

Universidad Nacional Autónoma de México

Facultad de Ingeniería



**Desarrollo de Aplicaciones Para Dispositivos
Móviles Bajo la Plataforma Palm OS**

Tesis que para obtener el grado de Ingeniero en
Computación Presenta:

René Pascual Gazano Cancino

Director: Ing. Manuel Manríquez Miranda

A mis Padres y Hermanos

A Karen

A mis amigos

*No sólo no hubiera sido nada sin ustedes sino con toda la gente
que estuvo a mi alrededor desde el comienzo,
algunos siguen hasta hoy,
Gracias...Totales.*

Índice

INTRODUCCIÓN	5
CAPÍTULO I	7
ANTECEDENTES	7
1.1 TIPOS DE COMPUTADORAS	7
1.2 SISTEMAS OPERATIVOS	10
1.3 LA PROGRAMACIÓN	18
1.3.1 Ingeniería de Software	20
1.3.2 Tipos de programación	21
1.3.3 Proceso de desarrollo de un programa	23
1.3.4 Desarrollo Rápido de Aplicaciones	23
CAPÍTULO II	26
LOS ASISTENTES PERSONALES DIGITALES	26
2.1 HISTORIA DE LOS PDA	26
2.2 FUNCIONAMIENTO DE LOS PDA	31
2.2.1 Hardware	31
2.2.2 Software	34
2.3 SISTEMAS OPERATIVOS PARA PDA	34
2.3.1 Windows Mobile	34
2.3.2 Symbian OS	35
2.3.3 Sistemas basados en Unix	35
2.3.4 Palm OS	35
Control de acceso	36
Cifrado	36
Sistema de Archivos	36
Administración de memoria	37
2.3.4.1 Historia del Palm OS	37
2.3.4.2 Los dispositivos móviles que utilizan Palm OS	38
CAPÍTULO III	41
HERRAMIENTAS DE DESARROLLO EN PALM OS	41
3.1 LA PROGRAMACIÓN EN DISPOSITIVOS MÓVILES	41
3.1.1 Conductos (Conduits)	42
3.1.2 La Sincronización	42
3.1.3 Tipos de archivo en Palm OS	44
3.2 EL DISEÑO DE APLICACIONES	45
3.2.1 Diseño de interfaces de usuario	45
3.2.2 Modelos de Conexión	45
3.2.3 La importancia de los tipos de datos	46
3.2.4 Manejo de Base de Datos en el desarrollo de aplicaciones.	47
3.3 LENGUAJES Y ENTORNOS DE PROGRAMACIÓN PARA PALM OS	47

CAPÍTULO IV **49**

DISEÑO DE UNA APLICACIÓN DE BASE DE DATOS PARA OBTENER INFORMACIÓN EN LA INVESTIGACIÓN DE CAMPO	49
4.1 LA INVESTIGACIÓN DE CAMPO	49
4.2 DEFINICIÓN DEL PROBLEMA	51
4.3 ALTERNATIVAS DE SOLUCIÓN	57
4.3.1 PDA Toolbox	57
4.3.2 CASL	63
4.3.3 HB++ (Handheld Basic ++)	70
4.3.4 NS Basic	77
4.3.5 Satellite Forms	82
4.3.6 Superwaba	87
4.3.7 HandDBase	94

CAPÍTULO V **98**

CONCLUSIONES	98
5.1 ANÁLISIS DE SOLUCIONES	98
5.2 MEJORAS AL SISTEMA	100
5.3 EL FUTURO DE LOS PDA	100
Ambiente Médico	101
Almacenamiento y distribución	101
Ganadería y Agricultura	101
Ventas	102
Automatización de trabajo en equipo	102
Perspectiva del Palm OS	102

APÉNDICE: **105**

CÓDIGO UTILIZADO PARA EL PROYECTO EN CASL	105
CÓDIGO UTILIZADO PARA EL PROYECTO EN HB++	111
CÓDIGO UTILIZADO PARA EL PROYECTO EN NSBASIC	117
CODIGO UTILIZADO PARA EL PROYECTO EN SUPERWABA	122

FUENTES: **128**

Introducción

Con la intención de ayudar al hombre moderno en la tarea de administrar su vida cotidiana, así como de proveer cierta información que no le es posible almacenar en su mente, se desarrollaron a mediados de los ochentas los Asistentes Personales Digitales (PDA por sus siglas en inglés *Personal Digital Assistant*). El propósito principal de un PDA es actuar como un organizador electrónico o planeador de actividades diarias que es portátil, fácil de usar y capaz de compartir información con la PC.

Los PDA, también llamados handhelds o palmtops, han evolucionado a lo largo de los años. No sólo pueden manejar información personal como contactos, citas o tareas sino que los dispositivos de hoy pueden conectarse a Internet, convertirse en posicionadores GPS o ejecutar software multimedia. Además, los fabricantes han combinado PDA's con teléfonos celulares, reproductores multimedia y otros aparatos. En general, estos dispositivos se han convertido en verdaderas herramientas para proveer soluciones en múltiples campos de trabajo.

Debido a lo anterior surge la necesidad de promover el desarrollo de software para PDA, de tal manera que se aprovechen al máximo las herramientas de hardware y la movilidad y versatilidad que éstos aportan, adaptándolas o extendiéndolas a las diferentes disciplinas de la vida cotidiana.

El presente trabajo tiene como objetivo presentar a la comunidad estudiantil, y principalmente a la comunidad de Ingeniería el uso de los PDA como una herramienta más para el desarrollo de sus tareas, tanto cotidianas como profesionales. Bajo este esquema, se presenta una síntesis de historia de los PDA, las plataformas en las que se puede trabajar, y las ventajas que los PDA aportan. Por último se exhiben algunas alternativas de programación que son las herramientas necesarias para que estos equipos se ajusten a las necesidades de cada usuario.

El desarrollo de software para la tecnología móvil es un campo que poco se ha explorado en nuestra comunidad universitaria, por lo cual los objetivos de este trabajo se basan en promover y fomentar las disciplinas de programación en estos dispositivos. Se busca que este trabajo sea una referencia para los estudiantes interesados en el tema.

Mediante el desarrollo de este trabajo se mostrará la elaboración de una aplicación de base de datos buscando dos objetivos. El primero de éstos será mostrar el proceso de desarrollo de software en las diferentes herramientas y lenguajes que se presentarán. El segundo objetivo será construir una aplicación útil para recabar información y que auxilie en disciplinas que requieren investigación de campo y que elaboren encuestas, estudios, evaluaciones o cuestionarios. Dicha aplicación debe sustituir el método tradicional de investigación obteniendo ventajas tales como eliminar el proceso de vaciado de información a una base de datos en un equipo fijo o bien evitar el desperdicio de recursos como el papel.

Observaremos brevemente los sistemas operativos más exitosos en PDA para enfocarnos en el Palm OS. Hoy, los principales sistemas operativos usados en PDA se resumen en tres: Palm OS, Windows Mobile y Symbian OS. Sin

embargo las plataformas mencionadas no son las únicas. Muchos desarrolladores de hardware poseen sus propios sistemas operativos como Sharp que utiliza sistemas basados en Linux para sus PDA muy exitosos en Japón.

Palm OS es un sistema operativo que ha demostrado ser una de las plataformas más fáciles de comprender a nivel de usuario ya que es una herramienta intuitiva y carente de complejidad. Es por ello que se ha elegido en especial esta plataforma.

Por otro lado, se mostrarán los diferentes lenguajes y entornos de programación para la plataforma Palm OS exponiendo sus ventajas y desventajas. Como ya se mencionó se desarrollará una aplicación de base de datos la cual consistirá en una herramienta para recabar información en la investigación de campo. Esta aplicación se diseñará bajo distintos lenguajes y entornos de programación. Se mostrará paso a paso el desarrollo de la aplicación de base de datos en los diferentes ambientes con el fin de exponer al programador lo interesante que puede resultar el trabajar bajo estos esquemas. Una vez terminada la aplicación en cada ambiente de desarrollo, se expondrán las conclusiones con el fin de analizar las ventajas y desventajas de programar con dicha herramienta.

Se observarán los avances tecnológicos en el desarrollo de los PDA. Cabe destacar las funciones básicas de estos dispositivos en el presente las cuales son manejo de ofimática, conectividad, almacenamiento de archivos, manejo de elementos multimedia y telefonía. Dentro de los avances esperados en el desarrollo de los PDA se esperan interfaces de diálogo, aumento en capacidad de almacenamiento, manejo de gráficos entre otros. Sea cual sea el camino que tome el desarrollo de los PDA seguramente, y debido a su conectividad, comenzarán a integrarse más en la vida diaria.

Se sabe que los PDA cuentan con más herramientas y elementos de comunicación con redes por lo que es fundamental conocer los dispositivos para explotar sus características al máximo. En la actualidad se pueden encontrar aplicaciones en los PDA que van desde el simple manejo de información como bases de datos hasta grandes herramientas en diversos campos del ambiente médico, educativo, técnico y comercial por mencionar algunos.

Muchas son las tendencias en el ámbito tecnológico que es difícil predecir cuales son las que van a sobresalir. La tecnología apuesta por la miniaturización, o al menos el ahorro de espacio físico. En las empresas se cambia la PC por equipos laptop, o se integran la impresora, scanner, copiadora y fax en una multifuncional. Bajo este contexto es muy probable que los PDA se integren de manera más activa en actividades de carácter laboral, es decir, no sólo en los campos ejecutivo o personal, sino aprovechar las características de los PDA para cualquier tarea.

Este trabajo pretende presentar el esquema de trabajo en los dispositivos PDA con sistema operativo Palm. La delimitación de este trabajo sobre Palm OS es simplemente con el fin de fijar un objetivo. Las marcas y productos se exponen con carácter informativo y en ningún momento se pretende agraviar o preferir a alguno en específico ni favorecer esta plataforma sobre otras.

Capítulo I

Antecedentes

Con el fin de ubicarse en el contexto correcto, en este capítulo se tratarán conceptos de manera general para dar paso en el capítulo II al análisis de los Asistentes Personales Digitales (PDA's). Se observará la evolución de las computadoras, los sistemas operativos y los lenguajes de programación.

1.1 Tipos de Computadoras

Existen un sinnúmero de términos para describir a las computadoras. La mayoría de éstos implican el tamaño, el uso o la capacidad de las computadoras. Mientras el término Computadora puede aplicarse virtualmente a cualquier dispositivo que contenga un microprocesador, la mayoría de la gente piensa en una computadora como un dispositivo que recibe instrucciones por medio de un dispositivo de entrada (Mouse, teclado), las procesa y muestra el resultado en una pantalla. A continuación se presenta la clasificación más trascendente.

Microcomputadoras

Una microcomputadora es una computadora que contiene un chip microprocesador como unidad central de procesamiento. Usualmente son denominadas computadoras personales (PC) porque están diseñadas para ser utilizadas por no más de una persona al mismo tiempo. Las PC's son empleadas comúnmente en el hogar, en la escuela o en los negocios. El uso popular para las microcomputadoras incluyen el procesamiento de palabra, navegar en Internet, enviar y recibir correo electrónico, manejo de hoja de cálculo, manejo de bases de datos, edición de gráficos y la reproducción de elementos multimedia tales como música o videojuegos.

Las computadoras personales se presentan en Computadoras de escritorio y portátiles (laptop).

Las computadoras de escritorio son las más conocidas. El gabinete de la computadora contiene la tarjeta madre, las unidades de disco, la fuente de energía, y las tarjetas de expansión. La computadora usualmente tiene por separado el monitor (CRT o LCD) aunque algunos modelos poseen una pantalla incorporada al gabinete. Un teclado y ratón por separado le permiten al usuario proporcionar instrucciones y comandos.





Las denominadas “laptop” o “notebook” son computadoras pequeñas y lo suficientemente ligeras para ser transportadas por el usuario. Poseen una batería como fuente de energía pero también pueden ser conectadas al suministro eléctrico. Poseen una pantalla LCD la cual se repliega por protección cuando es transportada. Integran un teclado y un dispositivo que hace las veces de ratón, llamado Touch Pad.

Tanto computadoras de escritorio como laptops poseen la capacidad de procesamiento necesaria para el usuario final. El costo es la principal diferencia entre estos equipos ya que los componentes para construir un equipo laptop son más pequeños y en el mercado su costo es mayor.

PDA, Palmtops, Handhelds

Un Asistente Personal Digital (PDA por sus siglas en inglés) es una microcomputadora de mano que traduce su éxito por su tamaño y su gran portabilidad. Utilizan una pantalla de cristal líquido (LCD por sus siglas en inglés) sensible al tacto como dispositivo tanto de entrada como de salida. El usuario dibuja caracteres y oprime íconos en la pantalla con un apuntador (parecido a un lápiz) denominado “Stylus”. Los PDA se comunican con las computadoras de escritorio y otros equipos por conexión de cable, infrarrojo, o tecnologías bluetooth o wi-fi. Son utilizadas normalmente para llevar una agenda, tareas, direcciones, tomar notas.



Algunos autores presentan como sinónimo PDA, Palmtops y Handhelds mientras que otros defienden que un equipo Palmtop o Handheld es una microcomputadora pequeña que sacrifica potencia por tamaño y portabilidad. Estos dispositivos semejan una pequeña laptop con pantalla replegable y teclado. Aún así, los PDA y palmtops modernos poseen la habilidad de formar parte de redes inalámbricas o conectarse a Internet.



Servidores y Estaciones de trabajo

Una estación de trabajo es una poderosa microcomputadora que contiene uno o más microprocesadores. Puede ser utilizada por un usuario que requiere realizar tareas que necesitan un procesamiento mayor al que puede aportar una PC común, tales como el manejo de gráficos complejos o intensivos cálculos científicos.

De manera alternativa, las estaciones de trabajo pueden ser utilizadas como servidores.



Estos proporcionan alojamiento a computadoras cliente en tecnologías de red. También pueden ser utilizadas para manejar el procesamiento de varios usuarios simultáneamente conectados a través de terminales.

A continuación se exponen algunos tipos de computadoras que no se consideran microcomputadoras. Esto incluye a las Supercomputadoras, Mainframes y Minicomputadoras.

Minicomputadoras

Una Minicomputadora es una computadora multiusuario, menos poderosa que un mainframe. Esta clase de computadoras surgen en los años 60's cuando la gran escala de circuitos integrados hicieron posible construir computadoras más baratas que los existentes mainframes.

El trabajo antes cubierto por la minicomputadora en gran parte ha sido asumido por las estaciones de trabajo.

Mainframes

Un mainframe es una computadora grande, poderosa que maneja el procesamiento simultáneo de varios cientos de usuarios. Los usuarios se conectan al mainframe utilizando terminales y envían sus tareas a ser procesadas al mainframe. Una terminal es un dispositivo que posee teclado y pantalla para entrada y salida de información, pero no posee su propio procesamiento (también son llamadas terminales tontas). El poder de procesamiento del mainframe es compartido entre todos los usuarios.



El costo de los mainframes puede llegar a los cientos de miles de dólares. Son utilizados cuando se quiere el poder de procesamiento y almacenamiento de información centralizado en un solo lugar. Los mainframes también son

utilizados como servidores de alta capacidad para redes con estaciones de trabajo como clientes.

Supercomputadoras

Una supercomputadora es un mainframe cuyas velocidad y poder de procesamiento han sido optimizadas. Las supercomputadoras son utilizadas para tareas que requieren cálculos extremadamente intensivos. El costo llega a varios millones de dólares.

Recientemente se han construido supercomputadoras conectando un gran número de unidades de procesamiento individual.

La última supercomputadora adquirida por la Universidad Nacional Autónoma de México tiene por nombre *Kan Balam*. Esta supercomputadora cuenta 1368



procesadores AMD Optaron de 2.6 GHz y 3016GB de memoria RAM distribuidos en 337 nodos de cálculo, cada uno con 8GB RAM y dos procesadores DUO y en 5 nodos especializados con 64 GB RAM. Posee un sistema de almacenamiento de 768 discos duros de 200 GB cada uno.

1.2 Sistemas Operativos

Reseña histórica.

La importancia de los sistemas operativos nace históricamente desde los años 50, cuando se hizo evidente que el operar una computadora por medio de tableros en la primera generación y luego por medio del trabajo en lote en la segunda generación se podía mejorar notablemente. El operador realizaba siempre una secuencia de pasos repetitivos, lo cual es una de las características contempladas en la definición de lo que es un programa. Es decir, se comenzó a ver que las tareas mismas del operador podían plasmarse en un programa, el cual a través del tiempo y por su enorme complejidad se le llamó "Sistema Operativo". Así, se tiene entre los primeros sistemas operativos al Fortran Monitor System (FMS) y al IBSYS Operating System.

Posteriormente, en la tercera generación de computadoras nace uno de los primeros sistemas operativos con la filosofía de administrar una familia de computadoras: el OS/360 de IBM. Debido al alto costo de las computadoras era necesario idear un esquema de trabajo que mantuviese a la unidad central de procesamiento más tiempo ocupada, así como la cola de trabajos para su lectura hacia los lugares libres de memoria o la escritura de resultados. Sin embargo, se puede afirmar que los sistemas durante la tercera generación siguieron siendo básicamente sistemas de procesamiento por lotes.

En la cuarta generación la electrónica avanza hacia la integración a gran escala, pudiendo crear circuitos con miles de transistores en un centímetro cuadrado de silicón y ya es posible hablar de las computadoras personales y las estaciones de trabajo. Surgen los conceptos de interfaces amigables intentando así atraer al público en general al uso de las computadoras como herramientas cotidianas. Se hacen populares el MS-DOS y UNIX. También es común encontrar clones de computadoras personales y una multitud de empresas pequeñas ensamblándolas por todo el mundo.

Para mediados de los 80's, comienza el auge de las redes de computadoras y la necesidad de sistemas operativos en red y sistemas operativos distribuidos. La red mundial de Internet se va haciendo accesible a toda clase de instituciones y se comienzan a dar muchas soluciones (y problemas) al querer compartir recursos fijos en computadoras con sistemas operativos diferentes. Para los 90's el paradigma de la programación orientada a objetos cobra auge, así como el manejo de objetos desde los sistemas operativos. Las aplicaciones intentan crearse para ser ejecutadas en una plataforma específica y poder ver sus resultados en la pantalla o monitor de otra diferente (por ejemplo, ejecutar una simulación en una máquina con UNIX y ver los resultados en otra con DOS). Los niveles de interacción se van haciendo cada vez más profundos.

Definición

Un sistema operativo es un conjunto de programas destinados a permitir la comunicación del usuario con una computadora y gestionar sus recursos de

una forma eficaz. Comienza a trabajar cuando se enciende la computadora, y gestiona el hardware de la máquina desde los niveles más básicos.

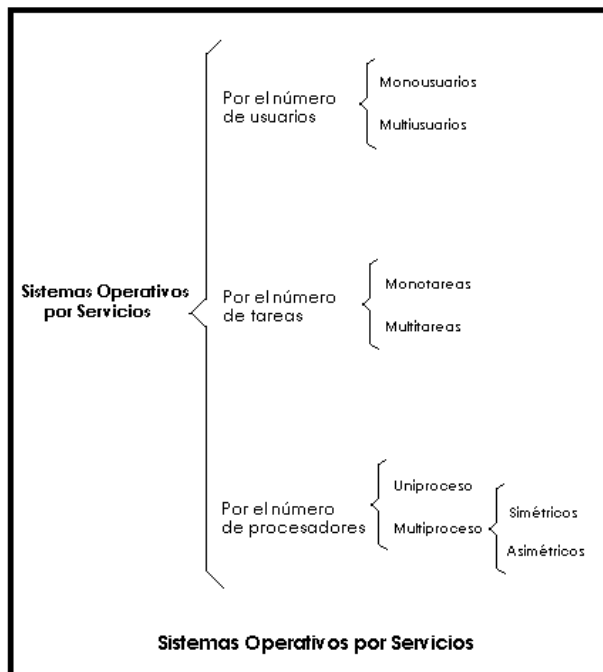
Un sistema operativo explota los recursos del hardware de uno o más procesadores, para ofrecer un conjunto de servicios a los usuarios del sistema. El sistema operativo también gestiona la memoria secundaria y los dispositivos de entrada/salida (E/S) en nombre de los usuarios.

Actualmente los sistemas operativos se pueden clasificar de la siguiente manera:

- i. Por los servicios que ofrecen (visión externa)
- ii. Por su estructura (visión interna)
- iii. Por la forma en que ofrecen sus servicios.

i. Sistemas Operativos por Servicios(Visión Externa).

Esta clasificación es la más comúnmente usada y conocida desde el punto de vista del usuario final. Esta clasificación se comprende fácilmente con el cuadro sinóptico que a continuación se muestra:



i.1) Por el Número de Usuarios:

i.1.a) Sistema Operativo Monousuario.

Los sistemas operativos monousuarios son aquellos que soportan a un usuario a la vez, sin importar el número de procesadores que tenga la computadora o el número de procesos o tareas que el usuario pueda ejecutar en un mismo instante de tiempo. Las computadoras personales típicamente se han clasificado en este renglón.

i.1.b) *Sistema Operativo Multiusuario.*

Los sistemas operativos multiusuarios son capaces de dar servicio a más de un usuario a la vez, ya sea por medio de varias terminales conectadas a la computadora o por medio de sesiones remotas en una red de comunicaciones. No importa el número de procesadores en la máquina ni el número de procesos que cada usuario puede ejecutar simultáneamente.

En esta categoría se encuentran todos los sistemas que cumplen simultáneamente las necesidades de dos o más usuarios, que comparten mismos recursos. Este tipo de sistemas se emplean especialmente en redes.

i.2) Por el Número de Tareas:

i.2.a) *Sistema Operativo Monotarea.*

Los sistemas monotarea son aquellos que sólo permiten una tarea a la vez por usuario. Puede darse el caso de un sistema multiusuario y monotarea, en el cual se admiten varios usuarios al mismo tiempo pero cada uno de ellos puede estar haciendo solo una tarea a la vez.

Los sistemas operativos monotareas son más primitivos y, solo pueden manejar un proceso en cada momento o que solo puede ejecutar las tareas de una en una.

i.2.b) *Sistema Operativo Multitarea.*

Un sistema operativo multitarea es aquél que le permite al usuario estar realizando varias labores al mismo tiempo.

Es el modo de funcionamiento disponible en algunos sistemas operativos, mediante el cual una computadora procesa varias tareas al mismo tiempo. Existen varios tipos de multitareas. La conmutación de contextos (context Switching) es un tipo muy simple de multitarea en el que dos o más aplicaciones se cargan al mismo tiempo, pero en el que solo se esta procesando la aplicación que se encuentra en primer plano (la que ve el usuario). En la multitarea cooperativa las tareas en segundo plano reciben tiempo de procesamiento durante los tiempos muertos de la tarea que se encuentra en primer plano (por ejemplo, cuando esta aplicación esta esperando información del usuario), y siempre que esta aplicación lo permita. En los sistemas multitarea de tiempo compartido cada tarea recibe la atención del microprocesador durante una fracción de segundo.

Un sistema operativo multitarea se distingue por su capacidad para soportar la ejecución concurrente de dos o más procesos activos. La multitarea se implementa generalmente manteniendo el código y los datos de varios procesos simultáneamente en memoria y multiplexando el procesador y los dispositivos de E/S entre ellos.

i.3) Por el Número de Procesadores:

i.3.a) *Sistema Operativo de Uniproceto.*

Un sistema operativo uniproceto es aquél que es capaz de manejar solamente un procesador de la computadora, de manera que si la computadora tuviese más de uno le sería inútil.

i.3.b) Sistema Operativo de Multiproceso.

Un sistema operativo multiproceso se refiere al número de procesadores del sistema, que es más de uno y éste es capaz de usarlos todos para distribuir su carga de trabajo. Generalmente estos sistemas trabajan de dos formas: simétrica o asimétricamente.

i.3.b.1) Asimétrica.

Cuando se trabaja de manera asimétrica, el sistema operativo selecciona a uno de los procesadores el cual jugará el papel de procesador maestro y servirá como pivote para distribuir la carga a los demás procesadores, que reciben el nombre de esclavos.

i.3.b.2) Simétrica.

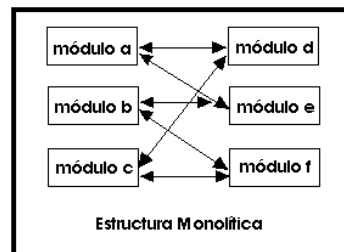
Cuando se trabaja de manera simétrica, los procesos o partes de ellos (threads) son enviados indistintamente a cualquiera de los procesadores disponibles, teniendo, teóricamente, una mejor distribución y equilibrio en la carga de trabajo bajo este esquema.

Se dice que un thread es la parte activa en memoria y ejecutándose de un proceso, lo cual puede consistir de un área de memoria, un conjunto de registros con valores específicos, la pila y otros valores de contexto.

ii. Sistemas Operativos por su Estructura (Visión Interna).

ii.1) Estructura Monolítica.

Es la estructura de los primeros sistemas operativos constituidos fundamentalmente por un solo programa compuesto de un conjunto de rutinas entrelazadas de tal forma que cada una puede llamar a cualquier otra. Las características fundamentales de este tipo de estructura son:



- * Construcción del programa final a base de módulos compilados separadamente que se unen a través del ligador.

- * Buena definición de parámetros de enlace entre las distintas rutinas existentes, que puede provocar mucho acoplamiento.

- * Carecen de protecciones y privilegios al entrar a rutinas que manejan diferentes aspectos de los recursos de la computadora, como memoria, disco, etc.

Generalmente están hechos a medida, por lo que son eficientes y rápidos en su ejecución y gestión, pero por lo mismo carecen de flexibilidad para soportar diferentes ambientes de trabajo o tipos de aplicaciones.

ii.2) Estructura Jerárquica.

A medida que fueron creciendo las necesidades de los usuarios y se perfeccionaron los sistemas, se hizo necesaria una mayor organización del



software, del sistema operativo, donde una parte del sistema contenía subpartes y esto organizado en forma de niveles.

Se dividió el sistema operativo en pequeñas partes, de tal forma que cada una de ellas estuviera perfectamente definida. Se constituyó una estructura jerárquica o de niveles en los sistemas operativos, el primero de los cuales fue denominado THE¹ que se utilizó con fines didácticos. Se puede pensar también en estos sistemas como si fueran "multicapa".

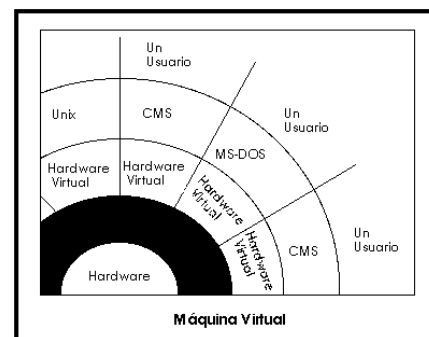
En la estructura anterior se basan prácticamente la mayoría de los sistemas operativos actuales. Otra forma de ver este tipo de sistema es la denominada de anillos concéntricos o "rings".

En el sistema de anillos, cada uno tiene una apertura, conocida como puerta o trampa (trap), por donde pueden entrar las llamadas de las capas inferiores. De esta forma, las zonas más internas del sistema operativo o núcleo del sistema estarán más protegidas de accesos indeseados desde las capas más externas. Las capas más internas serán, por tanto, más privilegiadas que las externas.

ii.3) Máquina Virtual.

Se trata de un tipo de sistemas operativos que presentan una interface a cada proceso, mostrando una máquina que parece idéntica a la máquina real subyacente. Estos sistemas operativos separan dos conceptos que suelen estar unidos en el resto de sistemas: la multiprogramación y la máquina extendida. El objetivo de los sistemas operativos de máquina virtual es el de integrar distintos sistemas operativos dando la sensación de ser varias máquinas diferentes.

El núcleo de estos sistemas operativos se denomina monitor virtual y tiene como misión llevar a cabo la multiprogramación, presentando a los niveles superiores tantas máquinas virtuales como se soliciten. Estas máquinas virtuales no son máquinas extendidas, sino una réplica de la máquina real, de manera que en cada una de ellas se pueda ejecutar un sistema operativo diferente, que será el que ofrezca la máquina extendida al usuario



ii.4) Cliente-Servidor (Microkernel).

El tipo más reciente de sistemas operativos es el denominado Cliente-servidor, que puede ser ejecutado en la mayoría de las computadoras, ya sean grandes o pequeñas. Este sistema sirve para toda clase de aplicaciones por tanto, es de propósito general y cumple con las mismas actividades que los sistemas operativos convencionales.

El núcleo tiene como misión establecer la comunicación entre los clientes y los servidores. Los procesos pueden ser tanto servidores como clientes. Por ejemplo, un programa de aplicación normal es un cliente que llama al servidor correspondiente para acceder a un archivo o realizar una operación de entrada/salida sobre un dispositivo concreto. A su vez, un proceso cliente puede actuar como servidor para otro. Este paradigma ofrece gran flexibilidad

¹ Technische Hogeschool Eindhoven, diseñado por Edsger Dijkstra

en cuanto a los servicios posibles en el sistema final, ya que el núcleo provee solamente funciones muy básicas de memoria, entrada/salida, archivos y procesos, dejando a los servidores proveer la mayoría que el usuario final o programador puede usar. Estos servidores deben tener mecanismos de seguridad y protección que, a su vez, serán filtrados por el núcleo que controla el hardware.

iii. Sistemas Operativos por la Forma de Ofrecer sus Servicios

Esta clasificación también se refiere a una visión externa, que en este caso se refiere a cómo el usuario accede a los servicios. Bajo esta clasificación se pueden detectar dos tipos principales: sistemas operativos de red y sistemas operativos distribuidos.

iii.1) Sistema Operativo de Red.

Los sistemas operativos de red se definen como aquellos que tienen la capacidad de interactuar con sistemas operativos en otras computadoras por un medio de transmisión con el objeto de intercambiar información, transferir archivos, ejecutar comandos remotos entre otras actividades. El punto crucial de estos sistemas es que el usuario debe saber la sintaxis de un conjunto de comandos o llamadas al sistema para ejecutar estas operaciones, además de la ubicación de los recursos que desee acceder. El primer Sistema Operativo de red estaba enfocado a equipos con un procesador Motorola 68000, pasando posteriormente a procesadores Intel como Novell Netware.

iii.2) Sistemas Operativos Distribuidos.

Los sistemas operativos distribuidos abarcan los servicios de los de red, logrando integrar recursos (impresoras, unidades de respaldo, memoria, procesos, unidades centrales de procesamiento) en una sola máquina virtual a la que el usuario accede en forma transparente. Es decir, ahora el usuario ya no necesita saber la ubicación de los recursos, sino que los conoce por nombre y simplemente los usa como si todos ellos fuesen locales a su lugar de trabajo. Entre los diferentes Sistemas Operativos distribuidos que existen se encuentran los siguientes: Sprite, Solaris-MC, Mach, Chorus, Spring, Amoeba, Taos, entre otros.

Procesos

Uno de los conceptos más importantes que gira entorno a un sistema operativo es el de proceso. Un proceso es un programa en ejecución junto con el entorno asociado (registros, variables, etc.)

El corazón de un sistema operativo es el núcleo, un programa de control que reacciona ante cualquier interrupción de eventos externos y que da servicio a los procesos, creándolos, terminándolos y respondiendo a cualquier petición de servicio por parte de los mismos.

El Núcleo del Sistema Operativo.

Todas las operaciones en las que participan procesos son controladas por la parte del sistema operativo denominada núcleo. El núcleo normalmente representa sólo una pequeña parte de lo que por lo general se piensa que es todo el sistema operativo, pero es tal vez el código que más se utiliza. Por esta razón, el núcleo reside por lo regular en la memoria principal, mientras que otras partes del sistema operativo son cargadas en la memoria principal sólo cuando se necesitan.

Los núcleos se diseñan para realizar "el mínimo" posible de procesamiento en cada interrupción y dejar que el resto lo realice el proceso apropiado del sistema, que puede operar mientras el núcleo se habilita para atender otras interrupciones.

El núcleo de un sistema operativo normalmente contiene el código necesario para realizar las siguientes funciones:

- Manejo de interrupciones.
- Creación y destrucción de procesos.
- Cambio de estado de los procesos.
- Despacho.
- Suspensión y reanudación de procesos.
- Sincronización de procesos.
- Comunicación entre procesos.
- Manipulación de los bloques de control de procesos.
- Apoyo para las actividades de entrada/salida.
- Apoyo para asignación y liberación de memoria.
- Apoyo para el sistema de archivos.
- Apoyo para el mecanismo de llamada y retorno de un procedimiento.
- Apoyo para ciertas funciones de contabilidad del sistema.

Dispositivos de Entrada y Salida.

El código destinado a manejar la entrada y salida de los diferentes periféricos en un sistema operativo es de una extensión considerable y sumamente complejo. Resuelve las necesidades de sincronizar, atrapar interrupciones y ofrecer llamadas al sistema para los programadores.

Los dispositivos de entrada salida se dividen, en general, en dos tipos: dispositivos orientados a bloques y dispositivos orientados a caracteres.

Dispositivos Orientados a Bloques.

Los dispositivos orientados a bloques tienen la propiedad de que se pueden direccionar, esto es, el programador puede escribir o leer cualquier bloque del dispositivo realizando primero una operación de posicionamiento sobre el dispositivo. Los dispositivos más comunes orientados a bloques son los discos duros, la memoria, discos compactos y, posiblemente, unidades de cinta.

Orientados a Caracteres.

Los dispositivos orientados a caracteres son aquellos que trabajan con secuencias de bytes sin importar su longitud ni ninguna agrupación en especial. No son dispositivos direccionables. Ejemplos de estos dispositivos son el teclado, la pantalla o display y las impresoras.

La clasificación anterior no es perfecta, porque existen varios dispositivos que generan entrada o salida que no pueden englobarse en esas categorías. Por ejemplo, un reloj que genera pulsos. Sin embargo, aunque existan algunos periféricos que no se puedan categorizar, todos están administrados por el sistema operativo por medio de una parte electrónica - mecánica y una parte de software.

Administración de Archivos.

Un archivo es un conjunto de información, que se encuentra almacenada o guardada en la memoria principal de la computadora, en el disco duro, o en cualquier otra unidad de almacenamiento.

Antes de que un archivo pueda leerse o escribirse en él, debe abrirse, momento en el cual se verifican los permisos. Estos archivos se abren especificando en la computadora la ruta de acceso al archivo desde el directorio raíz, que es la unidad principal del disco de la computadora. Entonces el sistema operativo visualiza el entorno al abrir un archivo.

Seguridad de un Sistema Operativo.

En los sistemas operativos se requiere tener una buena seguridad informática, tanto del hardware, programas y datos, previamente haciendo un balance de los requerimientos y mecanismos necesarios con el fin de asegurar la integridad de la información contenida.

Dependiendo de los mecanismos utilizados y de su grado de efectividad, se puede hablar de sistemas seguros e inseguros. En primer lugar, deben imponerse ciertas características en el entorno donde se encuentra la instalación de los equipos, con el fin de impedir el acceso a personas no autorizadas, mantener un buen estado y uso del material y equipos, así como eliminar los riesgos de causas de fuerza mayor, que puedan destruir la instalación y la información contenida.

En la actualidad son muchas las violaciones que se producen en los sistemas informáticos, en general por acceso de personas no autorizadas que obtienen información confidencial pudiendo incluso manipularla. En ocasiones este tipo de incidencias resulta grave por la naturaleza de los datos; por ejemplo si se trata de datos bancarios, datos oficiales que puedan afectar a la seguridad de los estados, etc.

Todos los mecanismos dirigidos a asegurar el sistema informático sin que el propio sistema intervenga en el mismo se engloban en lo que se denomina seguridad externa.

La seguridad externa puede dividirse en dos grandes grupos:

Seguridad física. Engloba aquellos mecanismos que impiden a los agentes físicos la destrucción de la información existente en el sistema; entre ellos se puede citar el fuego, el humo, inundaciones descargas eléctricas, campos magnéticos, entre otros.

Seguridad de administración. Engloba los mecanismos más usuales para impedir el acceso lógico de personas físicas al sistema.

Todos los mecanismos dirigidos a asegurar el sistema informático, siendo el propio sistema el que controla dichos mecanismos, se engloban en lo que se denomina seguridad interna.

1.3 La Programación

Un lenguaje de programación es un lenguaje artificial que puede ser utilizado para controlar el comportamiento de una máquina, particularmente de una computadora. Los lenguajes de programación, como los lenguajes humanos, son definidos a través de reglas semánticas y sintácticas que determinan su estructura y significado respectivamente.

Los lenguajes de programación son utilizados para facilitar la comunicación acerca de la tarea de organizar y manipular información y expresar algoritmos de manera precisa. Hoy se han creado miles de lenguajes de programación y cada año surgen nuevos.

Muchos lenguajes han sido diseñados, alterados para cubrir nuevas necesidades, combinados con otros lenguajes y eventualmente caído en desuso. Aunque ha habido intentos de diseñar un lenguaje “universal” que sirva todos los propósitos todos ellos han fallado. La necesidad de diversos lenguajes de programación surge de la diversidad de contextos en los cuales los lenguajes son utilizados:

- Los programas van desde pequeños “scripts” realizados por aficionados hasta grandes sistemas escritos por cientos de programadores.
- Los programadores van desde novatos quienes necesitan simplicidad, hasta expertos quienes exigirán mayor complejidad.
- Los programas deben considerar velocidad, tamaño y simplicidad de los sistemas que pueden ser desde microcontroladores hasta supercomputadoras.
- Los programas pueden ser escritos una vez y no sufrir cambios por generaciones, o bien pueden sufrir constantes modificaciones.
- Los programadores pueden diferir en sus gustos; pueden estar acostumbrados a discutir ciertos problemas y resolverlos en un lenguaje en particular.



Una moda común en el desarrollo de los lenguajes de programación ha sido agregar más habilidades para resolver problemas usando mayores niveles de abstracción. Los primeros lenguajes estaban muy ligados al hardware. Al tiempo que se han ido desarrollando nuevos lenguajes, se han añadido características que permiten a los programadores expresar ideas que se convierten en instrucciones traducidas al hardware.

La forma superficial de un lenguaje de programación es conocida como Sintaxis. La mayoría de los lenguajes son puramente textuales; usan secuencias de texto incluyendo palabras, números y puntuación, muy similar a los lenguajes naturales escritos. Por otro lado, hay algunos lenguajes que son

más gráficos, utilizan relaciones espaciales entre símbolos para especificar un programa.

La sintaxis de un lenguaje describe las combinaciones posibles de los símbolos que forman un programa sintácticamente correcto. El significado dado a la combinación de símbolos es manejado por la semántica.

La especificación de un lenguaje de programación proporciona una definición del lenguaje que los usuarios y diseñadores del lenguaje pueden usar para interpretar el comportamiento de los programas cuando lean su código fuente.

