type="prev" label="Previous">
<prev/>
</do>
</template>
```

2.5.3 La etiqueta *access*

La etiqueta *access* especifica la información del control de acceso para la baraja entera. Hay que advertir que es un error sintáctico en WML el que una baraja contenga más de un elemento *access*. Si una baraja no incluye un elemento *access*, el control de acceso está inhabilitado. Cuando el control de acceso está inhabilitado, las cartas de cualquier baraja pueden acceder a esta baraja.

- Sintaxis

Los atributos del elemento `access` se explican en la siguiente tabla:

| Atributo | Explicación |
|--------------|---|
| domain=cdata | Los atributos <i>domain</i> y <i>path</i> de una baraja especifican las otras barajas que pueden acceder a ella. Mientras el terminal del usuario navega de una baraja a otra, se realizan comprobaciones (chequeos) del control de acceso para determinar si la baraja destino permite el acceso desde la baraja actual. |

Ejemplo

Por ejemplo, dados los atributos de control de acceso siguientes:

```
domain="acmecorp.com"
path="/pub"
```

Las siguientes referencias a URLs deberían poder acceder a la baraja:

```
acmecorp.com/pub/stocks.cgi
www.acmecorp.com/pub/demos/packages.cgi
```

2.5.4 La etiqueta `do`

La etiqueta `do` da al usuario un mecanismo general para realizar acciones en la carta actual, esto es, un elemento de interfaz con el usuario a nivel de carta. El elemento `do` está mapeado a un único interfaz de usuario *widget*, el cual el usuario puede activar. Por ejemplo, el mapeado *widget* puede ser para un botón restaurado (formateado y presentado) gráficamente, para una tecla blanda o una tecla de función, una secuencia de comando activada por la voz

o cualquier otro interfaz que tenga una operación "de activar" sencilla con ningún estado persistente de íter operación.

- **Sintaxis**

Los atributos de la etiqueta *do* se explican en la siguiente tabla:

| Atributo | Explicación |
|-------------------------------|---|
| <code>type=cdata</code> | El tipo del elemento <i>do</i> . Este atributo proporciona un indicación al terminal del usuario sobre como el autor de WML propone que el elemento sea usado, y como debería ser mapeado a una construcción de interfaz de usuario física. |
| <code>label=vdata</code> | Este atributo especifica una cadena de texto para rotular el interfaz de usuario <i>widget</i> . Si un elemento no puede ser rotulado de forma dinámica, este atributo se ignora. |
| <code>name=nmtoken</code> | Este atributo especifica el nombre del enlace del suceso <i>do</i> . |
| <code>optional=boolean</code> | Si se coloca este atributo en verdadero (<i>true</i>), el terminal puede ignorar este elemento. Por defecto su valor es falso (<i>false</i>). |

Ejemplo

Este ejemplo muestra un elemento *DO* que incluye una tarea *go*. Cuando el usuario activa el elemento seleccionando *Next*, el terminal va a la carta 2 de la baraja actual y la muestra al usuario.

```

<card id="card1">
  <do type="accept" label="Next">
    <go href="#card2"/>
  </do>
  <p>
    Select <b>Next </b> to go to the next card.
  </p>
</card>

<card id="card2">
  <p>
    This is card 2.
  </p>
</card>

```

2.5.5 La etiqueta go

El elemento *go* determina una tarea *go* (“ir”), indicando la navegación a un URL. Si el URL nombra a una baraja o a una carta de WML, ésta es mostrada. Una tarea *go* ejecuta una operación de empuje sobre la pila del historial del *browser* (navegador).

- Sintaxis

Los atributos del elemento *go* se explican en la siguiente tabla:

| Atributo | Explicación |
|---------------------|--|
| href=href | Este atributo especifica el URI de destino, por ejemplo el URI de la carta a mostrar. |
| sendreferer=boolean | Si se coloca el atributo en <i>true</i> (verdadero), el terminal incluye el URL de la baraja que contiene esta tarea en el requerimiento de URI. |

Esto permite al servidor realizar una forma de control de acceso en los URIs,

`method=(post | get)`

Actualmente, los valores de *get* y *post* son aceptados y provocan que el terminal del usuario realice un *get* o un *post* de HTTP respectivamente.

`accept-charset=CDATA`

Este atributo especifica el listado de codificaciones de caracteres de datos que el servidor web debe aceptar cuando se realiza el procesado.

Ejemplo

En el siguiente ejemplo, el elemento *go* produciría una petición HTTP GET al URL `"/foo? x=1"`:

```
<go href="/foo">
<postfield name="x" value="1"/>
</go>
```

2.5.6 La etiqueta `img`

La etiqueta `img` instruye al dispositivo para mostrar una imagen, se debe tener en cuenta que no todos los dispositivos pueden mostrar imágenes.

- Sintaxis

Los atributos de esta etiqueta son:

| Atributo | Explicación |
|----------------|---|
| src | Lugar donde se encuentra la imagen. |
| alt | Texto alternativo. |
| Vspace /hspace | Espacio en blanco entre la imagen y el resto de contenidos. |
| Align | Alineación del texto con respecto a la imagen |
| height /width | Alto y ancho de la imagen (por este orden) |

Ejemplo

```

<wml>
<card>
<p>
This is a smileyface:
<br/>
</p>
</card>
</wml>

```

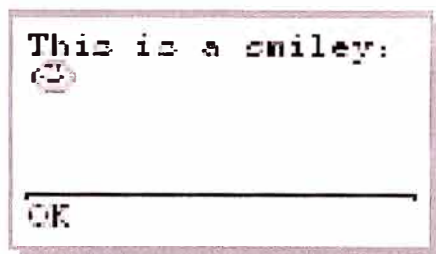


Fig. 2.4. Acción de la etiqueta img

2.5.7 La etiqueta input

El elemento *input* (entrada) especifica un objeto de entrada de texto. Se puede especificar el formato de la entrada del usuario con atributos *format* (formato) opcionales. El agente usuario no debe inicializar el objeto de entrada con ningún valor que no esté conforme con la máscara de entrada ligada.

- Sintaxis

Los atributos del elemento input se explican en la siguiente tabla.

| Atributo | Explicación |
|-------------------------|---|
| name=nmtoden | El atributo <i>name</i> especifica el nombre de la variable a colocar con el resultado de la entrada de texto del usuario. |
| value=vdata | El atributo <i>value</i> indica el valor por defecto de la variable nombrada en el atributo <i>name</i> . |
| type= (text password) | Este atributo especifica el tipo de área de entrada de texto. |
| text | Un control de entrada de texto. |
| password | Un control de entrada de texto. La entrada de cada carácter es repetida en una forma "oculta", de una manera apropiada al terminal del usuario. |
| format=cdata | Este atributo especifica una máscara para una entrada de usuario. La cadena consiste en los |

caracteres de control de la máscara y el texto estático que es mostrado en el área de entrada.

`emptyok=boolean`

Si se coloca este atributo como verdadero (*true*) el elemento *input* acepta la entrada aunque una cadena de formato que no esté vacía haya sido especificada.

`size=number`

Este atributo especifica la anchura, en caracteres, del área de entrada de texto.

`maxlength=number`

Este atributo especifica el número máximo de caracteres que el usuario puede entrar en el área de entrada de texto. El valor por defecto de este atributo es un número ilimitado de caracteres.

`title=vdata`

Este atributo especifica un título para el elemento *input*. Es título puede utilizarse en la presentación del objeto.

`tabindex=number`

El elemento *tabindex* especifica la posición de "tabbing" (de tabulación) del elemento actual. Esta posición indica el orden relativo en el cual los elementos son recorridos cuando se pasa por ellos dentro de una carta WML sencilla. Un valor *tabindex* numéricamente mayor indica un elemento que está después en la secuencia *tab* de un elemento con un valor menor.

Ejemplo

Especifica un elemento *input* que acepta cualquier carácter y muestra la entrada al usuario de una forma que el usuario puede leer. El número máximo de caracteres que pueden ser introducidos es 32, y la entrada resultante es asignada a la variable denominada X.

```
<input name="X" type="text" maxlength="32"/>
```

2.5.8 La etiqueta option

El elemento *option* especifica una opción de elección sencilla en un elemento *select*.

- Sintaxis

Los atributos del elemento *option* se explican en la siguiente tabla:

| Atributo | Explicación |
|-------------|--|
| value=vdata | Este atributo especifica el valor a usar cuando está fijada la variable <i>key</i> . Cuando el usuario selecciona esta opción, el valor resultante especificado en el atributo <i>value</i> se utiliza para colocar la variable <i>name</i> del elemento <i>select</i> . |
| title=vdata | Este atributo especifica un título para el elemento <i>option</i> , el cual puede usarse en la presentación de este objeto. |
| onpick=href | El suceso <i>onpick</i> se produce cuando el usuario selecciona o deshace la selección de esta opción. |

Ejemplo

El siguiente ejemplo especifica una lista sencilla de elección simple. Si el usuario eligiera la opción "*dog*", la variable "*X*" tomaría el valor "*D*".

2.5.9 La etiqueta `select`

El elemento `select` permite al usuario escoger de una lista de opciones. Cada opción está especificada por un elemento `option` que tiene una línea de texto formateado. Se puede organizar los elementos `option` dentro de grupos jerarquizados usando el elemento `optgroup`.

- Sintaxis

Los atributos del elemento `select` se explican en la siguiente tabla:

| Atributo | Explicación |
|-------------------------------|---|
| <code>multiple=boolean</code> | Si se coloca este atributo con valor <i>true</i> , la lista de selección acepta múltiples selecciones. El valor por defecto es <i>false</i> . |
| <code>name=nmtoken</code> | El atributo <i>name</i> indica el nombre de la variable que toma el valor del elemento elegido. |
| <code>value=vdata</code> | El atributo <i>value</i> indica el valor por defecto de la variable especificada por el atributo <i>name</i> . |
| <code>iname=nmtoken</code> | El atributo <i>iname</i> indica el nombre de una variable que contiene un número de índice. El terminal del usuario utiliza el número de índice para colocar la opción por defecto. El número 1 especifica el primer artículo, el número 2 el segundo, y así sucesivamente. |



Fig. 2.5. Una carta con una lista de elección múltiple.

2.5.10 La etiqueta *prev*

El elemento *prev* (de ‘previous’, previo, anterior) indica la navegación al URI anterior de la pila del historial. Una tarea *prev* produce una operación *pop* (de salto) en la pila del historial y elimina el URI actual de la pila. Si no hay URI anteriores en la pila del historial, el elemento *prev* no tiene efecto.

- Sintaxis

Los atributos del elemento *prev* se explican en la siguiente tabla:

id

class

Ejemplo

El siguiente ejemplo crea un *widget* de interfaz de usuario con el rotulo “Back” y va a la carta anterior cuando se activa.

