GESTIÓN DE PROCESOS INDUSTRIALES A TRAVÉS DE MÓVILES PARA MANEJAR Y CONTROLAR VARIABLES FÍSICAS

RICARDO REYES ALZATE
CHRISTIAN GARCIA LOPEZ

Presidente: Diego Fernando Quintero López

UNIVERSIDAD DE MANIZALES
FACULTAD DE INGENIERIA
INGENIERIA DE SISTEMAS Y TELECOMUNICACIONES
MANIZALES, FEBRERO DE 2008

GESTIÓN DE PROCESOS INDUSTRIALES A TRAVÉS DE MÓVILES PARA MANEJAR Y CONTROLAR VARIABLES FÍSICAS

RICARDO REYES ALZATE
CHRISTIAN GARCIA LOPEZ

UNIVERSIDAD DE MANIZALES
FACULTAD DE INGENIERIA
INGENIERIA DE SISTEMAS Y TELECOMUNICACIONES
MANIZALES, FEBRERO DE 2008

TABLA DE CONTENIDO

ANEXOS

LISTA ESPECIALES

GLOSARIO

API: Interfaz de programación de aplicaciones (Applications Programming Interface): una serie de funciones que están disponibles para realizar programas para un cierto entorno.

CDC: Está orientada a dispositivos con cierta capacidad computacional y de memoria. Por ejemplo, decodificadores de televisión digital, televisores con internet, algunos electrodomésticos y sistemas de navegación en automóviles.

CLDC: (Connected Limited Device Configuration) Configuración de dispositivos limitados con conexión, está orientada a dispositivos dotados de conexión y con limitaciones en cuanto a capacidad gráfica, cómputo y memoria. Un ejemplo de estos dispositivos son: teléfonos móviles, buscapersonas (pagers), PDAs, organizadores personales, etc.

DoJa: Se basa en el API de Java ME CLDC que se define en el JCP (Java Community Process). DoJa es un perfil definido por NTT DoCoMo para proporcionar comunicaciones y otras entradas-salidas de procesamiento, interfaz de usuario (GUI) y otras características y funciones exclusivas de i-mode, las bibliotecas y la ampliación definida por cada terminal de teléfono son los encargados de añadir funciones originales. Sin embargo, en contraste con otros perfiles como Java ME Mobile Information Device Profile (MIDP) o Módulo de Información de Perfil (IMP), DoJa no se ha definido como Java Specification Request (JSR).

GPRS: (Servicio General Paquetes por Radio). Servicio de comunicación de telefonía móvil basado en la transmisión de paquetes. Puede transmitir a una velocidad de 114 kbit/s y permite la conexión a Internet.

Hardware: Es un neologismo proveniente del inglés definido por la RAE como el conjunto de elementos materiales que conforman una computadora, sin embargo, es usual que sea utilizado en una forma más amplia, generalmente para describir componentes físicos de una tecnología, así el hardware puede ser de un equipo militar importante, un equipo electrónico, un equipo informático o un robot.

iPhone: Es un teléfono celular multimedia, con capacidad para navegar en Internet y tecnología GSM y EDGE, que fue desarrollado y es comercializado por Apple.

IMP: (Information Module Profile) Es una versión de MIDP pero sin interfaz de usuario.

J2ME: Java 2 Platform Micro Edition. Es la respuesta de Sun Microsystems para una plataforma de dispositivos inalámbricos. El J2ME permite que los desarrolladores usen Java y las herramientas inalámbricas J2ME para crear aplicaciones y programas para dispositivos móviles inalámbricos.

J2SE: Java 2, Standard Edition. Versión básica del conjunto de herramientas y APIs de Sun Microsystems destinadas a la creación de aplicaciones en plataforma Java.

MIDP: (Mobile Information Device Profile), Este perfil está construido sobre la configuración CLDC. Al igual que CLDC fue la primera configuración definida para J2ME, MIDP fue el primer perfil definido para esta plataforma.

OPC: (OLE for Process Control) es un estándar de comunicación en el campo del control y supervisión de procesos. Este estándar permite que diferentes fuentes de datos envíen datos a un mismo servidor OPC, al que a su vez podrán conectarse diferentes programas compatibles con dicho estándar. De este modo se elimina la necesidad de que todos los programas cuenten con drivers para dialogar con múltiples fuentes de datos, basta que tengan un driver OPC.

PLC: Programmable logic controller o Controlador lógico programable, los PLC no sólo controlan la lógica de funcionamiento de máquinas, plantas y procesos industriales, sino que también pueden realizar operaciones aritméticas, manejar señales analógicas para realizar estrategias de control, tales como controladores.

PERFIL: Es el que define las APIs que controlan el ciclo de vida de la aplicación, interfaz de usuario, etc. Más concretamente, un perfil es un conjunto de APIs orientado a un ámbito de aplicación determinado. Los perfiles identifican un grupo de dispositivos por la funcionalidad que proporcionan (electrodomésticos, teléfonos móviles, etc.) y el tipo de aplicaciones que se ejecutarán en ellos.

SSL: Secure Sockets Layer (SSL) y Transport Layer Security (TLS) -Seguridad de la Capa de Transporte-, su sucesor, son protocolos criptográficos que proporcionan comunicaciones seguras en Internet. Existen pequeñas diferencias entre SSL 3.0 y TLS 1.0, pero el protocolo permanece sustancialmente igual.

WEB: Servidor de información WWW. Se utiliza también para definir el universo WWW en su conjunto.

RESUMEN

El proyecto se centra en los aspectos de seguridad del J2ME* y su implicación en el mercado de los móviles. Tiene como objetivo diseñar e implementar una aplicación que permita manipular variables físicas como la temperatura a través de dispositivos móviles. Para lo anterior se instalará un servidor, el cual permitirá administrar los datos de la aplicación. Se construirá una aplicación móvil para manipular las variables de temperatura realizando una interpretación y una respuesta a los datos recibidos. Se montará un circuito electrónico (sensor de temperatura, PLC*), que posibilitará la captura de variables físicas. Esto será posible a través de un servidor WEB el cual nos dejará conectar los dispositivos móviles. En este se implementará una base de datos en la cual estarán los datos relacionados con los procesos que se van a manejar; en los dispositivos móviles solo se manejará la interfaz de usuario necesaria para manipularla.

La estructura del prototipo permitirá que estas variables físicas puedan ser enviadas y procesadas con móviles actuales y futuros, que tendrán implementadas más funcionalidades. Se pretende que la aplicación incluya los mecanismos de seguridad necesarios para garantizar la autenticidad y la integridad de los datos.

Con esta propuesta se presenta un análisis de las fuentes de riesgo que enfrentan el sector industrial, mediante las cuales los empleados se encuentran expuestos a peligros al manipular máquinas que manejan grandes niveles de temperatura, además de irregularidades en el manejo de la información, como son las lecturas que se obtienen al verificar los procesos.

Inicialmente se hace una introducción a la tecnología J2ME y se explican las motivaciones que han llevado a crear un estándar para dispositivos móviles; luego se verá la arquitectura de J2ME y las características de seguridad que se han definido en la especificación; a continuación se explica el montaje de la aplicación que permite manipular las variables físicas.

---

* Java 2 Platform Micro Edition. Es la respuesta de Sun Microsystems para una plataforma de dispositivos inalámbricos. El J2ME permite que los desarrolladores usen Java y las herramientas inalámbricas J2ME para crear aplicaciones y programas para dispositivos móviles inalámbricos. Consiste de dos elementos: configuraciones y perfiles. Las configuraciones proveen las librerías y una máquina virtual para una categoría de dispositivos inalámbricos.

* Programmable Logic Controller, los PLC no sólo controlan la lógica de funcionamiento de máquinas, plantas y procesos industriales, sino que también pueden realizar operaciones aritméticas, manejar señales analógicas para realizar estrategias de control, tales como controladores

ABSTRACT

The project focuses on the security aspects of J2ME and its involvement in the market for mobile. Its objective to design and implement an application that allows manipulate physical variables such as temperature via mobile devices. For the foregoing will install a server, which will manage application data. It will build a mobile application to manipulate variables temperature conducting an interpretation and a response to the data received. It mounted an electronic circuit (temperature sensor, PLC), which allow the capture of physical variables. This will be possible through a web server which will be leaving us to connect mobile devices. This will be implemented in a database which will include data related to the processes that are going to handle; In mobile devices will be handled single user interface to manipulate it.

The structure of the prototype will allow these physical variables can be sent and processed with current and future mobile, which will be implemented more features. It is intended that the application include adequate security arrangements to ensure the authenticity and integrity of data.

This proposal provides an analysis of the sources of risk faced by the industrial sector, through which employees are exposed to danger when handling machines that handle large temperature levels, as well as irregularities in the handling of information, such as the readings are obtained when checking processes.

Initially provides an introduction to technology J2ME and explains the motivations that have led to the creation of a standard for mobile devices; Then you will see the architecture of J2ME and the security features that have been defined in the specification; Then explains the assembly of the application that allows you to manipulate the physical variables.

INTRODUCCIÓN

J2ME es un subconjunto de Java orientado a dispositivos con recursos limitados, es decir, a un hardware en el que características como la memoria o la velocidad del procesador son muy inferiores a las de un ordenador convencional.

Al igual que ocurre con el J2SE, la principal virtud de J2ME es su habilidad para ejecutarse en distintas plataformas: casi cualquier móvil de hoy en día incorpora soporte Java (las excepciones más significativas son el iPhone y los móviles con Windows Mobile). Sin embargo, no todos los dispositivos tienen el mismo soporte de J2ME.