La implementación del lenguaje proporciona la forma para ejecutar un programa sobre una o más configuraciones de hardware y software. Hay dos prácticas de la implementación del lenguaje: La compilación y la interpretación. También es posible implementar lenguajes usando las dos técnicas.

En los años 40's fue creada la primer computadora digital. Las computadoras de los tempranos años 50, como la UNIVAC I y la IBM 701, usaban programas en lenguaje máquina. La primera generación de lenguajes de programación fue rápidamente sucedida por una segunda generación de lenguajes conocida como Lenguajes Ensambladores. Más tarde en los 50's, los lenguajes ensambladores los cuales habían evolucionado con el uso de macroinstrucciones fueron sucedidos por tres modernos lenguajes de programación: Fortran, Lisp y Cobol. Versiones actualizadas de estos lenguajes siguen utilizándose en nuestros días y han influenciado a la creación de otros tantos lenguajes. A finales de los 50's apareció Algol 60, y puede decirse que la mayoría de los lenguajes de programación modernos son en cierta forma descendientes de Algol.

El periodo entre 1960 y 1970 trajo el desarrollo de los paradigmas de lenguajes hoy en uso, aunque muchos aspectos fueron refinamientos de ideas en los primeros lenguajes de la Tercera generación

- ◆ APL introdujo la programación con arreglos e influenció la programación funcional
- ◆ En los años 60 Simula fue el primer lenguaje diseñado para soportar la programación orientada a objetos. A mediados de los 70 Smalltalk le siguió con el primer lenguaje "puramente" orientado a objetos.
- ◆ C fue desarrollado entre 1969 y 1973 como un lenguaje de programación de sistemas convirtiéndose en un lenguaje muy popular.
- ◆ Prolog fue diseñado en 1972 y fue el primer lenguaje de programación lógico.
- ◆ En 1978 ML construyó un sistema de tipificado polimórfico sobre Lisp, promoviendo los lenguajes tipificados estáticamente funcionales.

Cada uno de estos lenguajes posee una familia entera de descendientes y la mayoría de los lenguajes modernos tiene al menos uno de éstos lenguajes como ancestro.

En los años 60 y 70 se vivió el debate de los méritos de la programación estructurada, y si los lenguajes de programación deberían ser diseñados para soportarla. En este debate se argumentaba que las sentencias "GOTO" deberían ser eliminadas de todos los lenguajes de programación de alto nivel.

Los años 80 fueron años de relativa consolidación. C++ combinó la programación de sistemas y la orientación a objetos. Un avance importante en el diseño de lenguajes en la década de los 80 fue el incremento del enfoque en

los sistemas de programación a gran escala a través del uso de módulos, o unidades de código organizacional a gran escala.

El rápido crecimiento del Internet a mediados de los 90 creó oportunidades para nuevos lenguajes tales como Perl, una herramienta de “scripts” en Unix liberada en 1987 que se volvió común en el desarrollo de páginas web dinámicas. Java surgió orientado a la programación para el servidor. Estos esquemas no fueron fundamentalmente nuevos, fueron más bien refinamientos a paradigmas y lenguajes ya existentes y en gran parte basados en la familia de lenguajes de programación de C.

La evolución de los lenguajes de programación continúa tanto en la industria como en la investigación. Las nuevas corrientes apuntan a incluir seguridad, fiabilidad, nuevos tipos de modularidad e integración de bases de datos.

No existe un esquema reinante para la clasificación de los lenguajes de programación. Un lenguaje de programación dado comúnmente no tiene un solo ancestro. Los lenguajes nuevos comúnmente surgen de combinar varios elementos de lenguajes predecesores con nuevas ideas.

La tarea de clasificarlos se complica por el hecho de que los lenguajes pueden ser clasificados a través de múltiples parámetros. Por ejemplo, Java es un lenguaje orientado a objetos (ya que su organización es orientada a objetos) pero también un lenguaje de tipo concurrente (ya que tiene la capacidad de ejecutar múltiples procesos en paralelo). Por otro lado Python es un lenguaje de scripts orientado a objeto.

1.3.1 Ingeniería de Software

La ingeniería de software es el proceso de construir aplicaciones de tamaño o alcance prácticos, en las que predomina el esfuerzo del software y que satisfacen los requerimientos de funcionalidad y desempeño. La programación es una de las actividades de la ingeniería de software. La diferencia entre la programación y la ingeniería de software recae en la magnitud de los proyectos y en el conocimiento del personal requerido.

Existe una tendencia a identificar el proceso de creación de un programa informático con la programación, que es cierta cuando se trata de programas pequeños para uso personal, y que dista de la realidad cuando se trata de grandes proyectos. El proceso de creación de software desde el punto de vista de la Ingeniería tiene los siguientes pasos:

1. Entender la naturaleza del proyecto. Entender la magnitud del proyecto, tiempo, fondos y personal disponible. Esto ayuda a aclarar el alcance del proyecto.
2. Los proyectos requieren documentación desde el principio; es muy probable que esta documentación sufra muchos cambios. Este proceso es conocido como Administración de la configuración.
3. Desarrollar un plan global para el proyecto, incluida una programación del calendario.
4. Reunir los requerimientos para la aplicación. Debe quedar claro qué es lo que debe hacer el programa y para qué se necesita.
5. Diseño e implementación del producto. Consiste en realizar un diseño detallado, especificando completamente todo el funcionamiento del programa, tras lo cual la codificación debería resultar inmediata.

6. Implantar (instalar) el programa. Consiste en poner el programa en funcionamiento junto con los componentes que pueda necesitar (bases de datos, redes de comunicaciones, etc.). El producto inicial y el producto final deben probarse en forma exhaustiva.
7. Una vez entregado el producto entra el modo de “mantenimiento” que incluye reparaciones y mejoras. El mantenimiento llega a consumir hasta el 80% de los recursos de ingeniería de software.

La Ingeniería del Software se centra en los pasos de planificación y diseño del programa, mientras que antiguamente la realización de un programa consistía únicamente en escribir el código.

Es difícil construir aplicaciones de tamaño realista debido a su complejidad. El proceso bajo el cual se construyen es un factor crucial para administrar esta complejidad. Existen varios modelos del proceso de ingeniería de software, que lo abarcan y organizan entre los cuales se pueden mencionar el modelo del proceso en cascada, el modelo del proceso en espiral, el modelo del proceso por incrementos o el proceso de desarrollo de software unificado entre otros.

1.3.2 Tipos de programación

Los programas suelen subdividirse en partes menores (módulos), de modo que la complejidad algorítmica² de cada una de las partes sea menor que la del programa completo, lo cual ayuda al desarrollo del programa.

Se han propuesto diversas técnicas de programación, cuyo objetivo es mejorar tanto el proceso de creación de software como su mantenimiento. Entre ellas se pueden mencionar las siguientes:

- a) La Programación estructurada
- b) La programación modular
- c) La programación orientada a objetos
- d) La programación concurrente
- e) La programación funcional
- f) La programación lógica

a) Programación estructurada.

La programación estructurada esta compuesta por un conjunto de técnicas que han ido evolucionando, aumentando considerablemente la productividad del programa, reduciendo el tiempo de depuración y mantenimiento del mismo. Esta programación estructurada utiliza un número limitado de estructuras de control, reduciendo considerablemente los errores.

Esta técnica incorpora:

- Diseño descendente (top-down): el problema se descompone en etapas o estructuras jerárquicas.
- Recursos abstractos (simplicidad): consiste en descomponer las acciones complejas en otras más simples capaces de ser resueltas con mayor facilidad.

² Según Niklaus Wirth un programa está formado por algoritmos y estructura de datos.

- Estructuras básicas: existen tres tipos de estructuras básicas:
 - Estructuras secuenciales: cada acción sigue a otra acción secuencialmente. La salida de una acción es la entrada de otra.
 - Estructuras selectivas: en estas estructuras se evalúan las condiciones y en función del resultado de las mismas se realizan unas acciones u otras. Se utilizan expresiones lógicas.
 - Estructuras repetitivas: son secuencias de instrucciones que se repiten un número determinado de veces.

Las principales ventajas de la programación estructurada son:

- Los programas son mas fáciles de entender
- Se reduce la complejidad de las pruebas
- Aumenta la productividad del programador
- Los programas quedan mejor documentados internamente.

Un programa esta estructurado si posee un único punto de entrada y sólo uno de salida, existen de "1 a n" caminos desde el principio hasta el fin del programa y por último, que todas las instrucciones son ejecutables sin que aparezcan bucles infinitos.

b) Programación modular

La programación modular consta de varias secciones divididas de forma que interactúan a través de llamadas a procedimientos, que integran el programa en su totalidad.

En la programación modular, el programa principal coordina las llamadas a los módulos secundarios y pasa los datos necesarios en forma de parámetros. A su vez cada modulo puede contener sus propios datos y llamar a otros módulos o funciones.

c) Programación orientada a objetos (POO)

Se trata de una técnica que aumenta considerablemente la velocidad de desarrollo de los programas gracias a la reutilización de los objetos.

El elemento principal de la programación orientada a objetos es el objeto. El objeto es un conjunto complejo de datos y programas que poseen estructura y forman parte de una organización. Un objeto contiene varios datos bien estructurados y pueden ser visibles o no dependiendo del programador y las acciones del programa en ese momento.

d) Programación concurrente

Este tipo de programación se utiliza cuando se tienen que realizar varias acciones a la vez. Se suele utilizar para controlar los accesos de usuarios y programas a un recurso de forma simultánea. Se trata de una programación más lenta y laboriosa, obteniendo unos resultados lentos en las acciones.

e) Programación funcional

Se caracteriza principalmente por permitir declarar y llamar a funciones dentro de otras funciones.

f) Programación lógica

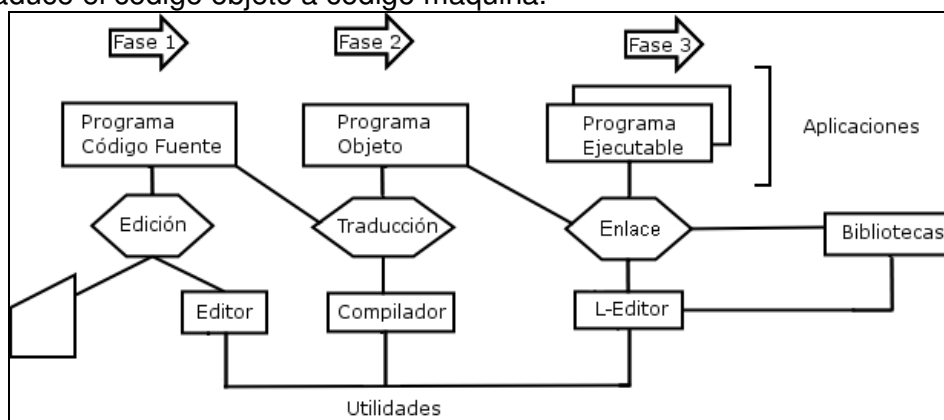
Se suele utilizar en la inteligencia artificial. Se trata de una programación basada en el cálculo de predicados (una teoría matemática que permite lograr que una computadora, basándose en hechos y reglas lógicas, pueda dar soluciones inteligentes).

1.3.3 Proceso de desarrollo de un programa

El programa escrito en un lenguaje de programación no es inmediatamente ejecutado en una computadora. La opción más común es compilar el programa, aunque también puede ser ejecutado mediante un intérprete informático.

El código fuente del programa se debe someter a un proceso de transformación para convertirse en lenguaje máquina, interpretable por el procesador. A este proceso se le llama compilación.

Normalmente la creación de un programa ejecutable conlleva tres fases. La primera fase se trata de la edición del programa, el código fuente. La segunda fase se llama compilación (propriadamente dicho) y traduce el código fuente escrito en un lenguaje de programación almacenado en un archivo a código en bajo nivel, (normalmente en código objeto no directamente al lenguaje máquina). La tercera fase se llama enlazado (del inglés link o linker); se junta el código de bajo nivel generado de todos los ficheros que se han mandado compilar y se añade el código de las funciones que hay en las bibliotecas del compilador para que el ejecutable pueda comunicarse con el sistema operativo y traduce el código objeto a código máquina.



Estos dos pasos se pueden mandar hacer por separado, almacenando el resultado de la fase de compilación en archivos objetos, para enlazarlos posteriormente, o crear directamente el ejecutable con lo que la fase de compilación se almacena sólo temporalmente. Un programa podría tener partes escritas en varios lenguajes (C, C++, Cobol, Java, Delphi, Asm, etc.), que se podrían compilar de forma independiente y enlazar juntas para formar un único ejecutable.

1.3.4 Desarrollo Rápido de Aplicaciones

El desarrollo rápido de aplicaciones (RAD, Rapid Application Development), es un proceso de desarrollo de software desarrollado inicialmente por James

Martin en 1980. El método comprende el desarrollo interactivo, la construcción de prototipos y el uso de utilidades CASE (Ingeniería de Software Asistida por Computadora). Tradicionalmente, el desarrollo rápido de aplicaciones tiende a englobar también la facilidad de uso, utilidad y la rapidez de ejecución.

Se suele utilizar para referirnos al desarrollo rápido, herramientas como Glade, o IDEs de desarrollo completas como Delphi, Foxpro o Net Beans entre otros. Uno de los programas más usados para elaborar aplicaciones bajo este esquema es Microsoft Visual Basic.

El desarrollo rápido tiene dos ventajas primarias:

- Velocidad del desarrollo: Los aumentos de la velocidad son debido al uso de la herramienta CASE.
- Calidad: según lo definido por el RAD, es el grado al cual un uso entregado resuelve las necesidades de usuarios así como el grado al cual un sistema entregado tiene costes de mantenimiento bajos. El RAD aumenta calidad con la implicación del usuario en las etapas del análisis y del diseño.

A continuación se mencionan ejemplos de herramientas RAD:

Herramientas RAD Multiplataforma

- NetBeans
- Revolution Studio Es una avanzada herramienta multiplataforma RAD que deriva ejecutables sobre Windows, Linux, Solaris, MacOS X Universal Binary and MacOS Classic.
- Lazarus Es un IDE multiplataforma similar a Borland Delphi.
- Real Basic Es un IDE multiplataforma similar a Visual Basic.
- Leonardi Es una herramienta avanzada multiplataforma RAD que deriva ejecutables sobre Windows, Linux, MacOS.
- Microsoft Visual Studio 2003 / 2005

Herramientas RAD para Escritorio

- AppBuilder 1 2 3
- Automated Architecture's Blue Ink
- Borland C++Builder
- Borland Delphi
- Gambas
- Microsoft Visual_Basic
- Ultimate++

Herramientas RAD para Bases de Datos

- Clarion Es una herramienta (ARAD) Advanced Rapid Application
- FileMaker Pro Advanced
- Omnis Studio
- Oracle Forms
- Sybase PowerBuilder
- Velneo

Herramientas RAD Orientadas a la WEB

- 37 Signals Ruby on Rails

- Adobe ColdFusion
- Symfony (PHP)
- Velneo
- Leonardi
- Microsoft Visual Studio 2003 / 2005

Capítulo II

Los Asistentes Personales Digitales

En el capítulo anterior se observó una clasificación de los tipos de computadoras entre las cuales se encuentran los Asistentes Personales Digitales (PDA) En esta sección se conocerá la evolución de estos dispositivos, el funcionamiento de un PDA, así como los principales sistemas operativos que imperan bajo estos esquemas.

2.1 Historia de los PDA

Antes de la aparición de los PDA en el mundo, existía el nutrido mercado de las agendas electrónicas. Sin seguir parámetros comunes, con diseños muy diversos, las distintas compañías ofrecían sus repertorios de modelos que podían variar desde calculadoras disfrazadas hasta artefactos más espectaculares que funcionales. A pesar de todo, el uso de estos aparatos y su comercio en general, siempre tuvo más pasado que futuro, pues el perfil de su destinatario estaba más tentado por los avances en las computadoras personales que por los adornos de estos inventos.

Es difícil precisar cual fue el primer PDA ya que muchos se adjudican ese logro. Sin embargo, entre los predecesores del PDA moderno se encuentran el Psion Organiser y el Sharp Wizard. Estos dispositivos tempranos, los cuales fueron intentos de computadoras portátiles, se originaron en la segunda mitad de la década de los ochenta. El Psion Organiser tenía un CPU de 8 bits y se podía programar en lenguaje ensamblador o en OPL. Estos dispositivos incluían un pequeño teclado como dispositivo de entrada, pequeña pantalla, y características básicas como calendario, alarma, calculadora y libreta telefónica. El Psion Organiser II, liberado en 1986, fue muy popular y vendió cerca de medio millón de ejemplares.

El término Personal Digital Assistant (PDA) fue acuñado en 1992 por John Sculley para dar nombre al producto Apple Newton el cual muchos autores consideran el primer PDA de la historia. Newton era una agenda electrónica que funcionaba con un lápiz (Stylus) con el cual se escribe en la pantalla y esta reconoce lo que la persona escribe. No tuvo mucho éxito por su alto costo y su tamaño que era de aproximadamente 20x10 cm. sin embargo ya proveía una solución para el reconocimiento de caracteres.

Fue hasta tres años más tarde que se lanzaron las PILOT 1000 y PILOT 5000 creadas por Jeff Hawkins para US Robotics, las cuales contaban con memoria de 256 KB y 512 KB respectivamente. Para este momento las PILOT no fueron muy famosas. Luego se lanzó al mercado la nueva línea de Palm Pilot la cual estaba compuesta por los modelos PERSONAL (500kb) y PROFESIONAL

(1mb y TCP/IP). A partir de estos modelos el mundo conoció las verdaderas capacidades de un asistente personal.

Las ventajas primordiales de las Palm Pilot en relación a la competencia fueron su sistema operativo de muy fácil uso (Palm OS 2.0), su pequeño tamaño, su ligero peso, larga duración de las baterías, el poder escribir en la pantalla sin necesidad de un teclado y algo muy importante que no tenían las agendas personales: comunicación directa con su computadora personal (HotSync Technology). En el caso de la Palm Pilot Professional esta logró la sincronización con el programa de correo electrónico de la PC para poder leer y escribir nuevos mensajes, esto es algo que ninguna agenda electrónica en el mercado podía hacer. Por esto fue que solo en los Estados Unidos se hayan vendido más de un millón de estos modelos.

Handspring fue un fabricante de PDA's bajo el sistema operativo Palm OS fundada por el inventor de la Palm Pilot Jeff Hawkins y los fundadores de Palm Computing Donna Dubinsky y Ed Colligan. Palm Computing se convirtió en una división de US Robotics la cual fue adquirida por 3Com. Cuando los fundadores se disgustaron por la forma en la que 3Com llevaba el rumbo de la empresa la abandonaron y fundaron Handspring en junio de 1998.

La empresa lanzó al mercado su línea de productos Handspring Visor en septiembre de 1999, los cuales a diferencia de la mayoría de los productos de Palm en ese tiempo, usó el puerto USB para sincronización con la PC e incluía un puerto de expansión. El soporte de USB hizo que fuera el primero de los productos de PDA capaces de trabajar con Macintosh. El puerto de expansión llamado Springboard Expansion Slot permitía la adición de módulos como juegos, eBooks, cámaras, memoria extra, etc. Sin embargo la principal aportación de Handspring fue el inicio de Treo, los llamados teléfonos inteligentes que hasta la fecha continúan teniendo éxito con Palm.



En el año 2000 Palm se separa de 3Com y se crea Palm Inc. la cual consistía en dos dependencias: Palm Source, una subsidiaria responsable del desarrollo y licenciamiento del Palm OS y el Palm Solutions Group, una empresa responsable de desarrollo, producción y comercialización de los dispositivos líderes en el mercado en ese entonces. En 2003 Handspring se fusionó con Palm Solutions Group (la división de hardware) para formar PalmOne. En 2005 PalmOne recupera el nombre de Palm.

A continuación se presentan algunos modelos y sistemas que distintas compañías se han encargado de desarrollar. Aunque no todos lograron llegar al mercado, algunos son pioneros de la tecnología que hoy se encuentra disponible.



Apple Newton

En lo que probablemente fuera uno de los proyectos tecnológicos más ambiciosos de nuestro tiempo, Apple dedicó centenares de millones de dólares a éstos dispositivos. Por desgracia, el Newton fue menospreciado por lo impreciso de su reconocimiento de escritura, y aunque mejoró en versiones posteriores, nunca llegó a estar a la altura de su reputación.

Magic Cap Sony

Magic Cap es un Sistema Operativo que utilizaba la metáfora de una oficina. Todos los elementos del usuario se disponían sobre una mesa, en forma de accesorios de escritorio, documentos y carpetas. Magic Cap era lento, sólo funcionaba en aparatos muy grandes (todavía mayores que el Newton) y era desproporcionadamente caro. Fue utilizado por los dispositivos de Sony HIX-300, PIC-1000 y PIC-2000 entre otros.



Los PDA con GEOS Casio



GEOS fue creado originalmente como interfase gráfico y sistema operativo para el Commodore 64 (Computadora personal que logró vender 30 millones de unidades entre 1982 y 1993). Era muy transportable y por eso fue convertido a muchos otros sistemas. Tandy y Casio decidieron unirse para fabricar un PDA llamado Zoomer, y encargaron la creación del software a quien más tarde sería precursor de Palm, Jeff Hawkins. El resultado fue interesante. Muchos apreciaron su interfase sencilla y rápida, pero era caro y ofrecía pocas ventajas respecto al Newton, más popular en ese tiempo. El Zoomer también carecía de reconocimiento de escritura, recurriendo a un teclado en pantalla.

Palm PC Microsoft

A medida que Palm iba ganando aceptación, Microsoft se resistía a ceder todo un mercado a una empresa recién llegada. Microsoft lanzó el Palm PC que funcionaba sobre Windows CE (Precursor de Windows Mobile). La interfase era parecida al de Windows 95, pero con unos iconos más grandes y simples. Era mucho más potente, ofrecía una vista de agenda, era ampliable y de precio asequible. No obstante, los dispositivos de Palm seguían siendo más baratos, y Microsoft no pudo estar presente en el mercado de los PDA sino hasta el lanzamiento del Pocket PC.



Motorola Marco

Al principio de la era del Newton, Apple cedió licencias del Newton OS a varios fabricantes, entre ellos Motorola. El clon del Newton era idéntico, salvo por el módem con teléfono móvil que incorporaba. El aparato era capaz de mandar y recibir faxes y mensajes de correo-e sin necesidad de cables.

VTECH Helio

El VTECH Helio fue lanzado en pleno apogeo de Palm y carecía de prestaciones que lo hicieran suficientemente atractivo. El Helio estaba equipado con un procesador muy rápido y contaba con una gran capacidad de almacenamiento. Su sistema operativo, el VTOS, era fiable y poseía una interfase elegante. VTECH abandonó el producto al cabo de tan sólo un año. Varios programadores lograron adaptar Linux y X11 al dispositivo



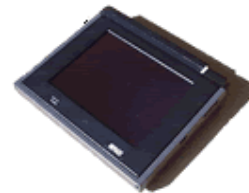
Royal LineaLX



El LineaLX de Royal era un aparato tan potente como sofisticado. Llevaba pantalla en color, ranuras de ampliación, módem incorporado y un sistema operativo basado en Linux. De hecho, fue ampliamente acogido como todo un triunfo de Linux, y contó con un gran respeto en el sector tecnológico. Pese a los elogios, el dispositivo nunca superó la fase de desarrollo, tras la cual fue archivado definitivamente.

GRiD PAD

GRiD llegó a comercializar varias portátiles y tablets³ que podían funcionar tanto con DOS como con su propio sistema operativo especial. Manufacturado por Samsung los tablets fueron adoptados por algunas grandes empresas y organismos oficiales, pero se vieron obstaculizados por su precio y su fragilidad. Propiamente el GRiD PAD no era un PDA sino una PC, sin embargo trabajaba también bajo el esquema del Stylus. GRiD PAD fue creada por Jeff Hawkins quien más tarde crearía las PILOT y fundaría Palm Inc.



HP 200LX



El HP 200LX era un prácticamente una portátil con pantalla en la tapa, que cabía en la palma de una mano y utilizaba una versión completa del sistema operativo MS-DOS. Posteriormente lograron adaptar Windows hasta su versión NT 4.0 Llegó a ser bastante popular, pero quedaba limitado por sus pocas posibilidades de sincronización con una PC.

Xircom REX

El Rex era un PDA diminuto, “embutido” en una tarjeta PC Card. Podía sincronizarse con la computadora por medio de un puerto PCMCIA/PC. EL REX no poseía reconocimiento de escritura, utilizaba un teclado POPUP. Para la sincronización utilizaba el TrueSync de Starfish, compañía que posteriormente sería adquirida por Motorola.



³ Tablet es una Laptop, la cual cuenta con una pantalla similar a los PDA que se utiliza como dispositivo de entrada y de salida, utilizando un stylus.

Cerca del año 2000 el mercado del PDA dio un primer salto y ganó una buena base de sus actuales usuarios. Los PDA de 2000 estaban muy lejos de los de hoy, aunque ya empezaban a resolver una parte importante de tareas: sincronización con datos desde el PC, compatibilidad con una buena variedad de documentos informáticos, agenda, calculadora, reloj, notas, portapapeles, contactos, escritura, etc. Con la llegada de las últimas tecnologías para teléfonos móviles y la posibilidad de incluir nuevas posibilidades a los PDA's, este invento se ha hecho muy popular. Las nuevas prestaciones (recursos inalámbricos, autonomía de la batería, resolución de pantalla, audio y vídeo, etc.) le han hecho al PDA tan novedoso como en sus días fueron las laptops: bonitos, potentes, portátiles y cada vez más baratos. El comprador y el posible comprador acaban viendo estos inventos más interesantes inclusive que sus equipos de escritorio.

Los modelos actuales de PDA permiten realizar una infinidad de actividades y existen miles de aplicaciones en los diferentes ambientes profesionales del hombre moderno.

Los PDA en la Actualidad

Los modernos PDA tienen como mínimo las siguientes características:

- Pantalla a color
- Interfaz de usuario amigable e intuitiva (fácil de aprender)
- Stylus como dispositivo de entrada
- Software estándar (calendario, agenda, directorio, etc)
- Sonido
- No menos de 16MB de memoria
- Conectividad inalámbrica (IR, Bluetooth, WiFi)
- Enlace con la PC
- Gran cantidad de software el cual puede ser instalado vía PC
- Puertos de expansión (para memoria extra, adjuntar otros dispositivos o para respaldar)
- Algunos, soporte para conexión a Internet por medio de un teléfono celular

PDA con conexión a redes móviles

Ha habido varios dispositivos capaces de trabajar con redes móviles (son PDA's pero ofrecen servicios de un teléfono móvil). El primer éxito fue de Nokia con el modelo 9110.

El siguiente real contendiente fue el XDA disponible en el Reino Unido en la red móvil de O2. Otro diseño similar es Blackberry de Research In Motion la cual es líder en servicios de correo electrónico por medio de red de telefonía móvil.

Smartphones

Los smartphones o teléfonos inteligentes son dispositivos muy similares a los PDA con funciones de red, solo que los smartphones son diseñados con el propósito principal de ser teléfonos con funciones subordinadas de PDA. Por ejemplo



Sony Ericsson P800 y P900, Nokia 6600, Motorola MPx200, Siemens MPx200 y otros nuevos teléfonos con tecnología 3G como Motorola A835 y NEC e808. Bajo el Palm OS se han destacado los Treo creados por Handspring y perfeccionados en PalmOne y Palm más tarde. Hoy Treo ofrece smartphones bajos las dos plataformas más populares: Palm OS y Windows Mobile.

Ya era un gran avance el que se llevaba a finales del siglo XX, sin embargo, durante estos primeros años ha habido un gran desarrollo de tecnologías de comunicación que se han integrado a los PDA haciéndolas a su vez las poderosas herramientas que conocemos. Por otro lado el desarrollo de teléfonos inteligentes cada vez se unifica más con el de los PDA, tan es así que se utilizan los mismos sistemas operativos y cada vez se desarrollan más dispositivos de este tipo.

Las funciones que prácticamente se vuelven básicas en los PDA son: agenda, reproductor multimedia, grabadora de sonidos, manejo y almacenamiento de archivos, tecnologías Wifi y bluetooth. Entre los dispositivos más sobresalientes se encuentran aquellos que adaptan estas funciones y además proporcionan servicios de telefonía o adaptan al PDA accesorios como cámara fotográfica o almacenamiento en disco duro.

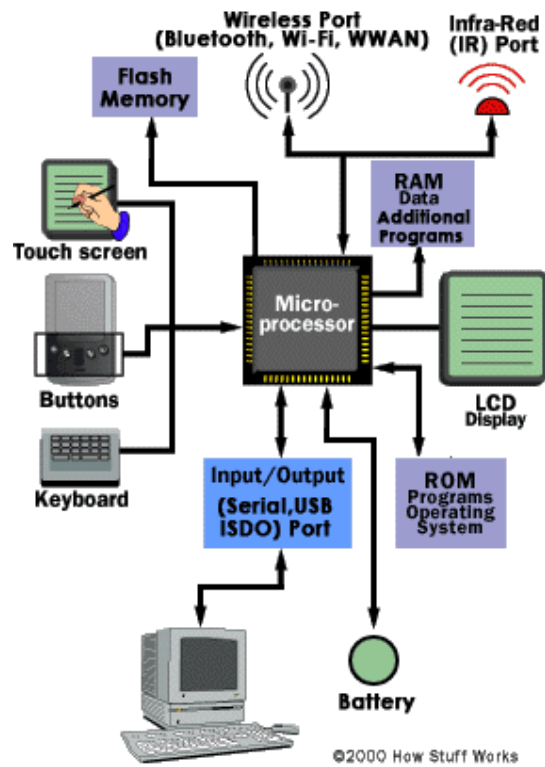
2.2 Funcionamiento de los PDA

2.2.1 Hardware

Microprocesadores y memoria

Como las PCs y laptops, el poder de los PDA se centra en un microprocesador. El microprocesador es el cerebro del PDA y este coordina todas las funciones del PDA de acuerdo con las instrucciones programadas. A diferencia de las laptops y PC, los PDA utilizan microprocesadores más pequeños y económicos. Aunque estos microprocesadores tienden a ser más lentos que los de las PC, son adecuados para las tareas que un PDA realiza. Cabe destacar a los procesadores Dragon Ball VZ de Motorola, y a otros fabricantes como Intel o Freescale.

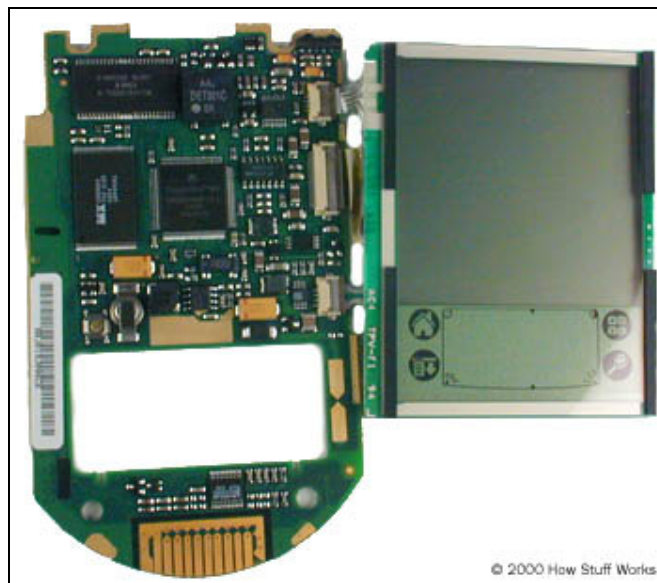
Un PDA no tiene disco duro (a excepción de LifeDrive de Palm). Esta almacena programas básicos (sistema operativo, tareas, calendario) en un chip de memoria de solo lectura ROM, la cual se mantiene intacta aun cuando la PDA se reinicia. Los datos y programas que se añaden por parte del usuario se almacenan en la memoria RAM del dispositivo. Debido a su diseño, los PDA



mantienen a salvo los datos en una RAM ya que siempre hay una pequeña cantidad de energía suministrándola de las baterías.

Los PDA menos poderosos tienen menor capacidad en RAM. Sin embargo, muchas aplicaciones toman un espacio significativo de memoria. Debido a esto los modelos actuales poseen mayor capacidad de memoria. También los dispositivos Pocket PC requieren mayores recursos y tienen aun más memoria RAM. Para proveer de mayor capacidad de memoria los PDA modernos utilizan puertos de expansión para memoria flash removible. Son muy útiles para el almacenamiento de archivos.

Algunos de los PDA modernos en lugar de usar memoria RAM, utilizan memoria Flash. La memoria flash es no volátil, lo que significa que las aplicaciones y datos que se almacenen se conservaran aun cuando la batería sea completamente drenada.



Vista interna de un PDA de Palm. En medio de la tableta se encuentra el microprocesador y a la izquierda y arriba los chips de memoria.

Baterías

Los PDA son alimentados por baterías. Los modelos más antiguos utilizaban baterías alcalinas, sin embargo ahora se utilizan baterías recargables de diversos materiales (litio, níquel cadmio, níquel e hidruro metálico) la vida de la batería depende del tipo de PDA que se tenga y de cómo se use. El drenado de la batería depende de los componentes del PDA que más utilizan energía los cuales son:

- El sistema operativo (la energía depende de los requerimientos de memoria)
- Más memoria
- Las conexiones inalámbricas como Wi-Fi o Bluetooth
- El uso de la pantalla.

La carga de las baterías varía de horas a meses, dependiendo del modelo de PDA y sus características. Algunos modelos poseen sistemas de administración de energía para extender el tiempo de uso de las baterías. Aún

cuando las baterías estén muy bajas usualmente hay suficiente energía para mantener la memoria RAM.

Si se descarga por completo la batería o se remueve, la mayoría de los dispositivos tiene una batería interna de respaldo que provee un corto tiempo de energía (normalmente 30 minutos o menos) hasta que se instale un repuesto. Si se descargan todas las fuentes de energía de un PDA se pierde toda la información en RAM. Esto hace extremadamente importante el respaldo o la sincronización con una PC.

Además de la batería muchos PDA tienen adaptadores de corriente AC.

Pantalla

Los PDA utilizan una pantalla de cristal líquido (LCD). A diferencia de las pantallas LCD para laptops o PC's las cuales son utilizadas únicamente como dispositivos de salida, los PDA usan su pantalla como entrada y salida de datos.

Las características de una pantalla de PDA son las siguientes:

- LCD para uso de entrada y salida
- Diferentes resoluciones en pixeles con altas resoluciones para mejor calidad
- Pantalla a color
- Luz de fondo para lectura en lugares poco iluminados

Entrada de Datos

Los PDA varían en la forma en como adquiere datos y comandos. Algunos dispositivos usan un stylus y una pantalla sensible en combinación con un programa de reconocimiento de escritura. Usando un stylus de plástico se dibujan caracteres sobre la pantalla o sobre el área dedicada para escritura. El software dentro del PDA convierte los caracteres a letras o números. En los dispositivos de Palm, este software se llama Graffiti. Graffiti requiere que cada letra se registre de cierta manera y se debe utilizar un alfabeto especial. Por ejemplo, para escribir una A solo se dibuja el arco, es decir, una especie de V al revés. Para ayudar a Graffiti a tener resultados más precisos se deben dibujar las letras en una parte de la pantalla y los números en otra.

Los PDA Pocket PC ofrecen tres formas de reconocimiento de escritura: transcriptor, reconocedor de letras y reconocedor de bloque. Los reconocedores son similares a graffiti y requieren un alfabeto especial. En contraste, el transcriptor reconoce la escritura regular siempre y cuando se sea muy legible.

El uso de la escritura en el PDA muchas veces es desdeñado por el uso de un pequeño teclado en pantalla. Se ve como un teclado regular excepto que las letras se "teclean" por medio del stylus. Algunos dispositivos incluyen un pequeño teclado QWERTY empotrado en la carcasa y requieren de un tecleo con los dedos. Se puede usar un teclado de talla normal conectándolo vía bluetooth, IR o USB. Cada modelo cuenta con diferentes botones de navegación para acceder rápidamente a algunas aplicaciones o usar como cursores.

2.2.2 Software

El Sistema operativo

El sistema operativo contiene las instrucciones preprogramadas para el microprocesador. Los sistemas operativos utilizados por los PDA no son tan complejos como los usados en las PC. Contienen menos instrucciones que requieren por tanto, menos memoria. En el siguiente apartado se profundizará al respecto mencionando los principales Sistemas Operativos para PDA.

Aplicaciones

Existe una gran cantidad de aplicaciones disponibles para PDA en sus diferentes sistemas operativos. Existen desde aplicaciones simples (bloc de notas, agendas) hasta complejas y modernas (localizadores GPS) pasando por bases de datos, juegos, aplicaciones multimedia, herramientas de comunicación e Internet entre otros rubros. El diseño de éstas aplicaciones debe ser proyectado en función de las capacidades del hardware, sin embargo, esto no es limitante para generar poderosas aplicaciones.

2.3 *Sistemas Operativos para PDA*

En esta sección haremos un análisis de las alternativas que se presentan en los PDA respecto a los Sistemas Operativos. Hay un gran número de ellos, sin embargo muchos no pasan de la etapa de desarrollo y el mercado es dominado por 2 plataformas: Palm OS y Windows CE (Windows Mobile). A continuación los veremos a detalle.

2.3.1 Windows Mobile

Windows Mobile (antes Windows CE) es un sistema operativo de Microsoft principalmente para PDA. La ideología ha sido hacerlo tan similar con Windows como sea posible. Las API's son tan grandes como las usadas en la programación tradicional de Windows. La razón de esto es permitir la fácil adaptación para los programadores de aplicaciones que tienen experiencia en Windows para PC y llevar hacia las aplicaciones existentes a los sistemas móviles.

Existe un mecanismo del control de acceso en Pocket PC que permite que los usuarios bloqueen los dispositivos. Para tener acceso al dispositivo bloqueado se debe proporcionar un NIP. Esto aplica para tener acceso al dispositivo a través de su propia interfaz de usuario así como para conectarlo con una PC. En otras palabras se identifica al usuario antes de autorizar el acceso a la información en el dispositivo, lo cual brinda confidencialidad e integridad.

Windows Mobile tiene una especie de sistema de ficheros jerárquico similar al utilizado en la plataforma Windows de la PC. Sin embargo, el sistema operativo es monousuario, por tanto no hay ningún control de acceso de archivos basado

en identidad de usuario. Además, los sistemas de ficheros no conceden las directivas de acceso para cada módulo individualmente. Windows Mobile puede conceder ciertos permisos a nivel de programa en vez de solamente a nivel de usuario.

Windows Mobile no soporta archivos que cifran y descifran a nivel del sistema operativo. Sin embargo, los programadores se proporcionan medios para crear software con criptografía. Hay API's para el cifrado y usan protocolos de seguridad igualmente.