```

<do type="accept" label="Back">
<prev/>
</do>
<p>
Hello, World!
</p>

```

2.5.11 El elemento wml

El elemento *wml* define una baraja y encierra toda la información y las cartas de la baraja.

- Sintaxis

El atributo *xml:lang* del elemento *wml* se explica en la siguiente tabla:

| Atributo | Explicación |
|--------------------|---|
| xml : lang=nmtoken | Este atributo especifica el lenguaje natural o formal en el cual el documento está escrito. |

Ejemplo

A continuación se muestra una baraja que contiene dos cartas, cada una representada por la etiqueta *card*. Después de cargar la baraja, el terminal del usuario muestra la primera carta. Si el usuario activa el elemento *do*, el terminal muestra la segunda carta.

```

<wml xml:lang="en-us">
<card id="card1" title="Card 1">
    <do type="accept">
    <go href="#card2"/>
    </do>

```



```

<p>
Hello world!
This is the first card...
</p>
</card>
<card id="card2" title="Card 2">
<p>
This is the second card.
Goodbye.
</p>
</card>
</wml>

```

La baraja genera la siguiente interfaz de usuario en el terminal:

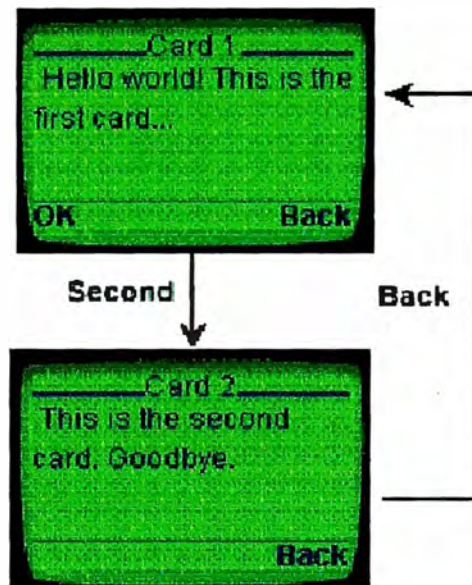


Fig. 2.6. Una baraja WML que contiene dos cartas.

CAPITULO III

DESCRIPCION DEL PROYECTO Y DE SUS COMPONENTES

3.1 Descripción del Proyecto de Aplicación Wap vía Nextel para Luz del Sur.

Tecsur es la empresa encargada de la supervisión de las obras de Luz del Sur las cuales se ejecutan con contratistas. Para el control y supervisión de estas obras por parte de Tecsur fue implementado un sistema informático, este sistema fue desarrollado por Luz del Sur, sistema llamado Sistema de Proyectos y Obras (SPO), el cual permite el control de las órdenes de trabajo (OTs).

Tiene instalado un servidor en Luz del Sur y otro en Tecsur. La información que se maneja con este sistema es bastante amplia contiene mas de 160 campos, desde los números de Ots, datos del cliente, descripción de la obra, fechas, calificación de obra, montos de proyectado en materiales y mano de obra así como montos de liquidación de obras, montos de facturación, etc. Todos estos datos se ingresa paso a paso desde la generación de la OT pasando por la planificación, ejecución, liquidación y facturación.

Luz del Sur tenía algunos problemas con el cliente por información no actualizada, vamos a comentar algunos de ellos:

- Fecha de inicio programada para una obra, por ejemplo en el sistema figuraba el inicio de la obra para el 15/02/2006, al inicio de una obra primero se hace una pre inspección; esto para el cliente significa que todavía no se inicia la obra, generándose una incomodidad en el cliente quien llamaba presentando su queja a lo que las operadoras entraban al sistema, y lo único que podían ver era la fecha de inicio programado y no el inicio real aún habiéndose ya comenzado la obra.

Este tipo de problema afectaba en algunas partes del desarrollo de la obra ya sea en la planificación, ejecución, liquidación y facturación de la OT.

- Así como se tenía problemas con fechas de inicio real de una obra, se tenía problemas en ver la manera como se estaba supervisando, la presencia del técnico en la obra. No se tenía una herramienta que pudiera medir directamente el tiempo de la supervisión, la administración de Luz del Sur y Tecsur necesitaba saber como estaba siendo supervisada una obra, esta información se necesitaba tener en línea, si el cliente llamaba quejándose durante la ejecución de la obra, la respuesta por parte de las operadoras de Luz del Sur estaba en función a los datos que encontraban en el sistema y aquellos datos no estaban actualizados.
- Otro problema que teníamos era no poder dar por terminada una obra en el sistema, esto consta de dos partes uno de ellos es poner la fecha de instalación del medidor y el otro es poner la fecha de término, la primera tarea lo hace el contratista (instalador del medidor) la segunda tarea lo hace el supervisor Tecsur, el encargado de instalar el medidor debía de llegar a la base para ingresar los datos de instalación del medidor y eso se daba al día siguiente o en algunos casos después de dos días si eran sábados y domingos. Después de instalar el medidor el cliente ya contaba con energía eléctrica en este momento ya debíamos de finalizar la orden de trabajo y para esto también teníamos que ir a una PC para ingresar los datos del fin de la obra, el supervisor según su recorrido de supervisión muchas veces antes de regresar a la base pasa por inspeccionar varias obras luego de ello recién en su PC coloca la fecha de término.

La supervisión de Tecsur necesitaba información en el sistema de manera actualizada es decir información en línea. Debido a estos inconvenientes la supervisión se veía involucrada y el problema era cada vez difícil. No teníamos información en línea, es decir con información actualizada, nos referimos sobre todo a datos de fechas, una obra tiene varias fechas entre ellas podemos mencionar las de inicio y fin programado, plazo ley (numero de días de plazo dado por OSINERG para la atención de una obra), fecha de inicio real (a veces es mayor a la fecha de inicio programado),

fecha de instalación del suministro (medidor), fecha de fin de obra (que no es necesariamente cuando instalan el medidor).

Por todo esto se buscó una manera de encontrar una solución, en este punto es cuando se presenta el SISTEMA WAP, sistema bastante útil, funcional y versátil, esta herramienta informática que nos permite el ingreso de fechas al sistema en línea que funciona interactuando con el servidor de Luz del Sur con una tabla intermedia nos da la facilidad de tener datos en línea.

El desarrollo del SISTEMA WAP estuvo a cargo de Nextel quién a su vez en coordinación con LDS y Tecsur lograron poner en producción los diversos aplicativos, uno de los mayores beneficios que logramos con estos desarrollos fue la atención efectiva en respuestas a los clientes por los reclamos con respecto a inicios y fin de obra, asimismo el control de las ordenes de trabajo nos permite hacer proyectos con tiempos mas reales, los supervisores muchas veces tenían que ir a la empresa los días sábados y domingos solo por ingresar fechas, lo cual actualmente lo hacen desde cualquier lugar.

3.2 Diagrama de Comunicación