J2ME se divide en configuraciones (configurations), perfiles (profiles) y APIs opcionales. Una configuración define un tipo de dispositivo en función de las características de su hardware: sus limitaciones, sus capacidades y le asigna una máquina virtual y un conjunto de APIs adecuados a ese hardware. En la actualidad existen dos configuraciones: CDC (Connected Device Configuration), utilizada sobre todo en sistemas de telemetría, automoción o domótica, y CLDC (Connected Limited Device Configuration) que es una versión más limitada y que es la que nos interesa por estar presente en la mayoría de los móviles.

Dentro de una configuración, un perfil nos define ciertas características concretas, como la interfaz de usuario. Existen tres perfiles para la configuración CLDC: MIDP (Mobile Information Device Profile), que es la usada en los teléfonos móviles y por tanto la que nos interesa, IMP (Information Module Profile) que es una versión de la anterior sin interfaz de usuario, y DoJa, destinado a un tipo de móviles japoneses.

Esto permite a este proyecto realizar de forma más versátil su aplicación móvil ya que al utilizar J2ME e integrarla con un cliente en java implementando el protocolo OPC Fundation mejora su comunicación y la forma en que se interpretan los datos de los dispositivos de control.

1. DESCRIPCIÓN Y PLANTEAMIENTO DEL PROBLEMA

Gran cantidad de procesos que se llevan a cabo en las industrias se ven afectados por causas externas e internas, además la poca movilidad que se tiene a la hora de estar monitoreándolos hace que estos sean difícil de controlar; muchas veces, los procesos industriales son afectados por el mal funcionamiento de las máquinas, las cuales trabajan continuamente sin tener descanso; cuando fallan, el personal encargado se da cuenta muy tarde; por lo tanto la producción se paraliza o la materia prima de las empresas se estropea.

La empresa donde se desea implementar el sistema como caso piloto, se encuentra ubicada en el oriente de la ciudad de Manizales, dedicada a la producción de Neveras, donde tienen las secciones de mantenimiento, pintura, metales, ensamble, bodegas, etc. El proyecto se aplicará en el proceso de mezcla de Poliol más R22, el cual presenta más fallas y se debe estar monitoreando constantemente para que la producción no se detenga.

En la actualidad los procesos donde manejan altas temperaturas son monitoreados por personas quienes se encargan de supervisar el buen funcionamiento; si el proceso es interrumpido producirá pérdidas materiales y económicas. El problema se presenta cuando hay fallas en las lecturas e interpretación de las temperaturas presentes en el sector industrial donde se pretende desarrollar el sistema para manejar estas variables; en este caso se centrará en la producción de láminas metálicas pintadas que son utilizadas para el ensamblaje de neveras; lo más importante del tratamiento de la materia prima es el manejo de las medidas de temperatura de los tanques y el horno por donde pasa la lamina.

Estos procesos son afectados por el mal manejo de la información de estas máquinas como son las lecturas que se obtienen al verificar las temperaturas, lo que permite justificar el planteamiento de un desarrollo para el control de las mismas utilizando la plataforma J2ME a través de móviles.

2. OBJETIVOS

2.1 OBJETIVO GENERAL

- Diseñar e implementar un sistema de información que permita monitorear y controlar mediante dispositivos móviles variables físicas como temperaturas, niveles, presiones, etc. en procesos industriales, utilizando para ello la tecnología J2ME.

2.2 OBJETIVOS ESPECIFICOS

- Diseñar e implementar una base de datos la cual almacenara la información de las mediciones de los procesos que manejan variables de temperatura en tiempo real.
- Diseñar e implementar la aplicación móvil para manipular las variables de temperatura realizando una interpretación y una respuesta a los datos recibidos.
- Realizar la interfaz J2ME que permita la conectividad entre los dispositivos móviles y la base de datos del servidor.
- Diseñar una aplicación de monitoreo y control para las variables físicas de los procesos industriales.
- Realizar pruebas funcionales para la puesta a punto e implementación del sistema.

3. JUSTIFICACIÓN

La gestión de procesos a través de móviles permitirá dar solución a la interpretación y manejo de las temperaturas en los procesos industriales. Dicha gestión mejorara el rendimiento y calidad de los procesos industriales en la empresa MABE Colombia, ya que surge la necesidad de supervisar y manipular de manera más cómoda y eficiente, beneficiar al empresario, evitar pérdidas económicas, tiempo de operación.

Esta propuesta optimizara la calidad de los procesos de máquinas, a través de sensores, actuadotes, y brindara a los usuarios una portabilidad y comodidad.

Con esta aplicación el funcionamiento y desarrollo de las industrias será más fácil de gestionar ya que se obtienen datos en tiempo real, lo cual permite que los supervisores de estos procesos puedan dar una respuesta oportuna a eventos inesperados.

En cuanto a la realidad de las empresas, es claro saber lo que pasa con los procesos, muchas veces son interrumpidos por fallas en tanques, hornos, etc. Esta dificultad repercute negativamente en el desarrollo de la empresa por carecer del personal suficiente para controlar el funcionamiento de las máquinas.

¿Cuál será el beneficio social?, ¿Cuál es la novedad de la propuesta?, de lo propuesto se brinda una alternativa de desarrollo tecnológico, mejora la seguridad industrial, evitando accidentes laborales, posee soporte estable, seguro y confiable; además el bajo costo lo hace asequible a las pequeñas y medianas industrias.

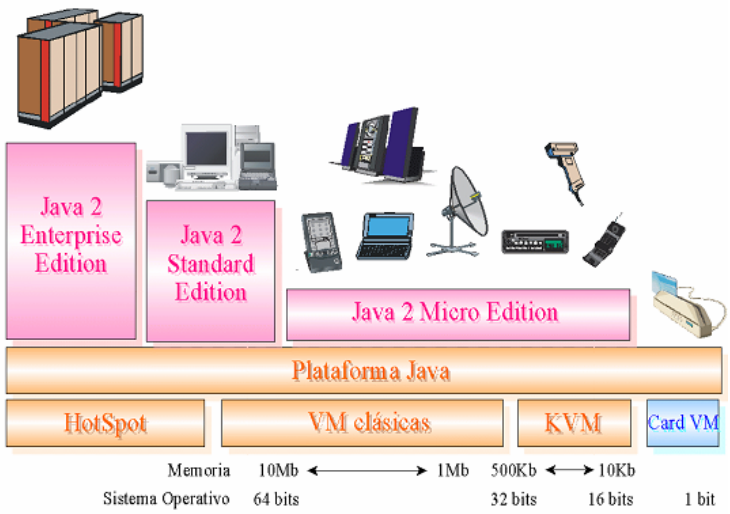
4. MARCO TEÓRICO

4.1 HERRAMIENTAS DE DESARROLLO DEL PROYECTO

4.1.1 Java 2 Platform, Micro Edition (J2ME)

Según GÁLVEZ R Sergio, ORTEGA D Lucas[1], esta versión de Java está enfocada a la aplicación de la tecnología Java en dispositivos electrónicos con capacidades computacionales y gráficas muy reducidas, tales como teléfonos móviles, PDAs o electrodomésticos inteligentes. Esta edición tiene unos componentes básicos que la diferencian de las otras versiones, como el uso de una máquina virtual denominada KVM (Kilo Virtual Machine, debido a que requiere sólo unos pocos Kilobytes de memoria para funcionar) en vez del uso de la JVM clásica, inclusión de un pequeño y rápido recolector de basura.

Figura 2. Arquitectura de la plataforma Java 2



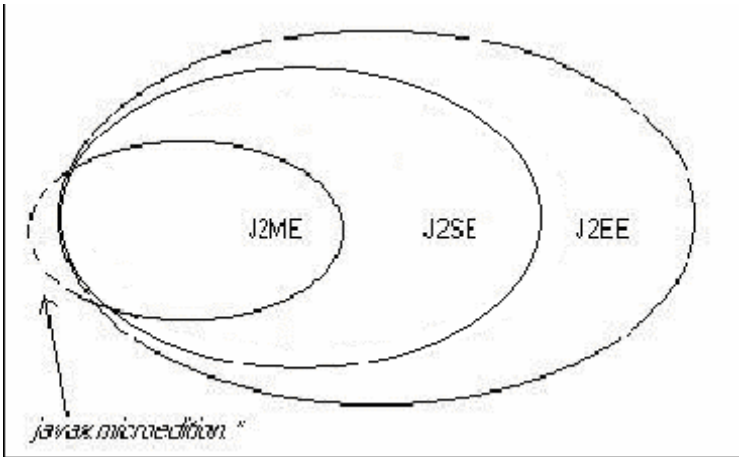
Fuente: GÁLVEZ R Sergio, ORTEGA D Lucas. Java a Tope: J2ME (Java 2 Microedition). [En línea]. Fecha de publicación: 2003. Universidad de Málaga. Disponible en: http://www.lcc.uma.es/~galvez/J2ME.html.

[1] GÁLVEZ R Sergio, ORTEGA D Lucas. Java a Tope: J2ME (Java 2 Microedition). [En línea]. Fecha de publicación: 2003. Consultado el 23-7-2007. Universidad de Málaga. Disponible en: http://www.lcc.uma.es/~galvez/J2ME.html

En la actualidad no es realista ver Java como un simple lenguaje de programación, si no como un conjunto de tecnologías que abarca a todos los ámbitos de la computación con dos elementos en común:

- El código fuente en lenguaje Java es compilado a código intermedio interpretado por una Java Virtual Machine (JVM), por lo que el código ya compilado es independiente de la plataforma.
- Todas las tecnologías comparten un conjunto más o menos amplio de APIs básicas del lenguaje, agrupadas principalmente en los paquetes java.lang y java.io.