Windows Mobile soporta las versiones 2,0 y 3,0 de Secure Socket Layer (SSL) que se pueden utilizar para las conexiones de red seguras.

En Windows Mobile, el manejo de la memoria es basado en memoria virtual y el sistema hace uso por completo de un MMU (Memory Management Unit). Esto significa en la práctica que los programas están funcionando en áreas de memoria protegidas. Porque cada aplicación utiliza sus propias direcciones virtuales y no pueden tener acceso a otras áreas de memoria.

2.3.2 Symbian OS

Symbian OS era conocido previamente como EPOC. El sistema operativo Symbian fue desarrollado inicialmente por Psion para sus propios dispositivos PDA. Principalmente es utilizado para telefonía y Smartphones y se encuentra en dispositivos de Ericsson, Motorola, Nokia, Panasonic y Psion.

Symbian promueve normas para la inter-operación de teléfonos móviles que permiten el manejo de datos por medio de redes móviles, uso de contenidos y servicios. Aporta un sistema avanzado, abierto y estandarizado de operaciones de servicios inalámbricos. Symbian OS es flexible y bastante escalable para ser usado en una gran variedad de teléfonos móviles.

2.3.3 Sistemas basados en Unix

Linux y Embedded xBSD, sistemas basados en Unix, se encuentran también como los sistemas operativos disponibles para los dispositivos PDA. Sin embargo, la configuración, la instalación e incluso el uso de tales dispositivos requiere un cierto conocimiento sobre sistemas operativos y principalmente de Linux. La mayoría de los usuarios de PDA no tienen el conocimiento y la pasión necesaria para operarlos. Por el contrario, quisieran que los dispositivos fueran tan fáciles de utilizar como sea posible.

El conocido grupo nipón Sharp lleva tiempo ofreciendo una gama de PDA's bajo sistemas basados en Linux llamados Zaurus. En principio estos PDA's soportan C y Perl como lenguajes de programación. Estos dispositivos son exitosos pero solo en el mercado local.

2.3.4 Palm OS

La ideología del Palm OS es que sea simple de utilizar y tan abierta como sea posible. En el diseño, el Palm OS no trata de hacer una réplica de una PC o de

cualquier tipo de plataforma. La meta ha sido hacer un PDA que sea fácil de utilizar y que tenga solo las aplicaciones necesarias. Palm ha mostrado que se puede tener un fácil acceso a la información en un corto tiempo, en comparación con el tiempo de acceso a información en una PC. Por otro lado Palm tiene un fuerte soporte, de hecho, hay centenares de aplicaciones disponibles para la plataforma. Así también, hay varias empresas fabricantes de hardware que fabrican diferentes dispositivos bajo Palm OS.

Palm OS fue desarrollado inicialmente para dispositivos con recursos restringidos. La idea era hacer dispositivos lo suficientemente baratos de modo que lo comprara tanta gente como fuera posible. Por lo tanto, el sistema operativo tuvo que ser diseñado para hacer frente a memoria y a procesamiento limitados.

Las últimas versiones de este Sistema Operativo son la Palm OS 5.4 también conocido como Garnet y el Palm OS 6.0 también conocido como Cobalt, el cual no se ha dado a conocer en ningún dispositivo en el mercado.

Control de acceso

Desde la versión 4.0 de Palm OS esta plataforma soporta el bloqueo para el PDA. Se necesita una contraseña para tener acceso al dispositivo. Después de incorporar la contraseña, el usuario puede tener acceso a todos los datos en el dispositivo. Es decir, Palm OS tiene un esquema muy simple de identificación y de autorización de usuario basado en identidad. Debido a esto, soporta solamente un usuario por lo que no se requiere un nombre de usuario para tener acceso al dispositivo, solamente contraseña. La identificación y la autorización protegen confidencialidad e integridad a cierto grado, porque los datos no se pueden leer o modificar si el usuario no ha sido identificado y por consiguiente autorizado

Cifrado

Palm OS soporta datos encriptados y los mantiene fácilmente accesibles. Por otra parte, hay varios productos de terceros que apoyan el cifrado disponible para versiones anteriores del sistema operativo. El cifrado de datos ayuda a asegurar datos confidenciales si alguien pudiese extraer información en bruto del dispositivo. Esta característica es especialmente importante, si las tarjetas de memoria o una cierta otra clase de almacenaje externo o removible se utilizan. Además, los dispositivos son vaciados típicamente al disco duro de la PC durante el proceso de la sincronización. El cifrado protege la información si alguien es capaz de acceder a los datos de respaldo que se almacenaron en la PC.

Sistema de Archivos

Palm OS no obliga cumplir los permisos de acceso para los archivos, de hecho, no tiene un sistema de ficheros jerárquico. Tanto los datos de las aplicaciones como los mismos ejecutables se almacenan como bases de datos supuestas del Palm OS en la memoria RAM. Las bases de datos son identificadas por una ID y el nombre de la base de datos. La ID identifica normalmente al creador del ejecutable que a su vez creó la base de datos. En

la práctica, sin embargo, cualquier aplicación puede abrir bases de datos con cualquier creador. Así que no hay control de permisos en los módulos del software en el sistema operativo.

Administración de memoria

En Palm OS, la memoria RAM es utilizada como memoria en tiempo de ejecución para aplicaciones y como almacenamiento permanente para los ejecutables y los datos. En el arranque de la aplicación, el programa no se carga a otra clase de memoria, sino que se ejecuta directamente de donde se almacena. No se utiliza ninguna memoria virtual. Todas las aplicaciones utilizan la misma área de memoria de modo que el acceso a memoria no está protegido contra los desbordamientos del buffer. De hecho, todo se almacena en la misma memoria, a excepción de bases de datos del sistema operativo, tales como los códigos binarios de aplicaciones estándar. Estos se almacenan en la ROM o la memoria flash dependiendo del hardware. Aunque esa memoria no puede ser escrita, puede accederse a ella con cualquier aplicación. Existe software que se puede utilizar para emitir las aplicaciones que se almacenan en la memoria ROM a otro dispositivo con Palm OS.

2.3.4.1 Historia del Palm OS

Originalmente Palm OS fue desarrollado por Jeff Hawkins para su uso en la exitosa PDA Pilot de US Robotics. La versión 1.0 fue comercializada con la Pilot 1000 y Pilot 5000 y la versión 2.0 la introdujo la PalmPilot Personal y Profesional. Con el desarrollo de la serie III de Palm, la versión 3.0 de Palm OS fue comercializada y ocurrieron grandes cambios incrementales con las versiones 3.1, 3.3 y 3.5 agregando soporte para color, múltiples puertos de expansión, nuevos procesadores entre otras cosas. La versión 4.0 se libera con la serie m500 de Palm y más tarde se convierte en una actualización para los dispositivos más antiguos. Esta versión adiciona una interfase estándar para el acceso a archivos externos (como Tarjetas SD) y mejora las bibliotecas para telefonía, seguridad.

Las versiones de Palm OS de la 1.0 a la 4.1 estaban basadas en un pequeño kernel desarrollado por Kadak. Estas versiones eran técnicamente capaces de un procesamiento multitareas, sin embargo había limitantes impuestas por el desarrollador del kernel.

La versión 5.0 fue liberada con la Tungsten T y fue la primer versión liberada para dar soporte a dispositivos ARM (Advanced RISC Machine). Las aplicaciones de Palm son ejecutadas en ambiente emulado llamado Ambiente de Compatibilidad de Aplicaciones Palm (PACE Palm Application Compatibility Environment) lo que hace a estos dispositivos capaces de ejecutar software escrito para versiones anteriores. Aun con lo complejo del PACE, las aplicaciones de Palm corren más rápido sobre dispositivos ARM que sobre las generaciones previas de hardware. El nuevo software toma la ventaja de los procesadores ARM, con pequeñas unidades de código también conocidas como "ARMlets". Fue en este tiempo cuando Palm comenzó a separar su hardware y su plataforma convirtiéndose en dos empresas PalmSource, Inc (Sistema Operativo) y PalmOne (hardware, ahora llamada Palm, Inc.)

La versión 5.4 agregó el Sistema de Archivos No Volátil (NVFS por sus siglas en inglés) y se utilizó el almacenamiento en memorias tipo Flash en lugar de DRAM previniendo la pérdida de información durante el drenado de la batería. Por varios años PalmSource desarrolló un moderno sucesor para el Palm OS 5. La licencia de Palm OS 6 fue liberada en enero de 2004. Permitía aplicaciones nativas de ARM con mejorado soporte multimedia. Sin embargo, ninguno de los desarrolladores de hardware puso a la venta algún dispositivo con esta versión. Esto fue el llamado Palm OS 6 Cobalt.

En septiembre de 2005 PalmSource anuncio que fue adquirida por ACCESS. Mas tarde ese mismo mes, Palm anunció su nueva Palm Treo 700w bajo Windows Mobile, señalando la diversificación de Palm en dos sistemas operativos.

El 24 de enero de 2007 ACCESS anunció un nuevo nombre para el actual Palm OS Garnet, el cual ahora es llamado Garnet OS.

Las aplicaciones más comunes que se incluyen en el Palm OS son Address book (Contactos), un Navegador que bien puede ser Blazer o Web Pro, Calc (Calculadora), Date book (Calendario), Expense(gastos), Hotsync (Aplicación para sincronizar con la PC), Memo Pad (Notas), Note pad (notas rápidas con escritura a mano), To do list (tareas). Modelos más recientes incluyen Photos (multimedia) e VoiceRecorder (Notas de voz). Para realizar ajustes y cambios en la configuración se utiliza Preferences (Preferencias).

Existen muchas aplicaciones que pueden ser instaladas en dispositivos Palm OS. Para agosto de 2005 había más de 20000 aplicaciones disponibles para Palm OS desarrolladas por terceros con licencias desde open-source hasta aplicaciones completamente comerciales.

2.3.4.2 Los dispositivos móviles que utilizan Palm OS

Existen diversos dispositivos que cuentan con la licencia de Palm OS. El enfoque de la mayoría de fabricantes es proveer soluciones de tipo ejecutivo o personal, sin embargo, debido a las características con las que cuentan, es posible diseñar aplicaciones que aprovechen los recursos de hardware que proporcionan los fabricantes y los estándares de tecnologías de información que permiten una mayor compatibilidad entre dispositivos de diferentes tipos.

Entre los dispositivos que destacan en el presente se encuentra el llamado LifeDrive de Palm. Es el primer dispositivo PDA en el mercado que integra un disco duro. La capacidad del disco duro es de 4 GB además de contar con los estándares inalámbricos bluetooth y wi-fi. Por otro lado Palm cuenta con una gama de PDA entre las que destaca su línea Tungsten la cual está enfocada al ambiente ejecutivo ofreciendo estándares de tecnología móvil como bluetooth o wi-fi además del manejo de ofimática.

Recientemente, Motorola adquirió a la compañía Symbol Technologies la cual es una de las principales desarrolladoras de PDA empresariales e industriales. Symbol tuvo la certeza de desarrollar dispositivos tanto para Windows Mobile como para Palm OS. Entre los PDA con Palm OS se encuentran los modelos SPT1550 y SPT1800 los cuales son PDA de tipo empresarial. El SPT1550 cuenta con un scanner básico de código de barras, un puerto serial RS-232 y comunicaciones por puerto infrarrojo. El modelo





SPT1800 cuenta con un scanner mucho más sofisticado, puerto infrarrojo, estándar de comunicaciones wi-fi (IEEE 802.11b) y está construido bajo las normas ambientales IP54⁴ lo que sugiere una mayor resistencia y duración al trato del dispositivo. Bajo esta misma línea de PDA empresariales se encuentra Medzura de la empresa Aceeca. Este dispositivo cuenta con un puerto de expansión de memoria Compact Flash, puerto infrarrojo y está construido bajo las normas ambientales IP67⁵. Es el primer dispositivo PDA a prueba de agua.

La compañía Garmin produce dispositivos con diferentes tareas y aplicaciones en diversos campos. Entre ellos destaca el PDA iQue3000 el cual integra un completo posicionador global GPS. Además de las ventajas que aporta el Palm OS, este dispositivo cuenta con un puerto de expansión para memoria micro SD. Una función que llama la atención es que la agenda y la libreta de contactos del Palm OS se sincronizan con el posicionador GPS de tal manera que se puede manejar de mejor manera un itinerario.



La compañía norteamericana Janam también ha diseñado dispositivos PDA de carácter industrial como los modelos XP20 y XP30 los cuales están planteados para trabajo rudo.

El XP30 de Janam cuenta con conectividades bluetooth, wi-fi, y puerto infrarrojo, cuenta con teclado numérico y aunque aparenta ser un teléfono, se trata de un PDA muy funcional.

Estos solo son algunos ejemplos de dispositivos, algunos de carácter ejecutivo, algunos otros de carácter industrial. Cabe destacar las características con las que cuentan para determinar los campos en los cuales podrían aportar mejores soluciones.

Aunque con un costo un poco más elevado los teléfonos inteligentes ganan terreno en el mercado de los PDA. Los teléfonos inteligentes Treo, desde su desarrollo realizado por Handspring hasta la fecha producidos por Palm son los smartphones más sobresalientes bajo la plataforma Palm OS. Como ejemplo existen los modelos Treo 650 y el llamado de bajo costo Treo 680. Este último más reciente y bajo la misma plataforma ahora llamada Garnet OS por ACCESS. Además de integrarse a la red telefónica ofrecen servicios de correo electrónico, ofimática y reproductor multimedia.



Por otro lado, se pueden encontrar otros smartphones bajo Palm OS a nivel mundial como PiTech de iQool, varios modelos de Samsung como el SPH-I500SS o el SPH-I550.

Fundamentalmente los teléfonos inteligentes darán mucho de que hablar en el ambiente ejecutivo, sin embargo en otros campos los PDA pueden aportar más y adaptarse mejor.

En definitiva, el sistema operativo Palm OS es una referencia fundamental en el estudio de los dispositivos móviles, sobre todo PDA's. Cabe destacar el trabajo

⁴ Protegido contra viento, polvo y lluvia

⁵ Protegido contra polvo y a prueba de agua (inmersiones de entre 15 cm y 1 m)

desarrollado hasta la fecha por desarrolladores de hardware y software, para hacer de esta plataforma una de las mejores a nivel usuario. En el ambiente profesional muchos consideran la gran facilidad de manejo del Palm OS, la cual evita grandes gastos en la capacitación de los empleados. Por otro lado la compatibilidad con las plataformas de escritorio, la portabilidad y el valor estratégico para la recolección de información son ventajas que han otorgado un gran crecimiento económico a Palm, pero sobre todo que han dado prestigio y fama al sistema operativo. Sin duda habrá mucho más de que hablar.

Capítulo III

Herramientas de Desarrollo en Palm OS

Para la programación en dispositivos móviles existen herramientas muy similares a las utilizadas en la programación común. A continuación se presentará este enfoque de programación, los diferentes conceptos que se manejan en el desarrollo de aplicaciones y se determinarán los entornos de desarrollo en los que se trabajará en el siguiente capítulo.

3.1 La Programación en dispositivos móviles

Dentro de los tipos de programación utilizados en dispositivos móviles sobresale la programación orientada a objetos gracias a la posibilidad de reutilizar objetos en el desarrollo aplicaciones y a la existencia de entornos de programación que utilizan estos objetos y los adaptan a diversas plataformas. Para la construcción de aplicaciones móviles se necesitan crear tres grandes procesos:

- El cliente móvil. Este debe cumplir, en lo posible, con los requerimientos del cliente en cuanto a procesos en el dispositivo móvil.
- La aplicación que es la que procesa el trabajo de los clientes móviles.
- La sincronización. Este es el mecanismo por el cual se mantiene la consistencia e integridad de datos entre los clientes móviles y la aplicación de servidor.

En cuanto a métodos se refiere existen básicamente dos maneras de programar en los dispositivos móviles. Una es desarrollando las aplicaciones en un equipo PC de escritorio, haciendo las pruebas pertinentes en emuladores o simuladores y exportando las aplicaciones ya concluidas al dispositivo. La otra manera es desarrollando las aplicaciones directamente sobre el dispositivo móvil con herramientas llamadas "on-board", compilarlas y ejecutarlas sobre la misma plataforma móvil.

Para Palm OS no es la excepción, existen herramientas para desarrollar aplicaciones bajo los dos esquemas mencionados. Lo más recomendable es desarrollar las aplicaciones en un equipo de escritorio y probar las aplicaciones en el emulador de Palm OS(Palm OS Emulator) o simuladores de la plataforma a fin de evitar los riesgos que implica el probar las aplicaciones directamente sobre el PDA. Esta manera de desarrollar software es la que ha tenido más progreso y éxito, sin embargo la programación directa sobre el PDA también es interesante y existen un gran número de herramientas para practicarla.

Las principales herramientas para el desarrollo de aplicaciones móviles son clasificadas como herramientas RAD ya que se ayudan de elementos visuales para el diseño y comprenden un desarrollo interactivo con la interfaz de usuario.

A continuación se presentan conceptos básicos del Palm OS necesarios para proceder al diseño de aplicaciones.

3.1.1 Conductos (Conduits)

Un conducto es un módulo que provee un enlace entre una aplicación ejecutándose en un dispositivo bajo la Plataforma Palm OS y una particular aplicación en un equipo de escritorio. Un conducto es responsable de los datos de la aplicación durante la sincronización entre el PDA y la computadora de escritorio. El conducto requiere:

- Abrir y cerrar las bases de datos en el dispositivo Palm
- Determinar que datos serán solo cargados, sólo descargados o una combinación de datos proveniente de ambos dispositivos.
- Agregar, borrar o modificar apropiadamente los registros en el PDA y en la PC
- Ser capaz de trabajar con un ambiente multiusuario donde más de un PDA puede sincronizarse a la misma red o PC (aunque no necesariamente al mismo tiempo).
- Convertir los datos en los registros de la base de datos de la aplicación en el PDA a las adecuadas estructuras de datos de la computadora de escritorio.
- Opcionalmente, aunque es recomendado, comparar registros para sincronizar solo aquellos que hayan sido modificados.

3.1.2 La Sincronización

La sincronización es el proceso por el cual se lleva a cabo un intercambio o complemento de información entre un dispositivo PDA y una PC. El programa que transfiere la información entre el PDA de Palm y una PC se llama HotSync Manager. Este programa utiliza los conductos para obtener las instrucciones sobre como intercambiar información entre los componentes. Gracias al Hotsync Manager se pueden tener cuatro opciones de sincronización: de la Palm a la PC, de la PC a la Palm, hacia ambas direcciones o no realizar intercambio de información.

La sincronización se hace por medio de protocolos, donde ambos componentes "conocen" la forma de intercambio de datos.

La importancia de la sincronización radica en el mantenimiento de la confiabilidad e integridad de la información.

Entre los protocolos se encuentran los siguientes:

Slow sync : En términos generales cuando se va a sincronizar con este protocolo el PDA transfiere todos los datos hacia la PC y compara uno a uno los registros para ver cuales han sufrido cambios, si se han borrado o han sido adicionado nuevos registros. Entonces cada uno de los datos existentes en la Base de datos o archivo del PDA pasa a sincronizar a la PC, por eso se denomina slow sync.

Un ejemplo de datos puede ser la sincronización de pedidos, los cuales se requieren que pasen todos para su posterior facturación.



Fast sync : Este protocolo de sincronización es mucho más rápido que el anterior y se basa en que sólo pasan al servidor los datos que han tenido modificaciones, en otras palabras, los datos que no han sufrido cambios, borrados o inserción de nuevos registros no se sincronizan.



Un ejemplo típico de sincronización con este protocolo es el mantenimiento de una agenda de citas, donde no se interesa sincronizar las citas que no han sufrido modificaciones.

Este protocolo de sincronización utiliza una "bandera" de estado que permite "marcar" los registros que han sufrido modificaciones desde la última fecha de sincronización. Cuando se sincronizan los datos de la PC y el PDA pueden ocurrir los siguientes eventos:

Inserción: el registro nuevo en la PDA es insertado al servidor

Modificación : el registro de la PDA reemplaza al del servidor.

Borrado : el Registro en el servidor es borrado

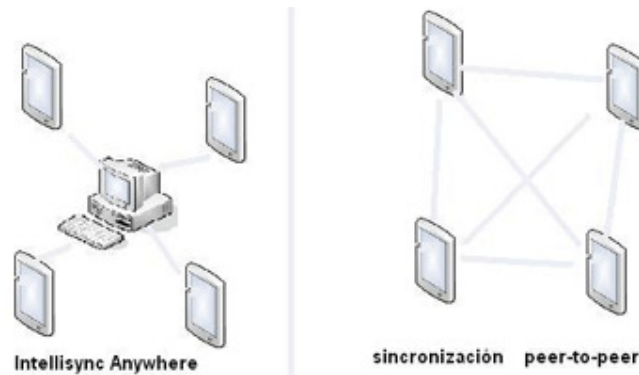
Guardado : el Registro se guarda en el servidor.

Cuando termina la sincronización las "banderas" de estado se anulan o regresan a su estado inicial (Reset) y se guarda la fecha de sincronización.

El Protocolo Intellisync anywhere está dedicado para la sincronización en red de diferentes dispositivos, donde los dispositivos móviles se sincronizan con un servidor central.

Además de los dos protocolos anteriores existe otro llamado SyncML (Synchronous Markup Language). El objetivo de este protocolo es la sincronización "abierta" entre dispositivos de diferentes familias. Varias compañías unen esfuerzos para que sus equipos trabajen con este protocolo de sincronización. Se puede decir que se busca un protocolo de sincronización universal.

Este protocolo asume que cada dispositivo guarda banderas de estado de los registros de los demás. Por tanto, necesita bastante capacidad de almacenamiento para mantener la integridad de la información y las banderas de estado.

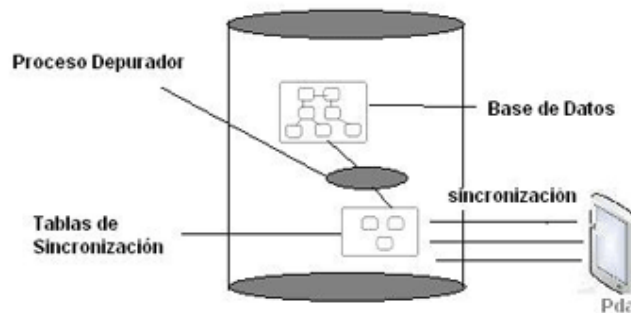


Peer-To-Peer: es otro protocolo de sincronización donde todos y cada uno de los dispositivos se sincronizan sin la necesidad de una computadora central o servidor como ocurre en el caso del protocolo Intellisync anywhere.

No está demás hacer énfasis en que la sincronización es un proceso diferente al de transmisión de información.

En algunas aplicaciones no se sincroniza directamente sobre la base de datos. Para este fin se utilizan tablas temporales donde se guarda la información y posteriormente con un proceso se puede depurar y hacer correcciones.

Lo anterior se realiza por motivos de seguridad.



Un ejemplo de esto es la sincronización en un sistema de pedidos donde lo que se necesita es descargar los pedidos y que el proceso de facturación se encargue del resto.

Claro está que se puede sincronizar directamente sobre la base de datos y hacer la depuración en el proceso de sincronización

3.1.3 Tipos de archivo en Palm OS

Para que las aplicaciones puedan ejecutarse en el Palm OS, éstas deben ser presentadas al sistema operativo por medio de archivos .prc y .pdb:

PRC.- Los archivos .prc son aplicaciones de Palm OS, mismas que se pueden instalar y ejecutar.

PDB.- Los archivos .pdb son archivos de base de datos de Palm. Trabajan específicamente con las aplicaciones (.prc) para los que fueron creados.

En Palm OS existen aplicaciones capaces de ejecutar una gran cantidad de archivos, por mencionar algunos: txt, html, xls, doc, jpg, bmp, gif, mp3, aac, zip, entre otros, cuyos formatos son los más conocidos por los usuarios de computadoras de escritorio y que cada vez adquieren universalidad debido a su compatibilidad con nuevos dispositivos electrónicos. Sin embargo, la ejecución

y el manejo de éstos archivos en Palm OS siempre se realiza por medio de una aplicación (archivos .prc).

3.2 El diseño de aplicaciones

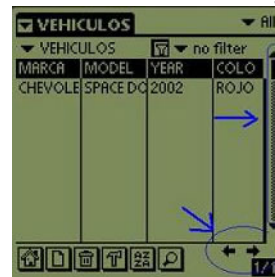
A continuación se presentan las consideraciones necesarias para el diseño de aplicaciones en el Palm OS.

3.2.1 Diseño de interfaces de usuario

Es muy importante señalar que en el proceso de diseño de las pantallas de usuario se debe considerar el tamaño de la ventana para la que se esta diseñando. Es necesario ser un poco recursivo para que los usuarios finales tengan comodidad y agilidad al momento de trabajar con la aplicación móvil. En lo posible se debe tratar de cumplir con los requerimientos de funcionalidad y diseño esperados por el cliente. También es necesario llegar a acuerdos con el cliente en cuanto al numero de campos, desplazamientos, eventos y carga de proceso a tener en el dispositivo móvil.

Las ventanas de los dispositivos móviles poseen barras de desplazamiento vertical y horizontal, pero estos desplazamientos pueden resultar incómodos y poco prácticos para el usuario final.

Otro aspecto importante es definir el modelo de conexión que tendrá el dispositivo móvil con el servidor.



3.2.2 Modelos de Conexión

Con el paso del tiempo y el desarrollo de la tecnología, cada día es más fácil construir y configurar una red de área local, más aún si es una red inalámbrica. Entre las grandes ventajas que ofrecen este tipo de redes es su bajo costo y la movilidad. Sin embargo, para implementar una solución móvil no necesariamente se debe construir sobre una red inalámbrica. A continuación se presentan los tres principales modelos de conexión para el trabajo con dispositivos móviles.

Trabajo bajo el modelo desconectado

Este modelo es muy utilizado actualmente. Consiste en que el PDA realiza la captura y algunos procesos básicos como validaciones y cálculos. Después de tener los datos, éstos se sincronizan con el servidor donde se realizan los grandes procesos. Se denomina desconectado porque el PDA contiene la información necesaria para realizar los procesos básicos que se le han programado.

Un ejemplo de este modelo es un vendedor que visita a sus clientes para tomarles el pedido y al final del día o de su jornada se sincroniza con el servidor donde se realiza el proceso de facturación.

El vendedor solo necesita en el PDA la información de los productos, precios y la información de los clientes. Entonces el PDA se comunica con el servidor después de un determinado tiempo donde se sincroniza y todo se procesa.

Trabajo bajo el modelo Semi-conectado.

En el modelo semi-conectado el PDA tiene todo lo necesario para realizar un proceso básico como en el modelo desconectado pero la diferencia radica en que aquí no se dispone de la conexión al servidor hasta el final del proceso. Suponiendo que un médico visita varias clínicas, en cada una de ellas existe una red donde se pueden conectar los dispositivos y los médicos pueden actualizar y consultar la historia clínica de los pacientes. En determinados momentos por diversos factores tales como el salir del área de cobertura, fallas en la red, o mantenimiento la conexión se pierde. En resumen, el sistema semi-conectado se idea en casos donde se puede estar sin conexión temporalmente.



Trabajo bajo el modelo conectado.

Teniendo en cuenta los dos modelos anteriores se puede afirmar que en este modelo se confía en que la conexión con el servidor es permanente y que siempre se estará en el área de cobertura de la red a la que pertenece el dispositivo. Un ejemplo para este modelo es el de un restaurante donde los meseros hacen la toma de pedidos en el dispositivo móvil y estos pedidos son procesados directamente en el servidor.

Lo ideal es tener un sistema semi-conectado o conectado debido al respaldo de información que garantizan.

Bajo estos modelos de conexión se puede implementar una aplicación móvil. Es importante decidir que procesos se realizarán en el dispositivo móvil, cuales procesos se trabajarán en el servidor, y en caso de conflictos de datos cual será la alternativa a seguir.

3.2.3 La importancia de los tipos de datos

En lo que se refiere al almacenamiento de información, es de gran ayuda considerar el tamaño de los tipos de datos con los que se van a construir las tablas en los dispositivos móviles. El tamaño de los tipos de datos influye directamente en la capacidad de almacenamiento secundario del PDA.

Algunos de los tipos de datos disponibles generalmente en los entornos de programación para dispositivos móviles son: Integer, Long, Text, Float, Date y Memo. La utilización de alguno de estos tipos de datos depende de los requerimientos de la aplicación particular a construir.

3.2.4 Manejo de Base de Datos en el desarrollo de aplicaciones.

El espacio y el poder de procesamiento esta sujeto a los avances tecnológicos de los PDA. Sin embargo, en la actualidad se utilizan técnicas que funcionan para mejorar el procesamiento a nivel de búsquedas y presentación de información.

Aún cuando la tecnología avance y se fabriquen procesadores mas rápidos y memorias de mayor capacidad, reducir la complejidad de procesamiento reducirá la carga y el "esfuerzo" del dispositivo.

Cuando se trata del manejo de bases de datos, en la programación enfocada a poderosos procesadores y memorias de gran capacidad, se pregona a favor de la normalización y sus tres formas normales⁶ con el fin de minimizar los problemas de lógica, el ahorro de espacio de almacenamiento y la rapidez en las consultas. No obstante, en el desarrollo de aplicaciones para dispositivos móviles se debe pensar en la manera de administrar mejor el limitado hardware. Dentro de este marco, existe una especie de proceso inverso a la normalización, que consiste en tratar de almacenar la información en el menor número de tablas posible de tal manera que el procesador realice un menor esfuerzo en términos de consultas de base de datos.

No esta de más aclarar que la aplicación de la "desnormalización" puede depender de cada aplicación y diseño de base de datos particular y es opcional. Aunque la desnormalización es un proceso que trae sus beneficios y es de gran utilidad cuando de dispositivos móviles se trata, también tiene sus desventajas, tales como duplicidad o anomalías de actualización de información.

3.3 Lenguajes y Entornos de Programación para Palm OS

Existen diferentes herramientas de desarrollo que permiten escribir aplicaciones para Palm OS, desde auténticos ensambladores de 68K hasta sofisticadas paqueterías. Existen diferentes herramientas basadas en lenguajes como Basic y un creciente numero de Máquinas Virtuales de Java.

Enumerar todas las herramientas de desarrollo para Palm OS resulta imposible, y más aún cuando la lista sigue creciendo. En el presente trabajo se trabajará con 7 herramientas que se encuentran entre los lenguajes y entornos más representativos y que han alcanzado cierta popularidad debido a sus particularidades.

La mayoría de las aplicaciones son desarrolladas y compiladas en una PC, sin embargo no todos los ambientes de desarrollo poseen un compilador y frecuentemente requieren de herramientas que compilen las aplicaciones, principalmente las desarrolladas en C y C++. Existen dos principales ambientes de desarrollo para C y C++: uno que es la elección de los desarrolladores de aplicaciones comerciales, CodeWarrior Development Studio para PalmOS y una herramienta de código abierto llamada Prc-Tools basada en una antigua

⁶ Algunos autores manejan 4 formas normales.

versión de gcc⁷. En el Palm OS Developer Suite⁸ se incluye una versión de PRC Tools.

Hay herramientas de desarrollo disponibles para la programación en Palm que no requieren un mínimo de programación, otras que se desarrollan para programadores en C/C++, Visual Basic o Java como ya se mencionó.

A continuación se presentan las herramientas con las que se trabajará y una breve descripción antes de profundizar en ellas:

CASL. Es un ambiente de desarrollo fácil de usar para Windows. Con CASL se pueden crear aplicaciones para Palm OS, Windows Mobile y Windows. Utiliza un lenguaje de scripts el cual se compila con gcc por medio de Cygwin⁹

HB++. Handheld Basic es un entorno de desarrollo similar a Visual Basic para Windows, nutrido en bibliotecas (que incluyen acceso a bases de datos, desempeño en redes, elementos gráficos) y que genera ejecutables sin necesidad de instalar un entorno de ejecución.

NS Basic. Entorno de desarrollo para Windows con interfaz gráfica de usuario, y con Basic como lenguaje de scripts.

PDA Toolbox. Un entorno de desarrollo con la particularidad de que no requiere de ningún lenguaje de programación. Esto hace a PDA Toolbox la herramienta más sencilla de utilizar pero limita el alcance de las aplicaciones.

Satellite Forms. Se trata de un ambiente de desarrollo de software el cual crea aplicaciones para dispositivos con Palm OS y Windows Mobile. Permite crear aplicaciones muy útiles con muy pocas líneas de código. (Requiere aprobación de la empresa para enviar versión de prueba. En espera de dicha aprobación)

Superwaba. El lenguaje que se usa en Superwaba es un subconjunto de Java que usa la biblioteca estándar de Waba en lugar de la de Sun. Se instala una máquina virtual en los dispositivos móviles para ejecutar las aplicaciones.

HanDBase. Herramienta diseñada principalmente para la creación de aplicaciones con bases de datos directamente sobre el dispositivo.

En el siguiente capítulo se mencionarán las características y procedimientos para crear aplicaciones mediante estas herramientas.

⁷ GCC.- GNU Compiler Collection es un conjunto de compiladores creados por el proyecto GNU. Estos compiladores se consideran estándar para los sistemas operativos derivados de UNIX, de código abierto o también de propietarios, como Mac OS X. Originalmente GCC significaba GNU C Compiler (compilador GNU para C), porque sólo compilaba el lenguaje C. Posteriormente se extendió para compilar mucho otros.

⁸ Palm OS Developer Suite es un conjunto de herramientas disponibles para el desarrollo de aplicaciones en Palm OS anteriormente disponibles en PalmSource.com. Ahora se puede obtener una versión más actualizada en www.access-company.com previo registro.

⁹ Cygwin es un emulador Linux en Windows, desarrollado por la empresa RedHat y que es actualmente software libre GNU. <http://www.cygwin.com>

Capítulo IV

Diseño de una Aplicación de Base de Datos para Obtener Información en la Investigación de Campo

El objetivo principal del presente trabajo es desarrollar una aplicación para dispositivos móviles la cual sirva en el ejercicio de recaudar información en el proceso de investigación. En los siguientes apartados se pretende sentar las bases del proceso de investigación de campo por medio de definiciones para posteriormente plantear un problema que requiera la investigación de campo como principal ejercicio para la búsqueda de una solución. En el tercer apartado de este capítulo se presentarán las alternativas de programación del cuestionario que guiará hacia la solución del problema planteado.

4.1 La Investigación de Campo

La investigación es un proceso que tiene por fin ampliar el conocimiento científico. Una investigación se caracteriza por ser un proceso:

- * **Sistemático:** A partir de la formulación de una hipótesis u objetivo de trabajo, se recogen datos según un plan preestablecido que, una vez analizados e interpretados, modificarán o añadirán nuevos conocimientos a los ya existentes, iniciándose entonces un nuevo ciclo de investigación. La sistemática empleada en una investigación es la del método científico.

- * **Organizado:** Todos los miembros de un equipo de investigación deben conocer lo que deben hacer durante todo el estudio, aplicando las mismas definiciones y criterios a todos los participantes y actuando de forma idéntica ante cualquier duda.

- * **Objetivo:** Las conclusiones obtenidas del estudio se basan en hechos que se han observado y medido, y que en su interpretación se evita cualquier prejuicio que los responsables del estudio pudieran tener.

Tipos de investigación

- * **Investigación básica:** También llamada investigación fundamental o investigación pura. Se suele llevar a cabo en los laboratorios.

- * **Investigación aplicada:** Es la utilización de los conocimientos en la práctica, para aplicarlos, en la mayoría de los casos, en provecho de la sociedad. Un ejemplo son los protocolos de investigación clínica.

Para iniciar el proceso de investigación se deben tener claramente precisos los siguientes conceptos:

Definición: La definición del problema, es el instrumento adecuado para descubrir con mayor efectividad y comprensión los objetivos y contenido de la investigación.

Justificación: Justificar una investigación es exponer las razones por las cuales se quiere realizar. Toda investigación debe realizarse con un propósito definido. Debe explicar porque es conveniente la investigación y qué o cuáles son los beneficios que se esperan con el conocimiento obtenido.

Objetivos: La fijación de un objetivo o de varios objetivos es necesaria ya que a través de ellos se conoce que es lo que se pretende con el esfuerzo investigativo. El objetivo es la razón de ser y hacer en la investigación.

Los objetivos debe ser claros en su redacción, medibles y alcanzables. No deben permitir desviaciones durante el proceso de la investigación. Son las guías de estudio durante el proceso de la investigación.

Los objetivos deben mostrar una relación clara y consistente con la descripción del problema y, específicamente, con las preguntas y/o hipótesis que se quieren resolver.

Con el logro de los objetivos específicos se espera, entre otros, encontrar respuesta a una o más de las siguientes preguntas: ¿Cuál será el conocimiento generado si el trabajo se realiza? ¿Qué solución a que problema se espera desarrollar?

La creación de conocimiento no es más que la sola generación u obtención de datos nuevos, sino avanzar en la comprensión de un fenómeno interpretando los mismo.

Marco teórico de la investigación

El marco teórico es una síntesis del contexto general (local, nacional y mundial) en el cual se ubica el tema de la propuesta, estado actual del conocimiento del problema, brechas que existen y vacío que se quiere llenar con el proyecto; porqué y cómo la investigación propuesta, a diferencia de investigaciones previas, contribuirá, con probabilidades de éxito, a la solución o comprensión del problema planteado.

Las funciones principales del marco teórico son las siguientes:

- * Orienta sobre el conocimiento del tipo de investigación
- * Amplia el horizonte de estudio, pues da opciones de puntos de vista
- * Conduce al establecimiento de hipótesis
- * Inspira líneas de investigación
- * Prevé la manera de interpretar los datos provenientes de la investigación

Hipótesis

Las hipótesis son proposiciones tentativas acerca de las relaciones entre dos o más variables y se apoyan en conocimientos organizados y sistematizados. Las hipótesis contienen variables; éstas variables poseen propiedades que pueden ser medidas mediante un proceso empírico.

Las hipótesis surgen normalmente del planteamiento del problema y la revisión de la literatura y algunas veces de las teorías. Pueden referirse a una situación real. Las variables contenidas deben ser precisadas, concretas y poder observarse en la realidad. Asimismo, las hipótesis deben estar vinculadas con técnicas disponibles para probarlas.

Delimitación del Área Geográfica en estudio

La delimitación del área geográfica en estudio, no es más que la identificación del lugar, zona o conjunto de zonas que se plantean o visualizan para ser estudiadas.

Método de Investigación de Campo

Es la recopilación de datos en el área a investigar. La ventaja de esta manera de investigar es que proporciona información mas exacta, un alto grado de confiabilidad y por consecuencia un bajo margen de error. Entre las desventajas se encuentran que es costoso, tardado y requiere de personal especializado, instalaciones y equipo adecuado.

Fuentes de Información

Se trata de las personas sobre las cuales se puede obtener información a través del método de investigación. Por ejemplo consumidores finales, personal de una organización, personal de organismos gubernamentales y no gubernamentales o personal especializado en el tema.