La arquitectura del Sistema de Comunicación se inicia en el equipo Nextel, equipo desde el cual se envía la información tal como datos del cliente, descripción de la obra, fechas, calificación de obra, montos de proyectado en materiales y mano de obra, así como montos de liquidación de obras, montos de facturación, etc, dicha información se transmite por la Red de Nextel teniendo como ruta la nube de Internet, llegando a los servidores de Luz del Sur y posteriormente a los servidores de Tecsur, los cuales utilizan como plataforma Linux y base de datos Informix.

Una vez que la información es enviada y actualizada en el Sistema, las distintas áreas de Luz del Sur tienen acceso al sistema tanto para las tareas de atención de reclamos, verificación de avance de la obra, autorización para efectuar la liquidación, etc.

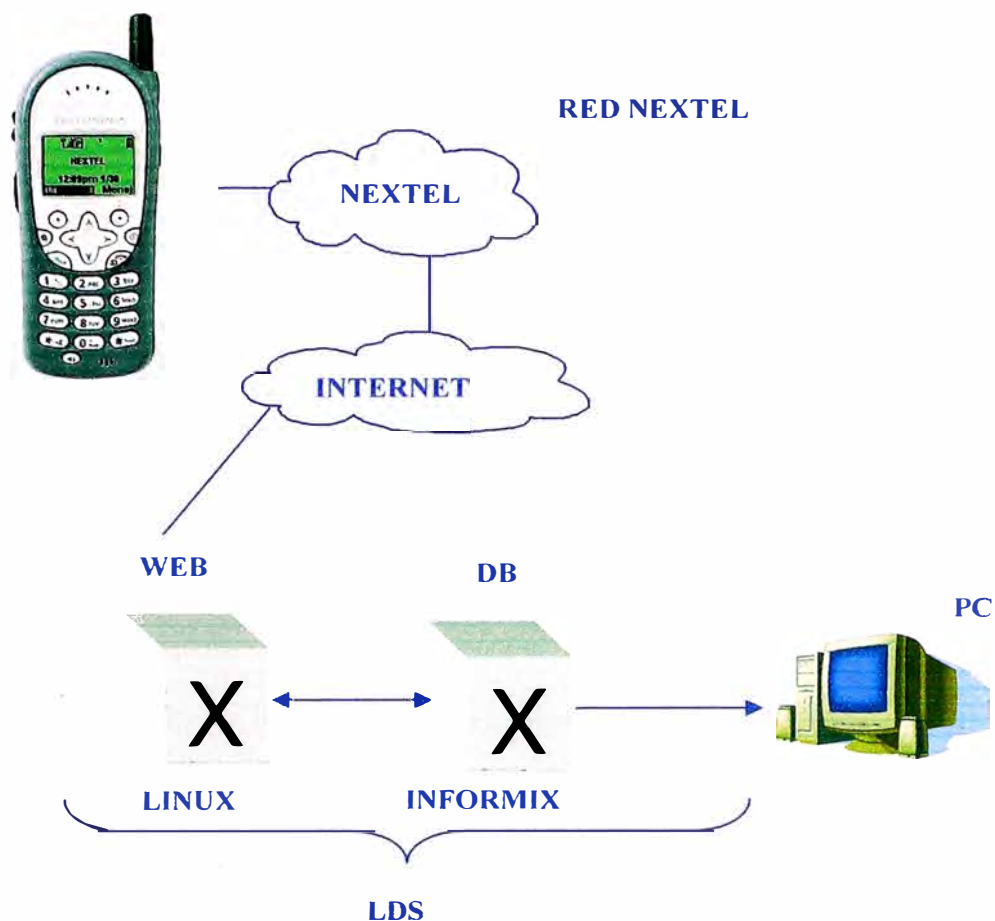


Fig. 3.1. Diagrama de Recorrido de la Información

3.3 Definición de Nextel

Nextel utiliza la tecnología iDEN (Integrated Digital Enhanced Network), red digital de servicios integrados de radiocomunicación desarrollada por Motorola. La misma tecnología que es usada por las compañías de comunicación más prestigiadas alrededor del mundo.

La compañía tiene la concesión para ofrecer el mejor servicio de trunking. Tecnología que integra varias opciones de comunicación en el mismo equipo:

- Conexión Directa.
- Acceso a la Red Telefónica.
- Mensajes.
- Nextel Online.

3.4 Definición de iDEN

Esta tecnología, desarrollada por la compañía Motorola, es conocida como (**iDEN**) **I**ntegrated **D**ispatch **E**nhanced **N**etwork. Actualmente ésta es utilizada por millones de usuarios en todo el mundo. El servicio en los Estados Unidos es brindado por compañías como **Nextel**.

A diferencia de sistemas celulares, que contemplan brindar el servicio de Walkie-Talkie, el sistema **iDEN** está diseñado para ser utilizado por el sector comercial, industrial y gubernamental. Conveniente para usuarios que requieren comunicación rápida, privada y confiable.

Entre las ventajas que provee la tecnología iDEN se encuentran:

- Integración del servicio digital de Walkie-Talkie y teléfono celular.
- Con el servicio (*I-2-1*) puede comunicarse de forma privada con otra unidad.
- El sistema **iDEN** provee el servicio (**Multi-Talk**) comunicación en simultánea con múltiples unidades.
- Rapidez en conexión. La conexión típica de unidades **iDEN** es de ½ segundo.
- Equipos diseñados con especificaciones industriales. Unidades intrinsically safe disponibles.
- Y el servicio al cliente personalizado

3.5 Servidores Utilizados por Luz del Sur

Como se comento en el punto 3.1 en Luz del Sur se utiliza una variedad de servidores de diferente tipo de la marca IBM lo mismo que Tecsur, para nuestro caso los servidores que se utilizan son los siguientes:

- **Sistema IBM eServer OpenPower 710**

El OpenPower 710 expande las ofertas de servidores de IBM diseñados y ajustados para los ambientes de computación en **Linux** utilizando tecnologías líderes inspiradas en mainframe. El sistema OpenPower 710 ofrece el

procesador de 64-bit más avanzado de IBM, el POWER5™ y la flexibilidad de las tecnologías de sistemas opcionales del IBM Virtualization Engine™, como micro-partición e I/O Virtual incluido en un paquete de rack de 2U montable.

El sistema IBM OpenPower 710 ofrece hasta dos procesadores POWER5 de 64-bit siendo ejecutados a 1.65GHz, la capacidad de configurar 20 particiones dinámicas lógicas a través de la micro-partición opcional, capacidad máxima de memoria de 32GB y más de 0.5 TB de almacenamiento interno en disco. Combinando esta posibilidad de expansión con la capacidad opcional del I/O Virtual de compartir unidades de disco y adaptadores de comunicaciones, los servidores OpenPower™ ofrecen la potencia de computación Linux que usted necesita ahora y la flexibilidad de adaptarse rápidamente y de forma eficaz en cuanto a costo para que su negocio prospere en el mundo.

Funcionalidad avanzada, menor tamaño y precio más bajo, llevan mayor flexibilidad y conveniencia al mercado de Linux

Al proporcionar recursos de clase corporativa como Virtualización OpenPower™ Avanzada y micro-partición al espacio de servidores de nivel de partida, los sistemas OpenPower™ pueden ayudar a mejorar la utilización del sistema, reducir el tiempo parado y rebajar los costos de administración de los sistemas.



Fig. 3.2. Servidor Open Power 710

- **Sistema IBM eServer iSeries p5 590**

La gran ventaja de consolidar sus servidores Linux en iSeries es que ésto los torna mucho más sencillos de administrar y más eficaces en cuanto a costo. Hasta 31 ambientes separados de **Linux** son soportados en un servidor iSeries.

Ventajas clave

IBM tiene el compromiso de apoyar su elección de plataforma y sistemas operativos un compromiso que hemos ampliado para incluir Linux, el sistema operativo de código abierto.

- Partición lógica dinámica
- Capacidad on demand
- Virtualización de recursos
- Integración de OS/400

IBM eServer p5 590

El servidor IBM eServer p5 590 utiliza procesadores de quinta generación **IBM POWER5™ de 64-bit** en configuraciones de SMP (symmetric multiprocessing) de hasta 32-vías, para proporcionar la **potencia, flexibilidad, escalabilidad y seguridad** necesarias para aplicativos de misión crítica a un precio muy accesible.

| | |
|---|---|
| Procesador | POWER5 8-vías a 32-vías |
| Ritmo de reloj (Min/Máx) | 1,65GHz / 1 TB |
| Memoria del sistema (Estándar/Máx) | 8GB / 512GB |
| Almacenamiento interno(Estándar/Máx) | 72,8GB / 18,7TB (con cajones I/O opcionales) |
| Rendimiento (amplitud de rPerf)* | 41,68 a 151,72 |



Fig. 3.3. Servidor eServer p5 590

3.6 Definición de Linux

El programa WML utiliza por plataforma el programa SUSE de Linux. Linux es un Sistema Operativo, una versión de Unix libremente distribuible e independiente.

Es una implantación de la especificación POSIX con la cual cumplen todas las verdaderas versiones de Unix. La mayoría de los programas disponible para Linux es desarrollado por el proyecto GNU de la Free Software Foundation (Fundación de Programas Libres).