Figura 3. Relación entre las APIs de la plataforma Java.



Fuente: GÁLVEZ R Sergio, ORTEGA D Lucas. Java a Tope: J2ME (Java 2 Microedition). [En línea]. Fecha de publicación: 2003. Universidad de Málaga. Disponible en: http://www.lcc.uma.es/~galvez/J2ME.html.

Componentes que forman parte de la Tecnología J2ME.

Por un lado se tiene una serie de máquinas virtuales Java con diferentes requisitos, cada una para diferentes tipos de pequeños dispositivos.

- Configuraciones, que son un conjunto de clases básicas orientadas a conformar el corazón de las implementaciones para dispositivos de

características específicas. Existen 2 configuraciones definidas en J2ME: Connected Limited Device Configuration (CLDC) enfocada a dispositivos con restricciones de procesamiento y memoria, y Connected Device Configuration (CDC) enfocada a dispositivos con más recursos.

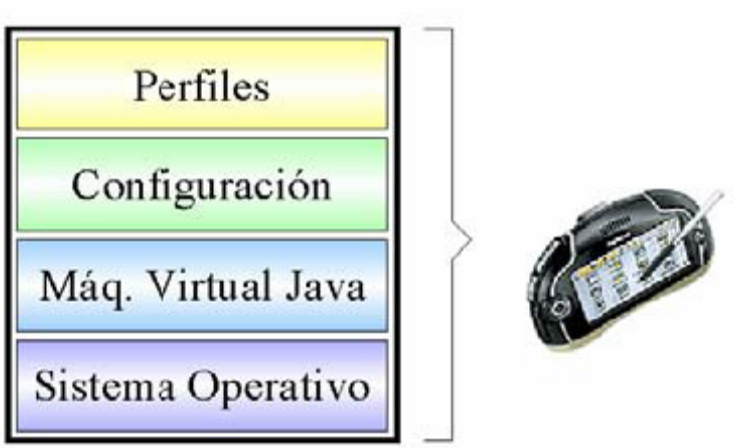
- Perfiles, que son unas bibliotecas Java de clases específicas orientadas a implementar funcionalidades de más alto nivel para familias específicas de dispositivos.
- Un entorno de ejecución determinado de J2ME se compone entonces de una selección de:
    a) Máquina virtual.
    b) Configuración.
    c) Perfil.
    d) Paquetes Opcionales.

Máquinas Virtuales J2ME

Una máquina virtual de Java (JVM) es un programa encargado de interpretar código intermedio (bytecode) de los programas Java precompilados a código máquina ejecutable por la plataforma, efectuar las llamadas pertinentes al sistema operativo subyacente y observar las reglas de seguridad y corrección de código definidas para el lenguaje Java. De esta forma, la JVM proporciona al programa Java independencia de la plataforma con respecto al hardware y al sistema operativo subyacente. Las implementaciones tradicionales de JVM son, en general, muy pesadas en cuanto a memoria ocupada y requerimientos computacionales. J2ME define varias JVMs de referencia adecuadas al ámbito de los dispositivos electrónicos que, en algunos casos, suprimen algunas características con el fin de obtener una implementación menos exigente[2].

[2] GÁLVEZ R Sergio, ORTEGA D Lucas. Java a Tope: J2ME (Java 2 Microedition). [En línea]. Fecha de publicación: 2003. Universidad de Málaga. Disponible en: http://www.lcc.uma.es/~galvez/J2ME.html.

Figura 4. Entorno de ejecución.



Fuente: GÁLVEZ R Sergio, ORTEGA D Lucas. Java a Tope: J2ME (Java 2 Microedition). [En línea]. Fecha de publicación: 2003. Universidad de Málaga. Disponible en: http://www.lcc.uma.es/~galvez/J2ME.html.

Como se mencionó anteriormente existen 2 configuraciones CLDC y CDC. En consecuencia, cada una requiere su propia máquina virtual. La VM (Virtual Machine) de la configuración CLDC se denomina KVM y la de la configuración CDC se denomina CVM. A continuación se mencionan las características principales de cada una de ellas:

- **KVM**

  Se corresponde con la Máquina Virtual más pequeña desarrollada por Sun. Su nombre KVM proviene de Kilobyte (haciendo referencia a la baja ocupación de memoria, entre 40Kb y 80Kb). Se trata de una implementación de Máquina Virtual reducida y especialmente orientada a dispositivos con bajas capacidades computacionales y de memoria. La KVM está escrita en lenguaje C, aproximadamente unas 24000 líneas de código, y fue diseñada para ser:

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

Sin embargo, esta baja ocupación de memoria hace que posea algunas limitaciones con respecto a la clásica *Java Virtual Machine* (JVM):

1. No hay soporte para tipos en coma flotante. No existen por tanto los tipos double ni float. Esta limitación está presente porque los dispositivos carecen del hardware necesario para estas operaciones.
2. No existe soporte para JNI (*Java Native Interface*) debido a los recursos limitados de memoria.
3. No existen cargadores de clases (*class loaders*) definidos por el usuario. Sólo existen los predefinidos.
4. No se permiten los grupos de hilos o hilos *daemon*. Cuando queramos utilizar grupos de hilos utilizaremos los objetos *Colección* para almacenar cada hilo en el ámbito de la aplicación.
5. No existe la finalización de instancias de clases. No existe el método Object.finalize().
6. No hay referencias débiles[*].
7. Limitada capacidad para el manejo de excepciones debido a que el manejo de éstas depende en gran parte de las APIs de cada dispositivo por lo que son éstos los que controlan la mayoría de las excepciones.
8. Reflexión[*].

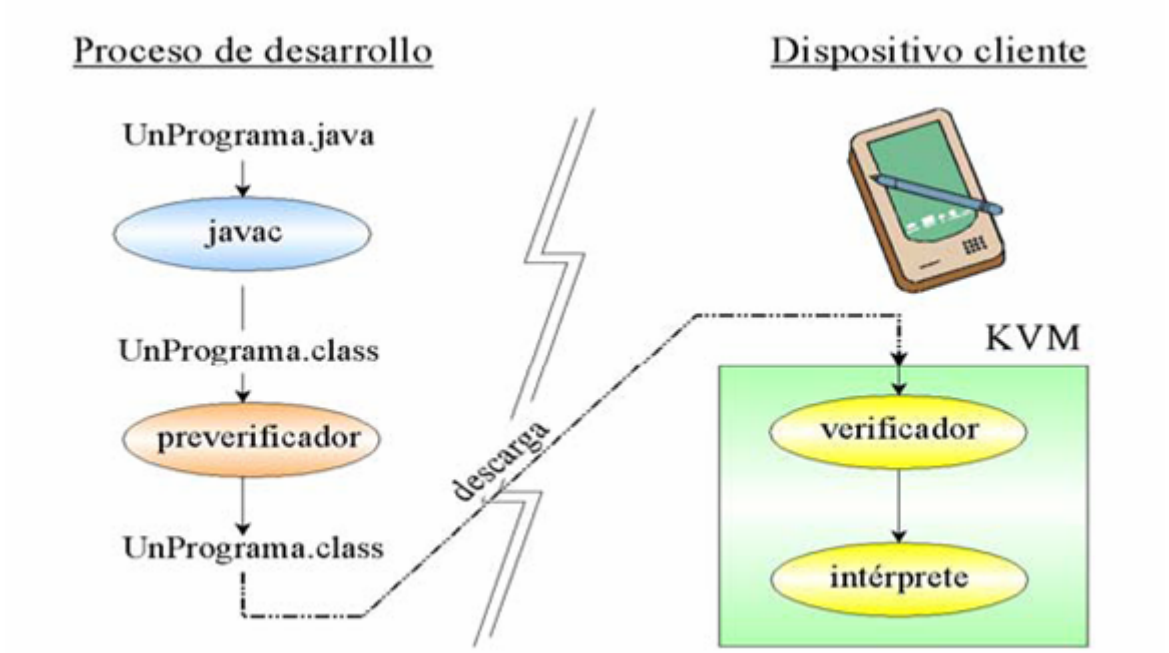
Aparte de la no inclusión de estas características, la verificación de clases merece un comentario aparte. El verificador de clases estándar de Java es demasiado grande para la KVM. De hecho es más grande que la propia KVM y el consumo de memoria es excesivo, más de 100Kb para las aplicaciones típicas. Este verificador de clases es el encargado de rechazar las clases no válidas en tiempo de ejecución. Este mecanismo verifica los *bytecodes* de las clases Java realizando las siguientes comprobaciones:

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

[*] Un objeto que está siendo apuntado mediante una referencia débil es un candidato para la recolección de basura. Estas referencias están permitidas en J2SE, pero no en J2ME.

[*] La reflexión es el mecanismo por el cual los objetos pueden obtener información de otros objetos en tiempo de ejecución como, por ejemplo, los archivos de clases cargados o sus campos y métodos.

Figura 5 Preverificación de clases en CDLC/KVM.



Fuente: GÁLVEZ R Sergio, ORTEGA D Lucas. Java a Tope: J2ME (Java 2 Microedition). [En línea]. Fecha de publicación: 2003. Universidad de Málaga. Disponible en: http://www.lcc.uma.es/~galvez/J2ME.html.

Por esta razón los dispositivos que usen la configuración CLDC y KVM introducen un algoritmo de verificación de clases en dos pasos. Este proceso puede apreciarse gráficamente en la Figura 5.