Lugar de Obtención de la Información

Es el lugar en el que se realiza la investigación de acuerdo con la delimitación del área geográfica

Procedimientos para obtener la información

Son las técnicas diseñadas para la recopilación de información. Entre ellos se encuentran la observación, el sondeo de opinión, las entrevistas personales, entrevistas por correo, por teléfono, entre otras.

Determinación de la Unidad de Muestreo

El muestreo se divide en:

No Probabilístico: La elección de un elemento de población para que forme parte de la muestra se basa en gran parte en el criterio del investigador o del entrevistador de campo

Probabilístico: Cada elemento de la población tiene una oportunidad real de ser seleccionado para la muestra. Nos permite calcular el grado posible al que puede llegar el valor de la muestra para diferenciarse del valor de interés de la población, a esto se le llama error muestral.

4.2 Definición del Problema

Para el presente trabajo se requiere de un problema planteado y cuya solución se halle en la investigación de campo mediante la realización de una encuesta. El problema que se presenta fue planteado y desarrollado pensando en un tema que puede arrojar beneficios a las empresas, siempre y cuando se estudien y se determinen sus causas: la insatisfacción laboral. La insatisfacción laboral es un problema que muestra uno de los tantos conceptos que involucran al desarrollo de una organización y las repercusiones que puede tener para la empresa en general. Esto puede ser muy útil al entender la relación satisfacción-productividad en los trabajadores. Para el presente trabajo lo más importante será lograr desarrollar las aplicaciones que contengan el cuestionario y obtener de manera más rápida la información así como el vaciado de ésta hacia una computadora.

El recurso mas importante de una empresa es precisamente el recurso humano, de ahí parte la importancia de mantener un clima laboral bueno para que el empleado sea motivado y trabaje con mayor eficacia.

La eliminación de las fuentes de insatisfacción conlleva en cierta medida a un mejor rendimiento del trabajador, reflejado en una actitud positiva frente a la organización. Existiendo insatisfacción en el trabajo, estaremos en presencia de una ruptura en las relaciones sindico - patronales.

La satisfacción en el trabajo puede ser definida como una actitud general del individuo hacia su trabajo. Para muchos la satisfacción en el trabajo es un motivo en sí mismo, es decir, el trabajador mantiene una actitud positiva en la organización laboral para lograr ésta. Para otros, es una expresión de una necesidad que puede o no ser satisfecha.

Por medio del estudio de la satisfacción, la gerencia de una empresa podrá saber los efectos que producen las políticas y disposiciones generales de la organización en el personal. Así se podrán mantener, suprimir, corregir o reforzar las políticas de la empresa, según sean los resultados que se obtengan.

La satisfacción en el trabajo puede ser determinada por el tipo de actividades que se realizan, es decir, que el trabajo tenga la oportunidad de mostrar las habilidades del trabajador y que ofrezcan un cierto grado de desafío para que exista el interés. Que los empleados sean bien recompensados a través de sus salarios y sueldos acorde obviamente a las expectativas de cada uno. Que las condiciones del trabajo sean adecuadas, no peligrosas o incómodas lo cual hace mejor su desempeño. Además los empleados buscan dentro del trabajo que su jefe inmediato sea amigable y comprensible y que los escuche cuando sea necesario.

La insatisfacción en el trabajo se refleja en la salida inminente de los empleados o en constantes gestos que expresen la necesidad de mejorar las relaciones obrero-empresa. En muchos casos también se ve reflejada en la forma leal de los trabajadores al esperar que las condiciones mejoren.

La insatisfacción laboral es un tema que preocupa a la mayoría de los empresarios actualmente. Cada vez es más importante preocuparnos para eliminar las fuentes de insatisfacción laboral para evitar problemas posteriores. La insatisfacción de cada persona puede contribuir involuntariamente a deteriorar la imagen de la empresa, la calidad de sus productos y servicios e influir en el crecimiento y el desarrollo organizacional bajando los niveles de productividad y calidad. La situación generada por la crisis generalizada requiere de la organización, de un plan estratégico para salvar la integridad empresarial y la de los empleados. La insatisfacción y el malestar en el trabajo suele expandirse a otros grupos como familia o amigos. Es por eso la necesidad de preocuparse por este tema tan importante y realizar políticas dirigidas a conciliar vida familiar y laboral.

Los métodos más ampliamente utilizados para la medición de la satisfacción laboral son la escala global y la calificación de la suma formada por numerosas facetas del trabajo. El primer método consiste en nada más pedirle a los individuos que respondan a una pregunta semejante a ésta: Considerando todo ¿cuán satisfecho está con su trabajo? Entonces los participantes contestan encerrando un número entre 1 y 5 que corresponden a las respuestas desde "altamente satisfecho" hasta "altamente insatisfecho". El segundo método es más complejo. Este identifica los elementos clave en un trabajo y pregunta a un empleado acerca de sus sentimientos sobre cada uno. Los factores típicos que estarían incluidos son la naturaleza del trabajo, la supervisión, el salario actual, las oportunidades de ascenso y las relaciones con los compañeros de trabajo.

Estos factores se estiman sobre una escala estandarizada y luego se suman para crear una calificación total sobre la satisfacción en el trabajo.

La bibliografía indica que los factores mas importantes que contribuyen a la satisfacción en el trabajo, son el reto del trabajo, los premios equiparables, las condiciones de trabajo favorables y colegas que gusten apoyar. A esta se agregan la importancia de una buena personalidad, la compatibilidad en el trabajo y la disposición genética del individuo:

- * Trabajo mentalmente desafiante.
- * Recompensas justas
- * Condiciones favorables de trabajo.
- * Colegas que brinden apoyo.
- * Compatibilidad entre la persona y el puesto.
- * Cuestión de genes.

En la presente investigación se tratará de entender prácticamente los efectos de la satisfacción laboral con la productividad, ausentismo y rotación.

Mediante el estudio de la satisfacción laboral se pretende entender el efecto que producen las políticas, normas, procedimientos y disposiciones generales de la empresa para los empleados. De esta manera la presente investigación podrá ser un útil conocimiento para la gerencia de la empresa para mantener, suprimir, corregir o reforzar las políticas de la empresa según sea el resultado obtenido. Por lo tanto la eliminación de las fuentes de insatisfacción conllevará a un mayor rendimiento del trabajador.

En base el problema planteado anteriormente y al objetivo que persigue la presente investigación, se procede a enunciar la hipótesis:

Hipótesis.- "La insatisfacción laboral tiene repercusiones negativas para la eficiencia organizacional"

Objetivo general.- Comprobar la repercusión negativa que tiene la insatisfacción laboral para la eficiencia organizacional.

Objetivos particulares.- Analizar algunas actividades en la empresa que provocan insatisfacción en el trabajo.

Encontrar los aspectos de la empresa afectados por la insatisfacción laboral.

Para este trabajo de investigación se utilizará la investigación aplicada ya que nos daremos a la tarea de buscar la información en el área a investigar. Se utilizará el método de campo ya que éste método aunque es más costoso y tardado, proporciona mayor objetividad en la información obtenida, proporciona mayor confiabilidad y un menor margen de error. La fuente de información será el personal de una organización y se realizará dentro de la organización y los procedimientos para obtener información serán la observación y sondeo de opinión por medio de un cuestionario.

Como muestreo se podrá elegir entre el probabilístico y el no probabilística dependiendo del número de empleados en la empresa y sobre todo aquellos que se presten para el estudio.

Diseño del cuestionario

Un cuestionario es un instrumento sistemático que permite obtener información sobre un tema o asunto determinado. El buen diseño de un cuestionario

determina el fracaso o éxito de Investigación ya que es el que obtiene la información requerida para emitir un juicio.

* Es el elemento mas importante o base de la investigación, de una manera sistematizada quedan planteadas las preguntas y respuestas.

* Por medio del cuestionario se obtiene información suficiente y efectiva.

En este estudio, el cuestionario elaborado para los empleados de la organización consta de 10 preguntas, número que se considera suficiente para obtener resultados favorables que ayuden a presentar deducciones positivas respecto a la investigación en proceso. Está estructurado por preguntas abiertas (ayudan a descubrir las preferencias y pensamientos de la persona entrevistada, y para interpretar los resultados finales de una encuesta), de opción múltiple (ayudan a evitar respuestas deformadas y simplifica el proceso de análisis de tabulación y codificación de resultados) y de batería.(permite seguir la secuencia en el cuestionario y no desubicarse al momento de contestarlo)

Normalmente una cuestionario de este tipo se vería impresa en una hoja de papel como se muestra en la siguiente página.

No. De Cuestionario: _____
Fecha de aplicación: _____
Hora: _____

NOMBRE DE LA EMPRESA
Encuesta

Nombre del empleado: _____
Puesto: _____
Sexo: F ___ M ___

1.- ¿Cuán satisfecho está con su trabajo?
 Altamente satisfecho satisfecho Regularmente satisfecho insatisfecho
 altamente insatisfecho

2.- ¿Estás satisfecho con las actividades que realizas en tu trabajo?
 Sí No

3.- ¿En caso de contestar NO en la pregunta anterior, enuncia 3 actividades de tu trabajo que te produzcan insatisfacción laboral:
1. _____
2. _____
3. _____

4.- ¿Cómo es tu relación con tu supervisor inmediato?
 Buena Regular Mala

5.- Cuáles son los principales puntos por los que la relación con su supervisor sea buena o mala?

6.- Cree que según las actividades que realiza en su trabajo, su salario es:
 Bueno Regular Malo

7.- ¿Cuenta con oportunidades de ascenso en su empleo?
 SI NO

8.- ¿Las relaciones laborales con sus compañeros de trabajo son?:
 Muy buenas Buenas Regulares Malas Muy malas

9.- Enumere 3 razones por las cuales creé Usted que sus compañeros anteriores presentaron renuncia a la empresa:
1. _____
2. _____
3. _____

10.- ¿Cómo considera las condiciones físicas de trabajo?

Como ya se mencionó, el objetivo primordial del presente trabajo es realizar la captura de información por medio de dispositivos PDA y reducir considerablemente el tiempo de captura y de vaciado de información para su posterior análisis.

Se desarrollará una aplicación para recolectar la información con este cuestionario por medio de dispositivos PDA con Palm OS. Esta aplicación se desarrollará por medio de las herramientas que se mostrarán en el siguiente apartado y los resultados se analizarán en el último apartado de este mismo capítulo.

4.3 Alternativas de Solución

Existen un sinnúmero de herramientas para el desarrollo de aplicaciones en Palm OS entre las cuales se halla software comercial y software libre. Bajo ambos tipos de licencia, a continuación se muestra el diseño de una aplicación por medio de 7 herramientas distintas que cuentan con cierta popularidad entre la comunidad de programación para la plataforma. Dichas herramientas presentan características destacables que se observarán durante el desarrollo. Sobre la aplicación se requiere que sea ejecutable en Palm OS, que permita capturar las respuestas del cuestionario presentado e ir acumulando registros para poder ser vaciados a un equipo de escritorio por medio de la sincronización.

4.3.1 PDA Toolbox

www.pdatoolbox.com

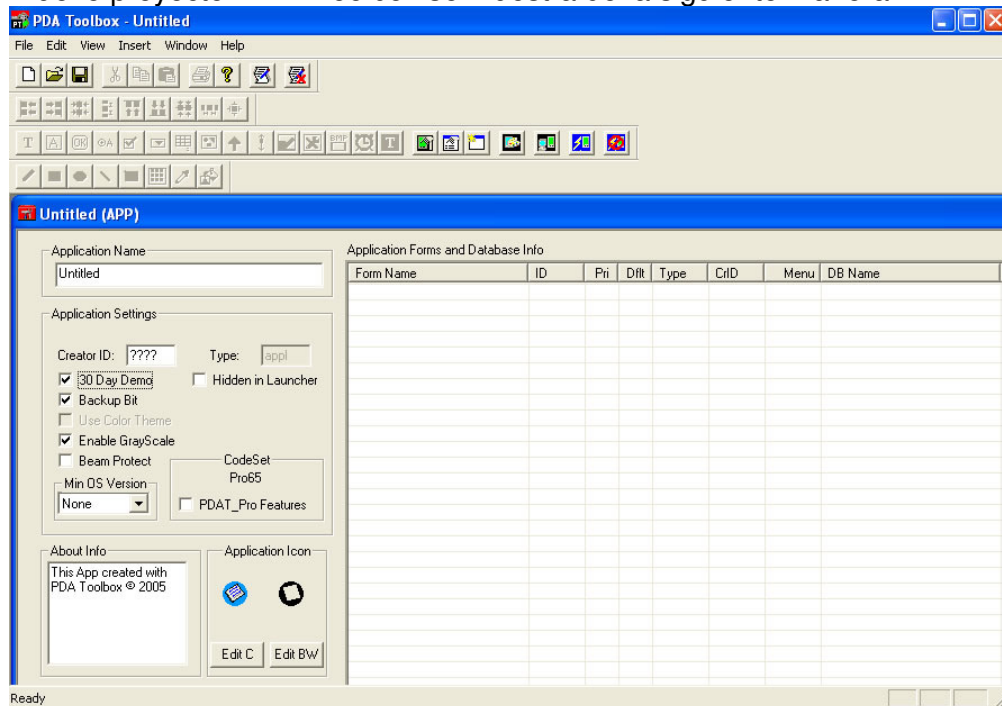
PDA Toolbox es un entorno de desarrollo que en su versión estándar de prueba no tiene opción para escribir código. A diferencia de otras herramientas de programación, PDA Toolbox no requiere conocimiento previo.

Este entorno de desarrollo provee de herramientas predefinidas que permiten crear aplicaciones estándar para Palm OS. El enlace entre los formularios y los objetos se hace de manera muy sencilla, y se puede lograr la creación de una aplicación de base de datos sin utilizar una sola línea de código, es decir, sólo con relacionar los formularios y los objetos. Así también, PDA Toolbox no requiere de instalar algún ambiente de ejecución (Runtime) en el dispositivo sino que genera un solo archivo ejecutable .prc. Sin embargo, este entorno no se limita a las características predefinidas, sino que en su versión Profesional ofrece la opción de crear o editar scripts aumentando las capacidades de acción de los objetos. Entre otras habilidades que proporciona el adquirir la versión profesional se encuentra la de crear aplicaciones que interactúen por medio del puerto Infrarrojo o capturar firmas digitales.

PDA Toolbox permite el manejo de bases de datos de hasta 32000 registros y la sincronización de éstas con Archivos de Microsoft Access y Microsoft SQL Server por medio del conducto PDA Toolbox HS Conduit (PDATHS). Se requiere conocimiento y licencia de Visual Basic 6.0 para crear las aplicaciones de escritorio con las cuales será sincronizada la base de datos del PDA

Existen muchas herramientas que se adaptan a PDA Toolbox y las aplicaciones creadas por éste entorno de desarrollo que están disponible (algunas de manera gratuita) en el sitio web, que van desde editores de bases de datos, de archivos pdb, conductos, y manejadores de bases de datos.

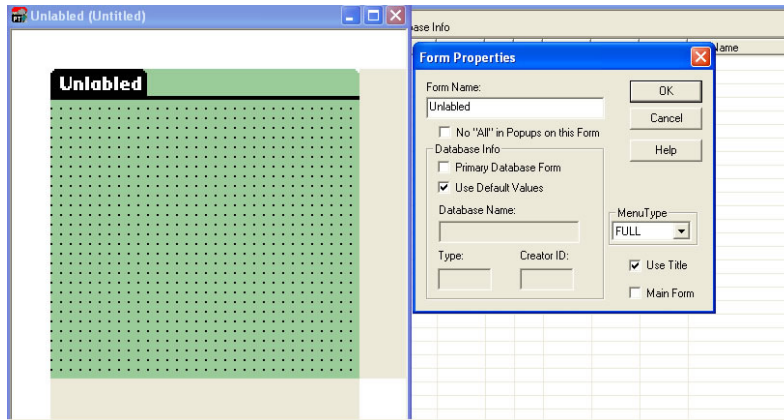
Una vez instalado PDA Toolbox se ejecuta el entorno de desarrollo y se crea un nuevo proyecto. PDA Toolbox se muestra de la siguiente manera:



Bajo este esquema se modificarán las características generales de la aplicación a desarrollar como con el Nombre de la aplicación, el ID de creador¹⁰, la posibilidad de limitar las aplicaciones desarrolladas a periodos de prueba de 30 días, permitir o evitar su difusión mediante puerto infrarrojo, decidir si se hace o no respaldo de la aplicación mediante la sincronización con la PC, elegir una versión mínima de Palm OS para ejecutar la aplicación desarrollada y el diseño de un icono para identificar la aplicación entre otras opciones.

Por medio del menú Ver (View) se activan las barras de herramientas necesarias para el desarrollo de las aplicaciones, siendo la Barra de Formularios (field-form toolbar) la más importante ya que es la que contiene los elementos necesarios para el diseño, compilación y ejecución de las aplicaciones. Por medio de esta barra agregamos un Formulario (New Form) y modificamos las propiedades de dicho formulario. PDA Toolbox está ideado principalmente para elaborar aplicaciones con Bases de datos por lo que durante el desarrollo de las aplicaciones se presentarán opciones para nombrar, o editar dichas bases. Entre las propiedades que se podrán modificar de los formularios se encuentra la de señalar el formulario principal en la aplicación, el formulario principal en la base de datos, el nombre de la base de datos y nuevamente el ID de creador (esté corresponderá a la base de datos y no a la aplicación, aunque puede ser el mismo). En la siguiente imagen se muestra un nuevo formulario con el cuadro de propiedades en espera de definir las.

¹⁰ El ID de creador es un identificador de 4 caracteres que permite a los programadores de aplicaciones bajo Palm OS realizar sus proyectos bajo esta firma. El registro de un ID de creador se hace de manera oficial en www.access-company.com



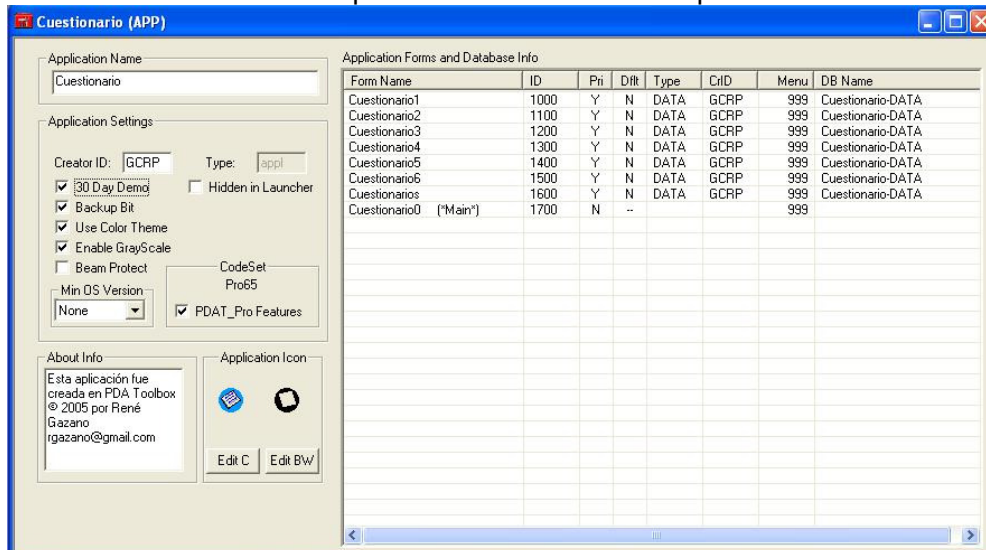
Una vez declaradas las propiedades del formulario se procede al diseño del mismo, es decir, colocar los elementos que contendrá dicho formulario por medio la barra de Herramientas “field-form toolbar”.



Dentro de los elementos predefinidos con los que puede diseñarse un formulario se encuentran las etiquetas, campos de texto, botones, botones de opción, casillas de verificación (checkbox), listas desplegables (popup), y tablas entre otros.

Para agregar dichos elementos al formulario basta con dar clic sobre la herramienta necesaria y el control aparecerá en la parte superior izquierda del formulario, listo para ser “arrastrado” hasta la posición en la que se desee colocar.

Las propiedades de cada elemento se pueden observar dando doble clic sobre el mismo. A continuación se presenta el diseño de la aplicación Cuestionario.



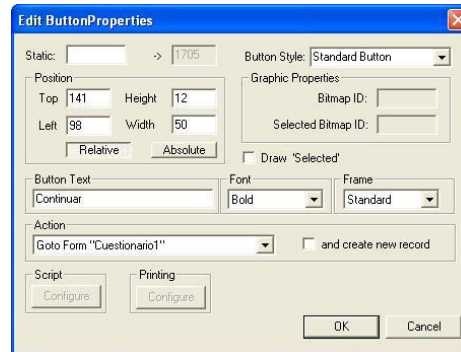
Esta aplicación cuenta con 8 formularios. Por medio de la tabla se puede observar que el formulario “Cuestionario0” es el formulario por el cual inicia la aplicación, aunque el menos importante ya que no se relaciona con la base de datos.



En el formulario de inicio solo muestra la bienvenida; contiene etiquetas, una imagen (la cual se copia y se pega directamente sobre el formulario, cuidando el tamaño) y dos botones de comando. Una limitante de PDA Toolbox es el manejo de menús. En este caso solo muestra la información del autor.

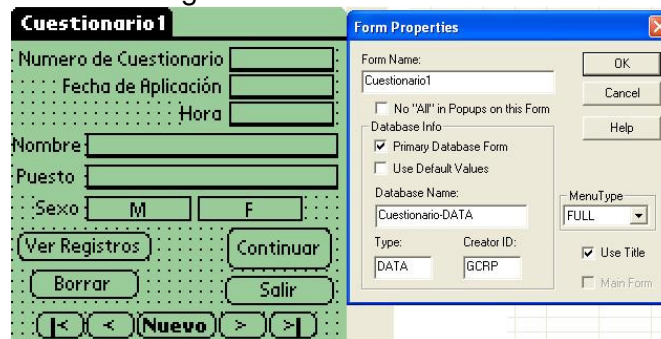
En las propiedades de los botones de comando cabe señalar las acciones que ejecutarán. PDA Toolbox cuenta con acciones predefinidas, las cuales simplemente se eligen mediante el selector *Action*. En

el caso del botón Salir, la acción elegida es *Exit Application*. Para el botón Continuar la acción elegida es que vaya al siguiente formulario *Goto form "Cuestionario1"* en el cual comienza la encuesta. Es necesario crear primero el formulario destino aunque no contenga elemento alguno. El resto de propiedades corresponden a cuestiones de estética y opciones que veremos más adelante. Para las opciones de *Script* y *Printing* es necesario contar con una versión registrada de PDA Toolbox.



Para el diseño del siguiente formulario debemos señalar en las propiedades del mismo la información de la base de datos. Puede haber uno o más formularios interactuando como Formularios principales de la base de datos. Se pueden usar los valores por defecto o bien proporcionarlos. Es importante tener presentes dichos valores ya que se utilizarán durante todos los formularios que inserten información en la base de datos, así como en la tabla final en la cual se concentrarán los registros para ser consultados.

El formulario *"Cuestionario1"* contiene la información general del estudio, es decir, el número de encuesta, fecha, hora, nombre del encuestado, puesto que ocupa en la empresa y sexo. Así también contiene los botones de acción que permiten el control de los registros como se muestra a continuación.

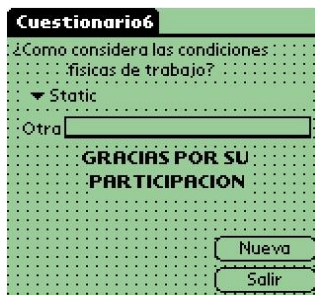
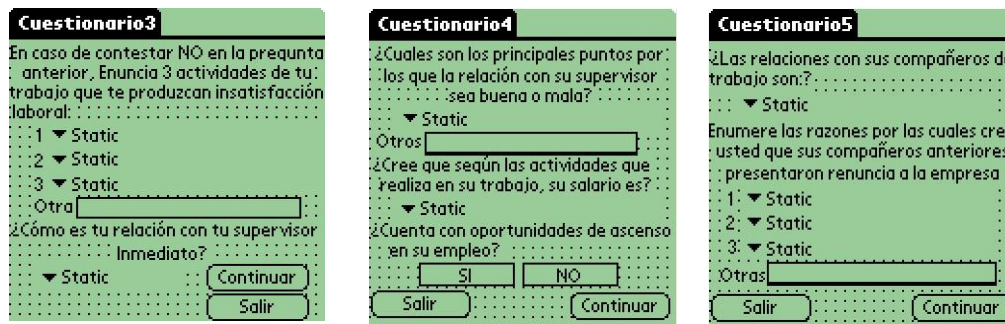


Cada control que implique información, como los campos de texto, los botones de opción o las listas desplegables, debe ser nombrados con un identificador de 4 caracteres, de esta manera la información se relaciona con la base de datos. En este caso se nombró Cuestionario-DATA, el tipo DATA (usualmente se utiliza este tipo de datos pero se puede cambiar por un tipo de registro definido por el usuario) y el ID de creador. Los campos del formulario relacionados con la base de datos son número de cuestionario (numc), fecha de aplicación (date), hora de aplicación (hora), nombre (name), puesto (psto) y sexo

(sexm y sexf agrupados mediante el Group ID 1). Los botones de acción, corresponderán a la acción descrita en su texto; en este caso los botones de la parte inferior ayudan a la exploración de registros por medio de las acciones *Goto first record*, *goto prev record*, *goto next record* y *goto last record*. El botón Nuevo crea un nuevo registro (*Create new record*). El botón Ver Registro muestra el formulario Cuestionarios (*Goto form "Cuestionarios"*) que es donde se muestra una tabla con los registros recabados. Para borrar el registro actual se usará el botón Borrar con la acción *Delete Current Record*. Por último el botón Continuar el cual sigue con la encuesta hacia el formulario *Cuestionario2*. Para el formulario Cuestionario2 se inician con las preguntas del estudio: Para la primera pregunta se utiliza un "Popup" o lista desplegable del cual se puede definir sus propiedades con doble clic.

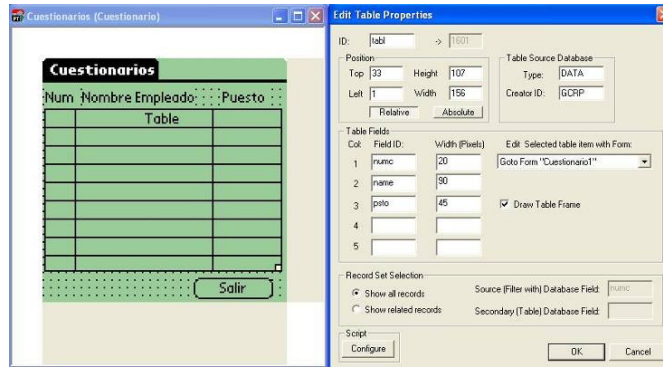


Las propiedades de la lista desplegable muestran las opciones de respuesta que bien pueden relacionarse con algún campo de otra base de datos, o una lista estática definida como es el caso. Para la pregunta 2 se utiliza nuevamente botones de opción (radio buttons) agrupados. De manera similar se diseñan los siguientes formularios:



Cabe señalar que para los casos en los que se piden respuestas que pueden variar a las esperadas, además de las opciones definidas en las listas desplegables se ha colocado un campo de texto para aquellas alternativas. En el siguiente y ultimo formulario de la aplicación se muestran los registros ordenados por numero de cuestionario y mostrando el nombre y puesto del encuestado.

Lo destacable de este formulario es el manejo de la tabla. Las propiedades de muestran en la figura. Se define de donde provienen los datos por medio del tipo de datos y el ID de creador y en los campos de la tabla se coloca el identificador de 4 caracteres perteneciente al campo que se asignó en la aplicación.



En este caso *numc*, *name* y *psto* son los datos con los que se rellena la tabla y se filtran por medio del numero de cuestionario (*numc*). Dentro de la definición de los campos que se mostrarán en la tabla se define el ancho de la columna.

Es conveniente guardar el proyecto de PDA Toolbox con la extensión pfa. Una vez terminada la aplicación se procede a su compilación y por tanto la creación del archivo ejecutable (prc). Existen 2 opciones al final de la barra de herramientas de formularios: crear la aplicación y probarla. La primera opción genera un archivo prc el cual puede ser instalado directamente en el PDA por medio de Hotsync, sin embargo no es conveniente instalar las aplicaciones sin haber sido probadas. La segunda opción genera el ejecutable y lo prueba en el emulador Palm OS Emulator o en el simulador Palm OS Simulator. Es recomendable probar las aplicaciones en el emulador o el simulador de Palm OS antes de sincronizarla con algún PDA. A continuación se observa la aplicación ejecutándose en el emulador.



Como se pudo observar, las complicaciones en PDA Toolbox son mínimas para el desarrollo de aplicaciones de este tipo. En este caso solo se requiere del manejo del entorno y el manejo de la base de datos debido a que todo se encuentra predefinido.

4.3.2 CASL

<http://www.caslsoft.com>

CASL (Compact Application Solution Language) es un ambiente integrado de desarrollo fácil de usar para Windows. Con CASL se pueden crear aplicaciones para Palm OS, Windows Mobile y Windows, usando un mismo conjunto de código, es decir, se pueden crear aplicaciones con el objetivo de ejecutarlas en diferentes sistemas operativos usando un lenguaje de programación de alto nivel. Se diseña la aplicación y se compila eligiendo la plataforma en la que se creará el ejecutable teniendo la capacidad de generar 3 tipos de archivos: .prc para Palm OS, .cce o .ccm para Windows Mobile (o Windows CE)¹¹ y .csp para ejecutar las aplicaciones directamente en Windows.

Cabe señalar que CASL es una herramienta libre de costo para el desarrollo de aplicaciones no comerciales (aficionados, desarrollo de freeware, aprendizaje). Para su instalación se requieren descargar 2 archivos: el instalador de CASL¹² y el archivo de instalación de componentes de CASL (CASL Components Setup). Una vez instalado CASL es muy importante instalar los componentes, ya que contienen las herramientas necesarias para la compilación y creación de archivos prc. A continuación se enlista el contenido de CASL Components:

- Cygwin.- Cygwin es un emulador Linux para Windows que provee herramientas de usuario y la facilidad para crear aplicaciones prc. Es gracias a gcc que PRC Tools puede crear archivos prc, esto es posible en Windows con Cygwin.
- PRC-Tools.- Contiene todas las herramientas gcc necesarias para crear aplicaciones para Palm OS.
- Palm OS SDK.- Contiene encabezados y bibliotecas de C requeridas por gcc para la creación de archivos prc.
- CASL Productivity Pak.- permite a los desarrolladores en CASL explorar y editar archivos cdb/cdi¹³. Permite la extracción de datos de manera sencilla y manejarlos con Microsoft Access.

Las capacidades de CASL incluyen

- Soporte para alta resolución en pantalla para Palm OS y Pocket PC(320X320)
- Biblioteca Bluetooth para el desarrollo de aplicaciones con dicho estándar de comunicaciones. Manejo de puerto infrarrojo.
- Biblioteca de eventos de Palm OS para el manejo de eventos.
- Conjunto completo de objetos con controles dinámicos
- Conjunto completo de funciones, eventos, funciones predefinidas, funciones definidas por el usuario.
- Archivos de base de datos indexados
- Manejo de conductos para la sincronización de la aplicación del PDA con la PC. (incluye soporte para Hotsync (Palm OS) y ActiveSync (Windows Mobile)

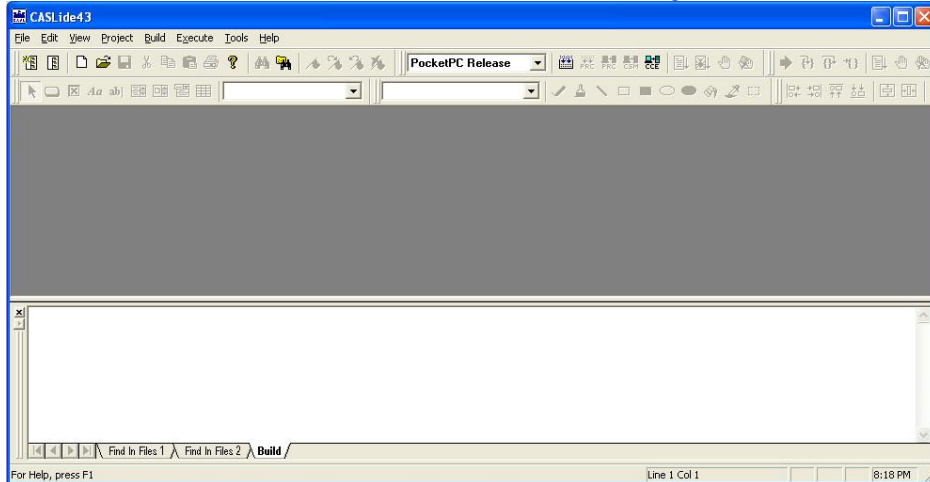
¹¹ Previa instalación del Ambiente de Ejecución de CASL, sólo para Pocket PC

¹² Se trabajó con la versión 4.3 de CASL, por lo que el archivo de instalación (setup) es CASL43

¹³ Archivo de base de datos de CASL, en Palm OS son representados como pdb.

- Socket para acceso a redes (http o TCP)
- Manejo de tecnologías de código de barras o puerto serial para equipos que cuentan con lectores incluidos.
- Manejo de software de impresión para Palm OS

Al ejecutar el entorno de desarrollo se muestra de la siguiente manera:



Antes de iniciar un proyecto cabe mencionar los tipos de archivos que se manejan con CASL:

CASL Project Files (.cpj). Archivo de proyectos de CASL. Identifica a todos los archivos usados por el proyecto que se está creando.

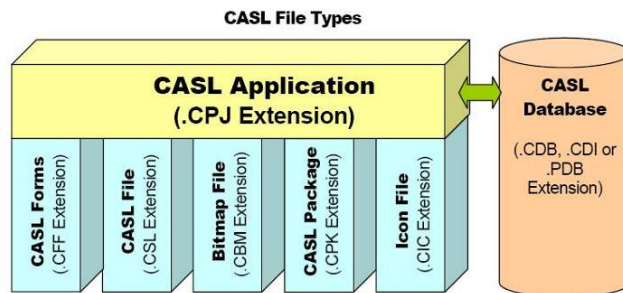
CASL Database Files (.cdb, .cdi o .pdb). Archivos de base de datos, .cdb y .cdi son archivos de base de datos de CASL. En Palm se representan como .pdb

CASL Forms (.cff). Estos archivos identifican a los Formularios¹⁴ dentro del proyecto con todos los objetos contenidos en éstos.

CASL File (.csi). Es el principal archivo de código fuente en CASL. Normalmente contiene las funciones de inicio (startup) y salida (shutdown) del programa y las sentencias "include" referentes a las bibliotecas (archivos .cpk).

Bitmap File (.cbm). Contienen los gráficos usados en la aplicación¹⁵.

CASL Package Libraries (.cpk). Éstos archivos son utilizados para descomponer el código CASL en componentes lógicos y utilizarlos desde el código principal como bibliotecas.

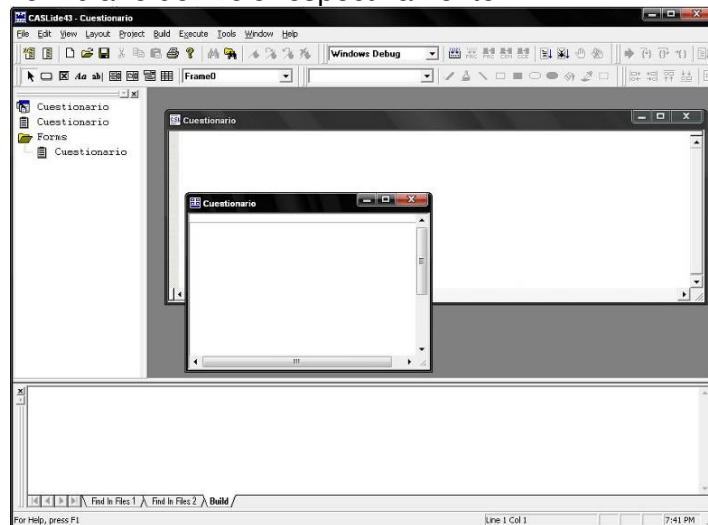


¹⁴ También llamados Frames en CASL

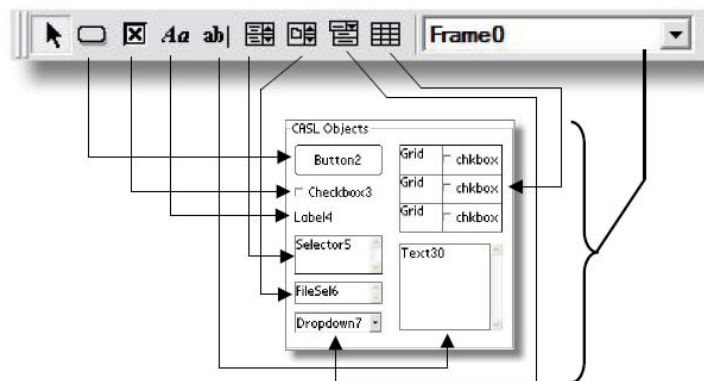
¹⁵ CASL utiliza esta extensión, sin embargo basta renombrar los archivos .bmp con la extensión .cbm

CASL Icon Files (.cic). Los archivos de íconos definen imágenes en íconos para la aplicación. Estos archivos son utilizado solamente por la plataforma Palm OS.

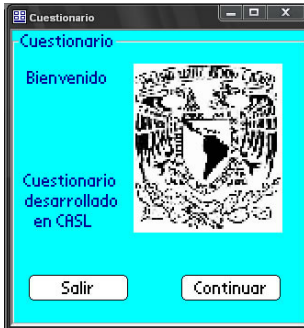
Al crear un proyecto será necesario indicar la carpeta de trabajo. Una vez creado CASL muestra el proyecto listo para ser desarrollado. En este caso se crea un proyecto llamado Cuestionario, el cual incluye de manera predeterminada los archivos Cuestionario.csl y Cuestionario.cff que identifican el código y el formulario de inicio respectivamente.



Para el diseño de formularios se debe tener activada la barra de herramientas Form Controls Toolbar la cual contiene los objetos necesarios para la interacción con el usuario.



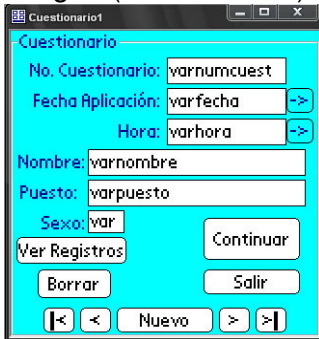
Los objetos disponibles en la barra de herramientas de formularios son: Botones, checkbox, etiquetas, campos de texto, selectores, selectores de archivo, listas desplegables(dropdown) y tablas(grid). En esta barra de herramientas también se muestra el nombre del formulario en el que se está trabajando.