SUSE LINUX Enterprise Server 9 es la primera descarga del kernel Linux 2.6 para el POWER processor, SLES 9 es diseñado para proporcionar una mejor partición y capacidades virtuales, una alta escalabilidad y una excelente fiabilidad, en algunos casos mejora la acción global comparado con el funcionamiento de los sistemas del kernel Linux 2.4.

Per-system precisada por el sistema SLES 9 en conjunto con IBM POWER processor lleva la virtualización a un nivel totalmente nuevo y permite que los administradores hagan la sintonía fina de las asignaciones de recursos del sistema para atender los requisitos cambiantes del negocio.

Además con el SLES 9 on POWER processor muestra los sistemas basado en Linux, las compañías pueden usar la capacidad de micro-partición del procesador POWER5 para crear 10 particiones por procesador y usar un única licencia SLES 9 por los 8 procesadores además ellos le permite hacer correr un alto numero de servidores virtuales sin ningún costo adicional

3.7 Definición de Informix

La base de datos esta dada en el programa Informix Servidor Dinámico (IDS) versión 10. Informix es un Gestor de base de datos creado por Informix Software Inc.

Informix fue adquirido por la empresa IBM, aquí es el por qué IBM Informix y Linux hacen una combinación poderosa:

- **La flexibilidad.** Nuestro compromiso para abrir normas combinadas con la aplicación de Linux y software de IBM por los medios de las plataformas heterogéneos usted puede escoger el mejor hardware y software para cada tarea, y fácilmente integrar las nuevas aplicaciones con los sistemas existentes.
- **La fiabilidad.** IBM Informix y Linux ya han demostrado su escalabilidad y disponibilidad en los ambientes de los clientes reales. Juntos ellos pueden permitir una solución total, ser más robusto, mientras la productividad va en crecimiento y la responsabilidad del cliente.

3.7.1 Informix® el Servidor Dinámico (IDS)

IBM Informix® el Servidor Dinámico (IDS) nos proporciona las siguientes características:

Confiabilidad

- Actualización sin problemas
- Reconstrucción de índices en línea
- Backup y Recuperación de datos flexible

Rendimiento

- Mayor velocidad de acceso que antes 5% mas que 9.4.
- Usa las capacidades del kernel Linux 2.6, i/o async

Escalabilidad

- Tamaños de pagina configurable
- Disponible desde Intel, hasta Mainframe
- Autenticacion PAM

Seguridad

- Encriptación a nivel de columnas
- Permite OpenSSL para cifrar datos cliente/servidor

Fácil admón.

- Instalación muy rápida
- Nuevo modo monousuario para administración

Desarrollo de software hacia web/escritorio

- Nuevo utilitario para convertir de Informix 4GL hacia EGL
- Disponible con Rational Developer V6

Alta Disponibilidad: Capacidad de recuperación ante fallas

- La replicación HDR se hace a nivel de base de datos.
- Si falla un nodo o el almacenamiento externo el otro nodo toma el control.
- Solamente en IDS Enterprise y WE

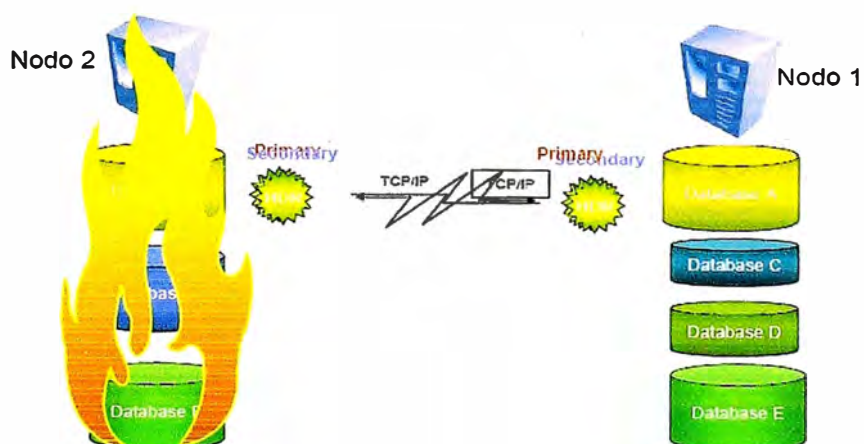


Fig. 3.4. Proceso de Almacenamiento

3.7.2 Plataformas Soportadas

| IDS 10.0 | | | | | | |
|------------------------|---|----------------------|----------------------------|----------------------|-------------------------|--|
| Product versión | Platform | Distributions | Base Kernel versión | Glibc versión | Compiler versión | Comments |
| IDS 10.00.UC4 | Intel x86 | SUSE SLES 9 | 2.6.5 | 2.3.3 | 3.3.3 | Install glibc-devel-2.3.3-98.28 or newer (required by ISM) |
| IDS 10.00.FC4 | x86_64 (EM64T/AMD64) | SUSE SLES 9 | 2.6.5 | 2.3.3 | 3.3.3 | Install glibc-devel-2.3.3-98.28 or newer (required by ISM) |
| IDS 10.00.FC1 | IBM zSeries | SUSE SLES 9 | 2.6.5 | 2.3.3 | 3.3.3 | Install glibc-devel-2.3.3-98.28 or newer (required by ISM) |
| IDS 10.00.FC1 | POWER (pSeries, iSeries, OpenPower, JS20 Blades) | SUSE SLES 9 | 2.6.5 | 2.3.3 | 3.3.3 | Install glibc-devel-2.3.3-98.28 or newer (required by ISM) |

Fig. 3.5. Plataformas del IDS 10.00

3.7.3 Requerimientos Mínimos

| | IBM Informix Dynamic Server 10 | Oracle 10g |
|------------|---|-----------------------|
| RAM | 256 MB | 750 MB |
| Tamaño | 80 MB | 600 MB |
| Disco duro | 500 MB* | 1000 MB |
| Overhead | 22 Bytes/Page | 53 Bytes/Page |

Fig. 3.6. Tabla de Requerimientos Mínimos

3.7.4 Integración con otros productos

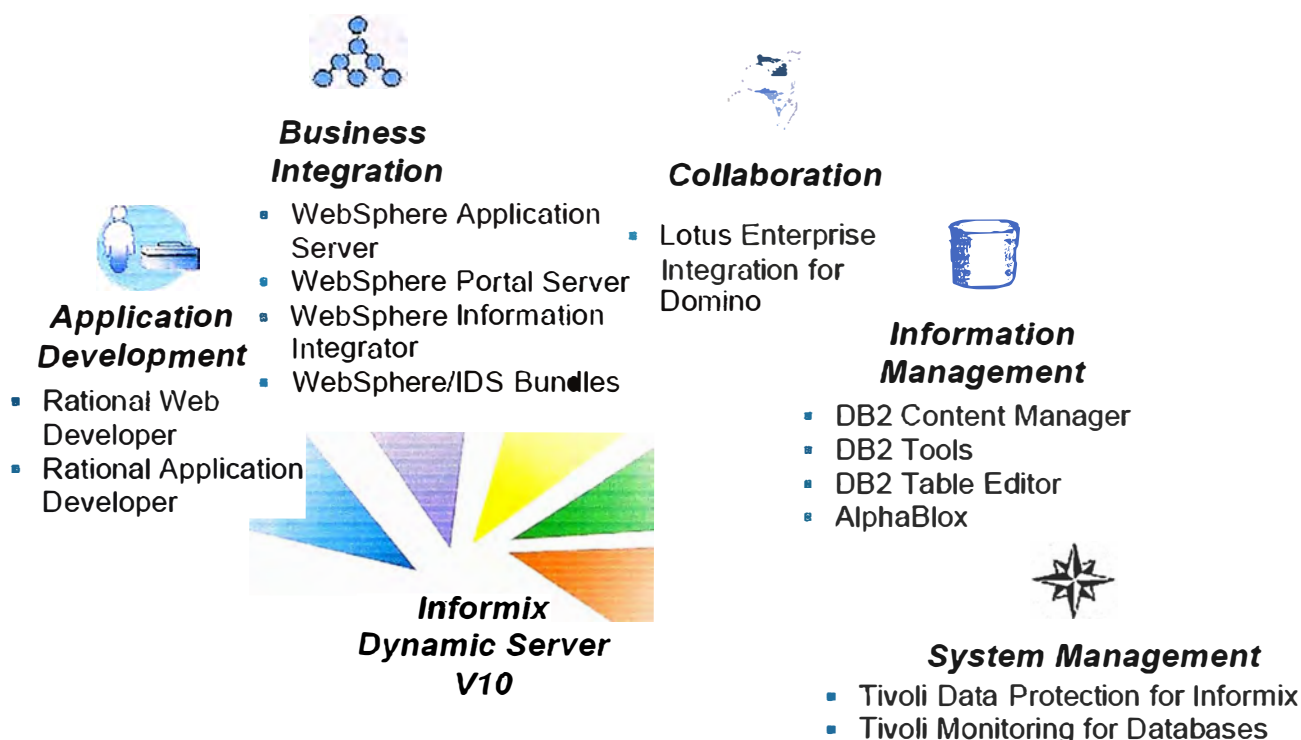


Fig. 3.7. Integración de Informix con otros Productos

3.8 Diferencias de Informix con ORACLE

Oracle siempre ha sido considerada una base de datos para uso más general que Informix. Informix por su lado, se especializó más en aplicaciones tipo GIS (datos geográficos), Datawarehouse y Datamining. Sin duda a los gurús, les agrada más Informix que Oracle.

En cuanto a precios, Informix tiene tendencia a ser más caro que Oracle en configuraciones parecidas.

En la practica, como buenos enemigos acérrimos, ambas tienen parecidas características y funcionalidades. Cada una tiene las típicas ampliaciones que permiten especializar la base de datos a un cierto tipo de aplicación (en Informix se llaman DataBlades. Para competir, en cada nueva versión que sacaban, las dos iban añadiendo de serie muchas de las extensiones que en la versión anterior eran

opcionales (de pago por separado). En el momento de escribir este artículo las dos están en la versión 9.X.

De cara a una instalación y configuración básicas, las diferencias frente a Oracle que el lector va a notar son dos:

En Oracle, tienes que definir los usuarios dentro la base de datos (gestión interna de usuarios). En cambio, Informix utiliza los mismos usuarios de Linux (los que creas con adduser), simplificando la administración. Para conectar externamente vía TCP/IP a Oracle, hacía falta colgar un daemon llamado listener de un puerto. Luego el listener traducía las llamadas al SQLNet para hablar con la base de datos. Luego había dos procesos: el daemon de la base de datos (oracle), y el daemon que escuchaba el puerto (listener). En Informix, el mismo daemon de la base de datos (ONINIT) atiende los puertos. De esta manera Informix ocupa menos memoria y recursos. Con estas diferencias ya podemos ver, que Informix es una base de datos más moderna y integrada con Linux que Oracle.

CAPITULO IV

EVOLUCION DEL PROCESO DE EJECUCION DE OBRAS POR CONTRATISTAS

4.1. Introducción