La KVM puede ser compilada y probada en 3 plataformas distintas:

1. Solaris Operating Environment.
2. Windows
3. PalmOs

- **CVM**

La CVM (Compact Virtual Machine) ha sido tomada como Máquina Virtual Java de referencia para la configuración CDC y soporta las mismas características que la Máquina Virtual de J2SE. Está orientada a dispositivos electrónicos con procesadores de 32 bits de gama alta y en torno a 2Mb o más de memoria RAM. Las características que presenta esta Máquina Virtual son:

1. Sistema de memoria avanzado.
2. Tiempo de espera bajo para el recolector de basura.

3. Separación completa de la VM del sistema de memoria.
4. Recolector de basura modularizado.
5. Portabilidad.
6. Rápida sincronización.
7. Ejecución de las clases Java fuera de la memoria de sólo lectura (ROM).
8. Soporte nativo de hilos.
9. Baja ocupación en memoria de las clases.
10. Proporciona soporte e interfaces para servicios en Sistemas Operativos de Tiempo Real.
11. Conversión de hilos Java a hilos nativos.
12. Soporte para todas las características de Java2 v1.3 y librerías de seguridad, referencias débiles, Interfaz Nativa de Java (JNI), invocación remota de métodos (RMI), Interfaz de depuración de la Máquina Virtual (JVMDI).

Configuraciones

Una configuración es el conjunto mínimo de APIs Java que permiten desarrollar aplicaciones para un grupo de dispositivos. Éstas APIs describen las características básicas, comunes a todos los dispositivos:

- Características soportadas del lenguaje de programación Java.
- Características soportadas por la Máquina Virtual Java.
- Bibliotecas básicas de Java y APIs soportadas.

Como ya se ha mencionado anteriormente existen dos configuraciones en J2ME: CLDC, orientada a dispositivos con limitaciones computacionales y de memoria y CDC, orientada a dispositivos con no tantas limitaciones. Ahora veremos un poco más en profundidad cada una de estas configuraciones.

• **Configuración de dispositivos con conexión, CDC** (*Connected Limited Configuration*)

La CDC está orientada a dispositivos con cierta capacidad computacional y de memoria. Por ejemplo, decodificadores de televisión digital, televisores con internet, algunos electrodomésticos y sistemas de navegación en automóviles. CDC usa una Máquina Virtual Java similar en sus características a una de J2SE, pero con limitaciones en el apartado gráfico y de memoria del dispositivo. Ésta Máquina Virtual es la que hemos visto como CVM (Compact Virtual Machine). La CDC está enfocada a dispositivos con las siguientes capacidades:

- Procesador de 32 bits.

- Disponer de 2 Mb o más de memoria total, incluyendo memoria RAM y ROM.
- Poseer la funcionalidad completa de la Máquina Virtual Java2.
- Conectividad a algún tipo de red.

La CDC está basada en J2SE v1.3 e incluye varios paquetes Java de la edición estándar. Las peculiaridades de la CDC están contenidas principalmente en el paquete javax.microedition.io, que incluye soporte para comunicaciones http y basadas en datagramas. La Tabla 1 nos muestra las librerías incluidas en la CDC.

Tabla 1. Librerías de configuración CDC

Nombre de paquete CDC	Descripción
Java.io	Clases e interfaces estándar de E/S
Java.lang	Clases básicas del lenguaje
Java.lang.ref	Clases de referencia
Java.lang.reflect	Clases e interfaces de reflection
Java.math	Paquete de matemáticas
Java.net	Clases e interfaces de red
Java.security	Clases e interfaces de seguridad
Java.security.cert	Clases de certificados de seguridad
Java.text	Paquete de texto
Java.util	Clases de utilidades estándar
Java.util.jar	Clases de utilidades para archivos jar
Java.util.zip	Clases y utilidades para archivos ZIP y comprimidos.
Javax.microedition.io	Clases e interfaces para conexión genérica CDC

J2ME y las comunicaciones

Los dispositivos que hagan uso de J2ME más específicamente CLDC-MIDP, necesitan poseer conexión a algún tipo de red, por lo que la comunicación de estos dispositivos cobra una gran importancia. Ahora vamos a ver cómo participan las distintas tecnologías en estos dispositivos y cómo influyen en el uso de la tecnología J2ME. Para ello vamos a centrarnos en un dispositivo en especial: los teléfonos móviles. De aquí en adelante el estudio y la creación de la aplicación para este proyecto se realizarán para este dispositivo. Esto es así debido a la rápida evolución que han tenido los teléfonos móviles en el sector de las comunicaciones, lo que ha facilitado el desarrollo, por parte de algunas empresas, de herramientas que usaremos para crear las aplicaciones.

Uno de los primeros avances de la telefonía móvil en el sector de las comunicaciones se dio con la aparición de la tecnología WAP. WAP proviene de *Wireless Application Protocol* o Protocolo de Aplicación Inalámbrica. Es un protocolo con el que se ha tratado de dotar a los dispositivos móviles de un pequeño y limitado navegador web. WAP exige la presencia de una puerta de enlace encargado de actuar como intermediario entre Internet y el terminal. Esta puerta de enlace o *gateway* es la que se encarga de convertir las peticiones WAP a peticiones web habituales y viceversa.

Las páginas que se transfieren en una petición usando WAP no están escritas en HTML, si no que están escritas en WML, un subconjunto de éste. WAP ha sido un gran avance, pero no ha resultado ser la herramienta que se prometía. La navegación es muy engorrosa (la introducción de URLs largas por teclado es muy pesada, además de que cualquier error en su introducción requiere que se vuelva a escribir la dirección completa por el teclado del móvil). Además su coste es bastante elevado ya que el pago de uso de esta tecnología se realiza en base al tiempo de conexión a una velocidad, que no es digamos, muy buena.

Otra tecnología relacionada con los móviles es SMS. SMS son las siglas de *Short Message System* (Sistema de Mensajes Cortos). Actualmente este sistema permite hacer una comunicación de manera rápida y barata sin tener que establecer una comunicación con el receptor del mensaje. Con ayuda de J2ME, sin embargo, se puede realizar aplicaciones de chat o mensajería instantánea.

Los últimos avances de telefonía móvil llevan a las conocidas cómo generación 2 y 2´5 que hacen uso de las tecnologías GSM y GPRS respectivamente. GSM es una conexión telefónica que soporta una circulación de datos, mientras que GPRS es estrictamente una red de datos que mantiene una conexión abierta en la que el usuario paga por la cantidad de información intercambiada y no por el tiempo que permanezca conectado. La aparición de la tecnología GPRS no hace más que favorecer el uso de J2ME y es, además, uno de los pilares sobre los que se asienta J2ME, ya que podemos decir que es el vehículo sobre el que circularán las futuras aplicaciones J2ME.

Otras tecnologías que favorecen la comunicación son Bluetooth y las redes inalámbricas que dan conectividad a ordenadores, PDAs y teléfonos móviles. De hecho, una gran variedad de estos dispositivos disponen de soporte bluetooth. Esto nos facilita la creación de redes con un elevado ancho de banda en distancias pequeñas (hasta 100 metros).

Es de esperar que todas estas tecnologías favorezcan el uso de J2ME en el mercado de la telefonía móvil.

Requerimientos Funcionales

Los dispositivos deben proporcionar mecanismos mediante los cuales se puedan encontrar los *MIDlets* que se desean descargar. En algunos casos, se encuentra los *MIDlets* a través de un navegador WAP o a través de una aplicación residente escrita específicamente para identificar *MIDlets*. Otros mecanismos como Bluetooth, cable serie, etc, pueden ser soportados por el dispositivo.

El programa encargado de manejar la descarga y ciclo de vida de los *MIDlets* en el dispositivo se llama Gestor de Aplicaciones o AMS (*Application Management Software*).

Un dispositivo que posea la especificación MIDP debe ser capaz de:

- Localizar archivos JAD vinculados a un *MIDlet* en la red.
- Descargar el MIDlet y el archivo JAD al dispositivo desde un servidor usando el protocolo HTTP 1.1 u otro que posea su funcionalidad.
- Enviar el nombre de usuario y contraseña cuando se produzca una respuesta HTTP por parte del servidor 401 (*Unauthorized*) o 407 (*Proxy Authentication Required*).
- Instalar el *MIDlet* en el dispositivo.
- Ejecutar *MIDlets*.
- Permitir al usuario borrar *MIDlets* instalados.

4.1.2 Java Servlets

Como dice Barrios Juan y Moreno José[3], los Servlets son módulos escritos en Java que se utilizan en un servidor, que puede ser o no ser servidor web, para extender sus capacidades de respuesta a los clientes al utilizar las potencialidades de Java. Los Servlets son para los servidores lo que los applets para los navegadores, aunque los servlets no tienen una interfaz gráfica.

Los servelts pueden ser incluidos en servidores que soporten la API de Servlet. La API no realiza suposiciones sobre el entorno que se utiliza, como tipo de servidor o plataforma, ni del protocolo a utilizar, aunque existe una API especial para HTTP.

[3] BARRIOS Juan, MORENO José. Java Servlets. [En línea]. Chile. Fecha de publicación: 30 de Noviembre del 2001. Disponible en: http://www.dcc.uchile.cl/~jbarrios/servlets/index.html. . [Consulta: 26 julio 2007] .