Los objetos se crean y se ordenan por medio del ratón. Para acceder a las propiedades de los controles basta con dar doble clic. En el caso del formulario *Cuestionario* se definirá como el formulario de inicio bajo el siguiente esquema:

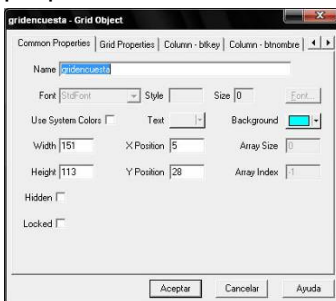
Objeto	Nombre	Display
Formulario	Cuestionario	Cuestionario
Etiqueta	Label1	Bienvenido
...
Botón	Cmdnextfr0	Continuar
Etiqueta	Lblimagen	
Botón	Cmdsalirfr0	Salir

En el caso de la etiqueta *Lblimagen* es necesario agregar al proyecto una imagen (Clic derecho en el proyecto>Add Bitmap File). Una vez agregada la imagen, se muestra del lado izquierdo (Barra de proyecto) dentro de la carpeta *Images*. En las propiedades de la etiqueta se proporciona el nombre de la imagen (sin extensión) en el campo *Bitmap*.



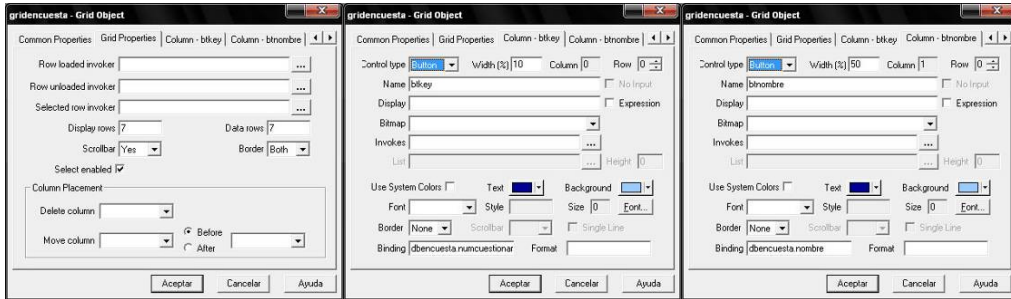
El diseño del formulario para capturar y mostrar los datos del encuestado, *Cuestionario1*, se compone de botones, etiquetas y campos de texto. Es importante tener en cuenta el nombre que se asigna tanto a los campos de texto como a los botones para su posterior codificación. En este formulario, cabe destacar, los botones “->” que se agregan a un costado de *varfecha* y *varhora*, en este caso llamados *cmdfecha* y *cmdhora* ya que servirán para elegir fecha y hora de manera automática.

El siguiente formulario en importancia se trata de aquel que contendrá la tabla para visualizar los registros. En este caso lo llamamos *Cuestionarios* y dentro de las propiedades a destacar es que en la tabla se insertan botones relacionados con la base de datos y los campos *varnumcuest*, *varnombre* y *varpuesto*. Para la creación de la tabla se crea un *Grid*, el cual se puede “rellenar” con botones, etiquetas, checkboxes, listas o cuadros de texto. Con el fin de que la tabla responda a un evento de búsqueda al seleccionar un registro se rellena la tabla con botones que mostrarán los campos mencionados. A continuación se muestra el cuadro de propiedades de la tabla.

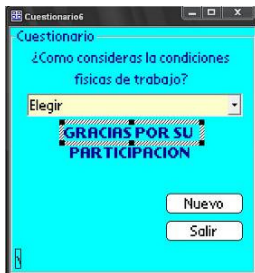
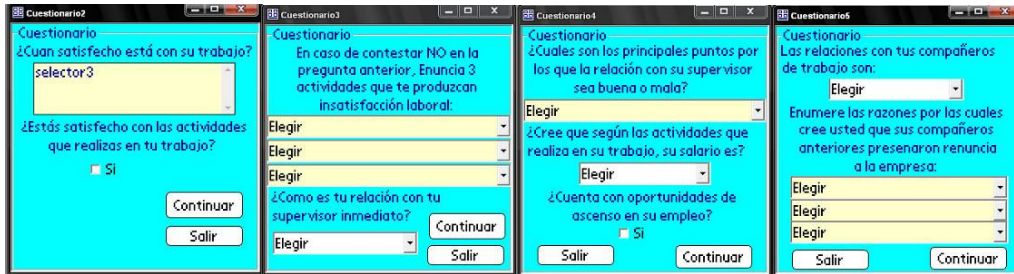


En la pestaña *Common Properties* de esta tabla se asigna el nombre de ésta y sus propiedades como medidas y posición. En la segunda pestaña *Grid Properties* se coloca el nombre de alguna función que puede ser invocada por la tabla, además de el número de columnas a mostrar, el tipo de bordes y se pueden eliminar o mover las columnas. La tercera pestaña servirá para agregar columnas. Una vez agregada se determina que tipo de control llevará esa columna (en este caso definimos que las tres columnas serán

botones), el porcentaje de la tabla que ocupará en ancho, el nombre del control (que servirá para relacionarlo con la base de datos), y entre todas las opciones de formato que se muestran cabe destacar la propiedad "Binding" ya que de esta manera cargará los registros de la base de datos proporcionando la referencia como *nombredebase.nombredecolumna*.

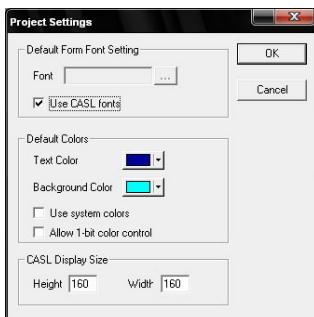


Se observa que el nombre de la base de datos ya se develó: dbencuesta, sin embargo, esta es definida y manipulada por medio del código. El resto de formularios se define de similar manera y se muestran a continuación.



Las opciones que mostrarán los selectores y cada una de las listas desplegadas deben ser definidas mediante código. En las propiedades de cada uno de estos controles se proporciona el nombre de la lista. Se recomienda llevar a cabo el diseño de los formularios y el desarrollo del código de manera simultánea ya que en ocasiones hay que declarar primero (como en el caso de las listas) y en otras primero debe existir el control para poder operarlo.

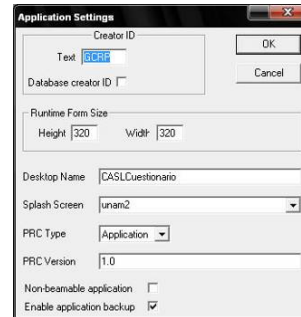
La totalidad del código se encuentra en el archivo .csl como ya se mencionó. En el apéndice se muestra el código comentado que corresponde al archivo Cuestionario.csl.



La estructura del código en CASL puede variar y puede presentarse en el orden deseado, la única excepción es que la sección de variables y las variables dentro de cada función deben aparecer antes que cualquier otra sentencia.

Antes de pasar a la compilación es importante ajustar las propiedades del proyecto y de la aplicación. En el Menú Project de CASL se encuentra dichas opciones:

Las propiedades del proyecto se refieren al formato de los formularios en el espacio de trabajo de CASL. Las propiedades de la aplicación se refieren a las propiedades que tendrá la aplicación una vez compilada y ejecutándose en el Palm OS. Se proporciona el ID de creador para la aplicación, se elige si es el mismo ID para la base de datos, el tamaño de pantalla en la que se ejecutará, el nombre de la aplicación en el escritorio de Palm, entre otras propiedades.



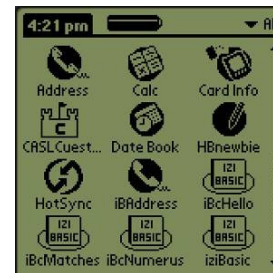
Una vez terminada la aplicación se procede a compilarla. Para liberar la aplicación en Palm OS se requiere activar la opción “C Build” del menú Build de CASL.¹⁶



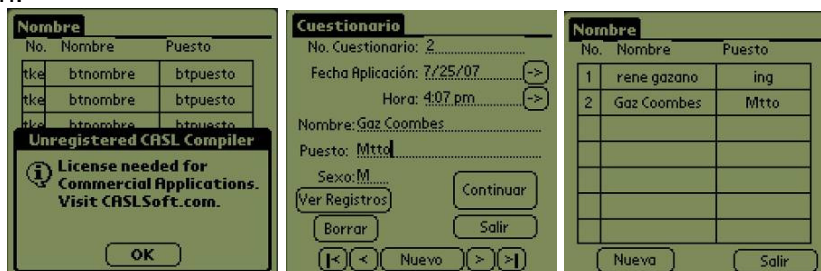
En la imagen se muestra la barra de Ejecución de CASL en la que se puede elegir para cual plataforma liberar la aplicación. En este caso se elige Palm OS Release¹⁷ y automáticamente se activan las 3 opciones siguientes que son:

- Compilar
- Crear el archivo PRC
- Instalar el archivo PRC

El proceso de compilación arroja distintos archivos dependiendo de la plataforma que se eligió. Posteriormente a la compilación, en el directorio de trabajo se creó la carpeta PalmOS Release el cual muestra todos los archivos implicados en el proceso de creación del archivo ejecutable y el propio archivo “Cuestionario.prc” listo para ser probado o instalado.



Entre las limitantes de la versión no registradas de CASL es un mensaje al inicio de la aplicación. Como se mencionó, para el desarrollo de aplicaciones comerciales CASL si tiene costo de licencia. A continuación se muestran algunas vistas de la aplicación en ejecución.



CASL es una herramienta poderosa debido a los alcances del lenguaje C aunque el código no se realice directamente en este lenguaje sino, como se vió, en un lenguaje de transición. Otra de las ventajas con las que cuenta es

¹⁶ CASL genera archivos con código en C que mediante las herramientas GCC y PRC Tools crean el archivo ejecutable .prc

¹⁷ Antes de probar la aplicación en Palm OS (sea mediante un emulador o bien sobre un dispositivo) y para mayor rapidez se puede liberar la aplicación para Windows con la opción Windows Release exactamente con el mismo código. Después de compilar se activan las últimas cuatro opciones de la barra de Ejecución que son: Ejecutar, depurar, pausar o detener la ejecución.

que no requiere de licencia para trabajar el entorno (a menos que sea para uso comercial). Por último, la capacidad de realizar con el mismo código aplicaciones para Windows y las dos principales plataformas en PDA, sin el requerimiento de instalar algún entorno de ejecución, hacen de CASL una opción muy conveniente.

4.3.3 HB++ (Handheld Basic ++)

<http://www.handheld-basic.com>

HB++ (Handheld Basic ++) es un entorno de desarrollo bastante similar a Microsoft Visual Basic con la capacidad de generar poderosas aplicaciones para Palm OS. HB++ es muy fácil de utilizar ya que contiene un manual de usuario muy completo además de ofrecer en su sitio web, una aplicación de aprendizaje de gran utilidad como guía para nuevos usuarios.

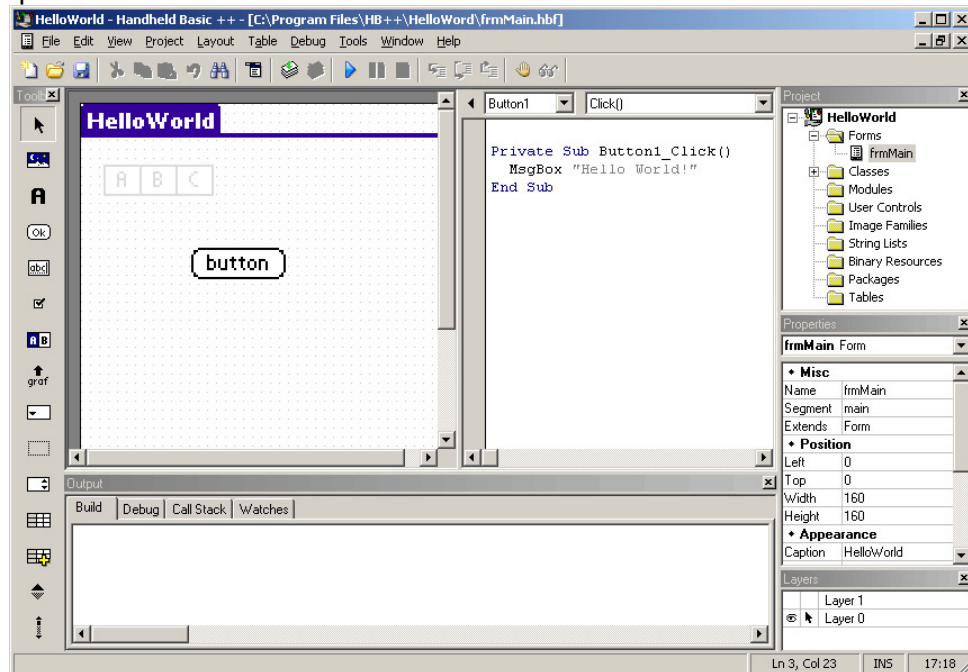
La instalación se realiza por medio del archivo de instalación disponible en línea, el cual proporciona una versión de prueba con todas las características del entorno liberadas. Sin embargo todas las aplicaciones compiladas por esta herramienta mostrarán un mensaje recordando que éstas fueron desarrolladas como prueba.

Entre las características más importantes de HB++ se encuentran las siguientes:

- Lenguaje similar a Microsoft Visual Basic con un compilador y soporte para APIs de Palm OS
- No requiere ambiente de ejecución (Runtime)
- Clases que encapsulan cada función de Palm OS: Transferencias por Bluetooth, Serial e IR, trabajo en redes, sistemas de archivos, telefonía, escaneo de código de barras, controles de interfaz de usuario
- Motor de base de datos, generador de conductos.
- Soporte para bibliotecas compartidas, llamadas a APIs externas y ARMIlets
- HB++ soporta desde la versión 3.0 hasta la denominada Cobalt de Palm OS.
- El proyecto es compilado en un único archivo .prc, no requiere Ambiente de ejecución (Runtime).
- El código puede llegar a ser 10 veces más rápido que otros lenguajes
- Utiliza Lenguaje Basic orientado a objetos altamente estructurado. Se puede usar código de Visual Basic con muy pocos cambios.
- Permite reutilizar los formularios, módulos, clases, tablas o imágenes de mapa de bits.
- Permite configurar el Palm OS Emulator para probar las aplicaciones y depurarlas al instante, es decir, sin esperar a crear el archivo prc y llevar a cabo una sincronización.
- Se pueden usar objetos de interfaz de usuario como botones, menús, listas, selectores, campos de texto, etiquetas, etc. pero además se pueden crear controles propios.
- Permite llamar rutinas externas en lenguaje C almacenadas en bibliotecas, código ARM o funciones de la API de Palm OS.
- Utiliza el estándar XML para almacenar los datos de la aplicación.
- Se pueden utilizar funciones avanzadas para manipulación de datos como sentencias se pueden escribir registros o datos de otras aplicaciones instaladas en el dispositivo, inclusive de aplicaciones estándar de Palm OS como (Agenda, gastos, etc)
- HB++ Permite generar conductos, creando procesos de sincronización con distintos manejadores de base de datos en equipos de escritorio.

- En comunicaciones y redes es posible trabajar con conexiones de módem, GPRS o Wi-fi: ajustando parámetros, crear y abrir conexiones, usar estándares de sockets TCP/IP, conectar a Internet mediante la clase http, conectar a ftp o servidores de correo con las clases FTP, SMTP y POP
- En cuanto a seguridad: cualquier flujo de información, sea de una base de datos o una conexión por socket puede ser cifrado. Las conexiones por socket pueden ser seguras usando SSL

A continuación una vista general del IDE de HB++ con el fin de mostrar sus componentes:



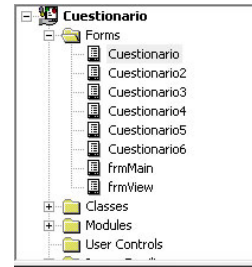
En la imagen anterior se pueden observar las partes que componen el IDE de HB++:

- El explorador de proyecto (Project). Presenta los diferentes elementos que componen el proyecto. Permite añadir, modificar o eliminar cada uno de ellos.
- La ventana de propiedades (Properties). Muestra las diferentes propiedades de un elemento seleccionado.
- La ventana de capas (Layers). Lista las capas de los formularios.
- La barra de herramientas (Toolbox). Contiene los diferentes controles que pueden utilizarse en el diseño de formularios.
- La ventana de salida (Output). Contiene 4 pestañas cuya función se observa en el proceso de compilación.
- La parte central del IDE sirve como editor de formularios y de código, de tablas, de clases, de módulos según sea el caso.

Una vez instalado HB++ se procede a su ejecución. Como primer paso hacia la creación de un proyecto se elige Nuevo proyecto en blanco al inicio de HB++ y se asigna un nombre al proyecto en la barra de propiedades. El nombre del proyecto que se utilizará es Cuestionario.

En el menú Tools > Options se pueden realizar ajustes para el espacio de trabajo como las características generales del ambiente, opciones de formato, elegir un editor de imágenes externo a HB++ y elegir Palm OS Emulador, Palm OS Simulator o un PDA como herramienta para depurar.

En el menú Project se añade un nuevo formulario (Add Form) con el nombre Cuestionario, el cual se alojará en el explorador de proyectos y será el formulario de inicio de la aplicación. Antes de codificar es recomendable crear todos los formularios que se utilizarán con el fin de codificar y crear referencias a formularios existentes. A continuación se muestra el diseño de los formularios y algunas características sobresalientes:

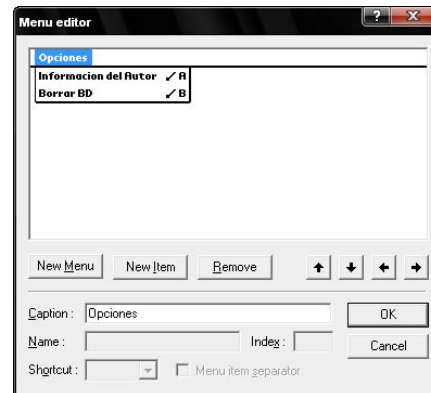


El formulario Cuestionario está compuesto por etiquetas, dos botones de comando y un control Picture. Se añade una imagen al proyecto mediante el menú Project> Add Images y se relaciona al control Picture por medio de la propiedad Image. Las propiedades Caption de las etiquetas y Text de los botones son las que contienen el texto visible en la interfaz de usuario. Es importante el nombre que se asigna a los botones por su referencia en el código.

Control	Nombre	Propiedad	Valor
Label	Label1	Caption	Bienvenido
Label	Label2	Caption	Cuestionario desarrollado en HB++
Button	cmdsalirFr0	Text	Salir
Button	cmdnextFr0	Text	Continuar
Picture	Picture1	Image	UNAM (imagen previamente añadida al proyecto)

Propiedades principales del formulario Cuestionario

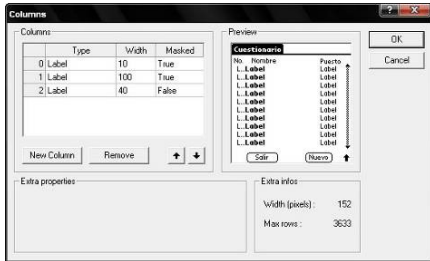
Para cada formulario se puede crear un menú distinto con el Menú Editor del menú herramientas (tools). Otra opción es crear un menú general, copiarlo y pegarlo en todos los formularios. A las opciones de los menús se les asigna un nombre para codificarlos.



En el presente proyecto, los formularios más importantes son: el que contiene la tabla con la lista de encuestados (frmMain) y el que captura y muestra la información de un registro (frmView).

Al añadir una cuadrícula (GridEx) la propiedad Columns ofrece la posibilidad de llenar la tabla con Etiquetas, Checkboxes y listas desplegables (Popup) y distribuir el

ancho de cada columna. Para el caso mostrado en la siguiente imagen se observan las tres columnas elegidas como etiquetas. Aun no se relacionan los valores de las etiquetas con los de la base de de datos ya que ésta aun no es definida. Sin embargo se sabe que la cuadrícula contendrá el número de cuestionario, nombre y puesto del encuestado para distribuir el espacio de las columnas.



Control	Nombre	Propiedad	Valor
Label	Labelx	Caption	No.
...
Button	cmdsairFr7	Text	Salir
...
GridEx	grdView	Columns	*Ver imagen

Propiedades principales del formulario frmMain

El formulario llamado frmView inicia la captura de nuevos registros, muestra los registros guardados, borra registros y accede al formulario frmMain para observar la tabla de registros. A continuación se muestra el diseño de frmView y sus propiedades más importantes.

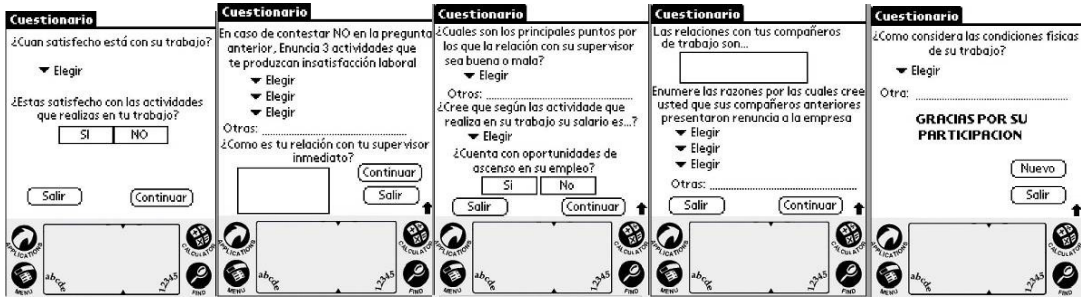


Control	Nombre	Propiedad	Valor
Field	Fidnumcuest	Numeric	True
Selector	Selfecha	Mode	1.-
Selector	Selfecha	Format	dd/mm/yy
Selector	Selhora	DialogTitle	Fecha de aplicación
Selector	Selhora	Mode	2.-
Selector	Selhora	Format	hh:nn
Selector	Selhora	DialogTitle	Hora de aplicación
Field	Txtnombre	Editable	True
Field	Txtpuesto	Editable	True
PushButton	pushsexM	Text	M
PushButton	pushsexF	Text	F
PushButton	pushsexM	Group	1
PushButton	pushsexF	Group	1
Graffiti	Graffiti1		

El resto de controles se componen de etiquetas y botones cuyas propiedades caption y text se modifican de acuerdo con lo mostrado. En HB++ cualquier formulario que implique captura de texto requiere de un indicador de Graffiti¹⁸ o bien, mediante código se agrega una línea de excepción.

El diseño de los siguientes formularios completa esta fase de desarrollo de la aplicación en HB++. Dentro de lo remarcable se encuentran los controles Popup los cuales muestran Elegir como propiedad Text, sin referirse al algún listado de opciones, y los controles List que tampoco se relacionan a alguna lista.

¹⁸ Graffiti es la utilidad de reconocimiento de escritura de Palm OS



Como siguiente paso se define la tabla que generará la base de datos. En el menú Project se agrega una nueva tabla (Add Table). Dicha tabla puede localizarse en el explorador de proyecto.

Las propiedades de la tabla consisten en asignar nombre e ID de creador. Existen 3 propiedades muy interesantes para las tablas las cuales se explican a continuación:

- Backup. Cuando el valor de esta propiedad es True, se realiza un respaldo de la tabla en el equipo de escritorio cada vez que se lleve a cabo una sincronización.
- Bundled. Cuando el valor de esta propiedad es True, se puede transmitir y compartir la tabla junto con la aplicación a otro dispositivo, siempre y cuando el ID de creador de la tabla sea el mismo que el de la aplicación.
- PDB. Esta propiedad define si el compilador crea o no un archivo .pdb para la aplicación y si lo instala para un proceso de depuración.

Properties	
tblEncuesta Table	
+ Misc	
Name	tblEncuesta
Creator	GCRP
Type	DATA
+ Behavior	
Backup	False
Bundled	False
PDB	Create & Install always

Al crear una tabla HB++ define 4 campos que no pueden ser modificados:

Field Name	Data Type	Sync Field	Sync Options	Comment
1 UniqueID	Long	*		Unique identifier
2 Category	Integer	*		Category
3 Dirty	Boolean	*		Modified since last HotSync
4 Secret	Boolean	*		Private record
5 Numcuest	String	*	String:3	numero de cuestionario
6 Fecha	Date	*	Date/Time	fecha de captura
7 hora	Long	*		hora de captura
8 nombre	String	*	String:30	nombre del encuestado
9 puesto	String	*	String:30	puesto del encuestado dentro de la empresa
10 sexo	String	*	String:1	sexo del encuestado
11 preg1	String	*	Memo	respuesta de la pregunta 1
12 preg2	Boolean	*		respuesta de la pregunta 2 Si o NO
13 preg3a	String	*	Memo	primer respuesta de la pregunta 3
14 preg3b	String	*	Memo	...
15 preg3c	String	*	Memo	...
16 preg4	String	*	Memo	...
17 preg5	String	*	Memo	...
18 preg6	String	*	Memo	...
19 preg7	Boolean	*		...
20 preg8	String	*	Memo	...
21 preg9a	String	*	Memo	...
22 preg9b	String	*	Memo	...
23 preg9c	String	*	Memo	...
24 preg10	String	*	Memo	...

- UniqueID. Valor numérico único que identifica cada registro
- Secret. Valor que indica si un registro es visible o no.
- Dirty. Valor que indica si el registro fue modificado desde la última sincronización. El valor de este campo ayuda a un conducto a

determinar que tipo de sincronización llevar a cabo.

- Category. Campo que sirve de ayuda para la búsqueda y ordenamiento de registros.

Los registros correspondientes a la base de datos del Cuestionario se añaden como se muestra en la imagen. Una vez terminado del diseño de la tabla queda

lista para introducir datos (si se desea que la aplicación inicie con registros precargados) en la pestaña Data.

Antes de codificar directamente los formulario conviene definir clases y módulos en el proyecto. Como para todos los componentes del proyecto, se añaden por medio del menú Project.

En el proyecto se definen una clase y un módulo de la siguiente manera:

Clase	clsApplication	Extends: Application	Instantiable: True
Módulo	Module1	Segment: main	

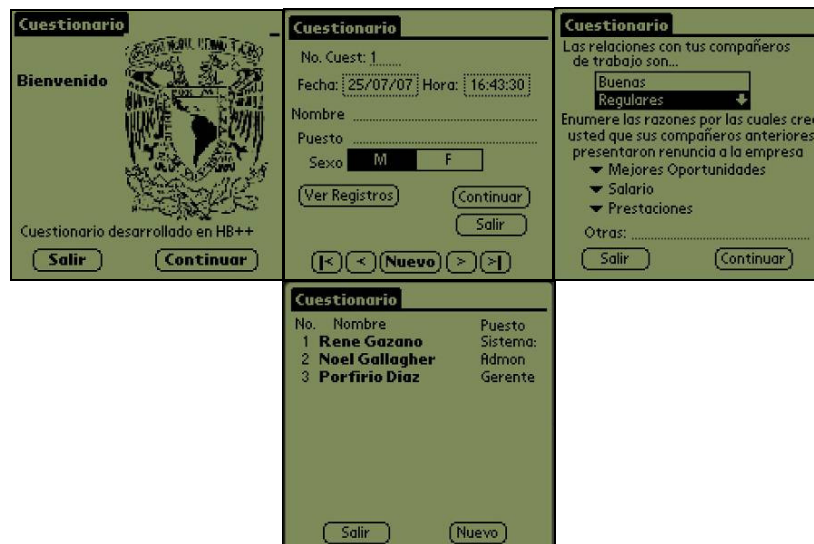
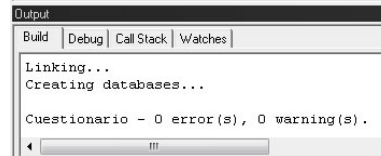
El código que corresponde a este proyecto se encuentra y se describe en el apéndice.

Al configurar el Palm OS Emulador o Simulator la aplicación terminada puede probarse con el comando Execute (F5) y HB++ mostrará la aplicación trabajando sobre la herramienta elegida.

Antes de crear el archivo ejecutable conviene revisar las opciones de la aplicación por medio del Menú Project > Settings donde se puede elegir el nombre para la aplicación, la versión mínima de Palm OS para ejecutarse, el nombre de la base de datos y el ID de creador de la aplicación entre otras propiedades.

Una vez concluido el proyecto se encuentra listo para crear el ejecutable mediante el menú Project > Build (F7). Los resultados de la compilación se muestran en la ventana de salida que se encuentra en la parte inferior del IDE.

El archivo .prc creado por HB++ se encuentra en el directorio de trabajo. También se crea un archivo .PDB que contiene la tabla con la que trabajará la aplicación. Ambos archivos deben instalarse al dispositivo para que la aplicación se ejecute correctamente.



Vistas de la aplicación en ejecución

Debido a la forma similar de programación con herramientas como Visual Basic, HB++ ha ganado muchos adeptos. HB++ es una herramienta muy

completa ya que explota muy bien las habilidades del Sistema Operativo de Palm como el hardware en el que residen la aplicaciones. Otro factor importante es que el trabajo con los formularios, tablas, controles y otras clases que conforman un proyecto, se encuentra bien estructurado y siempre administrado por el programador.

El entorno de HB++ cuenta con un gran índice de ayuda y documentación. Si bien no es un manual de usuario, cuenta con mucha información para resolver las dudas en cuanto a sintaxis o al uso de funciones.

Las aplicaciones desarrolladas con HB++ sin licencia también se generan como versiones de prueba. Sin embargo, el IDE permite utilizar el entorno sin restricciones dejando probar todas sus características, lo que permite evaluar la herramienta de mejor manera antes de realizar alguna adquisición.

4.3.4 NS Basic

<http://www.nsbasic.com>

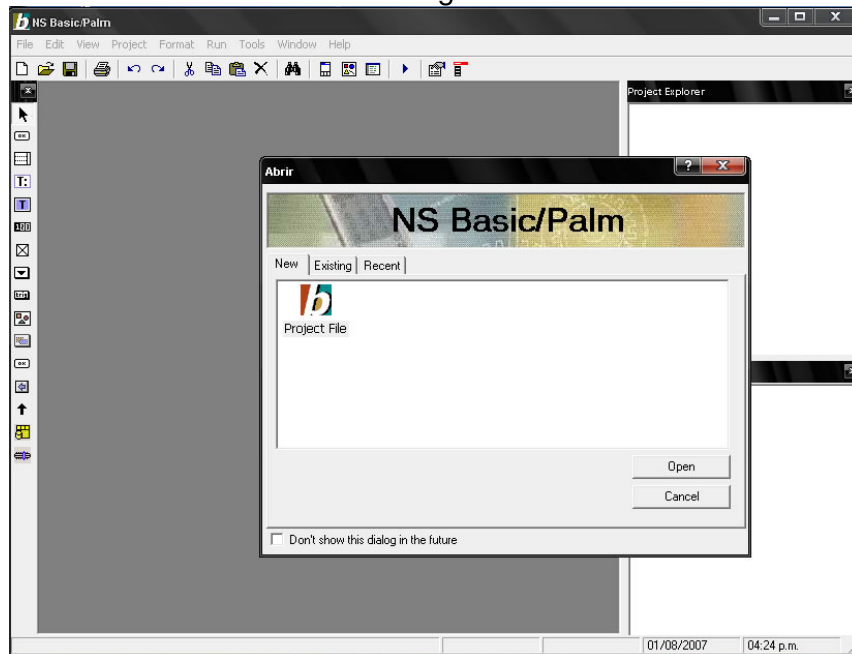
NS Basic Corporation ofrece herramientas para el desarrollo de aplicaciones en 4 plataformas: 1 de escritorio (Windows) y 3 móviles (Windows Mobile, Newton y Palm OS).

Dentro de las características principales de NS Basic para Palm OS se encuentran las siguientes:

- Completamente optimizado para dispositivos con Palm OS 5+
- Lenguaje BASIC estándar y estructurado
- Manejo de comunicaciones serial, infrarrojo e Internet
- Soporte para funciones matemáticas y trigonométricas
- Libre de regalías con respecto a las aplicaciones desarrolladas
- Soporte para código de barras
- Considera el soporte para hardware de Palm, Handspring, Symbol, Kyocera, Sony y Samsung entre otros dispositivos.

Las aplicaciones desarrolladas no requieren ambiente de ejecución (Runtime). Otra de las aportaciones interesantes que conlleva la instalación de NS Basic es que otorga una versión de Palm OS Emulator y Palm OS Simulator para probar las aplicaciones desarrolladas y listas para trabajar, es decir, las rutas de instalación son configuradas automáticamente por NS Basic.

Al ejecutar NS Basic se muestra de la siguiente manera:



Con un entorno de desarrollo muy similar a HB++, NS Basic se divide en tres zonas principales: El explorador de proyecto (parte superior derecha), la ventana de propiedades (parte inferior derecha) y la ventana de diseño acompañada de la barra de herramientas (parte izquierda).

La barra de herramientas de NS Basic muestra los mismos controles que ya se han analizado (etiquetas, botones, listas, campos de texto, selectores, cuadrículas). Sin embargo, es importante conocer que NS Basic asigna un ID

numérico a cada control creado y que podrá cambiarse el nombre del control, pero no el ID asignado por NS Basic. Este ID se utilizará para codificar los procedimientos del control correspondiente.

Al crear un nuevo proyecto NS Basic muestra un formulario en blanco listo para ser diseñado. En las propiedades del proyecto se señala la ruta en la que se almacenará (NS Basic guarda el proyecto en un solo archivo con extensión prj). Así también se indica el ID de creador y la versión de la aplicación además del nombre que llevará en el escritorio de Palm OS una vez instalada.

En NS Basic es importante señalar las instrucciones que se deben ejecutar al iniciar y al terminar la aplicación. Para ello se editan el código de inicio (*Startup Code*) y el código de salida (*Termination Code*).

Los componentes del proyecto se organizan dentro de su clase en el explorador de proyectos.

La siguiente imagen muestra el diseño del formulario de bienvenida a la encuesta, llamado *Cuestionario*. También muestra la organización de los formularios dentro del proyecto y las propiedades del Proyecto



En NS Basic es conveniente realizar la totalidad del diseño y dejar la codificación como parte final. Bajo este esquema se realiza el diseño de las clases, en este caso formularios y el menú.

El diseño del formulario de bienvenida *Cuestionario* consiste básicamente en botones y etiquetas. En el caso del control Bitmap, los pasos para manejar imágenes en NS Basic son dos:

1. En el explorador de proyecto se agrega un archivo de imagen (Add Bitmap). Una vez agregada, NS Basic asigna un ID a dicho objeto.
2. En el formulario se inserta un control Bitmap. Para relacionar la imagen agregada con el control Bitmap se utiliza el ID mencionado en el primer paso y se asigna a la propiedad *Bitmap ID* del control.

Para el diseño del formulario de captura *Cuestionario1* se añaden etiquetas, botones, selectores, campos de texto y un par de *push buttons*. La tabla siguiente comenta las propiedades más importantes.

Control	Nombre	ID	Propiedad	Valor
Field	Fidnumcuest	1034	Numeric	True
Selector	Selfecha	1036	Label	
Selector	Selhora	1037	Label	

Field	Fldnombre	1038	Numeric	False
Field	Fldpuesto	1040	Numeric	False
Pushbutton	PushsexF	1043	Label	F
Pushbutton	PushsexM	1042	Selected	True
Button	Cmdshowregs	1051	Label	Ver registros
Button	Cmdnextfr1	1053	Label	Continuar
Button	Cmdborrar	1050	Label	Borrar
Button	Cmdsalirfr1	1070	Label	Salir
Button	Cmdfirstrec	1044	Label	<
Button	Cmdprevrec	1054	Label	<
Button	Cmdnuevo	1049	Label	Nuevo
Button	Cmdnextrec	1056	Label	>
Button	Cmdlastrec	1055	Label	>

El ID de cada control lo asigna NS Basic llevando el orden en el que fueron creados.