Lo que se intenta conseguir en este capitulo es describir como se realiza el proceso de la ejecución de una obra por parte de Luz del Sur, realizando una comparación entre la manera como se realizaba anteriormente y como se realiza actualmente, tratando de demostrar con esta comparación los beneficios que nos dará el ejecutar este proyecto

4.2. Proceso de ejecución de obras: Anterior de la Obra

- Se le solicitaba a Luz del Sur la realización o ejecución de una obra para la instalación de un nuevo servicio de alumbrado, una nueva instalación eléctrica o cambio de líneas eléctricas por tiempo de uso.

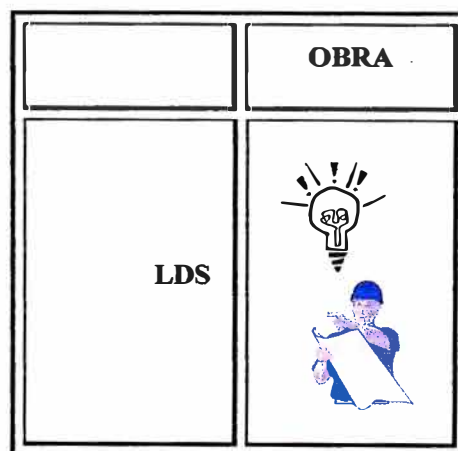


Fig. 4.1. Evaluación de Ejecución de Obras

- Una vez determinada el tipo de obra a realizar se ingresa en el sistema en el cual se designa el costo de la ejecución de la obra, la fecha de inicio, la fecha de término, el área encargada de la ejecución de la obra.

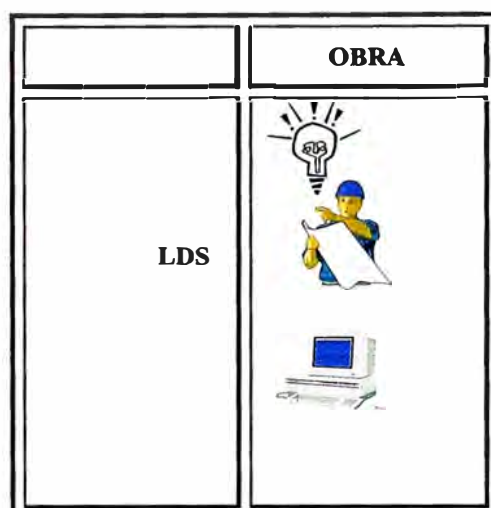


Fig. 4.2. Ingreso de Ejecución de Obras al sistema

Se le informaba al cliente el costo de la ejecución de la obra, la fecha de inicio, la fecha de término. Si el cliente esta de acuerdo con el tiempo de ejecución, y el costo de la ejecución de la obra se firma un contrato con todos los términos legales.

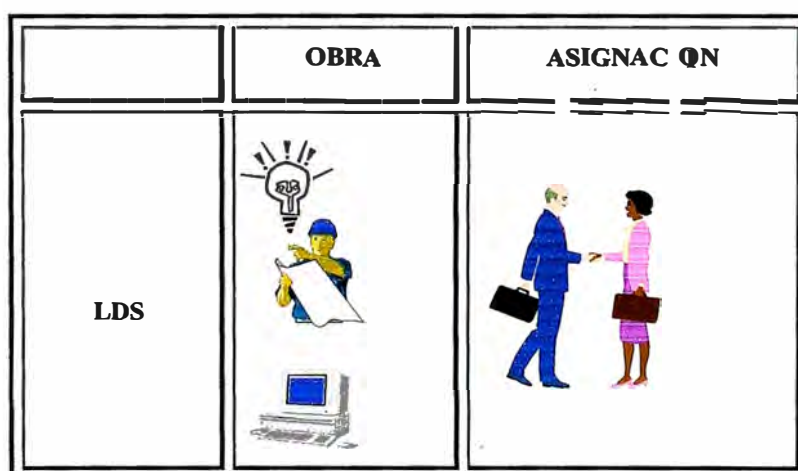


Fig. 4.3. Coordinaron con el cliente

- Una vez que se llega a un acuerdo con el cliente, se ingresa la información a la base de datos de Luz del Sur, se designa el contratista encargado de la ejecución de la obra, el supervisor de la obra.

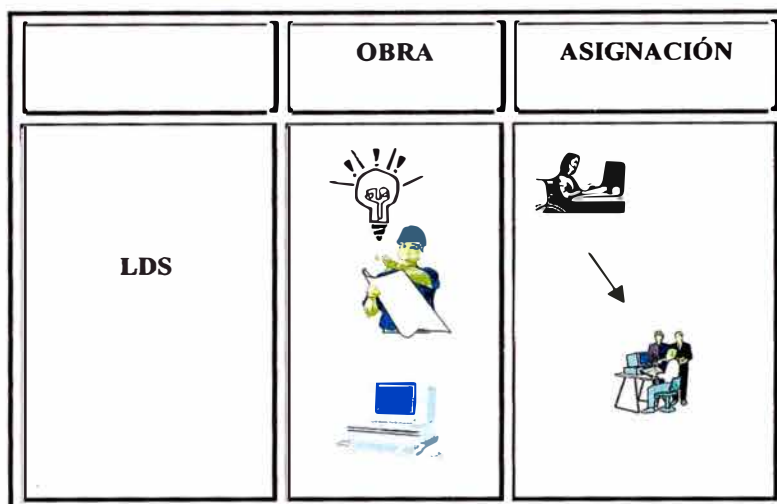


Fig. 4.4. Ingreso de la información de la obra al sistema

- Cuando se tiene la conformidad de la asignación de la obra Luz del Sur se puede dar inicio a la ejecución de esta en el campo de trabajo.

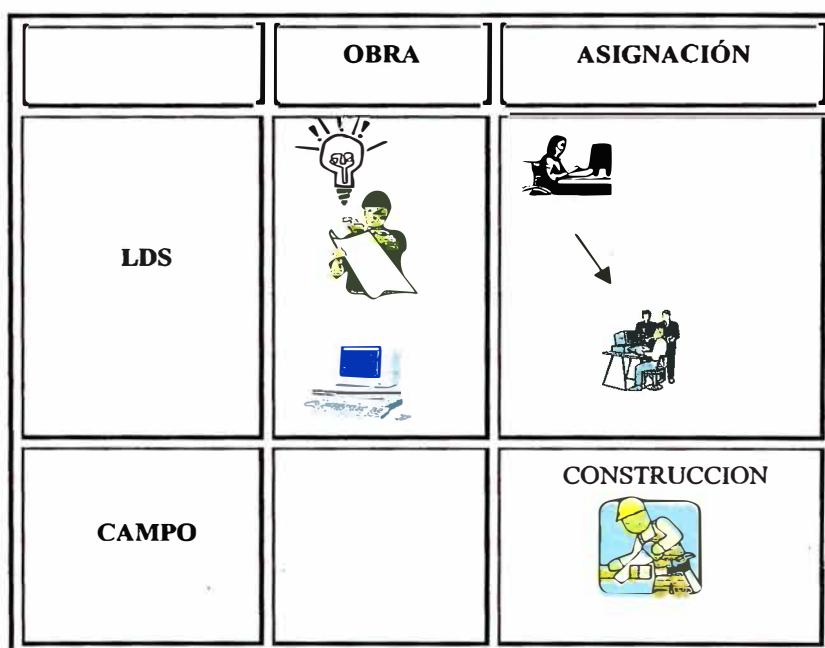


Fig. 4.5. Inicio de la Obra

- Luz del Sur asigna un supervisor en el campo de trabajo, quien será el representante de la empresa en la obra, teniendo una comunicación permanente con los encargados de la ejecución de la obra y el cliente tratando de solucionar cualquier inconveniente o reclamo que surja en el tiempo de ejecución de la obra.

También será el encargado de controlar el avance de la obra para preparar le preliquidación.



Fig. 4.6. Supervisión de la Obra

- Una vez terminada la obra, Luz del Sur prepara la preliquidación la cual debe de ser aprobada, para posteriormente proceder a liquidar el término del trabajo.

| | OBRA | ASIGNACIÓN | PRELIQUIDACIÓN | LIQUIDACIÓN |
|-------|---|---|--|---|
| LDS |  |  |  |  |
| CAMPO | | CONSTRUCCION  | SUPERVISION  | |

Fig. 4.7. Preparación de la pre liquidación

- Una vez aprobada la liquidación se ingresa al sistema para su almacenamiento en la base de datos.


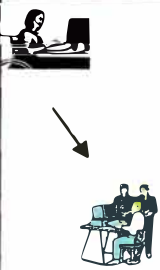



| | OBRA | ASIGNACIÓN | PRELIQUIDACIÓN |
|-------|---|---|--|
| LDS |  |  |  |
| CAMPO | | CONSTRUCCION  | SUPERVISION  |

Fig. 4.8. Ingreso de la pre liquidación a la base de datos

- Se envía la preliquidación ya ingresada en la base de datos para su aprobación por los jefes encargados de la obra.



Fig. 4.9. Envío de la liquidación para su cobro

- Una vez aprobada la liquidación se procede a realizar la facturación para su posterior cobro.

| | OBRA | ASIGNACIÓN | PRELIQUIDACIÓN | LIQUIDACIÓN | FACTURACIÓN |
|-------|---|---|--|---|---|
| LDS |  |  |  |  |  |
| CAMPO | | CONSTRUCCION  | SUPERVISION  | | |

Fig. 4.10. Facturación de la Obra

- Una vez enviada la facturación se realiza el cobro por el termino de la obra









| | OBRA | ASIGNACIÓN | PRELIQUIDACIÓN | LIQUIDACIÓN | FACTURACIÓN | PAGO |
|-------|---|---|--|--|---|---|
| LDS |  |  |  |  |  |  |
| CAMPO | | CONSTRUCCION  | SUPERVISION  | | | |

Fig. 4.11. Cobro de la Obra

4.3. Proceso de ejecución de obras: Nuevo

- En el nuevo proceso de la ejecución de obras la intención es llevar un nuevo orden en el cual se designara funciones específicas por parte de Luz del Sur sobre el trabajo en el campo de trabajo y las funciones a realizar por el contratista.

Luz del Sur se encargara del planeamiento de la obra, de la facturación y será el encargado de recibir el pago por el trabajo y a la vez realizar el pago al contratista.

| | OBRA | ASIGNACIÓN | PRE LIQUIDACIÓN | LIQUIDACIÓN | FACTURACIÓN | PAGO |
|-------------|--|---|-----------------|-------------|--|--|
| LDS |  | | | |  |  |
| CAMPO | | CONSTRUCCION  | | | | |
| CONTRATISTA | | | | | | |

Fig. 4.12. Asignación de Funciones por parte de Luz de Sur

- LDS se encarga de asignar la ejecución de la obra a una de las contratistas designada por la Gerencia de Operaciones los cuales. Este contratista contara con la supervisión de Luz del Sur.

| | OBRA | ASIGNACIÓN | PRELIQUIDACIÓN | LIQUIDACIÓN | FACTURACIÓN | PAGO |
|-------------|---|---|----------------|-------------|---|---|
| LDS |  |  | | |  |  |
| CAMPO | | CONSTRUCCION  | | | | |
| CONTRATISTA | | | | | | |

Fig. 4.13. Asignación de Funciones por parte de Luz de Sur

- Una vez que se determina el tipo de obra a ejecutar se le comunica al contratista todos los detalles para la ejecución de obra y para que puedan organizarse de la manera más apropiada para que puedan seguir con el tiempo de ejecución de la obra.

| | OBRA | ASIGNACIÓN | PRELIQUIDACIÓN | LIQUIDACIÓN | FACTURACIÓN | PAGO |
|-------------|---|---|----------------|-------------|---|---|
| LDS |  |  | | |  |  |
| CAMPO | | CONSTRUCCION  | | | | |
| CONTRATISTA | |  | | | | |

Fig. 4.14. Asignación de Contratista por Luz del Sur

- En esta etapa Luz del Sur asignara un Supervisor el cual se encargara de la supervisión del supervisor de la obra y del personal en general. En esta etapa es cuando entra a tallar gran parte de la aplicación, en esta etapa es donde el supervisor utiliza el equipo Nextel para comunicarse vía conexión WAP al sistema de la empresa enviando en tiempo real la información de la obra ya sea el inicio de la obra, el tipo de obra que se esta realizando, el termino de la obra, la autorización para la liquidación, etc. **Esta ruta de acceso se puede observar en el apéndice A, la cual detalla la manera de usar el equipo Nextel para poder acceder al sistema.**

| | OBRA | ASIGNACIÓN | PRELIQUIDACIÓN | LIQUIDACIÓN | FACTURACIÓN | PAGO |
|-----------------|--|---|--|-------------|--|--|
| DS |  |  | | |  |  |