Los Servlets son un reemplazo efectivo para los CGI en los servidores que los soporten ya que proporcionan una forma de generar documentos dinámicos utilizando las ventajas de la programación en Java como conexión a alguna base de datos, manejo de peticiones concurrentes, programación distribuida, etc. Por ejemplo, un servlet podría ser responsable de procesar los datos desde un formulario en HTML como registrar la transacción, actualizar una base de datos, contactar algún sistema remoto y retornar un documento dinámico o redirigir a otro servlet u alguna otra cosa.

¿Cómo es un Servlet?[4]

Un pequeño servlet de ejemplo es el siguiente:

```
public class SimpleServlet extends HttpServlet {

   // Maneja el método GET de HTPP para
   // construir una sencilla página Web.

   public void doGet (HttpServletRequest request, HttpServletResponse
response)
            throws ServletException, IOException {
   PrintWriter out;
   String title = "Simple Servlet Output";

   // primero selecciona el tipo de contenidos y otros campos de cabecera de la
respuesta
   response.setContentType("text/html");
   // Luego escribe los datos de la respuesta
   out = response.getWriter();
   out.println("<HTML><HEAD><TITLE>");
   out.println(title);
   out.println("</TITLE></HEAD><BODY>");
   out.println("<H1>" + title + "</H1>");
   out.println("<P>This is output from SimpleServlet.");
   out.println("</BODY></HTML>");
   out.close();
   }
}
```

[4] Ibid.

En negrita se ha resaltado los detalles importantes que son característicos de un Servlet. Este Servlet puede ser puesto en un servidor web ya que utiliza el protocolo HTTP para comunicarse.

Primero es necesario señalar que el servlet será del tipo HTTP por lo que se extiende de la clase **HttpServlet**. Al extender de esta clase es necesario definir el método **doGet** para responder la petición. Este método recibe los parámetros dados por el cliente a través de la clase **HttpServletRequest** y encapsula la respuesta que se le dará al cliente a través de la clase **HttpServletResponse**. El servlet puede retornar al cliente cualquier tipo de información, desde texto plano hasta un ejecutable, por lo que es necesario señalar inicialmente qué tipo de respuesta se dará a través del método **setContentType**. Luego se obtiene el objeto para poder escribir texto al cliente a través del método **getWriter** con el cual se puede retornar una página web llamado sucesivamente el método **println** hasta terminar con **close**.

Propiedades

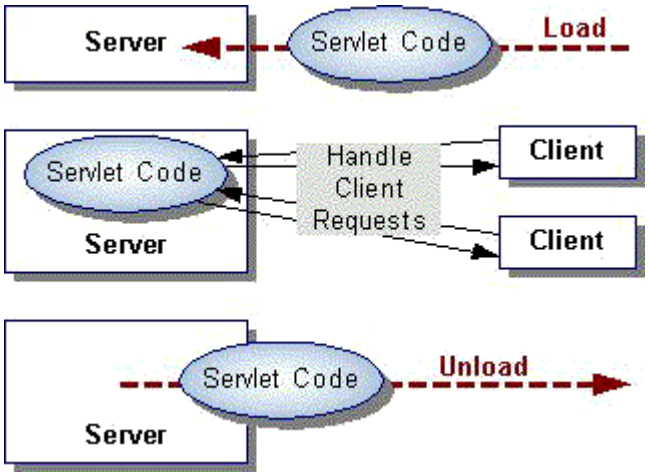
- **Manejo de Sesiones:** Se puede hacer seguimiento de usuarios a través de distintos servlets a través de la creación de sesiones.
- **Utilización de Cookies:** Las *cookies* son pequeños datos en texto plano que pueden ser guardados en el cliente. La API de servlets permite un manejo fácil y limpio de ellas.
- **Multi-thread:** Los servlets soportan el acceso concurrente de los clientes, aunque hay que tener especial cuidado con las variables compartidas a menos que se utilize la interfaz **SingleThreadModel**.
- **Programación en Java:** Se obtienen las características de multiplataforma o acceso a APIs como JDBC, RMI, etc.

El Ciclo de Vida[5]

Cada servlet tiene el mismo ciclo de vida:

- Un servidor carga e inicializa el servlet.
- El servlet maneja cero o más peticiones de cliente.
- El servidor elimina el servlet.

[5] Ibíd.



Fuente: BARRIOS Juan, MORENO José. Java Servlets. [En línea]. Chile. Fecha de publicación: 30 de Noviembre del 2001. Disponible en: http://www.dcc.uchile.cl/~jbarrios/servlets/vida.html. [Consulta: 26 julio 2007].

Inicializar un Servlet

Cuando un servidor carga un servlet, ejecuta el método **init** del servlet. La inicialización se completa antes de manejar peticiones de clientes y antes de que el servlet sea destruido.

Aunque muchos servlets se ejecutan en servidores multi-thread, los servlets no tienen problemas de concurrencia durante su inicialización. El servidor llama sólo una vez al método **init** al crear la instancia del servlet, y no lo llamará de nuevo a menos que vuelva a recargar el servlet. El servidor no puede recargar un servlet sin primero haber destruido el servlet llamando al método destroy.

Interactuar con Clientes

Después de la inicialización, el servlet puede manejar peticiones de clientes. Estas respuestas son manejadas por **la misma instancia** del servlet por lo que hay que tener cuidado con acceso a variables compartidas por posibles problemas de sincronización entre requerimientos concurrentes.

Destruir un Servlet

Los servlets se ejecutan hasta que el servidor los destruye, por cierre el servidor o bien a petición del administrador del sistema. Cuando un servidor destruye un servlet, ejecuta el método destroy del propio servlet. Este método sólo se ejecuta una vez y puede ser llamado cuando aún queden respuestas en proceso por lo

que hay que tener la atención de esperarlas. El servidor no ejecutará de nuevo el servlet, hasta haberlo cargado e inicializado de nuevo.

4.1.3 HyperText Markup Language (HTML)

Como dice Gracia[6], es un lenguaje de programación que se utiliza para la creación de páginas en la WWW (*World Wide Web*). Está compuesto por varios comandos, que son interpretados por el programa que se usa para navegar por el WWW.

HTML no permite definir de forma precisa la apariencia de una página. La presentación de la página es muy dependiente del navegador utilizado. HTML se limita a describir la estructura y el contenido de un documento y no el formato de la página y su apariencia.

El navegador es un programa mediante el cual se accede a Internet; se comunica con un servidor y comprende el lenguaje de todas las herramientas que manejan la información de Web. Este requiere de un servidor, el cual proporciona al navegador los documentos y medios que éste solicita. Utiliza un protocolo HTTP para atender las solicitudes de archivos por parte de un navegador.

El Protocolo de transferencia de hipertexto (HTTP), es el protocolo que los servidores de *World Wide Web* utilizan para mandar documentos HTML a través de Internet.

El Localizador Uniforme de Recursos (URL) es la dirección que localiza una información dentro de Internet.

Una página WWW incluye tres tipos de información: texto, gráficos e hipertexto. Un hipertexto es el texto que aparece resaltado en la página y que el usuario activa para cargar otra página WWW. La diferencia entre un documento hipertexto y un documento normal consiste en que el hipertexto posee enlaces con otros documentos que tienen la misma relación; de manera que el usuario puede pasar de una página a otra y volver a la página principal mediante cada enlace.

Está compuesto por varios comandos, interpretados por el programa que se utilice para navegar por el www. Este, es el encargado de ejecutar todas las órdenes contenidas en el código HTML, de manera que para visualizar correctamente el

[6] GRACIA, Joaquin. HTML fácil. [en línea]. Madrid. Fecha de publicación: Agosto 2006. HTML / Contenido del manual / conceptos básicos. Disponible en: http://www.webestilo.com/html/cap1a.phtml

código HTML, se debe poseer un software adecuado, el cual está limitado por la empresa y la versión que se tenga instalada en la máquina donde se ejecuta el código.

> Una de las claves del éxito de WWW, aparte de lo atractivo de su presentación es sin duda, su organización y coherencia. Todos los documentos WWW comparten un mismo aspecto y una única interfaz, lo que facilita enormemente su manejo por parte de cualquier persona. Esto es posible porque el lenguaje HTML, en que están escritos los documentos, no solo permite establecer hiperenlaces entre diferentes documentos, sino que es un "lenguaje de descripción de página" independiente de la plataforma en que se utilice. Es decir un documento HTML contiene toda la información necesaria sobre su aspecto y su interacción con el usuario, y es luego el browser que utilicemos el responsable de asegurar que el documento tenga un aspecto coherente, independientemente del tipo de estación de trabajo desde donde estemos efectuando la consulta.[7]

4.1.4 Java Server Pages (JSP)

Según Merelos[8], es una tecnología para generar páginas Web de forma dinámica en un servidor, basado en *scripts* que usan el lenguaje Java.
Los *scripts* JSPs y *servlets* se ejecutan en una máquina virtual java, lo cual permite que se puedan usar en cualquier tipo de ordenador, siempre que exista una máquina virtual java para él.
La principal ventaja de JSP frente a otros lenguajes de programación, es que puede integrarse con clases Java (.class) y permite separar en niveles las aplicaciones Web, almacenar en clases java las partes que consumen más recursos y las que requieren más seguridad.

---

[7] DOMÍNGUEZ, Octavio. Conceptos Básicos. [en línea]. Las Palmas. Fecha de publicación: Marzo 2002. Guías / Cursos / Tutorial de HTML. Disponible en: http://gias720.dis.ulpgc.es/Gias/Cursos/Tutorial_html/indice.htm.