El siguiente formulario en importancia en el diseño del cuestionario es aquel donde se mostrarán los registros llamado *Cuestionarios*. Para ello se utiliza el control Grid modificando propiedades de formato como ancho y alto de la cuadrícula y el número de columnas (Cols). En el diseño de los formularios que contendrán las preguntas cabe destacar la propiedad List de los controles PopUp, ya que permite agregar las opciones deseadas directamente en la ventana de propiedades. Por otro lado, se agregan campos de texto en algunas preguntas en caso de que la respuesta recibida no se encuentre dentro de las opciones y se desee ampliar el rango de información. El diseño de los formularios se muestra a continuación:

Cuestionario

Numero de Cuestionario <fldnumcue>
Fecha: Hora:

Nombre <fldnombre>
Puesto <fldpuesto>

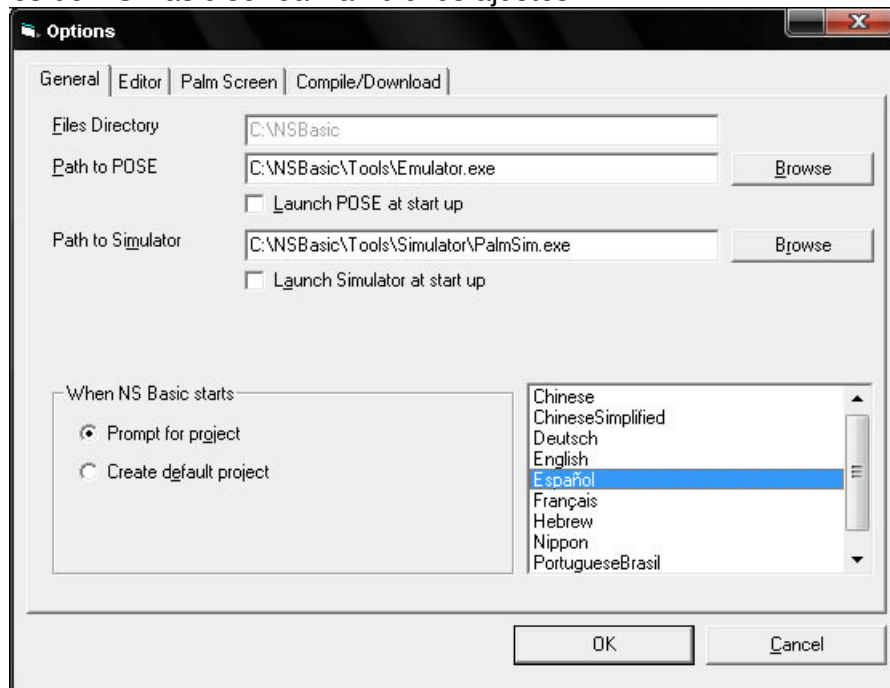
Sexo M F

<div style="border: 1px solid black; padding: 5px;"> <p>Cuestionario</p> <p>1.- ¿Cuan satisfecho está con su trabajo? ▼Elegir</p> <p>2.-¿Estas satisfecho con las actividades que realizas en tu trabajo? <input type="button" value="SI"/> <input type="button" value="NO"/></p> <p><input type="button" value="salir"/> <input type="button" value="Continuar"/></p> </div>	<div style="border: 1px solid black; padding: 5px;"> <p>Cuestionario</p> <p>En caso de contestar NO en la pregunta anterior, enuncia 3 actividades de tu trabajo que te produzcan insatisfacción laboral: ▼Elegir ▼Elegir ▼Elegir Otras: <fldpreg3d></p> <p>¿Como es la relación con tu supervisor inmediato? ▼</p> <p><input type="button" value="salir"/> <input type="button" value="Continuar"/></p> </div>	<div style="border: 1px solid black; padding: 5px;"> <p>Cuestionario</p> <p>¿Cuales son los principales puntos por los que la relación con su supervisor sea buena o mala? ▼Elegir Otros: <fldpreg5></p> <p>Cree que segun las actividades que realiza en su trabajo su salario es... ▼Elegir</p> <p>¿Cuenta con oportunidades de ascenso en su empleo? <input type="button" value="SI"/> <input type="button" value="NO"/></p> <p><input type="button" value="salir"/> <input type="button" value="Continuar"/></p> </div>																																	
<div style="border: 1px solid black; padding: 5px;"> <p>Cuestionario</p> <p>Las relaciones laborales con sus compañeros de trabajo son: ▼Elegir</p> <p>Enumere 3 razones por las cuales cree Usted que sus compañeros anteriores presentaron renuncia a la empresa: ▼Elegir ▼Elegir ▼Elegir Otras: <fldpreg3d></p> <p><input type="button" value="salir"/> <input type="button" value="Continuar"/></p> </div>	<div style="border: 1px solid black; padding: 5px;"> <p>Cuestionario</p> <p>¿Como considera las condiciones físicas de trabajo? ▼Elegir Otras: <fldpreg10></p> <p style="text-align: center;">GRACIAS POR SU PARTICIPACION</p> <p><input type="button" value="Nuevo"/> <input type="button" value="salir"/></p> </div>	<div style="border: 1px solid black; padding: 5px;"> <p>Cuestionarios</p> <table border="1" style="width: 100%; height: 100px; border-collapse: collapse;"> <tr><td> </td><td> </td><td> </td></tr> <tr><td> </td><td> </td><td> </td></tr> <tr><td> </td><td> </td><td> </td></tr> <tr><td> </td><td> </td><td> </td></tr> <tr><td> </td><td> </td><td> </td></tr> <tr><td> </td><td> </td><td> </td></tr> <tr><td> </td><td> </td><td> </td></tr> <tr><td> </td><td> </td><td> </td></tr> <tr><td> </td><td> </td><td> </td></tr> <tr><td> </td><td> </td><td> </td></tr> <tr><td> </td><td> </td><td> </td></tr> </table> <p><input type="button" value="salir"/> <input type="button" value="Nuevo"/></p> </div>																																	

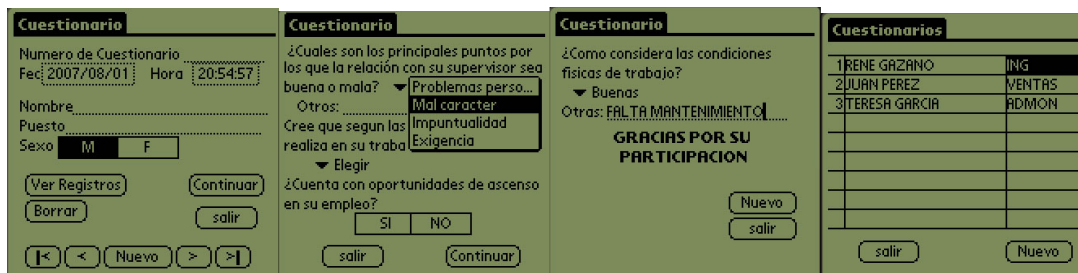
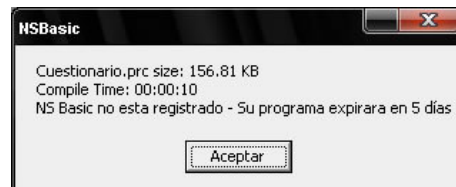
Por último, el diseño del menú (o los necesarios dependiendo de la aplicación) se realiza con el editor de menús de NS Basic. El nombre del menú es la referencia para cargar el menú pero medio del código.

Una vez terminada la fase de diseño se procede a editar el código. El código correspondiente a este proyecto se encuentra y se describe en el apéndice.

Una vez concluído el proyecto conviene revisar las opciones del IDE para observar dónde se realizarán las pruebas. En el menú Herramientas > opciones de NS Basic se realizan dichos ajustes.



El siguiente paso es la compilación (F5). Si la opción elegida para probar las aplicaciones es el Palm OS Emulator, este debe estar abierto antes de la compilación. En caso de presentarse alguna falla, NS Basic indica el error esperando su depuración. Si no hay errores de compilación se observa el siguiente mensaje:



Vistas de la encuesta en Palm OS Emulator

Dentro de las ventajas que ofrece NS Basic se encuentra el hecho de poder manejar todo el proyecto con un sólo archivo. Así mismo sólo genera un archivo prc y un archivo pdb que además es opcional.

Por otro lado el IDE de NS Basic cuenta con un índice de ayuda muy completo y algunos ejemplos de desarrollo en línea lo cual hace más sencillo el inicio en este ambiente.

4.3.5 Satellite Forms

<http://www.satelliteforms.net>

Satellite Forms es un entorno integrado de desarrollo que crea aplicaciones para dispositivos Palm y Pocket PC. Permite crear aplicaciones funcionales sin usar código. Además es posible desarrollar aplicaciones más sofisticadas o de mayor alcance implementando scripts o código de forma mínima.

Satellite Forms tiene cuatro componentes principales:

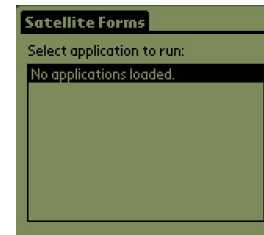
1. MobileApp Designer. Es el entorno de desarrollo de Sat Forms donde se diseñan las aplicaciones. Permite crear aplicaciones con múltiples formularios y tablas.
2. El conducto. Administra la transferencia de datos entre los dispositivos PDA y el equipo de escritorio.
3. Control ActiveX de Sat Forms. Trabaja con la tecnología de sincronización Hotsync para simplificar la integración de las aplicaciones con las bases de datos.
4. La máquina de de Sat Forms (runtime). Sobre esta máquina se ejecutan las aplicaciones en los dispositivos PDA.

Sat Forms soporta Palm OS desde la versión 3.5 hasta la Palm OS Garnet.

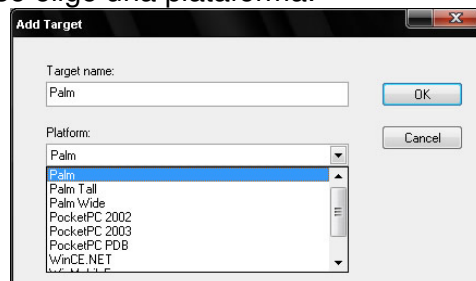
El archivo de instalación de Satellite Forms está disponible en el sitio web. Sin embargo, para obtener un periodo de prueba de Sat Forms es necesario registrarse y esperar una clave de activación. El periodo de prueba dura 30 días.

El proceso de instalación comprende 2 fases:

1. Instalar el MobileApp Designer (IDE de Satellite Forms) en el equipo de escritorio, siguiendo los pasos del asistente de instalación.
2. Instalar la máquina de Satellite Forms en el PDA (runtime). Esta máquina se encuentra en Satellite Forms 7\Runtime\Palm de la ruta de instalación. Puede instalarse como cualquier aplicación mediante HotSync. Una vez instalada en el Palm OS se muestra como en la siguiente imagen.

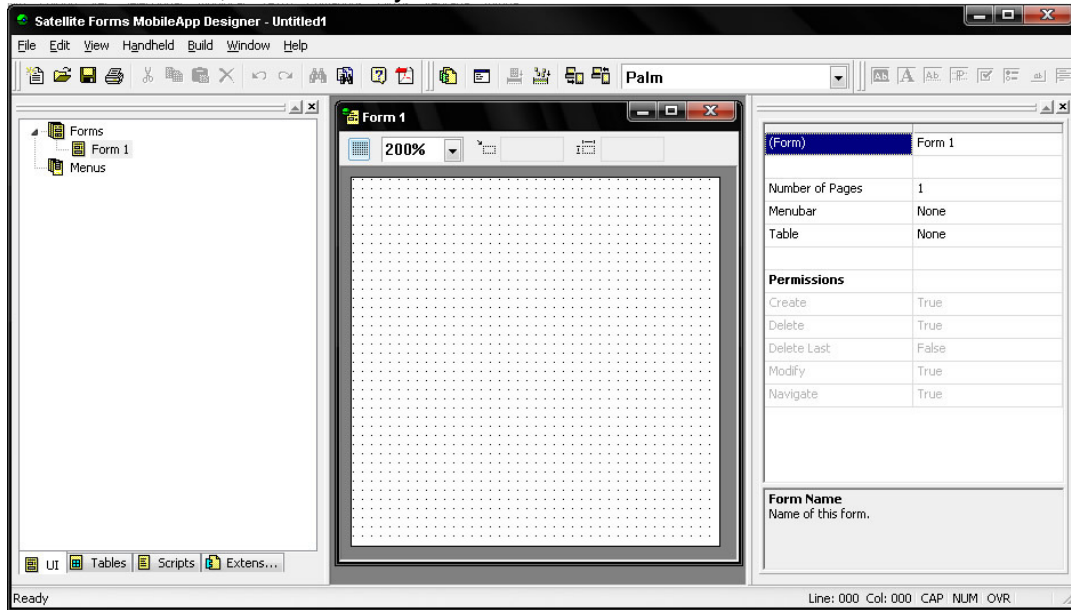


Por su parte, el MobileApp Designer se encuentra listo para trabajar. Se crea un nuevo proyecto y se elige una plataforma:



Una vez elegida la plataforma objetivo, el IDE de Satellite Forms se muestra como en la siguiente imagen. El IDE se divide principalmente en 3 partes:

1. El contenido del proyecto (Parte izquierda). Los componentes del proyecto (Formularios y Menús). Conforme avanza el desarrollo, los controles de los formularios se van mostrando a manera de árbol.
2. La zona de diseño de formularios (Parte central).
3. La paleta de propiedades (Parte derecha). Las propiedades de los controles, formularios y menús se listan en esta área.

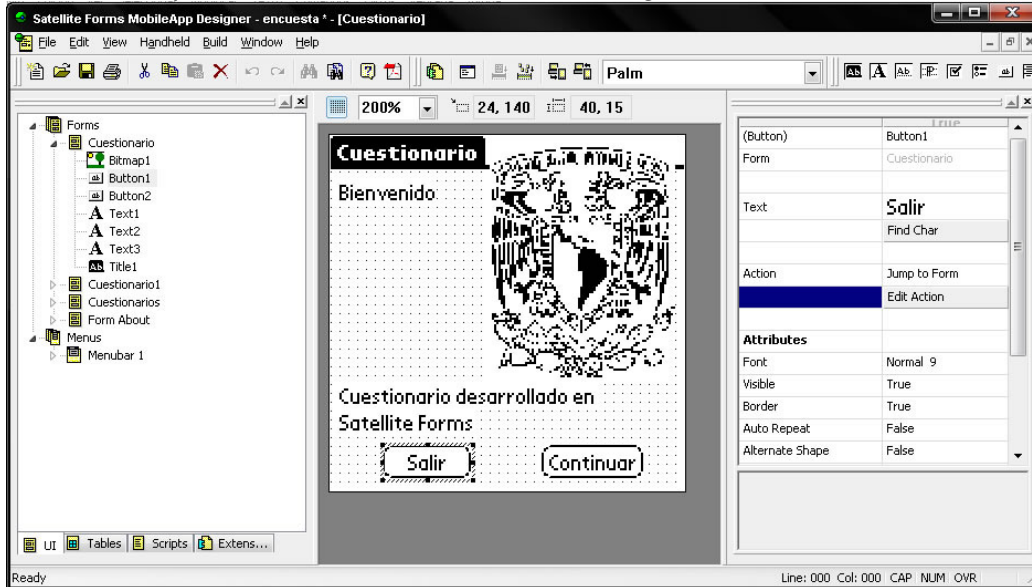


Vista general de Sat Forms.

La barra de herramientas de controles muestra los posibles elementos que pueden contener las aplicaciones, propiamente dicho, los formularios. Éstos elementos son los controles estándar que ya se han manejado como etiquetas (text control), botones, campos de texto (edit control), cajas de lista (listbox) y listas expandibles (drop list). En Sat Forms cabe destacar los siguientes controles:

- Title Control: mediante este control se asigna título a un formulario.
- Paragraph Control: Este control es de gran ayuda para notas o memos.
- Lookup Control: Sirve para realizar búsquedas de información entre los registros de la tabla principal.

El diseño del primer formulario se realiza de la siguiente forma:



Control	Nombre	Propiedad	Valor
Bitmap	Bitmap1	ImageFile	UNAM.bmp
Button	Button1	Action	Jump to form
Button	Button2	Action	Jump to form
Text	Text1	Text	Bienvenido
Form	Cuestionario	NumberOfPages	1
Form	Cuestionario	Menubar	Menubar1
Form	Cuestionario	Table	None

La propiedad Action de los botones contiene acciones comunes predefinidas (que con otras herramientas sería necesario codificar). Entre estas acciones se encuentran la creación, modificación, borrado y exploración de registros, dirigir a otro formulario, a otra página, correr un script o cargar otra aplicación.

La imagen relacionada a la propiedad ImageFile de Bitmap1 debe encontrarse en el mismo directorio donde se aloja el proyecto. Respecto a los botones, para dirigirse a otro formulario es necesario haber creado antes los formularios destino. Es el mismo caso de la barra de menú.

Cuando se diseña una aplicación como la deseada es muy común distribuirla en muchos formularios, sin embargo Sat Forms propone un manejo de páginas. Un formulario puede contener tantas páginas como sea necesario. En este caso, el diseño del cuestionario puede diseñarse en un solo formulario desde la primera hasta la última pregunta.

El primer paso para el diseño de este formulario, que contendrá la información principal de la aplicación, será el diseño de la tabla principal. En el contenido del proyecto se inserta una tabla (cambiando la pestaña de la parte inferior). Se asigna un nombre a la

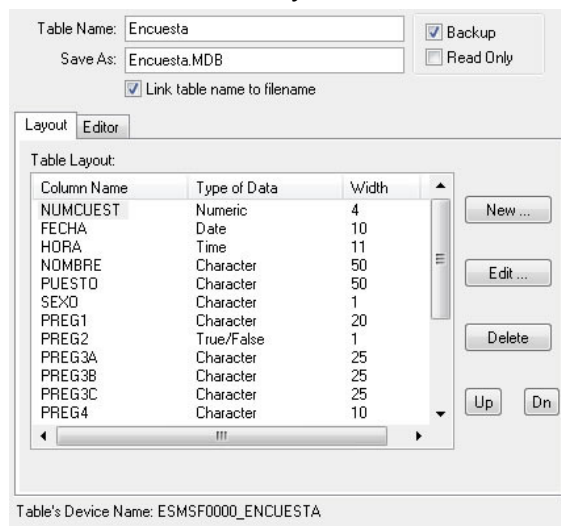
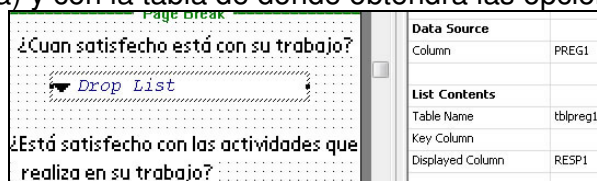


tabla (Encuesta), se nombra el archivo MDB¹⁹ que será el respaldo en el equipo de escritorio y se determina si la tabla será de lectura/escritura o sólo lectura. Por último se agregan las columnas, el tipo de datos que contendrán y el tamaño de la columna. Una vez terminado este proceso, los campos quedan listos para relacionarse con los controles de los formularios.

El diseño del formulario siguiente, Cuestionario1, contiene 6 páginas y se relaciona con la tabla Encuesta por medio de la propiedad Table. Se asigna verdadero a las propiedades de Crear, Borrar, Modificar y Navegar de este formulario.

Entre lo sobresaliente del diseño de este formulario y sus páginas se encuentran las siguientes propiedades:

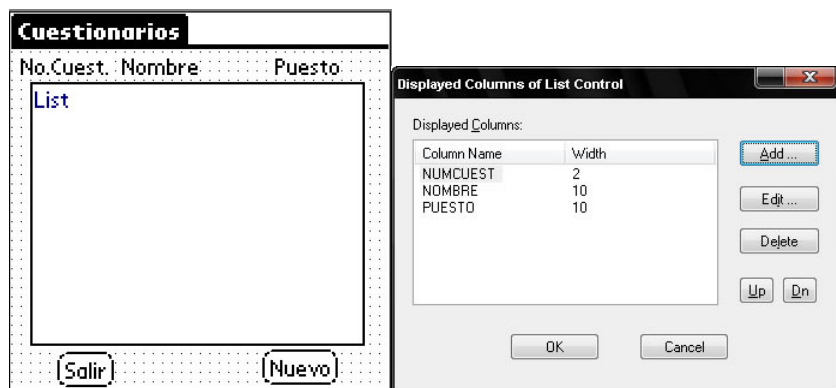
- La propiedad Autokeyboard permite elegir rápidamente una fecha, hora o número. Para los controles edit de Fecha y Hora el valor de Autokeyboard es Date y Time respectivamente.
- En cuanto a los controles Drop List se debe definir una tabla con los valores de las respuestas posibles a dichas preguntas. El control se relaciona con el campo en donde se almacenará la respuesta (Encuesta) y con la tabla de donde obtendrá las opciones.



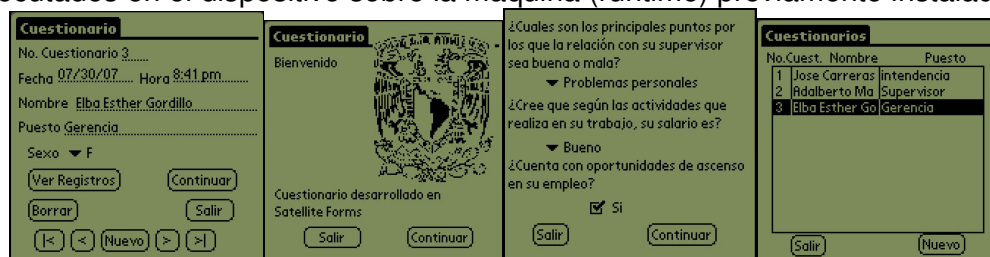
- La propiedad Action del botón salir lanza otra aplicación. Es necesario proporcionar el nombre del ejecutable como archivo.prc.

Para el formulario que muestra los registros en una tabla, Cuestionarios, se usa una caja de lista que se modifica para que contenga 3 columnas de la tabla Encuesta. Se proporcionan las columnas y el ancho que ocuparán al mostrarse.

¹⁹ Formato estándar de dBase y Microsoft Access



Al terminar el diseño se compila la aplicación mediante la opción Rebuild All (F7). Para probar el funcionamiento de la aplicación se elige Download App & Tables (F5). Cabe señalar que el proceso de compilación en Sat Forms genera además de los archivos pdb²⁰ y prc, las tablas con la extensión MDB para poder modificarlas en el escritorio con herramientas como dBase o Microsoft Access o bien, convertirlas y trabajar con cualquier otro DBMS. Es importante recordar que los archivos prc y pdb generados por Sat Forms sólo pueden ser ejecutados en el dispositivo sobre la máquina (runtime) previamente instalada.



Vistas de la encuesta en Palm OS Emulator

La rapidez para desarrollar aplicaciones con Sat Forms, es similar a la proporcionada por PDA Toolbox como herramienta RAD. Entre las ventajas que aporta Sat Forms es la capacidad de sincronizar las tablas con bases de datos de escritorio sin tener que crear un conducto previamente. Sin embargo también es una desventaja, al menos en la versión de prueba, que cada tabla genere un archivo pdb. Cuando el proyecto cuenta con muchas tablas, el proceso de compilación puede arrojar más de un archivo pdb, que son imprescindibles para que las aplicaciones de bases de datos funcionen correctamente.

En general es un ambiente amigable y fácil de usar. Para este caso no se requiere utilizar código, por lo que se prevee que el uso del mismo podría proporcionar aplicaciones más poderosas. A pesar del uso de un ambiente de ejecución, este presenta ventajas como la de proteger las aplicaciones de la empresa de posible fuga de información. En general Satellite Forms es una herramienta imprescindible en el aprendizaje de la programación de PDA's.

²⁰ Satellite Forms genera no sólo un archivo pdb, sino uno por cada tabla utilizada en el proyecto

4.3.6 Superwaba

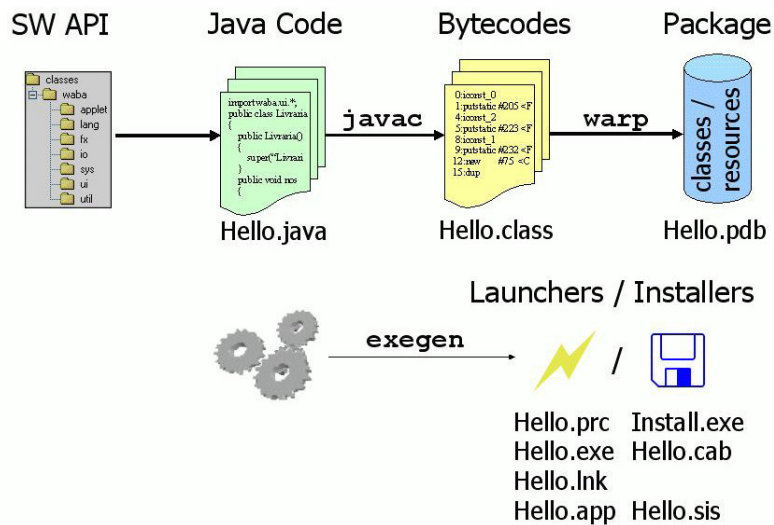
www.superwaba.com.br

Superwaba es una máquina virtual (similar a Java) para dispositivos móviles. El desarrollo de software se realiza usando las API's, accediendo por medio de bibliotecas asociadas (empaquetadas como archivos .jar) y pequeñas herramientas (que juntas hacen un SDK, conjunto de desarrollo de software) para crear aplicaciones que pueden ejecutarse sobre la máquina virtual en las plataformas soportadas.

Superwaba es un completo conjunto de desarrollo de software para sistemas operativos móviles. Entre los lenguajes y entornos de programación mencionados en este trabajo, cabe destacar a Superwaba como la mejor herramienta debido a:

- Su capacidad para elaborar grandes aplicaciones explotando los recursos posibles dentro de un dispositivo móvil
- Es una herramienta multiplataforma que cubre los tres principales sistemas operativos móviles (Palm OS, Windows Mobile y Symbian OS).
- Debido a su similitud, Superwaba resulta muy sencillo para los programadores de Java de Sun Microsystems.
- Posee una máquina virtual reducida que permite ejecutar aplicaciones en un dispositivo móvil aún mejor que la desarrollada por Java.
- Es compatible en un 99% con Java²¹ y puede trabajarse perfectamente con los ambientes de desarrollo Java más conocidos como Eclipse, Net Beans, Visual J++, Visual Café o J Builder. Además pueden tomarse los mismos IDE como ambientes de prueba.
- Es una herramienta Open-Source impulsada por los desarrolladores de Eclipse.

La siguiente imagen describe el proceso de desarrollo de aplicaciones con Superwaba:



Como se aprecia, el proceso de desarrollo es similar al de Java, codificando y generando las clases con .java, compilando y creando los archivos .class. Por

²¹ El único bytecode que no se implementa es Synchronized

medio de las herramientas Warp y Exegen se crean los archivos .pdb y .prc necesarios para ejecutar las aplicaciones en Palm OS.

El conjunto de herramientas de desarrollo (SuperwabaSDK) se encuentra disponible en www.superwaba.com.br Además del SuperwabaSDK se encuentran disponibles las máquinas virtuales (VM) para distintas plataformas. Cabe señalar que para compilar las aplicaciones y llevar a cabo la parte final, es decir, la creación de los archivos .prc y .pdb es necesario de un JDK²². Una vez instalada, Superwaba se presenta con los siguientes directorios:

Carpeta	Descripción
/bin	Se encuentran en ella paletas de colores, instaladores de las máquinas virtuales y las herramientas Warp y Exegen
/doc	Contiene la documentación de Superwaba, así como ejemplos. Contiene además el útil tutorial Superwaba Companion.
/lib	Se encuentran las bibliotecas .jar que se utilizan en el desarrollo de aplicaciones. En la misma se encuentran algunos ejemplos compilados.
/src	En esta carpeta yacen los paquetes y bibliotecas de Waba.

El API del lenguaje se compone de 7 paquetes con un total de 97 clases:

Paquete	Descripción
waba.fx	Este paquete contiene las bibliotecas de la parte gráfica y de sonido de Superwaba. Se encuentran las clases Color, Font, Image Pallette, Rect, Sound, SoundClip y Graphics.
waba.io	Es el conjunto de clases que se usan para la entrada y salida, así como la lectura de los archivos PDB (Catálogos) a través de ByteArrayInputStream, Catalog , DataStream, Socket, SerialPort y ResizeStream
waba.ui	Las clases más importantes, mismas que permiten diseñar la interfase grafica de las aplicaciones Button, Edit, Label, Container, ListBox, MessageBox, MenuBar, MainWindow .
waba.sys	Contiene métodos para conocer las características y configuración del S.O. También ayuda en el manejo de hilos
waba.util	Contiene clases que son útiles como el Vector, hashTable ,Date ,Random.
waba.lang	Este paquete se agrega implícitamente en toda clase que se este desarrollando la clase Object, RuntimeException, String,StringBuffer
waba.applet	Este paquete se usa para emular Superwaba cuando se ejecuta en el equipo de escritorio (no se ejecuta en el dispositivo)

Como se mencionó, es posible configurar Superwaba para trabajar con distintos ambientes de desarrollo. En este caso se eligió Net Beans por simple comodidad.

Como paso en común en todos los ambientes de desarrollo, es necesario agregar la biblioteca Superwaba.jar para poderla utilizar. Al crear un proyecto es necesario incluir esta biblioteca, utilizar como clase principal waba.applet.Applet y establecer la resolución de la aplicación. Su sintaxis es la siguiente :

²² Se recomienda la versión 1.2, sin embargo, también es compatible con 1.1 y para versiones 1.3 en adelante se debe añadir "-target 1.1" a la línea de comando.

/w val /h val1 /bpp val2 /scale val3 [paquete].[clase principal de la aplicación]

Las resoluciones más utilizadas en Palm OS se muestran a continuación:

Sony Clié 320x320:

/w 320 /h 320 /scale 1 /useSonyFonts /bpp 8 [paquete].[clase principal]

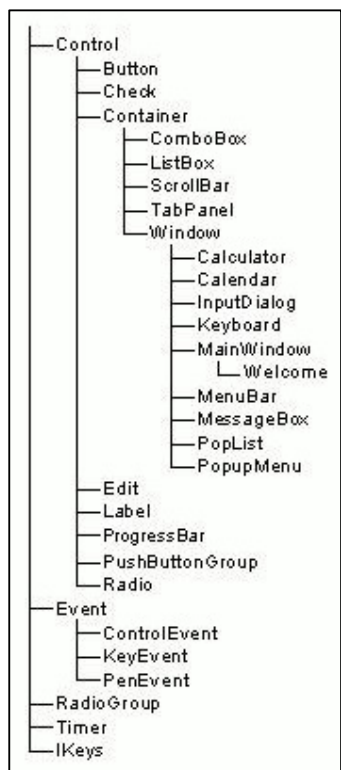
Palm OS 5:

/w 320 /h 320 /scale 1 /bpp 8 [paquete].[clase principal]

Palms zire 72:

/w 160 /h 160 / scale 1 / bpp8

Es importante señalar que con Superwaba no se dispone de una barra de herramientas en la cual se puedan elegir y agregar los controles, sino que se hace mediante código en un archivo .java.



En el esquema de la izquierda se observa la jerarquía de las clases en el paquete waba.ui. Como se aprecia, Superwaba posee más controles para diseñar aplicaciones que las herramientas anteriores. A continuación se describe brevemente cada uno de ellos.

- Button – Botón con texto o imagen.
- Calculator – PopUp usado cuando un campo tiene el modo CURRENCY (Moneda) y se llama el teclado de Palm abc/123.
- Calendar – PopUp utilizado cuando un campo tiene el modo DATE (Fecha) y se llama al teclado de Palm abc/123.
- Check – Muestra un control Checkbox con un texto.
- ComboBox – Muestra una lista expandible.
- Container – Es un control que puede contener a otros controles. En el desarrollo de una aplicación es muy importante ya que los formularios extienden a esta clase.
- Control – La clase madre de todos los controles de este paquete.
- ControlEvent – Encapsula el evento enviado por muchos controles.
- Edit – Campo de texto editable.
- Event – La clase madre de todos los eventos.
- Grid – Una tabla para mostrar datos.
- InputDialog – PopUp utilizado para obtener información proporcionada por el usuario.
- Label – Etiqueta con texto de solo lectura.
- ListBox – una lista de opciones.
- MainWindow – la ventana principal, donde todos los eventos son enviados, todos los programas en SuperWaba deben tener una y solo una ventana principal.
- MenuBar – Barra de menú que se muestra en la ventana principal.

- **MessageBox** – Ventana que muestra un texto y un conjunto de botones configurables (Con un botón OK por defecto).
- **PenEvent** – Evento manejado por la máquina virtual cuando se usa el stylus en la pantalla del dispositivo.
- **PopList** – Muestra una ventana con una lista dentro de ella. Es usado por la clase **ComboBox**.
- **PopupMenu** – Despliega un menú cuando su propiedad **Caption** es seleccionada en la barra de menú.
- **ProgressBar** – Barra utilizada para mostrar el progreso de alguna acción.
- **PushButtonGroup** – una matriz de botones seleccionables.
- **Radio** – un botón de radio con texto asociado.
- **RadioGroup** – Mantiene un conjunto de botones de radio relacionados y sincronizados de tal manera que solo se pueda elegir uno.
- **ScrollBar** – Barra de desplazamiento horizontal o vertical.
- **TabPanel** – Un contenedor con separadores. Cada separador tiene un contenedor donde se pueden añadir otros controles.
- **Timer** – Evento manejado por la máquina virtual cuando algún temporizador definido ha terminado su conteo.
- **Welcome** – La aplicación que se llama cuando el usuario da clic en el icono de Superwaba.
- **Window** – Padre de todas las demás ventanas y de la ventana principal. Solo debe ser extendida por ventanas tipo popup.

Cada programa de Superwaba debe tener una y solo una clase que extiende a `waba.ui.MainWindow`. `MainWindow` es la interfase entre la máquina virtual y el programa de Superwaba ya que recibe los eventos y los atiende.

Lo anterior se ilustra de la siguiente manera:

```
import waba.ui.*;
public class Ejemplo extends MainWindow
{
public Ejemplo()
    {
        super("Este es el titulo", TAB_ONLY_BORDER);
    }
public void onStart()
    {
        // añadir los controles aqui
    }
public void onExit()
    {
        // sentencias para cerrar la aplicación aqui
    }
}
```

La máquina virtual llama al constructor por defecto de `MainWindow` y después llama al método `onStart` (que pertenece a la clase `Container` y también puede ser usado para extender hacia estas). El método `onExit` es llamado cuando la máquina virtual termina en una situación normal, es decir, si el usuario indica el cierre de la aplicación. Sin embargo, si existe algún error fatal y termina la aplicación, el método `onExit` no es llamado.

El siguiente paso es agregar controles. Los controles siempre son añadidos a la clase `Container` o algún descendiente de ésta como las clases `Window` y `TabPanel`. Este contenedor es llamado el padre del control. Para definir los

límites del control se usa el método `setRect(x,y,ancho,alto)`. Antes de llamar al método `setRect` el control debe haber sido añadido y tener sus características.

Existe una forma más simple:

```
add(Control control, int x, int y):
```

que es lo mismo que:

```
add(control);
```

```
control.setRect(x, y, PREFERRED, PREFERRED);
```

Cada interacción del usuario con el programa genera un evento. Hay dos tipos de eventos básicos: eventos del stylus (Pen events) y de botones (key events). A continuación se describe como son manejados por la clase `Window` y enviados al control destino:

1. El usuario oprime un botón en el dispositivo. La máquina virtual intercepta la instrucción y la envía a la clase `Window` la cual crea un `KeyEvent` y lo envía al control que actualmente posee el enfoque.

2. El usuario presiona la pantalla del PDA generando un punto de coordenadas `x,y`. La máquina virtual envía el evento a la clase `Window` que a su vez busca el control cuyos límites contienen ese punto, convierte el punto a las coordenadas del control local y atiende un evento del stylus (`PenEvent`).

Los dos eventos descritos son manejados por controles que a su vez los convierten en otros eventos, definidos en la clase `ControlEvent(PRESSED, FOCUS_IN, FOCUS_OUT, TIMER, WINDOW_CLOSED, WINDOW_MOVED,)`.

Las aplicaciones normales de Superwaba necesitan mostrar controles en la pantalla basados en el estado actual de la aplicación. Debido a esto es importante el uso de contenedores lanzados por la ventana principal. Lo óptimo es diseñar formularios que extiendan a la clase `Container` con sus propios controles y métodos como se ilustra a continuación:

```
public class Cuestionario1 extends waba.ui.Container
{
    public void onStart()
    {
        // aquí se agregan los controles
    }
    public void onEvent(Event e)
    {
        // aquí se agregan instrucciones para los posibles eventos
    }
}
```

Para llamar a ese contenedor de la ventana principal se define de la siguiente forma:

```
public class Encuesta extends waba.ui.MainWindow
{
    static Cuestionario1 cuest1;

    public Encuesta()
    {
        super("Encuesta", TAB_BORDER);
        setDoubleBuffer(true);
    }
    public void onStart()
    {
```

```

Rect r = getClientRect();
cuest1 = new Cuestionario1();
cuest1.setRect(r);
swap(cuest1); // <-- aquí se llama al método onStart de cuest1
}
}

```

Bajo la lógica descrita se muestran y se comentan los aspectos más importantes de la aplicación Encuesta en Superwaba en el apéndice.

Una vez terminada la aplicación es conveniente realizar pruebas en el propio IDE (en este caso NetBeans) y una vez depurada y lista se utilizan los métodos Warp y Exegen para generar el archivo PDB y el archivo “lanzador” PRC. Como se mencionó, este paso requiere de una versión de Java Development Kit.

Warp genera un archivo PDB que contiene cualquier archivo de clase que tenga dependencias, solo habrá que especificar cual es la clase principal y todo será añadido automáticamente. Si no se especifica la clase principal, Warp busca un archivo .class con el mismo nombre que el archivo PDB que se está creando.

La sintaxis para Warp en la línea de comandos de java es la siguiente:

```
java superwaba.tools.Warp command [options] pdbfile [files]
```

donde:

command

c crea un nuevo archivo pdb

l lista el contenido de un archivo pdb

options (por mencionar algunas)

/c sobrescribe y asigna un ID de creador a la base de datos

/r Si se especifica un directorio recurre a cualquier directorio

pdbfile

es el nombre del archivo a crear

files

especifica la clase principal

Exegen genera el archivo .prc ejecutable en Palm OS. Estos ejecutables son utilizados para lanzar (iniciar) un programa de superwaba. Si no se especifica el nombre de un archivo .pdb, su nombre será el mismo que el ejecutable.

La sintaxis para Exegen en la línea de comandos es la siguiente:

```
java superwaba.tools.Exegen [options] exefile [main-class] [pdbfile]
```

donde:

options (por mencionar algunas)

/\$ Coloca el atributo de protección de copia de Palm OS

/INST Crea un archivo .exe para instalar la aplicación en dispositivos Palm OS.

/c sobrescribe y asigna un ID de creador a la base de datos

exefile

es el nombre del archivo a crear

main-class

especifica la clase principal

pdbfile

especifica el nombre del archivo pdb creado anteriormente con Warp.

- Es importante crear primero el archivo pdb con Warp
- Es importante incluir en el classpath del entorno de desarrollo
`/SuperWabaSDK/lib/SuperWaba.jar;/SuperWabaSDK/lib/SuperWabaTools.jar`

Antes de instalar los archivos creados en el paso previo al dispositivo se debe instalar la máquina virtual de Superwaba en éste. Dicho proceso se realiza instalando los archivos localizados en `\SuperWabaSDK\bin\installers\vm\palm` que se listan a continuación

1. Superwaba.prc (máquina virtual)
2. SWNatives.prc (métodos nativos)
3. Superwaba.pdb (Clases API localizado en el directorio xplat)

Hecho esto se procede a instalar los archivos pdb y prc generados por Warp y Exegen respectivamente y se ejecuta la aplicación.

Es necesario recalcar la importancia de Superwaba debido a la versatilidad que proporciona la máquina virtual en cuanto a plataformas. A diferencia de las herramientas anteriores, Superwaba se divisa como una herramienta con futuro debido a los cambios que surgen en las plataformas móviles. Hoy Superwaba genera aplicaciones multiplataforma destacando en los tres sistemas operativos de mayor trascendencia y gracias a ello puede ser considerada aún cuando haya cambios radicales como el que le espera al Palm OS.

Por otro lado, Superwaba permite explotar los recursos de los dispositivos móviles de manera óptima permitiendo usar controles propios tan útiles que algunas otras herramientas ya comienzan a enfocarse en ellos.

La mejor ventaja, tal vez, es la libertad de la licencia que permite probar y desarrollar aplicaciones sin ningún costo.

4.3.7 HandDBase

www.ddh-software.com

HandDBase es la herramienta más útil y sencilla en cuanto a bases de datos se refiere dentro del ambiente de dispositivos móviles. Más que un ambiente de desarrollo de aplicaciones, se trata de un manejador muy completo de bases de datos relacionales con versiones para Palm OS, Windows Mobile, Symbian. Así también es posible utilizar HandDBase como manejador de bases de datos en Windows y Macintosh.

Entre las características que posee HandDBase se encuentra la capacidad de crear bases de datos sobre el dispositivo móvil, y sobre éstas, construir formularios relacionados con los campos para la captura de información. Lo interesante de HandDBase es la facilidad de diseño de los formularios ya que por medio del dispositivo PDA y del stylus es posible diseñar grandes aplicaciones sin necesidad del equipo de escritorio.