| CAMPO | | CONSTRUCCION  | SUPERVISION  | | | |
| CONTRATIS CA | |  | | | | |

Fig. 4.15. Supervisión de la Obra por Luz de Sur

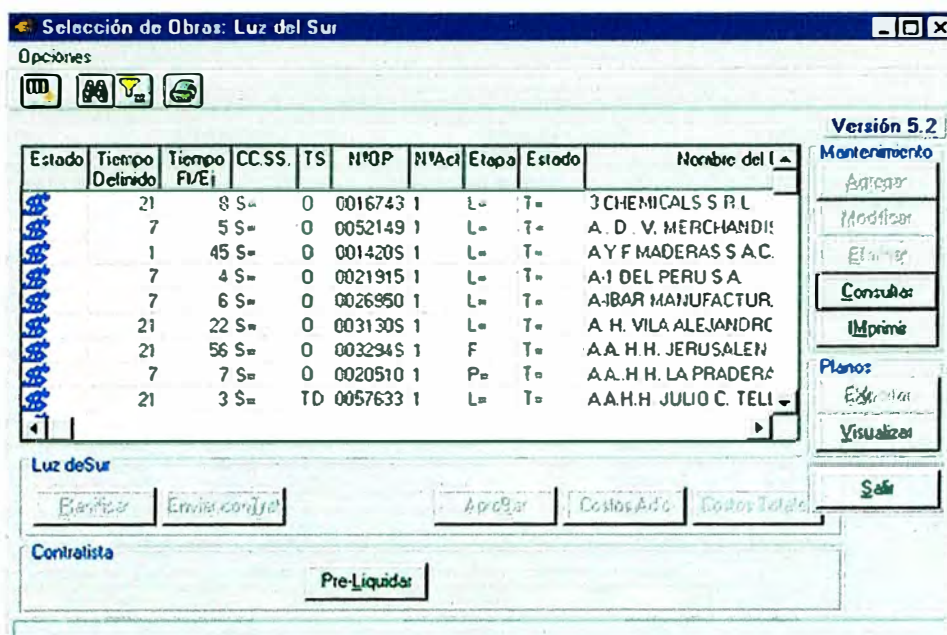


Fig. 4.16. Ingreso a Obras vía el Sistema

- El contratista se encargara de preparar la preliquidación del total de gastos y costos en los que se incurrió en la ejecución de la obra. Esto se realiza ingresando al sistema para verificar la información actualizada de la obra.

| | OBRA | ASIGNACIÓN | PRELIQUIDACIÓN | LIQUIDACIÓN | FACTURACIÓN | PAGO |
|-------------|---|---|--|-------------|---|---|
| LDS |  |  | | |  |  |
| CAMPO | | CONSTRUCCION  | SUPERVISION  | | | |
| CONTRATISTA | |  |  | | | |

Fig. 4.17. Preparación de la liquidación por parte del contratista

PROCESO DE PRELIQUIDACION: PASO EXTERNO CON CONTRATISTA

Departamento: S= SAN JUAN Tip.Solicitud: TD= PROYECTOS N° Solicitud: 0054718 N° Actualiz: 1

Opciones de Valorización
 Tipo: = TODO SubTipo: = TODO Costo: = TODO Contratista: = TODO Leer

| Detalle de Materiales | | | | Detalle de Mano de Obra | | | | Recepción de Obra | | | | |
|-----------------------|-----------------|-----------------------|-------------|-------------------------|------------|-----------|-------------------|-------------------|------------------|--------------------|------------------|------------------|
| VB | Código Material | Descripción | Tipo Valor. | STP Valor. | Tipo Costo | Proveedor | Precio Valorizado | Cant. Proy. Urd. | Total Proy. \$/. | Precio Liquidación | Cant. Ejec. Urd. | Total Ejec. \$/. |
| | 9335121 | TUBO PVC TP. SAP | 03. SUB | N. CONVE | 1. INVI | 20601841 | 0.61 | 2.00 | 1.220 | 0.61 | 5 | 0.00 |
| | 9335712 | TUBO PVC TP. SAP P. | 02. SUB | N. CONVE | 1. INVI | 20601841 | 1.08 | 6.00 | 6.480 | 1.08 | 0.00 | 0.00 |
| | 1317114 | COBRE EN FLETINA | 02. SUB | N. CONVE | 1. INVI | 20601841 | 27.65 | 0.30 | 8.295 | 27.65 | 0.00 | 0.00 |
| | 2111339 | PINTURA ESMALTE | 02. SUB | N. CONVE | 1. INVI | 20601841 | 3.27 | 1.00 | 3.270 | 3.27 | 0.00 | 0.00 |
| | 2112309 | PINTURA ESMALTE | 02. SUB | N. CONVE | 1. INVI | 20601841 | 3.27 | 1.00 | 3.270 | 3.27 | 0.00 | 0.00 |
| | 3112570 | PINTURA ESMALTE | 02. SUB | N. CONVE | 1. INVI | 20601841 | 3.27 | 1.00 | 3.270 | 3.27 | 0.00 | 0.00 |
| | 5011565 | CONDUCTOR | 02. SUB | N. CONVE | 1. INVI | 20601841 | 5.92 | 6.00 | 35.520 | 5.92 | 0.00 | 0.00 |
| | 5021243 | CONDUCTOR | 02. SUB | N. CONVE | 1. INVI | 20601841 | 2.04 | 4.00 | 8.160 | 2.04 | 0.00 | 0.00 |
| | 5021335 | CONDUCTOR | 02. SUB | N. CONVE | 1. INVI | 20601841 | 18.21 | 6.00 | 85.260 | 18.21 | 0.00 | 0.00 |
| | 5021411 | CONDUCT. SEC. IDT. 74 | 02. SUB | N. CONVE | 1. INVI | 20601841 | 0.82 | 10.00 | 8.200 | 0.82 | 0.00 | 0.00 |

Inversión Servicio Gasto Inversión Servicio Gasto
 Planif. (\$/.) 34701.97 0.00 0.00 Ejec. (\$/.) 25000000.00 0.00 0.00

Fig. 4.18. Pre liquidación de una Obra en el sistema

- Una vez aprobada la pre liquidación de la obra, esta es enviada a Luz del Sur para su aprobación y de esta manera se pueda efectuar la facturación del trabajo.

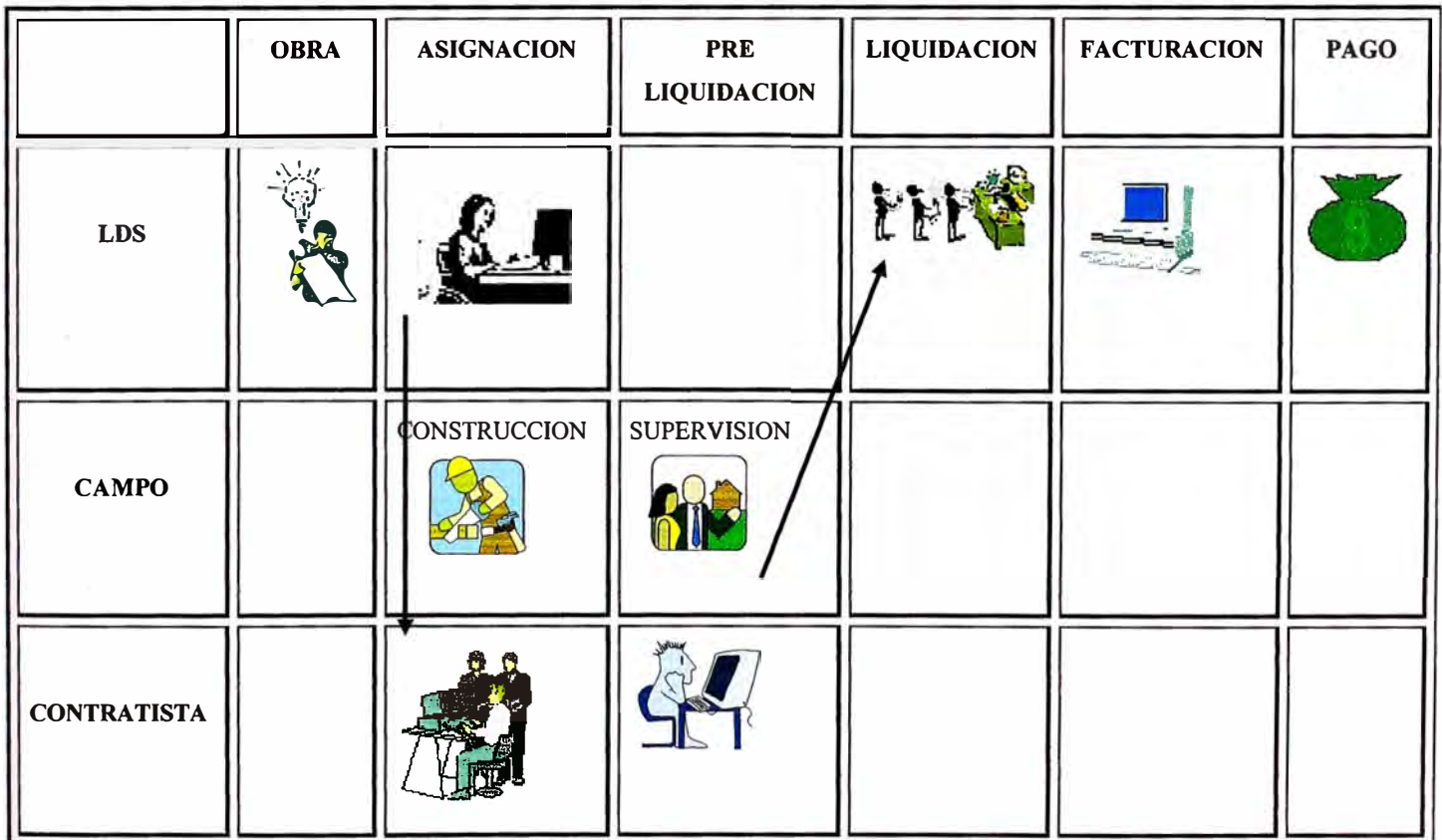


Fig. 4.19. Liquidación de la Obra

PROCESO DE LIQUIDACIÓN PASO EXTERNO CON CONTRATISTA

Departamento: [CONTRATA] Tipo de Obra: [OBRAS DE OBRAS] Nº de Liquidación: [000000] Nº de Contrato: []

Operación de Liquidación

Tipo: [1000] Subtipo: [1000] Causa: [1000] Contratación: [1000] [Lim]

| Detalle de Materiales | | | | Detalle de Mano de Obra | | | | Recepción de Obra | | | | |
|-----------------------|-----------------|------------------|------------|-------------------------|------------|------------|----------------|-------------------|----------------|-----------------|-----------------|------------|
| Nº | Código Material | Descripción | Tipo Valor | STP Valor | Tipo Costo | Porcentaje | Valor Validado | Costo Fijo Unit | Total Proyecto | Valor Liquidado | Costo Liquidado | Total Ejec |
| 1 | 10000 | TRABAJO DE OBRAS | 1000 | 1000 | 1000 | 100% | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 |
| 2 | 10000 | TRABAJO DE OBRAS | 1000 | 1000 | 1000 | 100% | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 |
| 3 | 10000 | TRABAJO DE OBRAS | 1000 | 1000 | 1000 | 100% | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 |
| 4 | 10000 | TRABAJO DE OBRAS | 1000 | 1000 | 1000 | 100% | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 |
| 5 | 10000 | TRABAJO DE OBRAS | 1000 | 1000 | 1000 | 100% | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 |
| 6 | 10000 | TRABAJO DE OBRAS | 1000 | 1000 | 1000 | 100% | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 |
| 7 | 10000 | TRABAJO DE OBRAS | 1000 | 1000 | 1000 | 100% | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 |
| 8 | 10000 | TRABAJO DE OBRAS | 1000 | 1000 | 1000 | 100% | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 |
| 9 | 10000 | TRABAJO DE OBRAS | 1000 | 1000 | 1000 | 100% | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 |
| 10 | 10000 | TRABAJO DE OBRAS | 1000 | 1000 | 1000 | 100% | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 |
| 11 | 10000 | TRABAJO DE OBRAS | 1000 | 1000 | 1000 | 100% | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 |
| 12 | 10000 | TRABAJO DE OBRAS | 1000 | 1000 | 1000 | 100% | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 |
| 13 | 10000 | TRABAJO DE OBRAS | 1000 | 1000 | 1000 | 100% | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 |
| 14 | 10000 | TRABAJO DE OBRAS | 1000 | 1000 | 1000 | 100% | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 |
| 15 | 10000 | TRABAJO DE OBRAS | 1000 | 1000 | 1000 | 100% | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 |
| 16 | 10000 | TRABAJO DE OBRAS | 1000 | 1000 | 1000 | 100% | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 |
| 17 | 10000 | TRABAJO DE OBRAS | 1000 | 1000 | 1000 | 100% | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 |
| 18 | 10000 | TRABAJO DE OBRAS | 1000 | 1000 | 1000 | 100% | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 |
| 19 | 10000 | TRABAJO DE OBRAS | 1000 | 1000 | 1000 | 100% | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 |
| 20 | 10000 | TRABAJO DE OBRAS | 1000 | 1000 | 1000 | 100% | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 |