[8] MERELOS, Juan. Programando con JSps. [en linea]. Granada. Fecha de publicación: Octubre 2004. Tutoriales diversos / Tutorial de java server pages, JSPs. Disponible en: http://geneura.ugr.es/~jmerelo/JSP/

4.1.5 Servidor Apache.

Según Hospedajes y dominios[9], es un servidor de red para el protocolo HTTP, de código fuente abierto, elegido para funcionar como un proceso *standalone*, sin necesidad de solicitar el apoyo de otras aplicaciones o directamente del usuario. Para poder hacer esto, Apache, una vez que se haya iniciado, crea unos subprocesos para poder gestionar las solicitudes; estos procesos, sin embargo, no podrán nunca interferir con el proceso mayor.

El servidor Apache está estructurado en módulos. La configuración de cada módulo se hace mediante la configuración de las directivas que están contenidas dentro del módulo. Los módulos del Apache se pueden clasificar en módulos base, los cuales cumplen con las funciones básicas del servidor Apache, los módulos multiproceso responsables de la unión con los puertos de la máquina, aceptando peticiones y enviando a los hijos a atenderlas y los módulos adicionales que añaden funcionalidades al servidor.

Una de las principales características que presenta Apache es que se ejecuta en plataformas virtuales muy utilizadas. Apache no solo funciona en la mayoría las versiones de Unix sino que, además, se puede implementar en Windows 2000/NT/9x y en muchos otros sistemas operativos de escritorio y de tipo servidor. "Apache presenta muchas otras características, entre ellas un elaborado índice de directorios, un directorio de alias, negociación de contenidos, informe de errores HTTP configurable, ejecución SetUID de programas CGI, gestión de recursos para procesos hijos, integración de imágenes del lado del servidor, reesc ritura de las URL, comprobación de la ortografía de las URL y manuales online"[10].

- **Otras características importantes del servidor apache**

  -Soporte del último protocolo HTTP 1.1.
  -Soporte para CGI (Common Gateway Interface).
  -Soporte de ***host*** virtuales.
  -Soporte de autentificación http.
  -Perl integrado.
  -Soporte de ***scripts*** PHP.
  -Soporte de ***servlets*** de Java.

---

[9] HOSPEDAJE Y DOMINIOS S.L. Apache: el comienzo. [en línea]. Murcia. Fecha de publicación: Julio 2005. Apache / Manual de Apache / Lista de características de Apache. Disponible en: http://www.hospedajeydominios.com/mambo/documentacion-manual_apache-pagina-45.html

[10] Ibíd.

-Servidor proxy integrado.
-Soporte de ***Secured Sockect Layer*** (SSL).

4.1.6 Apache Tomcat

Es un contenedor estable desarrollado bajo el proyecto Jakarta (Apache Software Foundation); se le considera como un servidor de aplicaciones y a la vez soporta los requerimientos demandados por servlets y especificaciones JSP.
Tomcat no funciona con cualquier servidor Web con soporte para servlets y JSPs. Tomcat incluye el compilador Jasper, que compila JSPs convirtiéndolas en servlets. El motor de servlets del Tomcat a menudo se presenta en combinación con el servidor Web Apache[11] .
Algunas de las características de Apache Tomcat se mencionan a continuación:

- Optimización en la ejecución y reducción de la colección de basura.
- Motor JSP rediseñado con Jasper
- Capa envolvente nativa para Windows y Unix para la integración de las plataformas
- Mayor escalabilidad y realce en las ejecuciones.
- Monitoreo completo del servidor usando JMX y el administrador de aplicación Web.
- Mejora en el manejo de librerías incluyendo los plugins de las etiquetas

4.1.7 Microsoft SQL Server 2000

Microsoft SQL Sever es un sistema de gestión de bases de datos relacional (SGBD), basado en SQL*, capaz de poner a disposición una gran cantidad de información de manera simultánea.

Se puede destacar entre sus características:

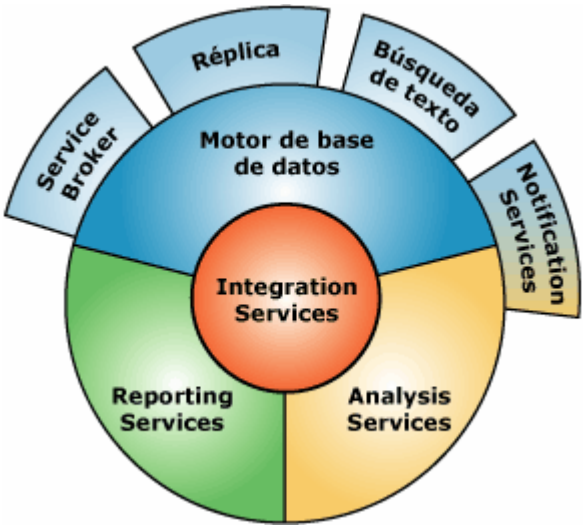
---

[11] WALES, Jimmy. Servidor Apache Tomcat.[en línea]. Fecha de publicación: Octubre 2005. Artículo / Servidor Apache Tomcat. Disponible en: http://es.wikipedia.org/wiki/Jakarta_Tomcat

* El Lenguaje de Consulta Estructurado (Structured Query Language) es un lenguaje declarativo de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones sobre las mismas. Una de sus características es el manejo del álgebra y el cálculo relacional permitiendo lanzar consultas con el fin de recuperar información de interés de una base de datos, de una forma sencilla. Es un lenguaje de cuarta generación (4GL). WINKIPEDIA.ORG.SQL.[en línea].Disponible en: http://es.wikipedia.org/wiki/SQL.

- Soporte de transacciones.
- Escalabilidad, estabilidad y seguridad.
- Soporta procedimientos almacenados.
- Incluye también un potente entorno gráfico de administración, que permite el uso de comandos DDL y DML gráficamente.
- Permite trabajar en modo cliente-servidor donde la información y datos se alojan en el servidor y las terminales o clientes de la red sólo acceden a la información.
- Además permite administrar información de otros servidores de datos

Componentes de M-SQLServer



Fuente: MICROSOFT Corporation. Información general de SQL Server. [en línea]. SQL Server/Documentación SQL Server 2005/ libro en pantalla de SQL Server 2005/ información general SQL Server. Disponible en: http://msdn2.microsoft.com/es-co/library/ms166352.aspx

Microsoft SQL Server 2000, según Microsoft[12] es una plataforma de base de datos que se utiliza en el procesamiento de transacciones en línea (OLTP) a gran escala, el almacenamiento de datos y las aplicaciones de comercio electrónico; es también una plataforma de Business Intelligence para soluciones de integración, análisis y creación de informes de datos.

[12] MICROSOFT Corporation. Información general de SQL Server. [en línea]. SQL Server/Documentación SQL Server 2005/ libro en pantalla de SQL Server 2005/ información general SQL Server. Disponible en: http://msdn2.microsoft.com/es-co/library/ms166352.aspx

Componentes de SQL Server 2000:

- Motor de base de datos
- Reporting Services
- Analysis Services
- Notification Services
- Integration Services
- Búsqueda de texto
- Réplica
- Service Broker

SQL Server 2000 introduce "estudios" que le ayudarán en las tareas de programación y administración: SQL Server Management Studio y Business Intelligence Development Studio. En Management Studio, se desarrolla y administra SQL Server Database Engine (Motor de base de datos de SQL Server) y soluciones de notificación, se administran las soluciones de Analysis Services implementadas, se administran y ejecutan los paquetes de Integration Services, y se administran los servidores de informes y los informes y modelos de informe de Reporting Services. En BI Development Studio, se desarrollan soluciones de Business Intelligence mediante proyectos de Analysis Services para desarrollar cubos, dimensiones y estructuras de minería; se crean proyectos de Reporting Services para crear informes; se crea el modelo de informes para definir modelos para los informes y se desarrollan proyectos de Integration Services para crear paquetes.

Los dos estudios están muy estrechamente relacionados con Microsoft Visual Studio y Microsoft Office System. Para obtener más información, vea Introducción a SQL Server Management Studio y Presentación de Business Intelligence Development Studio.
En los estudios, SQL Server 2000 proporciona las herramientas gráficas que necesita para diseñar, desarrollar, implementar y administrar bases de datos relacionales, objetos analíticos, paquetes de transformación de datos, topologías de réplica, informes y servidores de informes, y servidores de notificaciones. Además, SQL Server 2000 incluye utilidades del símbolo del sistema para realizar tareas administrativas desde el símbolo del sistema. Para obtener acceso rápidamente a temas especializados sobre las herramientas y utilidades, vaya a Mapa de documentación de las herramientas y utilidades.

SQL Server 2000 proporciona varias formas de enviar comentarios sobre el producto y la documentación, además de enviar automáticamente informes de errores y datos sobre el uso de las características a Microsoft. Para saber cómo

puede enviar comentarios y sugerencias, vaya a Comentarios sobre SQL Server 2000.

4.1.8 Java 2 Platform, Standard Edition (J2SE)

Esta edición de Java es la que en cierta forma recoge la iniciativa original del lenguaje Java. Tiene las siguientes características:

- Inspirado inicialmente en C++, pero con componentes de alto nivel, como soporte nativo de strings y recolector de basura.
- Código independiente de la plataforma, precompilado a bytecodes intermedio y ejecutado en el cliente por una JVM (Java Virtual Machine).
- Modelo de seguridad tipo *sandbox* proporcionado por la JVM.
- Abstracción del sistema operativo subyacente mediante un juego completo de APIs de programación.
  Esta versión de Java contiene el conjunto básico de herramientas usadas para desarrollar Java Applets, así como las APIs orientadas a la programación de aplicaciones de usuario final: Interfaz gráfica de usuario, multimedia, redes de comunicación, etc[13].