HandDBase sobre el Palm OS tiene la capacidad de importar y exportar datos de otras aplicaciones y soporte para código de barras y tarjetas de memoria.

Por otro lado, en el escritorio también es posible diseñar tablas y crear formularios para el PDA por medio de HandDBase Desktop, herramienta que además de las facilidades que aporta la herramienta móvil proporciona las siguientes:

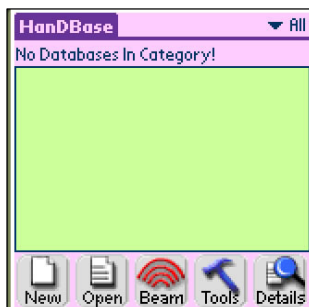
- Importar datos
- Exportar datos a archivos de Excel, Word, Access, archivos XML y HTML.
- Conversión de archivos .pdb a archivos de Excel, Word, Access, archivos XML y HTML y .CSV.

La versión de prueba de HandDBase cuenta con 30 días de permiso. El archivo de instalación se descarga del sitio web de DDH Software considerando las plataformas en las que se trabajará. Dicho archivo instala la aplicación HandDBase Desktop en el equipo de escritorio y las aplicaciones HandDBase y HB Forms en el PDA. En este caso se describirá la forma de trabajar sobre el PDA.

Una vez instalada la aplicación se observan como se mencionó, HandDBase y HB Forms sobre el escritorio del Palm OS.



Antes del diseño de formularios, lo primero es diseñar la base de datos que contendrá la los campos relacionados con los controles. Al ejecutar HandDBase se observa la aplicación lista para crear bases de datos.



Al crear una nueva base de datos (New) se asignan los datos generales de ésta como el nombre y se elige si se sincroniza y se realiza respaldo en el escritorio. Posteriormente HandDBase se prepara para la edición de campos en la tabla.

Al elegir un campo para modificar, se asigna el

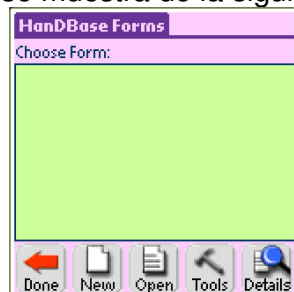
nombre del campo y el tipo de dato que contendrá. En algunos casos se describe el comportamiento del campo ante un posible evento, como el que el usuario elija la fecha y hora en lugar de asignar las actuales. Para las preguntas Si/No, se utiliza el tipo de dato Checkbox y para las preguntas con opción múltiple, se elige el tipo de dato PopUp y se editan los posibles valores. Como ejemplo, las respuestas de la pregunta 1 de la encuesta se muestran a continuación y corresponden al campo Preg1:



Una vez diseñada la tabla se observan los campos y al continuar se muestran las propiedades de la base de datos listas para ser editadas. Entre las propiedades destaca la seguridad de la base de datos y el modo de proteger la información. Por otro lado las vistas, que en realidad son consultas a la tabla.



El siguiente paso es el diseño de los formularios. Por medio del icono Forms de las propiedades, se usa dicha base de datos para diseñar los formularios por medio de los cuales se captura la información. En realidad, dicho icono ejecuta la aplicación HD Forms que se muestra de la siguiente manera:



La creación de formularios es un proceso sencillo de relación entre los controles y los campos de la base de datos diseñada previamente. Entre los controles disponibles se encuentran botones, casillas de verificación, campos de texto, imágenes, etiquetas, listas y listas expandibles(popup).

Al crear un formulario se asigna el nombre de éste y algunas propiedades como el título del formulario, y la capacidad de agregar o editar registros a la base. Posteriormente el formulario se muestra en blanco, listo para agregar controles con el botón nuevo (New) o ver las propiedades del control creado (Details). Los controles tienen 4 tipos de propiedades:

1. General. Se refiere al origen de datos del control y el texto a mostrar

2. Layout. Posición y tamaño del control en el formulario
3. Attributes. Se refiere a cuestiones de formato como fuentes y colores
4. Actions. Es la acción que realizará el control. Contiene instrucciones predeterminadas como moverse entre formularios, recorrer registros, exportar e importar datos, imprimir registros entre otras.

El diseño de los formularios se detalla a continuación:

Formulario: Cuestionario0

Boton Salir: Caption= Salir, Action= Save and Close
 Boton Continuar: Caption=Continuar,Action= Jump to form Cuestionario1
 Label Bienvenido, Caption=BIENVENIDO...

Formulario: Cuestionario1

Start Form: New Record=verdadero, Edit Record=verdadero
 Text Entry: Data Source= Numcuest
 Boton Fecha: CaptionSource=Field Value, Data Source=Fecha, Action= Select Date
 Boton Hora: CaptionSource=Field Value, Data Source=Hora, Action= Select Time
 Text Entry: Data Source= Nombre
 Text Entry: Data Source= Puesto
 Text Entry: Data Source= Sexo
 Boton Borrar: Action=Delete Record
 ...

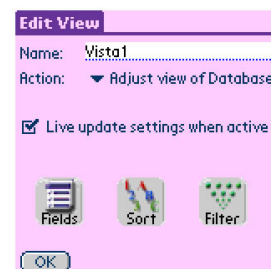
Formulario: Cuestionario2

Start Form: New Record=falso, Edit Record=verdadero
 Popup: Data Source= Preg1, CaptionSource=Field Value
 Checkbox: Data Source= Preg2, Caption Source="Si"
 Boton Continuar: CaptionSource=Continuar, Action= Jump to form>> Cuestionario3
 Boton Salir: CaptionSource=Salir, Action= Save and close

Con un esquema similar se diseñan los formularios restantes que completan las preguntas del cuestionario.

No es posible diseñar una tabla que muestre los registros, sin embargo, HandDBase proporciona una herramienta más útil: las vistas. En las propiedades de la base de datos se pueden editar las vistas que en realidad son consultas a la tabla Encuesta. Es posible elegir las columnas que se quieren mostrar en la vista, realizar ordenamientos y utilizar filtros de registros.

En el caso de la encuesta se requiere de una vista con las columnas Numcuest, Nombre y Puesto. HandDBase carga todos los registros con todas las columnas, sin embargo la vista se diseña para que se observen las columnas necesarias. Si se desean ver otras columnas sobre los registros, basta con hacer un desplazamiento hacia la derecha.



Encuesta			Vista1
N	Nombre	Puesto	Fecha
1	Rene Gazano	Supervisor	8/8/07
2	Artur Rojek	Gerente	8/8/07
3	Maria Gomez	Admon	8/8/07

Home New Tools Find Again

Otra de las habilidades interesantes es que se pueden crear reportes sobre un registro y aprovechar tales datos para realizar operaciones.

Es posible exportar los datos contenidos en la tabla hacia un Memo (Notepad) y bajo este formato ser enviados a algún dispositivo receptor. Este método es muy utilizado en hospitales o restaurantes.

Report for NumCuest		Memo 1 of 8	Sin archivar
Num Entries:	3	HandBase : Encuesta 8/8/07	
Min Value:	1	NumCuest: 1	
Max Value:	3	Nombre: Rene Gazano	
Sum:	6	Puesto: Supervisor	
Min Total:	1	Fecha: 8/8/07	
Max Total:	6	Hora: 4:40 pm	
Average:	2	Sexo: M	
		Preg1: Satisfecho	
		Preg2: TRUE	
		Preg3a: No Value	
		Preg3b: No Value	

Done Export New Report Done Details

HandDBase es una excelente herramienta de diseño para aplicaciones de base de datos que trabaja directamente sobre Palm OS. En cuanto a aprovechar el hardware, HandDBase adapta muy bien el precario espacio de trabajo que proporciona la pantalla de 160x160 de los dispositivos bajo Palm OS para diseñar formularios de manera sencilla con el stylus. El alcance en cuanto a conectividad en la versión profesional es el uso del puerto infrarrojo, suficiente para realizar grandes aplicaciones.

Capítulo V

Conclusiones

Se ha diseñado el cuestionario propuesto en 7 distintas herramientas aprovechando sus ventajas y se han obtenido resultados satisfactorios en todas y cada una de ellas. En cuanto a la utilidad, esta aplicación ahorra tiempo de captura de información, presenta mayor fluidez y se evita demasiada inversión de tiempo en el vaciado de información a un sistema de base de datos.

5.1 Análisis de soluciones

El cuestionario en si es una herramienta con las preguntas exactas y determinantes para realizar un buen estudio sobre la insatisfacción laboral. La interpretación de las encuestas conlleva a los expertos en el tema a realizar recomendaciones en las empresas en las que se realiza con el fin de erradicar esas insatisfacciones tomando las medidas oportunas mediante estrategias y decisiones.

En lo referente al presente trabajo, una encuesta realizada por medio de un PDA con una aplicación como la propuesta reduce el tiempo de captura del 50 al 80 por ciento. Bajo este cuestionario, una encuesta realizada con el método tradicional de recolección de información por medios impresos lleva un tiempo aproximado de 5 minutos. En el supuesto de que se requiera un mínimo de 100 muestras como población, para una persona laborando, el mejor de los casos se presentaría encuesta tras encuesta (sin tiempo de descanso) con un tiempo aproximado de 8 horas con 20 minutos. Para esta misma situación, es decir 100 encuestas, 1 persona laborando y encuesta tras encuesta utilizando una PDA se tiene un tiempo máximo de 4 horas. La aplicación Encuesta contiene muchas preguntas en las cuales se otorgan opciones de respuesta y que están basadas en las respuestas más comunes proporcionadas por la gente. En caso de que la respuesta recibida no se encuentre en las opciones puede tratarse de una respuesta casual o bien, considerar esa respuesta como habitual y añadir esa alternativa a la lista. En caso de que el entrevistado proporcione respuestas comunes el tiempo de la encuesta se reduce a 2 minutos y en caso de proporcionar una respuesta casual el tiempo se eleva a máximo 3 minutos.

El contexto anterior se describe como el mejor de los casos para un encuestador, sin embargo, por diversos factores es prácticamente imposible para una persona realizar encuesta tras encuesta, lo que obligaría a utilizar un mínimo de 2 días de trabajo a esa persona o bien utilizar más personas en la realización de dicha tarea. Con la ayuda de un PDA puede bastar con un día de labores para completar el objetivo descrito.

Generalmente se requiere que éste tipo de encuestas sean anónimas. El desarrollo de una encuesta como la presentada es óptimo para una empresa consultora que realice una investigación de campo y cuyo lugar de obtención de la información sea determinado por la empresa cliente.

En cuanto al vaciado de información se produce un ahorro que va del 90 al 100 por ciento de tiempo, dependiendo del ambiente en el que se haya desarrollado. Algunas herramientas de programación permiten crear inmediatamente un archivo de base de datos y sólo basta con apropiarlo al tipo de base de datos deseado. Otras herramientas de programación no permiten la sincronización con un archivo de base de datos, sin embargo, siempre es posible tomar el archivo PDB del respaldo de Palm en el equipo de escritorio y convertirlo o explorarlo con las aplicaciones que ofrecen dicho servicio.

En el caso de la recolección de información por el método tradicional, una vez decidido el tipo de base de datos a utilizar y diseñar las tablas, el proceso de vaciado suele ser largo y tedioso debido a que hay que capturar encuesta por encuesta. Para el caso descrito con 100 encuestas se requiere una inversión de tiempo aproximadamente un día de labor para el vaciado de información.

En el caso del vaciado de información desde la PDA la sincronización ocupa como máximo 10 minutos teniendo lista la información para presentarla en tablas, gráficos, o analizarla con software estadístico.

Como se pudo observar, se trabajó con herramientas sencillas que cumplieran con los estándares del desarrollo rápido de aplicaciones (RAD) y también con ambientes de programación que si bien resultaron un tanto complicados son los más poderosos en cuanto al alcance de las aplicaciones obtenidas.

Herramientas como PDA Toolbox y Satellite Forms requieren muy poco conocimiento en cuanto a programación. Más bien se trata de herramientas que apuestan por la familiarización del usuario con el Palm OS, es decir, el usuario que comprende la filosofía y el desempeño de las aplicaciones con las que ha trabajado en un PDA. El resultado son aplicaciones básicas pero útiles y eficientes. Si habría que destacar alguna herramienta como la más sencilla y por la cual un desarrollador de aplicaciones en Palm debe comenzar es sin duda PDA Toolbox. En cuanto a Satellite Forms es una herramienta más completa y como se mencionó, imprescindible para el aprendizaje, sin embargo requiere que se instale previamente en el dispositivo un entorno de ejecución lo cual genera confusión en el usuario final.

Por otro lado se presentaron dos herramientas muy similares entre si: HB++ y NS Basic. Ambos son ambientes de desarrollo muy completos e identificados con la manera de trabajar en Microsoft Visual Basic. En cuanto al diseño de formularios ambas herramientas se muestran muy sencillas de comprender y el momento de codificar los controles es también similar. Se destaca HB++ debido a que su versión de prueba permite analizar todas las propiedades del entorno y crear aplicaciones más elaboradas ya que explota de mejor forma el hardware disponible. La manera en que NS Basic administra todo un proyecto con un solo archivo también es notable, aunque para algunos puede resultar una desventaja. Ambos son ambientes muy bien documentados.

HandDBase se trata de una herramienta diferente, potente y hasta divertida. Como se pudo observar, consiste en el desarrollo de aplicaciones de bases de datos directamente sobre el dispositivo, es decir, sobre el Palm OS. Debido a que se trata de una aplicación gráfica, HandDBase permite diseñar las aplicaciones bajo la filosofía WYSIWYG y resulta un proceso muy entretenido.

Otra herramienta que llama mucho la atención es CASL. El diseño de formularios y aplicaciones es muy sencillo, sin embargo la codificación resulta un reto ya que se trata de una sintaxis local. CASL convierte dicho código para ser compilado por gcc y realmente la herramienta que crea el archivo

ejecutable PRC es prc-Tools. Debido a la gratuidad de su licencia para estudiantes y aficionados (en general para aplicaciones no comerciales) que permite analizar las propiedades de CASL, este ambiente de desarrollo ha ganado terreno entre sus competidores.

Sin duda la mejor herramienta que se ha visto en este trabajo es SuperWaba. Si bien no es un ambiente sencillo de utilizar, y aún cuando hay que instalar la máquina virtual en el dispositivo para poder ejecutar las aplicaciones sobre Palm OS, Superwaba permite crear aplicaciones potentes y eficientes aprovechando los recursos de los dispositivos con grandes bibliotecas de clases. El proyecto de Superwaba crece gracias a las ventajas que ofrece, como crear aplicaciones para Palm OS, Symbian y Windows Mobile, a que se trata de software libre y a que es casi 100% compatible con Java.

Es responsabilidad de la empresa o usuario de estos ambientes de programación realizar un estudio previo sobre que tipo de aplicación requiere y si la solución se implementa utilizando PDA's. Entre los lenguajes y ambientes presentados se encuentran herramientas de software comercial y software libre. Habrá que evaluar la solución y el desempeño que aporte cada una con el fin de realizar o evitar un gasto que puede ser innecesario.

5.2 Mejoras al sistema

Todas las herramientas presentadas resultaron interesantes, fueron capaces de realizar la aplicación de Encuesta y presentan otras características que si bien no fueron explotadas en este trabajo, se mencionaron para posibles mejoras como la utilización de los estándares de puerto infrarrojo y Bluetooth, la utilización de impresoras que trabajen en conjunto con los PDA o bien, trabajar directamente con una conexión a internet.

Muchos ambientes de programación restringen sus propiedades (algunas las más importantes) en sus versiones de prueba, sin embargo, se puede observar el alcance de las aplicaciones que es posible crear.

Existen muchos más lenguajes y ambientes de programación que bien vale la pena estudiar como HS Pascal, Palmphi, Metrowerks CodeWarrior, IziBasic, OnBoard C, entre otros proyectos que son herramientas también famosas y que pueden aportar soluciones interesantes. Por otro lado, herramientas como Java 2 Micro Edition mejoran y es posible ejecutar las aplicaciones creadas bajo este esquema en cada vez más dispositivos móviles que no sean teléfonos celulares.

Muchas mejoras pueden hacerse a la aplicación mostrada sin duda. Dependerá de la herramienta elegida, el PDA con el que se cuente y la imaginación del programador para explotar los recursos con la mayor eficiencia posible.

5.3 El futuro de los PDA

Los ambientes profesionales en los que un PDA puede aportar soluciones se extienden de manera vasta. La tendencia es seguir aprovechando las ventajas que aportan los dispositivos móviles. He aquí algunos ejemplos de la importancia que toman día a día estos dispositivos:

Ambiente Médico

Las autoridades médicas mundiales resaltan la importancia de la precisión cuando se trata del cuidado de pacientes. Los casos de negligencia médica abundan donde los pacientes reciben tratamientos incorrectos simplemente porque el personal médico utiliza información errónea o incompleta.

Ahora se utilizan diferentes tecnologías contra este mal. Con la proliferación de los PDA, doctores, paramédicos o enfermeras tienen la capacidad de capturar y almacenar datos de pacientes y rápidamente consultar esa información cuando se necesita. Otra tecnología utilizada en el campo médico es el código de barras. Capturando la información del paciente con un código de barras permite un eficiente manejo entre paramédicos, doctores de salas de emergencia, cirujanos y enfermeras en salas de recuperación. Reemplazando tablas de información escritas a mano por un brazalete con código de barras y un PDA, los requerimientos de cuidado de un paciente van a donde quiera que este vaya. Esta solución es perfecta para asegurar la precisión en el cuidado del paciente desde los paramédicos hasta que el paciente recibe el alta médica. El paramédico puede evaluar la situación del paciente y registrarlo usando una impresora de códigos de barra en el transcurso al hospital. En lugar de dar una explicación de la condición del paciente en una ruidosa sala de emergencia, el doctor puede simplemente escanear el código de barras del brazalete e ir directo al diagnóstico o la solución.

Existen un gran número de aplicaciones para el personal médico. Entre los problemas que se resuelven son el de medicación y dosis, valores pediátricos, guías de vacunación, nutrición y tratamiento de diabetes.

Almacenamiento y distribución

La logística de los sistemas de almacenamiento físico en la mayoría de los casos es compleja. Las complicaciones surgen cada vez que un nuevo objeto es agregado a un catálogo. Un almacén que trabaja de manera efectiva y exitosa requiere que la información del movimiento de los productos esté disponible en tiempo real. Existen soluciones para tener esta información disponible en tiempo real en almacenes o bodegas. La mayoría de estos lugares pueden ser considerados áreas peligrosas con grandes estantes desde el piso hasta el techo para asegurar la máxima utilización del suelo. Proveer al personal de almacenes con un PDA les permite hacer tomas de existencia, procesar pedidos y envíos y actualizar inventarios reduciendo el riesgo de mover o arruinar el producto.

Además de esto, muchos centros de almacenamiento y distribución utilizan la tecnología del código de barras para asegurar el correcto envío del producto y un seguimiento en tiempo real desde el almacén hasta el usuario final. Los servicios de entrega y los agentes de envío de pedidos pueden usar un PDA para registrar el progreso de los envíos.

Ganadería y Agricultura

La industria ganadera y agricultora se beneficia enormemente aplicando avances tecnológicos. Con la gran competencia, las duras reglas del mercado y la amenaza de enfermedades globales del ganado, tener información precisa al alcance de los dedos se convierte en prioridad para la mayoría de las industrias. Los agricultores comprenden la necesidad de reaccionar inmediatamente a los cambios atmosféricos, de ahí la necesidad de mantenerse informado del clima. Aquellos que poseen ganado vacuno conocen el peligro de una enfermedad y el impacto económico que puede tener. En 2001, el síndrome de las “vacas locas” diezmo seriamente el ganado alrededor del mundo. Desde entonces los gobiernos han observado más de cerca el desarrollo del ganado con estrictas reglas para los ganaderos. Las empresas de tecnología han realizado estudios de mercado potencial y han respondido con soluciones de alta tecnología a estos males.

Otra solución para muchos ganaderos es el control de su ganado por medio de código de barras. Algunos gobiernos han legislado sobre el hecho de que se utilicen códigos de barra para llevar un registro del ciclo de vida del ganado.

Ventas

Recientemente, los PDA y smartphones se convirtieron en las herramientas elegidas para las ventas profesionales. Se ha desarrollado software para ayudar a automatizar los procesos de ventas volviéndolos accesibles, disponibles y fáciles de usar. Estos paquetes de software regularmente se sincronizan con paquetes de contabilidad, lo que significa que las pequeñas empresas pueden beneficiarse de la implementación de este tipo de estrategias de ventas.

Automatización de trabajo en equipo

Para proveer exitosamente sus servicios a cuantos clientes les sea posible los plomeros, electricistas, constructores, personal de mantenimiento de equipos y otros técnicos profesionales, necesitan estar disponibles tan pronto como sea posible. Típicamente esto significa que los colaboradores se reportan a un departamento para recibir instrucciones del trabajo que se requiere durante el día. Recientemente, los PDA han refinado estos procesos. Algunos usuarios proveen a sus colaboradores de PDA's con sus tareas precargadas todos los días. Otros tienen la opción de la comunicación inalámbrica incorporada ya por medio de una sincronización con el teléfono celular, con la posibilidad de darle un seguimiento a las tareas. Cuando una tarea es terminada, el PDA registra el progreso, asegurándose de mantener satisfecho al cliente.

Perspectiva del Palm OS

Debido al costo de las licencias del Palm OS que imponía PalmSource (la división de software de Palm) esta plataforma fue perdiendo terreno frente a Windows Mobile. Recientemente la compañía japonesa ACCESS adquirió Palm Source. ACCESS se encargó de dar un giro radical al manejo del Palm OS. Como paso inicial ACCESS decidió renombrar al Palm OS versión 5.4 a Garnet OS. Los nuevos smartphones de Palm incluyen esta versión de Palm OS ahora

bajo el nombre de Garnet, además de que han diseñado dispositivos bajo la plataforma rival Windows Mobile.

Sin embargo, ACCESS ha decidido establecer una nueva plataforma móvil basada en Linux. Mediante la tecnología libre de Linux, la plataforma ACCESS Linux combina las tecnologías abiertas con nuevos componentes Linux desarrollados internamente y algunos otros privativos de ACCESS. ACCESS continúa la tradición de Palm OS, que permite crear dispositivos móviles con numerosas prestaciones y listos para su uso. La plataforma ACCESS Linux incluye aplicaciones para el sector de las comunicaciones, como telefonía, productividad y multimedia. También incluye la máquina virtual de Palm OS (Garnet VM). Garnet VM es compatible (siempre que estén correctamente diseñadas) con miles de aplicaciones para el sistema Palm OS. La plataforma ACCESS Linux también incorpora JV-Lite™2 de ACCESS, una solución Java para dispositivos móviles de última generación. Estos tres entornos de aplicaciones: ACCESS Linux Platform, Garnet VM y Java, ofrecen a los desarrolladores una oportunidad de crear y distribuir una amplia gama de aplicaciones para móviles y dispositivos PDA. Garnet VM permite funcionar sin trabas a miles de programas para Palm OS en cualquier aparato dotado de ACCESS Linux Platform y tiene a su disposición un completo juego de programas libres integrados para el desarrollo de aplicaciones.

En definitiva aunque el sistema operativo Palm OS ha adquirido un nuevo nombre con un nuevo dueño, la funcionalidad, los programas y proyectos que se desarrollen para este sistema operativo estarán disponibles bajo el ACCESS Linux Platform. Trabajar con código abierto y con esquemas de Linux, con el respaldo de una empresa se convierte en una práctica cada vez más común con el fin de ganar terreno en el mercado y ofrecer servicios de calidad en tecnologías de información. ACCESS realiza esta apuesta dejando atrás el desarrollo del Palm OS, pero considerandolo parte importante en la adaptación de su nueva plataforma.

En fechas más recientes Apple ha entrado al mercado con su producto Iphone, un teléfono inteligente basado en una versión modificada del sistema operativo Mac OS X. Este suceso ha marcado un cambio sustancial en la historia de los PDA y ha puesto en predicamento a los fabricantes que hasta el momento mantenían cierta hegemonía en el ambiente ejecutivo. Sin embargo, dicha tecnología resulta costosa para la implementación de soluciones en el ambiente profesional por lo que las soluciones aportadas por Palm OS continúan siendo más viables

Ante la pérdida de terreno con Windows Mobile, la adquisición y renovación del sistema operativo por parte de ACCESS es, en mi opinión, lo mejor que le pudo suceder al Palm OS. La apertura del código fuente llevará muchas mejoras a la plataforma y será tarea de los estrategas comerciales hacer que Palm OS vuelva a ocupar su lugar de privilegio en los equipos PDA, sobre todo en el ambiente productivo y no ejecutivo.

Ante tantos cambios cabe destacar la importancia que adquieren las herramientas de desarrollo multiplataforma. La adaptación de las aplicaciones a las plataformas líderes determinará su existencia o su deceso.

Apéndice:

Código utilizado para el proyecto en CASL

Cuestionario.csl

```
#Cuestionario CASL  
#Desarrollado por René Gazano
```

```
#Declaración de variables  
#se definen todas las variables que se necesitaran para la encuesta (campos de la BD)
```

```
variables;  
numeric numcuestionario;  
string fecha;  
string hora;  
string nombre;  
string puesto;  
string sexo;  
string preg1;  
numeric preg2;  
string preg3a;  
string preg3b;  
string preg3c;  
string preg4;  
string preg5;  
string preg6;  
numeric preg7;  
string preg8;  
string preg9a;  
string preg9b;  
string preg9c;  
string preg10;
```

```
#variables para procesos de fecha y hora
```

```
numeric StartofDay;  
numeric EndofDay;  
numeric CurrYear;  
numeric CurrMonth;  
numeric CurrDay;  
numeric CurrHour;  
numeric CurrMinute;  
string temporal;  
numeric numericval;  
string busqueda;  
string sDBName="cuestionario";  
numeric fabort;  
numeric frenameCDI;
```

```
#de esta manera se declaran las listas, como arreglos. He aquí las posibles respuestas a las preguntas
```

```
#señaladas
```

```
respuestas1[5] = "Altamente satisfecho", "Satisfecho", "Regularmente Satisfecho",  
                "Insatisfecho", "Altamente insatisfecho";  
respuestas3[8] = "Reportes", "Actividades extralaborales", "Ventas", "Atención a Clientes", "Inventarios",  
                "Trabajo de otras personas", "Intendencia", "Trabajar horas extra";  
respuestas4[3] = "Buena", "Regular", "Mala";  
respuestas5[4] = "Problemas Personales", "Mal carácter", "Impuntualidad", "Exigencia";  
respuestas6[3] = "Bueno", "Regular", "Malo";  
respuestas8[5] = "Muy Buenas", "Buenas", "Regulares", "Malas", "Muy malas";  
respuestas9[8] = "Mejores oportunidades", "Salario", "Prestaciones", "Incapacidad de crecimiento",  
                "Problemas personales", "Conflictos entre compañeros", "Conflictos con el jefe", "Falta  
                de pagos";  
respuestas10[7] = Excelentes, "Buenas", "Adecuadas", "Deficientes", "Malas", "Insalubres", "Inadecuadas";
```

```
#Se incluyen las librerías y archivos .cpk indicando su ubicación (por lo general C:\Archivos de  
#Programa\CASLSoft\CASL43\Samples\Libraries  
include "C:\...\ai_AuthorInfo.cpk";
```

```

include "C:\...\tp_TutorialPrice.cpk";
include "C:\...\Casler26.cpk";
include "C:\...\sml_ShowMsgLite.cpk";
include "C:\...\sal_ShowAboutLite.cpk";
include "C:\...\DataPath.cpk";
include "C:\...\DateTime.cpk";
include "C:\...\RENE>SelectDateTime.cpk";

```

#declaración del menú y submenús

```

menu_top mtabout;
display "Opciones";
end;

```

```

menu_item miacerca,mtabout;
display "Información del autor";
end;

```

```

menu_item miDeleteDB,mtabout;
display "Borrar DB";
end;

```

#definición del archivo de base de datos, los campos deben ser declarados con el mismo nombre que en la declaración de variables

```

dbfile dbcuestaa;
field numcuestionario;
field fecha;
field hora;
field nombre;
field puesto;
field sexo;
field preg1;
field preg2;
field preg3a;
field preg3b;
field preg3c;
field preg4;
field preg5;
field preg6;
field preg7;
field preg8;
field preg9a;
field preg9b;
field preg9c;
field preg10;
sync_pref pda_to_pc; #preferencia de sincronización, puede ser pc_to_pda o merge (en ambos sentidos)
data_source_name "CASLAccess"; #se puede definir un DSN con un archivo de Access y mediante este
#sincronizar la información recabada en la PDA y la base de datos en el equipo de escritorio
end;

```

#Funciones

```

function cmdfecha; #utiliza los paquetes SelectDateTime.cpk y DateTime.cpk para elegir fecha
variables;
string SelectType;
string t;
end;
SelectType = "day";
t = SelectDate(CurrYear, CurrMonth, CurrDay, SelectType, "CASLPro-Test");
if t <> "";
CurrYear = value(left(t,4));
CurrMonth = value(mid(t,4, 2));
CurrDay = value(right(t,2));
varfecha.display = GetDateStr(CurrYear, CurrMonth, CurrDay, 0);
end_if;
end;

```

```

function cmdhora; #utiliza los paquetes SelectDateTime.cpk y DateTime.cpk para elegir hora
variables;
string t;
end;
t = SelectOneTime(CurrHour, CurrMinute,"CASLPro-Test");

```

```

if t <> "";
    CurrHour = value(left(t,2));
    CurrMinute = value(right(t,2));
    varhora.display = GetTimeStr(CurrHour, CurrMinute, 0);
end_if;
end;

function GetCurrentDateTime; #utiliza el paquete DateTime.cpk para obtener fecha y hora actuales
    CurrYear = year();
    CurrMonth = month();
    CurrDay = day();
    CurrHour = hour();
    CurrMinute = minute();
    varhora.display = GetTimeStr(hour(), minute(), 0);
    varfecha.display = GetDateStr(year(), month(), day(), 0);
end;

function miacerca; #opción Información del Autor del Menú. Utiliza los archivos sal_ShowAboutLite.cpk,
ai_AuthorInfo.cpk y tp_TutorialPrice.cpk
    if not fAbort;
        sal_ShowAbout("Cuestionario", "1.0",
            "Cuestionario sobre Insatisfacción Laboral",
            "Programa basico con Base de Datos en CASL",
            ai_sAuthor, tp_sPaymentStatus, "");
    end_if;
end;

function miDeleteDB; #opción Borrar DB del menú
    if not fAbort;
        if sml_ffConfirmYes("¿Borrar cuestionarios?");
            sml_ShowMsg(0, "la BD será borrada "+
                "reiniciando el programa, ");
            fAbort=true;
            show Cuestionario1;
        end_if;
    end_if;
end;

function cmdnextFR0; #funciones de botón, muestran formulario siguiente
show Cuestionario1;
end;

function btkey; #función de botones dentro de la tabla para recuperar datos de la base y mostrarlos en
#el formulario Cuestionarios1
if not fAbort;
    temp.display=btKey[invokersub].display;
    get temp,numcuestionario;
    if numcuestionario=0;
        else;
            hide Cuestionarios;
            search dbencuesta,numcuestionario;
            muestra;
        end_if;
    end_if;
end;

function btnombre;
btkey;
end;

function btpuesto;
btkey;
end;

function cmdnuevoFR1; #Muestra el formulario Cuestionario1 listo para capturar nuevo registro
if not fAbort;
    hide Cuestionarios;
    varnumcuest.display="";
    varfecha.display="";
    varhora.display="";
    varnombre.display="";

```

```

                varpuesto.display="";
                varsexo.display="M";
            show Cuestionario1;
        end_if;
    end;

function cmdnuevoFR7;
cmdnuevoFR1;
end;

function cmdborrarFR1; #Borra registro seleccionado
get varnumcuest,numcuestionario;
errorcode=ce_no_error; #El manejo de errores se realiza con la biblioteca Casler26.cpk
search dbencuesta,numcuestionario;
if errorcode=ce_no_error;
temporal=string(numcuestionario,"##");
if sml_ffConfirmYes("Borrar Registro? \"+temporal+
"\", esta seguro?");
remove dbencuesta;
if errorcode=ce_no_error;
else;
sml_ShowMsg(2,ce_geterror+
" Borrando registro "+temporal);
end_if;
end_if;
else;
sml_ShowMsg(0,"Registro \"+temporal+
"\", no encontrado");
end_if;
hide Cuestionario1;
show gridencuesta;
show Cuestionarios;
end;

function cmdnextFR1; #se captura la información y se almacena el valor de los controles en las variables,
#hasta llegar al formulario Cuestionario 6 que es donde se inserta el registro completo a la Base de
#datos dbencuesta
get varnumcuest,numcuestionario;
fecha=varfecha.display;
hora=varhora.display;
nombre=varnombre.display;
puesto=varpuesto.display;
sexo=varsexo.display;
busqueda=varnumcuest.display;
hide Cuestionario1;
chkpregunta2.checkbox=0;
show Cuestionario2;
end;

function cmdnextFR2;
get selector3,preg1;
preg2=chkpregunta2.checkbox;
hide Cuestionario2;
show Cuestionario3;
end;

function cmdnextFR3;
preg3a=droprespuesta3a.display;
preg3b=droprespuesta3b.display;
preg3c=droprespuesta3c.display;
preg4=droprespuesta4.display;
hide Cuestionario3;
chkpregunta7.checkbox=0;
show Cuestionario4;
end;

function cmdnextFR4;
preg5=droprespuesta5.display;
preg6=droprespuesta6.display;
preg7=chkpregunta7.checkbox;
hide Cuestionario4;
show Cuestionario5;
end;

```

```

end;

function cmdnextFR5;
preg8=droprespuesta8.display;
preg9a=droprespuesta9a.display;
preg9b=droprespuesta9b.display;
preg9c=droprespuesta9c.display;
hide Cuestionario5;
    show Cuestionario6;
end;

function cmdsalirFR6;
preg10=droprespuesta10.display;
varnumcuest6.display=busqueda;
if busqueda<>"";
    errorcode=ce_no_error;
    search dbencuesta,varnumcuest6.display;
    if errorcode=ce_no_error;
        put_fields dbencuesta;
        if errorcode<>0;
            sml_ShowMsg(2,ce_geterror+
                " editando el registro existente");
        end_if;
    else;
        if errorcode=-12; # not found
            errorcode=ce_no_error;
            insert dbencuesta;
            if errorcode=ce_no_error;
                else;
                    sml_ShowMsg(2,ce_geterror+
                        " agregando nuevo registro");
                end_if;
            end_if;
        else;
            sml_ShowMsg(2,ce_geterror+
                " accedando a la BD");
        end_if;
    end_if;
end_if;

hide Cuestionario6;
show gridencuesta;
show Cuestionarios;
end;

function cmdnuevoFR6;
cmdsalirFR6;
cmdnuevoFR1;
end;

function cmdsalirFR0; #funciones para salir de la aplicación
launch "";
end;
function cmdsalirFR1;
launch "";
end;
function cmdsalirFR2;
launch "";
end;
function cmdsalirFR3;
launch "";
end;
function cmdsalirFR4;
launch "";
end;
function cmdsalirFR5;
launch "";
end;
function cmdsalirFR7;
launch "";
end;

```

```

function cmdshowregsFR1; #muestra los registros a partir del Botón Ver Registros
hide Cuestionario1;
show gridencuesta;
show Cuestionarios;
end;

function muestra; #asigna campos de la base de datos a los controles de Cuestionario1
get_fields dbencuesta;
varnumcuest.display=string(numcuestionario,"##");
    varfecha.display=fecha;
    varhora.display=hora;
    varnombre.display=nombre;
    varpuesto.display=puesto;
    varsexo.display=sexo;
    show Cuestionario1;

end;

function cmdfirstrec; #busqueda de registros, primero, anterior, siguiente y último
seek_start dbencuesta,0;
muestra;
end;
function cmdprevrec;
seek_here dbencuesta,-1;
muestra;
end;
function cmdnextrec;
muestra;
end;
function cmdlastrec;
seek_end dbencuesta,1;
muestra;
end;

function startup; #La función startup es muy importante ya que es la que inicia la aplicación
    sDBName=DataPath("", "")+sDBName; #utiliza el archivo DataPath.cpk para la creación del
#archivo de base de datos
    GetCurrentDateTime;
    StartOfDay = 8; # i.e. 8:00 am
    EndOfDay = 22; # i.e. 10:00 pm
    errorcode=ce_no_error;
    open dbencuesta,sDBName;
    if errorcode=ce_no_error;
        show Cuestionario;
        fAbort=false;
    else;
        sml_ShowMsg(3,ce_geterror+
            ". Borrar BD corrupta. Intentelo de nuevo");
        fAbort=true;
    end_if;
end;

function shutdown; #función de salida de la aplicación
close dbencuesta; #importante cerrar la base de datos al salir de la aplicación
if fAbort;
    delete sDBName; # la base de datos debe cerrarse antes
else;
    if fRenameCDI; # la base de datos debe cerrarse antes
        rename sDBName+".cdb",sDBName+".cdi";
        sml_ShowMsg(0,"Activa Hotsync para instalar "+
            "la BD \""+sDBName+"\" en tu dispositivo");
    end_if;
end_if;
end;

```

Código utilizado para el proyecto en HB++

Clase: clsApplication

'clase principal Aplicación

'destructor

```
Private Sub Terminate()  
    'usa la clase principal par cerrar la BD si es necesario  
    If Not dbEncuesta Is Nothing Then dbEncuesta.Close  
    'restaura la pantalla predeterminada  
    Set ScreenMode=DefaultScreenMode  
End Sub
```

'lanzador

```
Private Sub Application_NormalLaunch()  
    dim f as new Cuestionario  
    'modo de inicio en pantalla  
    InitScreenMode  
    'abre o crea la DB  
    Set dbEncuesta=OpenDatabase(hbModeReadWrite)  
    'carga la pantalla de inicio  
    f.Show hbFormModeless+hbFormGoto  
End Sub
```

'carga el registro seleccionado

```
Private Sub Application_GotoItem(ByVal itemGoto as ItemInfo)  
    dim f as new frmView  
    If Not SubCall Then  
        Set dbEncuesta=OpenDatabase(hbModeReadWrite)  
        InitScreenMode  
    End if  
    'inicia la búsqueda  
    dbEncuesta.LookupUniqueID itemGoto.Custom  
    f.Show hbFormModeless+hbFormGoto  
End Sub
```