| 21 | 10000 | TRABAJO DE OBRAS | 1000 | 1000 | 1000 | 100% | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 |
| 22 | 10000 | TRABAJO DE OBRAS | 1000 | 1000 | 1000 | 100% | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 |
| 23 | 10000 | TRABAJO DE OBRAS | 1000 | 1000 | 1000 | 100% | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 |
| 24 | 10000 | TRABAJO DE OBRAS | 1000 | 1000 | 1000 | 100% | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 |
| 25 | 10000 | TRABAJO DE OBRAS | 1000 | 1000 | 1000 | 100% | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 |
| 26 | 10000 | TRABAJO DE OBRAS | 1000 | 1000 | 1000 | 100% | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 |
| 27 | 10000 | TRABAJO DE OBRAS | 1000 | 1000 | 1000 | 100% | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 |
| 28 | 10000 | TRABAJO DE OBRAS | 1000 | 1000 | 1000 | 100% | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 |
| 29 | 10000 | TRABAJO DE OBRAS | 1000 | 1000 | 1000 | 100% | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 |
| 30 | 10000 | TRABAJO DE OBRAS | 1000 | 1000 | 1000 | 100% | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 |
| 31 | 10000 | TRABAJO DE OBRAS | 1000 | 1000 | 1000 | 100% | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 |
| 32 | 10000 | TRABAJO DE OBRAS | 1000 | 1000 | 1000 | 100% | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 |
| 33 | 10000 | TRABAJO DE OBRAS | 1000 | 1000 | 1000 | 100% | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 |
| 34 | 10000 | TRABAJO DE OBRAS | 1000 | 1000 | 1000 | 100% | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 |
| 35 | 10000 | TRABAJO DE OBRAS | 1000 | 1000 | 1000 | 100% | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 |
| 36 | 10000 | TRABAJO DE OBRAS | 1000 | 1000 | 1000 | 100% | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 |
| 37 | 10000 | TRABAJO DE OBRAS | 1000 | 1000 | 1000 | 100% | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 |
| 38 | 10000 | TRABAJO DE OBRAS | 1000 | 1000 | 1000 | 100% | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 |
| 39 | 10000 | TRABAJO DE OBRAS | 1000 | 1000 | 1000 | 100% | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 |
| 40 | 10000 | TRABAJO DE OBRAS | 1000 | 1000 | 1000 | 100% | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 |
| 41 | 10000 | TRABAJO DE OBRAS | 1000 | 1000 | 1000 | 100% | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 |
| 42 | 10000 | TRABAJO DE OBRAS | 1000 | 1000 | 1000 | 100% | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 |
| 43 | 10000 | TRABAJO DE OBRAS | 1000 | 1000 | 1000 | 100% | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 |
| 44 | 10000 | TRABAJO DE OBRAS | 1000 | 1000 | 1000 | 100% | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 |
| 45 | 10000 | TRABAJO DE OBRAS | 1000 | 1000 | 1000 | 100% | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 |
| 46 | 10000 | TRABAJO DE OBRAS | 1000 | 1000 | 1000 | 100% | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 |
| 47 | 10000 | TRABAJO DE OBRAS | 1000 | 1000 | 1000 | 100% | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 |
| 48 | 10000 | TRABAJO DE OBRAS | 1000 | 1000 | 1000 | 100% | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 |
| 49 | 10000 | TRABAJO DE OBRAS | 1000 | 1000 | 1000 | 100% | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 |
| 50 | 10000 | TRABAJO DE OBRAS | 1000 | 1000 | 1000 | 100% | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 |

Resumen de Liquidación

| Tipo Valor | STP Valor | Tipo Costo | Contratación | Tipo Requisito | Ref. Mº 9 | Estado Vig. | Ind. Aprob. | Fecha Aprob. | Ref. Aprob. | Mo. Fecha |
|------------|-----------|------------|--------------|----------------|-----------|-------------|-------------|--------------|-------------|-----------|
| 10000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 |
| 10000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 |

Indicador de Contratación
ESTO PARA ENVIAR A LUZ DEL SUR

Fig. 4.20. Liquidación vía sistema

- Como se puede observar que después de aprobada la liquidación por Luz del Sur este envía la liquidación al cliente para su aprobación y así de esta manera poder realizar el cobro del trabajo realizado y así de esta manera Luz del sur le realice el pago que le corresponde al contratista.

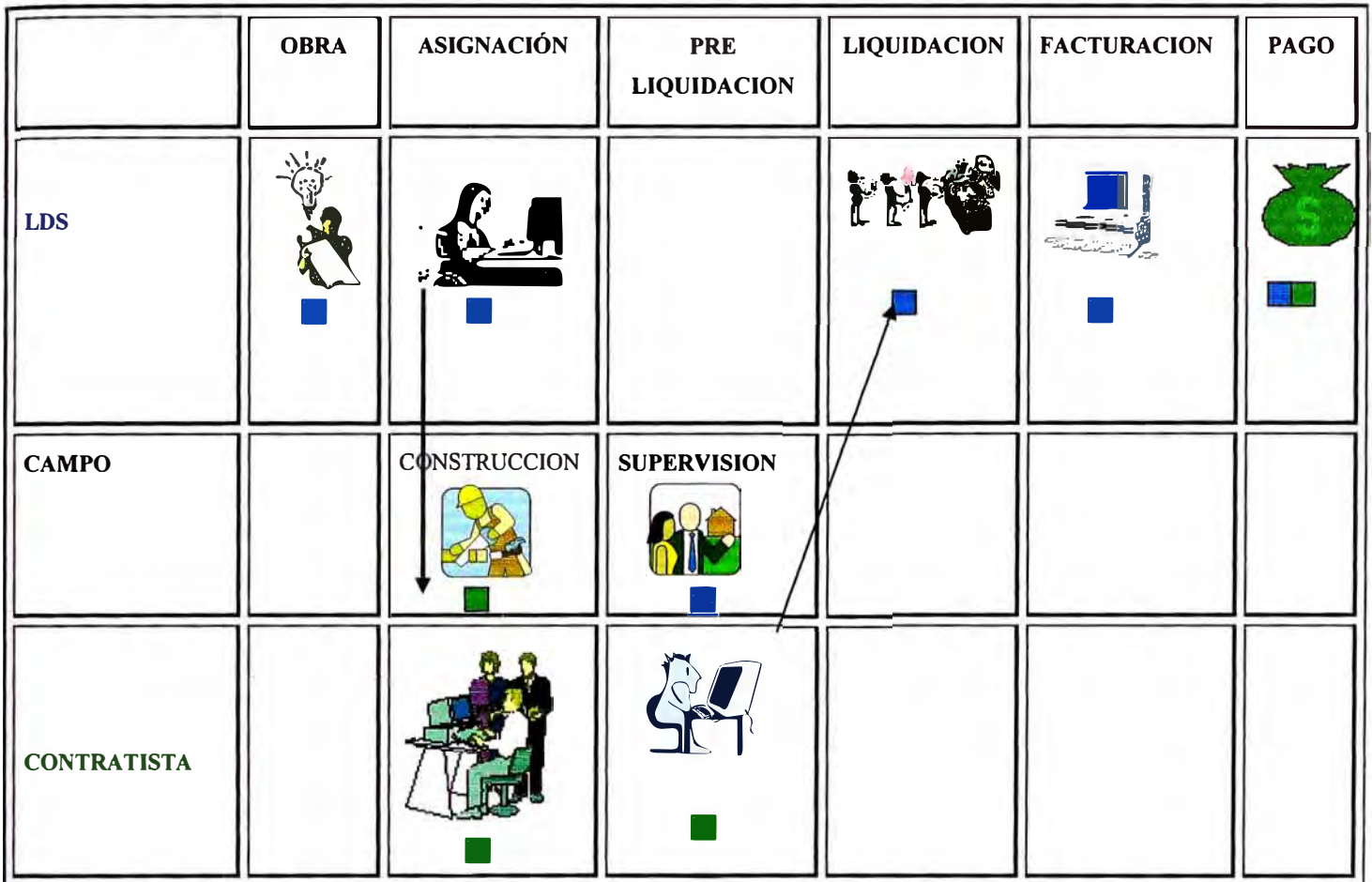


Fig. 4.21. Cobro de la obra

4.4 Tabla comparativa de los procesos Anterior, Actual, Nuevo.

En esta Tabla se intenta ver la diferencia entre los procesos anterior, actual y nuevo en comparación de los tiempos de ejecución y espacio.

Tabla 1: Tabla comparativa entre los procesos

| PROCESO | TIEMPO | ESPACIO |
|----------|-----------|-----------|
| ANTERIOR | DIFERENTE | DIFERENTE |
| ACTUAL | MISMO | MISMO |
| NUEVO | MISMO | DIFERENTE |

CARACTERISTICAS

- Las PC's en proceso de obsolescencia no necesitan potenciarse
- Desarrollo sobre plataforma conocida: Ahorra costos y reduce curva de aprendizaje.
- Proveedores no físicamente en LDS: Reduce tiempos y ahorra costos.
- Ingreso a todos los sistemas de LDS desde cualquier PC del Mundo con Internet: Brinda oportunidades.
- Desarrollo de Sistemas pensando no sólo en usuarios de LDS sino en Entorno: Cambio de paradigmas.

CONCLUSIONES

1. El mayor beneficio que se logra con este desarrollo es la atención efectiva en respuestas a los clientes por los reclamos con respecto a inicios y fin de obra.
2. Al lograr un mejor control de las órdenes de trabajo nos permite hacer proyectos con tiempos más reales.
3. El utilizar el protocolo WAP, se pudo observar que se trata de un sistema bastante útil, funcional y versátil, que nos permite el ingreso de fechas al sistema en línea que funciona interactuando con el servidor de Luz del Sur con una tabla intermedia nos da la facilidad de tener datos en línea.
4. A diferencia de GSM, GPRS y UMTS que son tecnologías de transmisión de información inalámbrica que se caracterizan por la velocidad que soportan, el tipo de pago y el tiempo de establecimiento de la conexión. WAP es un protocolo; no es comparable a GSM, GPRS y UMTS.