4.1.9 JDBC

Java es un lenguaje de programación orientado a objetos desarrollado por Sun Microsystems a principios de los años 1990. Las aplicaciones Java están típicamente compiladas en un bytecode, aunque la compilación en código máquina nativo también es posible. En el tiempo de ejecución, el bytecode es normalmente interpretado o compilado a código nativo para la ejecución, aunque la ejecución directa por hardware del bytecode por un procesador Java también es posible.

El lenguaje en sí mismo toma mucha de su sintaxis de C y C++, pero tiene un modelo de objetos más simple y elimina herramientas de bajo nivel como punteros.

[13] GÁLVEZ R Sergio, ORTEGA D Lucas. Java a Tope: J2ME (Java 2 Microedition). [En línea]. Fecha de publicación: 2003. Universidad de Málaga. Disponible en: http://www.lcc.uma.es/~galvez/J2ME.html.

Java posee un paquete con las clases necesarias para realizar las interfaces requeridas entre la aplicación y un origen de datos, comúnmente una base de datos relacional. El paquete es el JAVA.SQL el cual está conformado por:

JDBC es el acrónimo de Java Database Connectivity, un API que permite la ejecución de operaciones sobre bases de datos desde el lenguaje de programación Java independientemente del sistema de operación donde se ejecute o de la base de datos a la cual se accede utilizando el dialecto SQL del modelo de base de datos que se utilice.

El API JDBC se presenta como una colección de interfaces Java y métodos de gestión de manejadores de conexión hacia cada modelo específico de base de datos. Un manejador de conexiones hacia un modelo de base de datos en particular es un conjunto de clases que implementan las interfaces Java y que utilizan los métodos de registro para declarar los tipos de localizadores a base de datos (URL) que pueden manejar. Para utilizar una base de datos particular, el usuario ejecuta su programa junto con la librería de conexión apropiada al modelo de su base de datos, y accede a ella estableciendo una conexión, para ello provee en localizador a la base de datos y los parámetros de conexión específicos. A partir de allí puede realizar con cualquier tipo de tareas con la base de datos a las que tenga permiso: consultas, actualizaciones, creado modificado y borrado de tablas, ejecución de procedimientos almacenados en la base de datos, etc.

JDBC es apropiada para el desarrollo de este proyecto, ya que cuenta con una alta integración entre SQL y java, es decir se puede usar una variable creada en java en una sentencia SQL para recibir y enviar datos.

El JDBC ofrece una mejor compatibilidad ya que el ODBC[*] usa una interfaz en C, una traducción literal de la API ODBC en C, no es deseable. Por ejemplo Java no tiene punteros y la ODBC hace gran uso de ellos.
ODBC es muy difícil de aprender, hace una mezcla de acciones simples y avanzadas. Una API como JDBC es necesaria para poder desarrollar una solución puramente de Java.

Los tres componentes JDBC:

[*] ODBC son las siglas de Open DataBase Connectivity, que es un estándar de acceso a Bases de Datos desarrollado por Microsoft Corporation, el objetivo de ODBC es hacer posible el acceder a cualquier dato de cualquier aplicación, sin importar qué Sistema Gestor de Bases de Datos (DBMS por sus siglas en inglés) almacene los datos, ODBC logra esto al insertar una capa intermedia llamada manejador de Bases de Datos, entre la aplicación y el DBMS, el propósito de esta capa es traducir las consultas de datos de la aplicación en comandos que el DBMS entienda. WINKIPEDIA.ORG.ODBC.[en línea].Disponible en: http://es.wikipedia.org/wiki/ODBC.

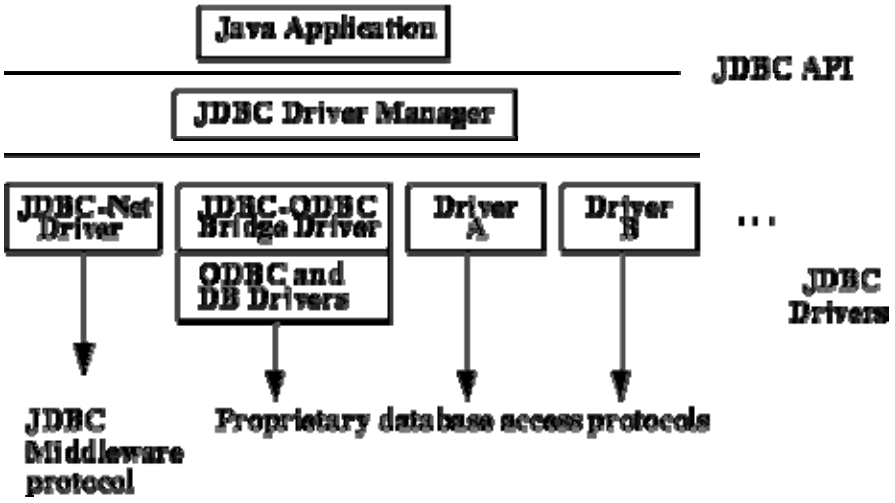
JDBC driver manager, JDBC driver test suite y Puente JDBC-ODBC.

El JDBC driver manager es el esqueleto de la arquitectura de JDBC. Realmente es bastante pequeño y simple; su función primaria es conectar las aplicaciones de Java al manejador de JDBC.

La JDBC driver test suite proporciona un poco de confianza en que drivers de JDBC ejecutarán su programa. Pueden designarse sólo drivers que pasan la JDBC driver test suite.

El puente de JDBC-ODBC les permite a los drivers de ODBC ser usado como drivers de JDBC. Y a largo plazo proporcionará una manera de acceder alguno del DBMSs menos popular si no se crean los drivers de JDBC para ellos.

Componentes de JDBC



Fuente: FERNÁNDEZ Felipe, MUÑOZ Yerko.JDBC. [en linea]. Disponible en: http://www.dcc.uchile.cl/~lmateu/CC60H/Trabajos/jfernand/

4.1.10 PROTOCOLO OPC

4.1.10.1 ORIGENES

Según la empresa Autómatas Industriales[14], el protocolo OPC* es un estándar de comunicación en el campo del control y supervisión de procesos. Este estándar permite que diferentes fuentes de datos envíen datos a un mismo servidor OPC, al que a su vez podrán conectarse diferentes programas compatibles con dicho estándar. De este modo se elimina la necesidad de que todos los programas cuenten con drivers para dialogar con múltiples fuentes de datos, basta que tengan un driver OPC.

OPC se basa en la tecnología OLE/COM*; Esta es la tecnología que permite que componentes de software (escritos en C, C++, java, .NET) sean utilizados por una aplicación. De esta forma se desarrollarán componentes en C, C++ y java que encapsulen los detalles de acceder a los datos de un dispositivo, de manera que quienes desarrollen aplicaciones empresariales puedan escribir código en Java o C que recoja y utilice datos de un PLC o tarjeta de adquisición de datos.

El diseño de las interfases OPC soporta arquitecturas distribuidas en red. El acceso a servidores OPC remotos se hace empleando la tecnología DCO M (Distributed COM) de Microsoft.

4.1.10.2 AQUITECTURA

Un servidor OPC se compone de varios objetos que se ajustan a la norma COM:

- El objeto servidor: contiene información sobre la configuración del servidor OPC y sirve de contenedor para los objetos tipo grupo.

- El objeto grupo: sirve para organizar los datos que leen y escriben los clientes (ej.: valores en una pantalla MMI o en un informe de producción). Se pueden establecer conexiones por excepción entre los clientes y los elementos de un grupo. Un grupo puede ser público, es decir, compartido por varios clientes OPC.

---

[14] AUTOMATAS INDUSTRIALES. OPC OLE for Process Control. [En línea]. Fecha de publicación: 2006. Disponible en: http://www.automatas.org/redes/opc.htm. Consulta: 31 Enero 2008].

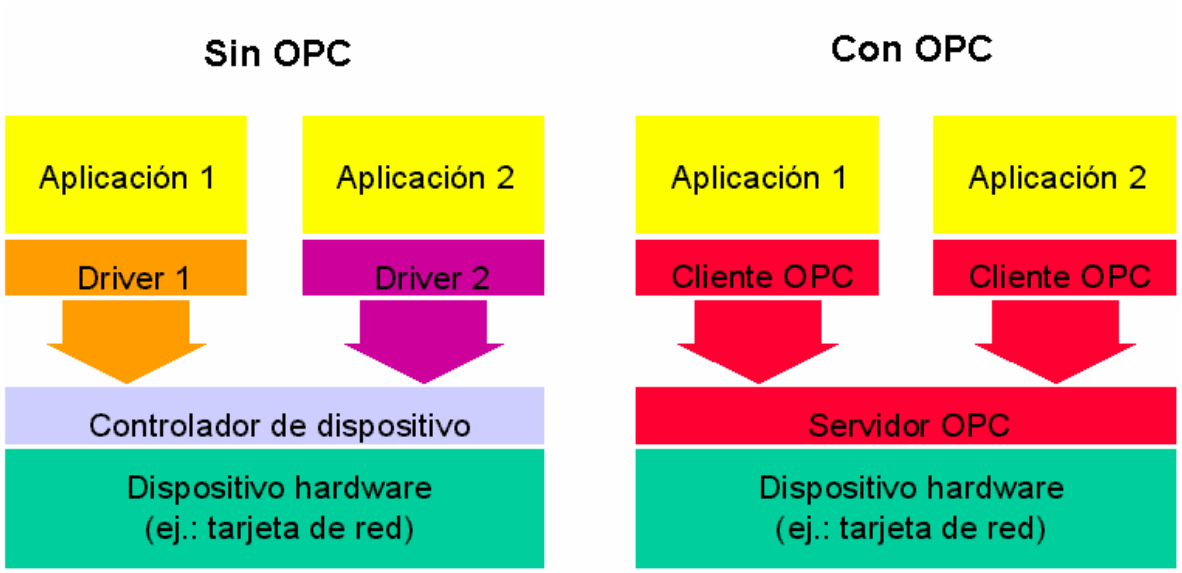
* OLE for Process Control

* Object Linking and Embedding / Component Object Model de Microsoft

- El objeto item: representa conexiones a fuentes de datos en el servidor (no son las fuentes de datos en sí). Tiene asociados los atributos Value, Quality y Time Stamp. Los accesos a los items OPC se hacen a través de los grupos OPC y los clientes pueden definir el ritmo al cual el servidor les informa sobre cambios en los datos.