'Busca registros

```
Private Sub Application_FindItem(ByVal sToFind As String, ByVal eAccessMode As HbMode, ByVal  
IRecordNum As Long, ByVal bContinuation As Boolean)  
    Dim db as tblEncuesta, ii as New ItemInfo  
    Dim bShowSecret as Boolean  
    If FindDrawHeader(App.Title) Then Exit Sub  
  
    Set db=OpenDatabase(hbModeReadOnly)  
    bShowSecret =(eAccessMode and hbModeShowSecret)=hbModeShowSecret  
    If db.RecordCount<>0 Then  
        If bContinuation Then  
            db.AbsolutePosition=IRecordNum  
        End if  
        While Not db.EOF  
            if db.Secret=False or (bShowSecret and db.Secret) then  
                If InStr(1,db.Numcuest,sToFind,hbCompareText)>0 Then  
                    ii.RecordNum=db.AbsolutePosition  
                    ii.Custom=db.UniqueID  
                    If FindMatch(db.Numcuest,ii) Then  
                        Exit While  
                    End if  
                End If  
            End if  
            If DoEvents Then Exit While  
            db.MoveNext  
        Wend  
    End If  
    db.Close  
End Sub
```

Módulo: Module1

'Variables globales y procedimientos

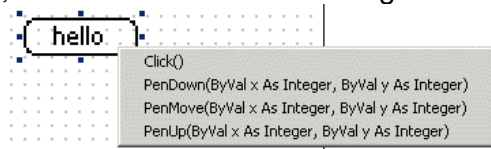
```
Public Const sFileExtension as String = ".agr"  
Public Const lMagicNumber as Long = &H2563A5F8  
Public dbEncuesta as tblEncuesta
```

'Abre una base de datos con la referencia de la tabla creada tblEncuesta

```
Public Function OpenDatabase(ByVal eAccessMode as HbMode) as tblEncuesta  
    Dim db as New tblEncuesta  
    Dim e as Integer  
  
    Try  
        db.OpenTable hbModeOpenExisting+eAccessMode  
        Set OpenDatabase=db  
    Catch  
        e=Err.Number  
        If e<>43 Then Err.Raise e  
        db.OpenTable hbModeCreateAlways+hbModeReadWrite  
        db.CategoryName(0)="Unfiled"  
        db.CategoryName(1)="Programmers"  
        db.AddNew  
        db.Numcuest=1  
        db.Fecha=DateSerial(2007,5,27,0,0,0)  
        db.Nombre="rene gazano"  
        db.puesto="ing"  
        db.sexo="M"  
        'al crear una nueva BD inserta un registro  
        db.Update  
        db.Close  
        db.OpenTable hbModeOpenExisting+eAccessMode  
        Set OpenDatabase=db  
    End Catch  
End Function
```

```
Public Sub InitScreenMode()  
    Dim sc as New ScreenMode  
    Dim b as Boolean  
    b=sc.FindFirstMode  
    While b  
        If sc.BitDepth<=8 Then  
            Set ScreenMode=sc  
            Exit Sub  
        End If  
        b=sc.FindNextMode  
    Wend  
End Sub
```

La codificación de los formularios se realiza a un costado de su diseño. Por medio de un clic derecho se pueden conocer los eventos a los cuales puede responder un control, como se muestra en la imagen:



Para el formulario de inicio, Cuestionario, se tiene la siguiente codificación:

```
Private Sub cmdnextFR0_Click() 'responde al botón continuar > carga el siguiente formulario  
    dim f as New frmView  
    f.Nuevo=True  
    f.show hbformgoto+hbformmodeless  
End Sub  
  
Private Sub cmdsalirFR0_Click() 'responde al botón salir > salir de la aplicación  
    App.Quit
```


End Sub

```
Private Sub mnuinfo_Click() 'Responde a la opción Información del autor en el menú.  
Msgbox "Cuestionario desarrollado \n & HB++ por René Gazano \n rgazano@gmail.com",  
vbMsgBoxDoneOnly+vbMsgBoxInformation,"Información del autor"  
End Sub
```

Formulario frmView:

'Variables globales

```
Public Nuevo as Boolean 'bandera para registros nuevos  
Public Modificado as Boolean 'bandera para modificación de registros
```

```
Private Sub Form_Load()  
dim ese as Integer  
Modificado=False  
If Nuevo Then 'carga formulario listo para capturar nuevo registro  
ese=dbEncuesta.RecordCount+1  
fldnumcuest.Text=cstr(ese) 'aunque solo admite números, es conveniente manejarlo  
'como cadena  
modificado=True  
selfecha.Date=(now)  
selhora.Format="hh:nn:ss"  
selhora.Time=Hour(Date())*CLng(3600)+Minute(Date())*60+Second(Date())  
txtnombre.Text=""  
txtpuesto.Text=""  
pushsexM.Value=True  
cmdDelete.Visible=False  
  
Else  
'Carga el registro existente  
Llenado 'llama función de llenado de los campos  
End If  
End Sub
```

```
Private function SaveChanges() as long  
If Modificado Then  
If Nuevo Then  
dbEncuesta.AddNew 'guarda nuevos registros  
Else  
dbEncuesta.Edit 'guarda registros modificados  
End If  
  
'Guarda los campos  
dbEncuesta.numcuest=Trim(fldnumcuest.Text)  
dbEncuesta.Fecha=selfecha.Date  
dbencuesta.hora=selhora.Time  
dbencuesta.nombre=txtnombre.Text  
dbencuesta.puesto=txtpuesto.Text  
if pushsexM.Value=True then  
dbencuesta.sexo="M"  
else  
dbencuesta.sexo="F"  
end if  
dbEncuesta.Update  
SaveChanges=dbEncuesta.UniqueID  
dbEncuesta.Requery  
Else  
If Nuevo Then  
SaveChanges=dbEncuesta.UniqueID  
dbEncuesta.Requery  
Else  
SaveChanges=dbEncuesta.UniqueID  
End If  
End If  
End Function
```

```
Private Sub fldnumcuest_Change() 'si se cambian valores como numero de cuestionario, fecha u hora  
'se activa la bandera de modificación  
Modificado=True  
End Sub
```

```

Private Sub selFecha_Change(ByVal bExitOnOK As Boolean)
    Modificado=True
End Sub

Private Sub cmdDelete_Click() 'Responde al botón Borrar para borrar el registro actual
    Dim r as HbMsgBoxReturn
    Dim f as new frmMain
    r=MsgBox("¿Borrar este registro?",hbMsgBoxYesNo+hbMsgBoxConfirmation)
    If r=hbMsgBoxYes Then
        dbEncuesta.Delete hbRecordDelete
        f.Show hbFormGoto+hbFormModeless
    End If
End Sub

Private Sub cmdDone_Click() 'Guarda los datos actuales y continua al siguiente formulario para seguir
                                'con la encuesta
    Dim f as new frmMain
    dim f2 as new Cuestionario2
    Dim IID as long
    IID=SaveChanges
    f.lLastUniqueID=IID
f2.Show hbFormGoto+hbFormModeless
End Sub

Private Sub mnuAbout_Click() ' Menu Opciones > información del autor
Msgbox "Cuestionario desarrollado\n& HB++ por René Gazano\nrgazano@gmail.com",
hbMsgBoxDoneOnly+hbMsgBoxInformation,"Información del autor"
End Sub

Private Sub cmdnuevo_Click() 'Crea un nuevo registro
Modificado=False
Nuevo=true
        Set Focus=fldnumcuest
        fldnumcuest.text=""
        selfecha.Date=(now)
        selhora.text=(now)
        txtnombre.Text=""
        txtpuesto.Text=""
        pushsexM.Value=True
        cmdDelete.Visible=False
End Sub

Private Sub cmdshowregs_Click() 'Carga el formulario frmMain para ver la cuadrícula de registros
    Dim f as new frmMain
    Dim IID as long
    IID=SaveChanges
    f.lLastUniqueID=IID
    f.Show hbFormGoto+hbFormModeless
End Sub

Private Sub cmdsalir_Click() 'Salir de la aplicación
app.Quit
End Sub

Private sub llenado() 'Procedimiento para cargar los datos de la consulta en los controles del formulario
    fldnumcuest.Text=dbEncuesta.Numcuest
    selfecha.Date=dbEncuesta.Fecha
    selhora.Time=dbencuesta.hora
    txtnombre.Text=dbencuesta.nombre
    txtpuesto.Text=dbencuesta.puesto
    if dbencuesta.sexo="M" then
        pushsexM.Value=True
    else
        pushsexM.Value=False
    end if
    cmdDelete.Visible=True
End Sub

Private Sub cmdfirst_Click() 'Exploración de registros
dbencuesta.MoveFirst

```

```
llenado
End Sub
```

```
Private Sub cmdprev_Click()
dbencuesta.MovePrevious
llenado
End Sub
```

```
Private Sub cmdnext_Click()
dbencuesta.MoveNext
llenado
End Sub
```

```
Private Sub cmdlast_Click()
dbencuesta.MoveLast
llenado
End Sub
```

Prescindiendo de los eventos ya vistos, el formulario frmMain contiene el siguiente código:

```
Public ILastUniqueID as Long
```

```
Private Sub Initialize()
ILastUniqueID=-1 'para considerar el registro 0
End Sub
```

```
Private Sub Form_Load() 'carga formulario con la tabla de registros
dim szSQL as string
grdView.Redraw=False
szSQL="UniqueID>0 ORDER BY Numcuest ASC " 'Sentencia SQL para consulta de los registros
dbEncuesta.OpenRecordset szSQL,hbModeReadWrite+hbModeOpenExisting,hbCompareText
grdView.Populate dbEncuesta,(eCurrentSecurityLevel=hbSecurityMaskRecords),true
grdView.Redraw=True
grdView.SelectItem grdView.FindItemData(ILastUniqueID),0
End Sub
```

```
Private Sub grdView_Click(ByVal iRow As Integer, ByVal iCol As Integer, ByVal IItemData As Long)
'Responde al evento clic en la tabla de registros mostrada en el formulario, cargando los datos del
'registro elegido
Dim f as New frmView
dbEncuesta.LookupUniqueID IItemData
f.Show hbFormGoto+hbFormModeless
End Sub
```

```
Private Sub cmdNew_Click() 'Para crear nuevo registro activa bandera
Dim f as New frmView
f.Nuevo=True
f.Show hbFormGoto+hbFormModeless
End Sub
```

```
Private Sub grdView_LoadItem(ByVal iRow As Integer, ByVal IItemData As Long, ByVal vValue() As Variant)
'Asigna los valores de las columnas requeridas de la base de datos a las etiquetas de la cuadrícula.
dbEncuesta.LookupUniqueID IItemData
vvalue(0)=dbEncuesta.Numcuest
vValue(1)=dbEncuesta.nombre
vValue(2)=dbencuesta.puesto
End Sub
```

El código de los formularios Cuestionario2 a Cuestionario6 simplemente continuarán con la encuesta y actualizarán el registro que se está modificando. Como ejemplo el formulario Cuestionario4 contiene el siguiente código en el que cabe señalar cómo se agregan las opciones en las listas y controles popup.

```

Private Sub Form_Load()
`Carga el formulario con los valores de las listas y el valor de la
`Pregunta Si/No con un valor por default
poppreg5.Redraw=False
poppreg5.Clear
poppreg5.AddItem "Problemas Presonales"
poppreg5.AddItem "Mal caracter"
poppreg5.AddItem "Impuntualidad"
poppreg5.AddItem "Exigencia"
poppreg5.Redraw=True
txtpreg5.text=""
poppreg6.Redraw=False
poppreg6.Clear
poppreg6.AddItem "Bueno"
poppreg6.AddItem "Regular"
poppreg6.AddItem "Malo"
poppreg6.Redraw=True
pushpreg7si.Value=True
End Sub

```

```

Private Sub cmdnextFR4_Click()
dim f as new Cuestionario5
dbencuesta.Edit `Abre el recordset para editarlo
if poppreg5.text="Elegir" then `Revisa si existe una nueva opción como respuesta en la pregunta 5
dbencuesta.preg5=txtpreg5.Text
else
dbencuesta.preg5=poppreg5.Text
end if

dbencuesta.preg6=poppreg6.Text

if pushpreg7si.Value then `Para asignar el valor de la respuesta a Pregunta 7
dbencuesta.preg7=True
else
dbencuesta.preg7=False
end if
dbencuesta.Update `Actualiza el recordset
dbencuesta.Requery
f.Show hbFormGoto+hbFormModeless
End Sub

```

Código utilizado para el proyecto en NSBasic

En el código de inicio del proyecto (Startup Code), se declaran las variables globales y funciones o procedimientos generales.

```
Sub Project_Startup()

Global mifecha as Date 'para manejar la fecha actual
Type dbEncuestaLayout 'declaración del contenido del recordset
    Numcuest as Integer
    fecha as Time
    hora as Date
    Nombre as String
    Puesto as String
    Sexo as String
    preg1 as String
    preg2 as String
    preg3a as String
    preg3b as String
    preg3c as String
    preg4 as String
    preg5 as String
    preg6 as String
    preg7 as String
    preg8 as String
    preg9a as String
    preg9b as String
    preg9c as String
    preg10 as String
End Type

Dim dbEncuestaRec as dbEncuestaLayout
Global dbEncuesta as Database 'declaración de la base de datos
Global registro as Integer 'contador de registros
Global nuevo as Integer 'bandera para registros nuevos
Dim result as Integer
Dim i as Integer
mifecha=today()
MenuDraw "Opciones" 'se elige el menú que se utilizará previamente diseñado
'crea BD si no existe
result=dbOpen(dbEncuesta, "Encuesta.pdb", 0)
If result=519 Then
    result=dbCreate(dbEncuesta, "Encuesta.pdb", 0, "GCRP") 'al crear la BD se
    'proporcina el ID de creador
    If result<>0 Then
        MsgBox "Error: no se pudo crear Encuesta " + str(result)
        Exit Sub
    End If
registro=dbgetnorecs(dbEncuesta)'contador de registros
nuevo=1
End If
End Sub

Sub displayGrid()
Dim EncuestaKey as Integer
Dim dbEncuesta1 as Database "Encuesta.pdb", dbEncuestaRec, dbEncuestaLayout,
EncuestaKey
Grid.bindToDatabase (dbEncuesta1, dbEncuestaRec.numcuest,
dbEncuestaRec.nombre, dbEncuestaRec.puesto)'relaciona la cuadrícula con las
'tres columnas de la BD
End Sub
```

El código de salida de la aplicación (termination code) sirve principalmente para finalizar instancias.

```
Sub Project_Termination()
Dim err as Integer
err=dbclose(dbEncuesta)'cierra la BD antes de salir
If err=0 Then
    MsgBox "BASE CERRADA CORRECTAMENTE"
End If
End Sub
```

Con respecto a los controles para el formulario de bienvenida, Cuestionario, se tiene lo siguiente:

```
Sub object1017()      'Botón Salir
Stop      'detiene la aplicación
End Sub
Sub object1018()      'Botón continuar
Nextform "Cuestionario1" 'Carga el siguiente formulario
End Sub
```

Cada formulario tiene la capacidad de ejecutar instrucciones antes, durante y después de ser utilizado. En el caso del formulario *Cuestionario1* (Form1020) se prepara el formulario para cargar datos o capturar nuevos registros:

```
Sub Form1020_Before()
    Dim t as Time
    Dim d as Date
    Dim result as Integer

    If registro<>0 And nuevo=0 Then 'para mostrar registros existentes

        result=dbread(dbEncuesta,registro,dbEncuestaRec)'lee datos de BD

        fldnumcuest.Text=str(dbEncuestaRec.Numcuest)
        d=dbEncuestaRec.fecha
        selfecha.Text=str(d)
        t=dbEncuestaRec.hora
        selhora.Text=str(t)
        fldnombre.Text=dbEncuestaRec.Nombre
        fldpuesto.Text=dbEncuestaRec.Puesto
        If dbEncuestaRec.Sexo="M" Then
            pushsexM.Status=True
        Else
            pushsexM.Status=False
        End If
    Else
        If registro=0 Then 'si no hay registros prepara para capturar uno
            fldnumcuest.Text=""
            d=today()
            selfecha.Text=str(d)
            t=now()
            selhora.Text=str(t)
            fldnombre.Text=""
            fldpuesto.Text=""
            pushsexM.Status=True
        Else
            If nuevo=1 Then 'prepara para capturar nuevo registro
                registro=registro+1
                fldnumcuest.Text=str(registro)
                d=today()
                selfecha.Text=str(d)
                t=now()
                selhora.Text=str(t)
                fldnombre.Text=""
                fldpuesto.Text=""
                pushsexM.Status=True
            End If
        End If
    End If
End Sub
```

Los controles *selfecha* y *selhora* muestran un calendario y un reloj respectivamente para facilitar la captura:

```
Sub object1036()
Dim result as Integer
result=popupdate(mifecha,"Fecha de aplicación")
selfecha.text=str(mifecha)
End Sub
```

```

Sub object1037()
    Dim t as Time
    t=now()
    selhora.Text=str(t)
End Sub

```

Para continuar con la captura de la encuesta, el botón *Continuar* salva la información capturada en el primer formulario en las columnas correspondientes.

```

Sub object1053()
Dim result as Integer
result=dbclose(dbEncuesta)'cierra y abre la BD para evitar errores
    result=dbOpen(dbEncuesta, "Encuesta.pdb", 0)
    If result<>0 Then
        MsgBox "Error: No se puede abrir Encuesta " + str(result)
        Exit Sub
    End If

    dbEncuestaRec.numcuest=val(fldnumcuest.text)'asignación de lo
capturado
                                'a las columnas de la BD
    dbEncuestaRec.fecha=todate(selfecha.Text)
    dbEncuestaRec.hora=totime(selhora.Text)
    dbEncuestaRec.nombre=fldnombre.Text
    dbEncuestaRec.puesto=fldpuesto.Text
    If pushsexM.Status=True Then
        dbEncuestaRec.sexo="M"
    Else
        dbEncuestaRec.sexo="F"
    End If

    registro=val(fldnumcuest.Text)'numero de registro
    result=dbInsert(dbEncuesta, registro, dbEncuestaRec) 'inserta el
registro
    If result<>0 Then
        MsgBox "Error: escribiendo registro " + str(result)
        Exit Sub
    End If
NextForm "Cuestionario2"
End Sub
Sub object1050() 'codigo del botón Borrar
Dim result as Integer
result=alert("Eliminar Registro","Desea eliminar el
registro?",1,"Borrar","Cancelar")
If result=0 Then
    result=dbdelete(dbEncuesta,registro)
end if
NextForm "Cuestionarios"
End Sub
Sub object1049() 'del botón Nuevo
registro=dbgetnorecs(dbEncuesta) 'obtiene el número de registros para generar
'el siguiente
nuevo=1 'bandera de nuevo
Redraw
End Sub
'Recorrido de registros
Sub object1044()'el recorrido se hace sobre el número de registro
Dim result as Integer
result=dbreset(dbEncuesta)'para leer el primer registro
result=dbreadnext(dbEncuesta,dbEncuestaRec.Numcuest,dbEncuestaRec)
registro=dbEncuestaRec.Numcuest
nuevo=0
Redraw
End Sub
Sub object1054()
registro=registro-1 'registro anterior

```

```

nuevo=0
Redraw
End Sub
Sub object1056() 'siguiente
registro=registro+1
nuevo=0
Redraw
End Sub
Sub object1055() 'ultimo registro
Dim result as Integer
registro=dbgetnorecs(dbEncuesta)
nuevo=0
Redraw
End Sub

```

Continuando con el contenido de la encuesta, los formularios que contienen las preguntas contienen código similar, es decir, guardan los cambios en las columnas correspondientes del registro actual y pasan al siguiente formulario. Como ejemplo el botón Continuar del formulario *Cuestionario2*

```

Sub object1089() 'botón continuar Cuestionario2
Dim result as Integer

    dbEncuestaRec.preg1=poppreg1.Text
    If pushpreg2si.Status=True Then
        dbEncuestaRec.preg2="Si"
    Else
        dbencuestaRec.preg2="No"
    End If

    result=dbupdate(dbEncuesta,registro, dbEncuestaRec)
    If result<>0 Then
        MsgBox "Error: escribiendo registro " + str(result)
        Exit Sub
    End If
NextForm "Cuestionario3"
End Sub

```

En *Cuestionario6* existe la opción de terminar la encuesta y presentar la cuadrícula o bien capturar un nuevo registro.

```

result=dbclose(dbEncuesta) 'para salir
NextForm "Cuestionarios"

```

O bien:

```

nuevo=1 'para agregar nuevo registro
NextForm "Cuestionario1"

```

Antes de que se ejecute el formulario *Cuestionarios* se prepara para cargar los datos

```

Sub Form1058_Before()
Grid.hide
grid.colwidth(1)=10
grid.ColType 1,"numeric","nnn"
grid.colwidth(2)=100
grid.ColType 2,"text",""
grid.colwidth(3)=50
grid.ColType 3,"text",""
displayGrid 'función definida en el código Startup
grid.ShowGridLines
grid.show
End Sub

```

Para cargar un registro elegido en la cuadrícula:


```
Sub object1059()  
Dim result as Integer  
registro=Grid.Row 'obtiene el número de registro  
nuevo=0  
    NextForm "Cuestionario1"  
End Sub
```

Código utilizado para el proyecto en Superwaba

```
/* Cuestionario.java
 * Created on 5 de agosto de 2007, 04:16 PM*/
package newpackage;
import waba.ui.*;
import waba.fx.*;
import waba.io.*;
/* @author Rene Gazano*/
public class Cuestionario extends MainWindow
{
//declaración de variables y clases
static Cuestionario0 cuest; //se define como Static para poder cambiar entre
static Cuestionario1 cuest1; // formularios ya que extienden a Container
static Cuestionario2 cuest2;
static Cuestionario3 cuest3;
static Cuestionario4 cuest4;
static Cuestionario5 cuest5;
static Cuestionario6 cuest6;
static Cuestionario7 cuests;
MenuBar menu;// se define la barra de Menú
/** Crea una nueva instancia de Cuestionario */
public Cuestionario() {
    super("Cuestionario",TAB_ONLY_BORDER);
    waba.sys.Settings.setUIStyle(waba.sys.Settings.PalmOS); //Se define el estilo
//de Palm en la interfaz de usuario
}
public void onStart() //método onStart de Cuestionario
{
    Rect r=getClientRect();
    cuest=new Cuestionario0(); //Cuestionario de 0 a 7 extienden a Container
    cuest.setRect(r);
    cuest1=new Cuestionario1();
    cuest1.setRect(r);
    cuest2=new Cuestionario2();
    cuest2.setRect(r);
    cuest3=new Cuestionario3();
    cuest3.setRect(r);
    cuest4=new Cuestionario4();
    cuest4.setRect(r);
    cuest5=new Cuestionario5();
    cuest5.setRect(r);
    cuest6=new Cuestionario6();
    cuest6.setRect(r);
    cuests=new Cuestionario7();
    cuests.setRect(r);
    swap(cuest);// el primer formulario que carga es el de bienvenida

    String columnaMenu[]={"Opciones","Acerca de...","Salir"}; //Menu
    menu=new MenuBar(new String[][]{columnaMenu});
    setMenuBar(menu);
}

public void onEvent(Event event) //Manejo de eventos para Cuestionario
{
    if(event.type==ControlEvent.WINDOW_CLOSED&& event.target==menu)
    {
        int sel=menu.getSelectedMenuItem();
        switch(sel)
        {
            case -1:break;
            case 001:
            {
                new MessageBox("SuperWaba","Cuestionario desarrollado por|René Gazano
Cancino|rgazano@gmail.com").popupModal();
                break;
            }
            case 002:
            {
                exit(0);
                break;
            }
        }
    }
}
}
```

El formulario de bienvenida se detalla a continuación:

```
package newpackage;
import waba.ui.*;
import waba.io.*;
import waba.fx.*;

public class Cuestionario0 extends Container{
    Button b1,b2,b3; //declaración de los controles a utilizar
```



```

Label e1,e2,e3;
public void onStart()
{
    b1=new Button("Salir"); //creación de controles
    b2=new Button("Continuar");
    Button.commonGap=2;
    String imgName="unam.bmp";
    b3=new Button(new Image("imgName"));
    e1=new Label("Bienvenido");
    e2=new Label("Cuestionario desarrollado en");
    e3=new Label("SUPERWABA");
    add(b1,25,130);//se agregan los controles
    add(b2,90,130);
    add(b3,70,10);
    add(e1,10,10);
    add(e2,5,100);
    add(e3,10,110);
}
public void onEvent(Event event)
{
    if (event.type==ControlEvent.PRESSED&&event.target==b2)
    {
        MainWindow.getMainWindow().swap(Cuestionario.cuest1);
// la sentencia anterior sirve para cambiar el formulario por medio de Swap a otro
//Container
    }
    if (event.type==ControlEvent.PRESSED&&event.target==b1)
    {
        MainWindow.getMainWindow().exit(0);//salir de la aplicación
    }
}
}
}

```

Para el formulario de captura es importante el uso de la clase Catalog utilizada para almacenar datos en el dispositivo. Esta clase representa la creación del archivo PDB. Es un sistema de bases de datos que consiste en un arreglo de registros. Cada registro puede tener un tamaño distinto, no hay índices, no hay definición de la tabla, de campos. Desde el punto de vista de Palm OS, el contenido del registro es solo una fila de datos.

Los catálogos tienen algunas restricciones:

- El número de registros es limitado a 65536
- El tamaño del registro está limitado a 64Kb menos 24 bytes.

A continuación la creación del formulario de captura

```

package newpackage;
import waba.ui.*;
import waba.io.*;

public class Cuestionario1 extends Container{
    Label e1,e2,e3,e4,e5,e6; //definición de controles
    Button b1,b2,b3,b4,b5,b6,b7,b8,b9;
    Edit ed1,ed2,ed3,ed4,ed5;
    ListBox lb1;
    Catalog cata;
    ResizeStream rs;
    DataStream ds;
    String cad;
    int iden=0;
    String numcuest=new String();
    String nombre=new String();
    String puesto=new String();
    String Fecha=new String();
    String Hora=new String();
    String Sexo=new String();
    Calendar cal;

    public void onStart() {
        b1=new Button("Salir");//se crean y se agregan los controles necesarios
        b2=new Button("Continuar");
        b3=new Button("Ver Registros");
        b4=new Button("Borrar");
        b5=new Button("<");
        b6=new Button("<");
        b7=new Button("Nuevo");
        b8=new Button(">");
        b9=new Button(">");
    }
}

```



```

e1=new Label("No.Cuestionario:");
e2=new Label("Fecha:");
e3=new Label("Hora:");
e4=new Label("Nombre:");
e5=new Label("Puesto:");
e6=new Label("Sexo:");

ed1=new Edit();
ed1.setMaxLength(3);
ed1.setValidChars("1234567890");
add(ed1);
ed1.setRect(80,5,30,10);

ed2=new Edit();
ed2.setMode(Edit.DATE);
add(ed2);
ed1.setValidChars("1234567890/");
ed2.setKeyboard(Edit.KBD_CALENDAR); //instrucción para definir por defecto el
//teclado de calendario para elegir fecha u hora
ed2.setRect(30,22,45,10);

ed3=new Edit();
ed3.setMode(Edit.DATE);
add(ed3);
ed1.setValidChars("1234567890:");
ed3.setKeyboard(Edit.KBD_CALENDAR);
ed3.setRect(105,22,45,10);

ed4=new Edit();
add(ed4);
ed4.setRect(40,39,110,10);
ed5=new Edit();
add(ed5);
ed5.setRect(35,55,110,10);
String []items1={"M","F"};
lb1=new ListBox(items1);
add(lb1);
lb1.setRect(40,72,30,20);
add(b1,120,113);
b1.setRect(120,113,25,13);
add(b2,110,95);
add(b3,0,95);
b3.setRect(0,95,60,13);
add(b4,0,113);
add(b5,5,130);
b5.setRect(5,130,20,13);
add(b6,30,130);
b6.setRect(30,130,20,13);
add(b7,55,130);
b7.setRect(55,130,30,13);
add(b8,90,130);
b8.setRect(90,130,20,13);
add(b9,115,130);
b9.setRect(115,130,20,13);
add(e1,5,5);
add(e2,5,22);
add(e3,80,22);
add(e4,5,39);
add(e5,5,55);
add(e6,15,72);
}

public void Agregar(String numcuest,String nombre,String puesto,String fecha,String
hora,String sexo) //Metodo que agrega los argumentos al catálogo
{
    cata=new Catalog("Encuesta.GCRP.DATA",Catalog.CREATE); //creación del catálogo
//Encuesta con ID de creador GCRP de tipo DATA
    if (cata.isOpen())
    {
        rs=new ResizeStream(cata,512);
        ds=new DataStream(rs);
        rs.startRecord();
        ds.writeString(numcuest); //asignación de valores al catálogo por medio de
//Datastream
        ds.writeString(nombre);
        ds.writeString(puesto);
        ds.writeString(fecha);
        ds.writeString(hora);
        ds.writeString(sexo);
        rs.endRecord(); //fin del registro
        cata.close(); // cierra el catálogo
    }
}

public void onEvent(Event event)
{
    if (event.type==ControlEvent.PRESSED&&event.target==b2)

```

```

    {
        //momento de la asignación cuando se oprime el botón continuar
        numcuest=ed1.getText();
        Fecha=ed2.getText();
        Hora=ed3.getText();
        nombre=ed4.getText();
        puesto=ed5.getText();
        Sexo=lbl1.toString();
        Agregar(numcuest, nombre, puesto, Fecha, Hora, Sexo); //método agregar
        ed1.setText("");
        ed2.setText("");
        ed3.setText("");
        ed4.setText("");
        ed5.setText("");
        lbl1.select(0);

        MainWindow.getMainWindow().swap(Cuestionario.cuest2); //continua con la
//encuesta
    }
    if (event.type==ControlEvent.PRESSED&&event.target==b1)
    {
        MainWindow.getMainWindow().exit(0);
    }
    if (event.type==ControlEvent.PRESSED&&event.target==b3)
    {
        MainWindow.getMainWindow().swap(Cuestionario.cuests);
    }
    if (event.type==ControlEvent.PRESSED&&event.target==b7)
    {
        MainWindow.getMainWindow().swap(Cuestionario.cuest1);
    }
}
}
}

```

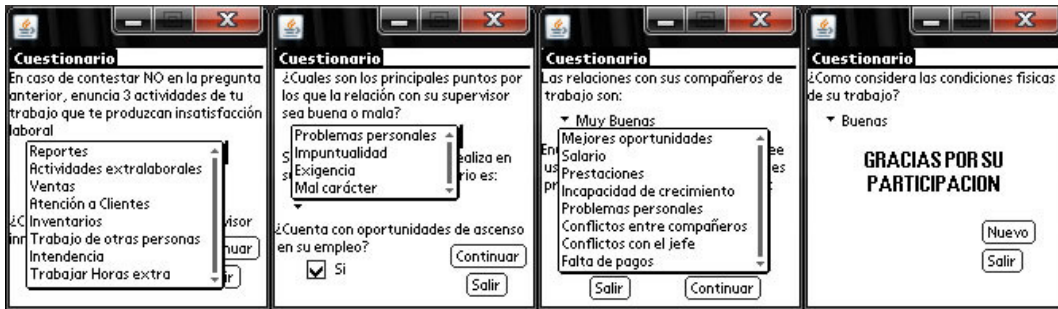
Para seguir con la encuesta el formulario Cuestionario2 se define de la siguiente forma:

```

package newpackage;
import waba.ui.*;
public class Cuestionario2 extends Container{
    Label e1,e2;
    Button b1,b2;
    ComboBox p1;
    Check ch1;
    public void onStart()
    {
        e1=new Label("¿Cuán satisfecho está con su trabajo?");
        e2=new Label("¿Estás satisfecho con las actividades que realizas en tu trabajo?");
        b1=new Button("Salir");
        b2=new Button("Continuar");
        ch1=new Check("Si");
        String []items1={"", "Altamente Satisfecho", "Satisfecho", "Regularmente Satisfecho", "Insatisfecho", "Altamente Insatisfecho"};
        p1=new ComboBox(items1); //definición de las preguntas con opción múltiple
        add(e1,1,1);
        add(e2,1,75);
        add(p1);
        add(ch1,20,105);
        add(b1,120,123);
        add(b2,110,105);
        p1.setRect(15,20,PREFERRED+10,PREFERRED);
    }
    public void onEvent(Event event)
    {
        if (event.type==ControlEvent.PRESSED&&event.target==b2)
        {
            MainWindow.getMainWindow().swap(Cuestionario.cuest3);
        }
        if (event.type==ControlEvent.PRESSED&&event.target==b1)
        {
            MainWindow.getMainWindow().exit(0);
        }
    }
}
}

```





La creación de los formularios Cuestionario2 a Cuestionario6 se definen de la misma forma debido a su similitud. El siguiente formulario en importancia es la tabla en la que se visualizan los registros, en este caso llamado Cuestionario7.

```

package newpackage;
import waba.ui.*;
import waba.fx.*;
import waba.io.*;

public class Cuestionario7 extends Container{
    Grid grid;//declaración de controles
    Button b1,b2;
    Catalog cata;
    ResizeStream rs;
    DataStream ds;
    String cad;

    public void onStart()
    {
        String []gridCaptions = {"No.Cuest.", "Nombre",
"Puesto"};
        int w=fm.getTextWidth("xxx");

        int www = fm.getTextWidth("xxxxxxxxxxxx");
        int ww=fm.getTextWidth("xxxxx");
        int gridWidths[] = {w, www, ww};
        int gridAligns[] = {LEFT,LEFT,LEFT};
        grid = new Grid( gridCaptions, gridWidths, gridAligns, false ); //generador del control
        //Grid el cual contiene los títulos de las columnas, el ancho de las mismas, alineación
        //de los textos contenidos en ellas, y si son modificables o no
        grid.verticalLineStyle = Grid.VERT_NONE;
        add( grid );
        grid.setRect(0,0,160,130);

        cata=new Catalog("Encuesta.GCRP.DATA",Catalog.READ_ONLY); //creación del catálogo para
        //solo lectura
        int tamaño=cata.getRecordCount();
        if(cata.isOpen())
        {
            rs=new ResizeStream(cata,512);
            ds=new DataStream(rs);
            String items[][]=new String[tamaño][10];
            for (int i=0;i<tamaño;i++)
            {
                for (int j=0;j<10;j++) //se define el Grid con 10 renglones
                {
                    cad=ds.readString();//lectura de los datos del catálogo
                    items[i][j]=cad;
                }
                cata.setRecordPos(i);
            }
        }
        grid.setItems( items );
        cata.close();
    }

    b1=new Button("Salir");
    b2=new Button("Nuevo");
    add(b1,30,130);
    add(b2,90,130);
}

    public void onEvent(Event event)
    {
        if (event.type==ControlEvent.PRESSED&&event.target==b2)
        {
            MainWindow.getMainWindow().swap(Cuestionario.cuest1);
        }
        if (event.type==ControlEvent.PRESSED&&event.target==b1)
        {
            MainWindow.getMainWindow().exit(0);
        }
    }
}

```



} }

Fuentes:

Tipos de computadoras

- <http://www.jegsworks.com/Lessons-sp/lesson1-2/lesson1-2.htm>
- <http://computer.howstuffworks.com/question543.htm>

Sistemas Operativos

- Goscinski, Andrej (1991). *Distributed Operating Systems: The logical Design*. Prentice-Hall. 2ª edición.
- Milenkovic, Milan (1994). *Sistemas Operativos: conceptos y diseño*. McGraw-Hill. 2ª edición.
- Stallings, William (2001). *Sistemas Operativos, Principios de diseño e interioridades*. Prentice Hall 4ª edición.
- http://es.wikipedia.org/wiki/Historia_y_evoluci%C3%B3n_de_los_sistemas_operativos

La programación

- Braude, Eric. *Ingeniería de Software, Una perspectiva orientada a objetos*. Ed. Alfaomega
- <http://www.desarrolloweb.com/articulos/2477.php>
- http://en.wikipedia.org/wiki/History_of_programming_languages
- http://es.wikipedia.org/wiki/Lenguaje_de_programaci%C3%B3n
- http://www.oreilly.com/pub/a/oreilly/news/languageposter_0504.html
- <http://www.blueink.biz/RapidApplicationDevelopment.aspx>

Historia de los PDA

- <http://www.palm.com/us/products/palmpilot/>
- http://es.wikipedia.org/wiki/Palm_OS
- http://www.medindia.net/pda/pda_history.htm
- <http://www.geocities.com/mywebbadepalm/historia.html>
- <http://www.canalpda.com/node/279>
- <http://www.deeptec.com/palmevolution/palmtree.html>

Funcionamiento de los PDA

- Carmen Carmack y Craig Freudenrich, Ph. D.
<http://communication.howstuffworks.com/pda.htm>

Sistemas Operativos para PDA

- <http://www.astalavista.com/media/directory06/uploads/jukka.ahonen.pdf>
Jukka Ahonen, *PDA OS Security: Application Execution*. Universidad Tecnológica de Helsinki.

Palm OS

- <http://www.palm.com>
- <http://www.access-company.com>
- Winton, Greg. *Palm OS Network Programming*. Ed. O'Reilly 1ra. Edición.

PDAs y Smartphones

- <http://www.palm.com/mx/>
- <http://www.symbol.com/products/mobile-computers>
- <http://www.aceeca.com/index.php>
- <https://buy.garmin.com/shop/shop.do?CID=140>
- <http://www.janam.com/handheld-products.php>
- http://www.pdaexpertos.com/modelos_pda/pitech_qool_qda-700.shtml

Programación en Dispositivos Móviles, Lenguajes y entornos de programación en Palm OS

- Winton, Greg. *Palm OS Network Programming*. Ed. O'Reilly 1ra. Edición.
- <http://www.access-company.com/developers/downloads/apptools.html#wireless>
- <http://mobile.eric-poncet.com/frames.html>
- http://www.nada.kth.se/projects/proj00/TV-guide/html_doc/Conduit.htm

El diseño de aplicaciones

- Viera, Victor. *Técnicas para el desarrollo de computación móvil (orientado a PDA)*. <http://www.mailxmail.com/curso/informatica/pda>
- Winton, Greg. *Palm OS Network Programming*. Ed. O'Reilly 1ra. Edición.

La investigación de Campo

- Sabino Carlos, (1992). *El proceso de investigación*. Disponible en <http://paginas.ufm.edu/Sabino/PI.htm>
- Pérez Bilbao Jesús, Fidalgo Vega Manuel. *Satisfacción Laboral: escala de satisfacción*. Disponible en http://www.mtas.es/insh/ntp/ntp_394.htm
- Torres Valencia Olivia Sugey. *Insatisfacción Laboral*. Disponible en <http://www.monografias.com/trabajos14/insatisf-laboral/insatisf-laboral.shtml>

PDA Toolbox

- <http://www.pdatoolbox.com>

CASL

- <http://www.caslsoft.com>

HB++

- <http://www.handheld-basic.com>

NSBasic

- <http://www.nsbasic.com>

Satellite Forms

- <http://www.satelliteforms.com>

Superwaba

- <http://www.superwaba.com.br>

HandDBase

- <http://www.ddhsoftware.com>

El Futuro de los PDA

- <http://www.access-company.com/spanish/>
- <http://www.aceeca.com/home/index.php>