5. Cuando hablamos de WAP, aun siendo un protocolo, en realidad nos estamos refiriendo a la capacidad de mostrar contenidos y servicios directamente en la pantalla de nuestro dispositivo móvil. Por tanto, WAP utiliza la tecnología actual de GSM y utilizará, sin duda, GPRS y UMTS como tecnologías de base.
6. Debido a que WAP no es más que un protocolo y, como tal, evolucionará con el tiempo. El debate se centra mayormente en los servicios y los lenguajes de más alto nivel (WML).
7. La ventaja de tener Informix a comparación de otras bases de datos es que ocupa menos memoria y recursos.

8. El trabajar con un equipo Nextel nos da varios beneficios que no podemos conseguir con otras empresas, como por ejemplo el costo es sólo una fracción de lo que cuesta una llamada entre o hacia celulares. Comunica el mensaje en menor tiempo que en una llamada telefónica. Usted decide quién inicia la llamada.

9. El hecho de que Luz del Sur siempre este en constante renovación de sus equipos (servidores, pc, etc.) nos permite observar que siempre trata de estar a la vanguardia de la tecnología, lo que hace presumir que las aplicaciones de este desarrollo va ir en avance y esto se vera en un tiempo corto debido a que WAP es una tecnología joven si miramos la historia de la tecnología, no encontraremos muchos ejemplos de novedades tecnológicas que se hayan impuesto en menos de 10 años (incluso Internet tiene una vida de 10 años y sólo es utilizada por el 15% de la población).

ANEXO A

CONFIGURACION DE UN EQUIPO NEXTEL PARA INGRESAR AL SISTEMA DE LUZ DEL SUR

Descripción del equipo Nextel vía WAP

En este capítulo mostraremos los diferentes ventanas que se utilizan para acceder a cada una de las opciones mediante la ejecución del programa WAP utilizando un equipo Nextel.

Esta figura me muestra el equipo móvil de la compañía Nextel en su estado de reposo.



Fig. A1: Equipo Nextel

1. Se ingresa al menú del equipo móvil Nextel en el que se puede visualizar las siguientes funciones:

- **Red**

Esta opción se encarga de acceder al Sistema de Red. Esta será la opción elegida.

- **Alerta**

En esta opción se puede configurar las diferentes alertas que pueda recibir o enviar el equipo a otros equipos móviles.

- **Configuración**

En esta opción se puede configurar todo lo referente al sonido de llamadas entrantes, llamadas salientes, mensajes, saludo inicial, etc.



Fig. A2: Acceso a la opción Red

2. Dentro de la Opción Red encontraremos las siguientes opciones

- **Mis Consumos**

En esta opción lo que se hace es verificar el consumo de paginas o sitios visitados.

- Favoritos

Lo que se logra conseguir en esta opción es ingresar a los sitios u opciones mas utilizadas. Esta será la opción elegida.

- Localizador

Optimice el trabajo de su personal de campo conociendo su ubicación de manera inmediata a través de Internet.

- Internet

Permite el acceso para poder navegar en Internet.

- Red de Negocio

Es el acceso para ver determinadas rutas, las cuales solo pueden ser vistas con determinados permisos.

- Ir a sitio

A través de los equipos Nextel ponemos todo el poder de Internet a su alcance.

- Mensaje 2 vías

Envíe y reciba mensajes de texto o correos electrónicos desde su equipo Nextel, la web o correo electrónico.



Fig. A3: Acceso a la opción Favoritos

3. Al ingresar a la Opción Favoritos se encuentra las siguientes opciones:

- **Supervisión**

Se observa la supervisión de las diferentes obras que se están ejecutando, pudiendo verificar el inicio, termino, avance o liquidación de la obra. Esta será la opción a elegir.

- **Fecha Puser**

Esta opción indica la fecha de instalación del medidor, el cual indica la instalación del suministro eléctrico.

- **Fecha Termino**

Esta opción indica la fecha de término de la obra.

- **Emergencia**

Esta opción se utiliza en caso se presente alguna emergencia, como por ejemplo algún accidente o incidente en el cual se necesite algún equipo de auxilio.



Fig. A4: Acceso a la opción Supervisión

4. Al ingresar a la Opción Supervisión el sistema nos pedirá un usuario y una contraseña el cual es asignado por Luz del Sur solo a los encargados de las obras, jefes y supervisores para poder elegir las opciones.



Fig. A5: Ingreso del usuario y contraseña al sistema

5. Una vez ingresado al sistema se escoge el número de OT (Orden de Trabajo) el cual se va a inspeccionar, liquidar, supervisar, etc.



Fig. A6: Acceso a la Orden de Trabajo

6. Después de ingresar la OT de la obra elegida se puede ver las siguientes opciones las cuales al elegir las enviara en tiempo real la acción que se esta realizando en el campo de trabajo:

- **Inspección Previa**

Lo que se realiza en este punto es inspeccionar la zona de la obra para poder hacer el planeamiento respectivo del inicio de la obra.

- **Supervisión de la Obra**

Se realiza la supervisión de la obra indicando el avance de esta.

- **Liquidación de la Obra**

Una vez que la obra se ha culminado desde el punto de trabajo se puede enviar la orden para proceder a la liquidación de la obra.

- **Maniobra en Baja Tensión**

Indica si se esta realizando un trabajo en Baja Tensión.

- Maniobra en la URF
Indica que se esta realizando una maniobra en una Unidad de Red Energizada
- Trabajo dentro de la Sub Estación
Da a conocer que se esta realizando un trabajo en una sub estación.
- Elaboración de IP
Que se esta realizando un Inspección Planeada.
- Elaboración de OP
Indica que se esta realizando una Observación Planeada.



Fig. A7: Opciones a es coger en la OT



Fig. A8: Conformidad de la actualización de la OT

BIBLIOGRAFÍA

1. **Desarrollo WAP con WML y WML SCRIPT**
Autor: Forta, B. / Lauver, K. D. / Fonte Paul /Juncker, R. M. / O'Leary Amy /
Mandel, Ronan / Bromby Dylan
2. **Tutorial de WAP**
Autor: <http://www.programacion.com/tutorial/wml/5/>
3. **Favorito WML Club Profesional**
Autor: <http://www.wmlclub.com/cgi>
4. **Informix, la base de datos relacional para Linux**
Autor: Abdellatif, a.
5. **WAP. Guía práctica**
Autor: http://www.mundotutoriales.com/tutorial_funciones_de_informix-mdtutorial1001647.htm
6. **IBM Informix Dynamic Server sobre Linux**
Autor: <http://www.ibm.com/IDS%2010%20informix.htm>
7. **IBM Servidores eServer**
Autor: <http://www.ibm.com>
8. **Manuales e Información del Desarrollo de la ejecución del Proyecto**
Autor: Área de Sistemas de Luz del Sur