El acceso a los objetos COM se hace a través de interfases, que son lo único que ven los clientes OPC. Los objetos descritos son representaciones lógicas que no tienen porqué coincidir con la implementación que se haga del servidor OPC.

Diferencia entre un sistema con OPC y sin el como se ve en la figura



Fuente: AUTOMATAS INDUSTRIALES. OPC OLE for Process Control. [En línea]. Fecha de publicación: 2006. Disponible en: http://www.automatas.org/redes/opc.htm. Consulta: 31 Enero 2008].

4.2 ANTECENDENTES

Una investigación realizada por Sergio Alcántara Segura con el nombre de:

4.2.1 Sistema de Control Domótico

> El objetivo del proyecto es diseñar e implementar un sistema que permita monitorizar los dispositivos de una vivienda de forma remota desde un Terminal móvil; ya sea una PDA, PocketPC, teléfono móvil o cualquier dispositivo con conexión a Internet. Permitiendo al usuario estar informado en todo momento de los eventos que sucedan en el hogar y poder actuar en consecuencia[15].

El trabajo anterior, se relaciona con nuestra propuesta por cuanto involucra la tecnología móvil para el caso de un sistema de control Domótico.

Otro proyecto realizado por la empresa de Telecomunicaciones COMCEL en el año 2004, con el nombre de:

4.2.2 Telemetría

> Consiste en un servicio en el que se puede medir y controlar en tiempo real e inalámbricamente su operación desde cualquier lugar y sin necesidad de desplazar personal para ello, reduciendo considerablemente los costos y el tiempo en la labor. Se podrá transmitir información de diferentes mediciones industriales como paso de fluidos, energía, temperaturas, presión y todos los valores que sean críticos para la eficiencia de su operación. Además, controlar remotamente procesos como el cierre de compuertas o el flujo de elementos desde su computador o teléfono celular[16].

Así mismo la Compañía de telefonía Celular COMCEL desarrolló una aplicación donde involucra todo el control de procesos como tiempo, temperatura, tomas desde un sensor que transmite a un PC para interpretar y analizar las muestras, estas se transmite a través de una antena. Esto se relaciona con nuestro proyecto ya que este aplicativo al igual que nuestra propuesta maneja este tipo de

[15]ALCÁNTARA Segura Sergio. Proyecto de Sistema de Control Domótico. Universidad Lasal le Bonanova de Barcelona. 2000. [Pagina Web en línea]. Disponible en: http://www.imarketing.es/pdf/sistema_domotico.pdf. [consulta: 19 marzo 2006].

[16]NUMPAQUE Pablo. Aplicación de Telemetría. DATUMCOMCEL. 2004. [Pagina Web en línea]. Disponible en: http://www.datumcomcel.com/. [Consulta: 19 marzo 2006].

variables. La diferencia dentro de nuestro proyecto consiste en el manejo de la tecnología J2ME y la interfaz en un dispositivo móvil.

Proyectos que han utilizado la tecnología WAP:

4.2.3 Diseño e implementación de una pasarela de signatura digital con tecnología WAP

Este proyecto habla de una aplicación que permita enviar mensajes de correo a través de teléfonos móviles[17]**.**

Un proyecto fin de carrera realizado por Luís Miguel Rodríguez Tudanca en el año 2002, con el nombre de:

4.2.4 Diseño e Implementación Software de una pasarela WAP para la navegación Web mediante teléfonos celulares

Para Luís Miguel Rodríguez: "E*ste proyecto se aborda el desarrollo software de una pasarela WEB/WAP. A través de dicho software se posibilitará la navegación Internet WEB genérica mediante teléfonos celulares WAP. La pasarela WAP/WEB permite al usuario visitar una página Web y visualizarla en formato WML aunque su formato original sea HTML"* [18]*.*

Otra aplicación realizada por la empresa SCI-TECH ScienceTech S.A., con el nombre de:

4.2.5 Soluciones de campo para automatización de fuerzas de ventas

Según ScienceTech: *"SellerWap le otorga una solución integral a la toma de pedidos en campo, mediante la aplicación de tecnología WAP, sus vendedores podrán acceder a su base de datos para actualizarla o consultar información mediante un celular WAP"*[19]*.*
Una aplicación WAP denominada:

---

[17] RIFÁ Pous Helena. Diseño e implementación de una pasarela de signatura digital con tecnología WAP. Universidad Politécnica de Barcelona. 2001. [Pagina Web en línea]. Disponible en: www.coit.es/pub/ficheros/philips_f68cd55b.pdf. [Consulta: 24 abril 2006].

[18] Ibíd.
[19] Ibíd.

4.2.6 Consulta de información online WAP de documentos de Identidad Civil

Para la empresa San Diego SoftWorks la solución es: ***"****Un complemento de la ya adquirida herramienta de consulta vehicular WAP que cuenta el Ministerio hoy en día, esta nueva solución se basa mediante el ingreso del número de documento de identidad por parte del policía, accediendo directamente a la base de datos del Ministerio del Interior de Uruguay****"***[20]*.*

Asimismo, un proyecto realizado por Joaquín Pérez y Sonia López llamado:

4.2.7 Prototipo de Comercio Electrónico Orientado A Dispositivos Móviles en México

El cual consiste en: "*desarrollar un prototipo de un sitio de comercio electrónico que pueda ser accedido mediante un teléfono celular.*"[21].

[20] Ibíd.

[21] LÓPEZ R. Sonia, PÉREZ O. Joaquín. Prototipo de Comercio Electrónico Orientado A Dispositivos Móviles en México. Centro Nacional de Investigación y Desarrollo Tecnológico de México. 2005. [Pagina Web en línea]. Disponible en: www.cenidet.edu.mx/subaca/web-dcc/web-sd/Pazos/CIICC05-ICI05/Art035.pdf. [Consulta: 24 abril 2006].

5. METODOLOGIA

Se desarrolló una aplicación para móviles que permite controlar procesos industriales. Para empezar debemos montar un servidor el cual dejará conectar los dispositivos móviles con él, en este servidor se monto una base de datos en la cual están todos los datos relacionados con los procesos que se manejan.

En el servidor se realizo todos los procedimientos que tienen que ver con el funcionamiento de la base de datos, para que el rendimiento en los celulares sea óptimo; en los dispositivos móviles solo se manejará la interfaz de usuario necesaria para manipular la base de datos por medio de una conexión inalámbrica; desde allí se pudo acceder a la base de datos, en la cual se logró consultar, actualizar e insertar registros.

5.1 TIPO DE TRABAJO

El sistema para la gestión de procesos industriales, GPIAM, es un trabajo de desarrollo de software, basado en un sistema de administración, el cual permite por medio del mismo llevar un control confiable de los estados de variables físicas.

5.2 PROCEDIMIENTO

En la elaboración del sistema de información motivo de este proyecto, se realizo las siguientes fases de desarrollo:

5.2.1 Requisitos de Documentación

Fase dedicada a la consulta de las fuentes de información y al levantamiento de información. Visitas constantes a MABE empresa donde se realizo las pruebas piloto del proyecto, se realizaron varios diálogos con el usuario directo del sistema jefe de Mantenimiento.

5.2.2 Análisis y Diseño

Elaboración del Análisis y Diseño del sistema utilizando el Lenguaje de Modelado Unificado UML (Diagrama de Clases)

Análisis requerido para la realizar la estructura de la base de datos. (Anexo A, Anexo B, Anexo C, Anexo D).

5.2.3 Servidor con base de datos

Fase dedicada a la búsqueda de un gestor de base de datos adecuado, para el manejo de la información de las variables físicas, se utilizo la herramienta Microsoft SQL Server para realizar el diseño e implementación de base de datos.

5.2.4 Aplicación móvil para manipular las variables

Fase dedicada al diseño de las interfaces para la aplicación móvil del usuario, este diseño fue realizado a través de la herramienta Netbeans con la tecnología J2ME.

5.2.5 Interfaz J2ME para la conectividad entre los Móviles y la BD

Fase dedicada a la implementación de la aplicación móvil con el diseño de interfaces previamente realizado, también se enfoca en realizar la conexión con la base de datos de la aplicación móvil al gestor de base de datos SQL Server.

5.2.6 Aplicación de monitoreo y control para las variables

Fase dedicada al diseño e implementación de un cliente que permita extraer información y controlar las variables físicas de los dispositivos de control programable (PLC), este cliente se diseño e implemento con ayuda de java y la herramienta Eclipse.

5.2.7 Pruebas para análisis de riesgos del sistema

En esta fase se realizan las pruebas necesarias para poner a marcha el sistema de información.
Se insertan datos imaginarios y reales de variables físicas, por medio de un emulador de un servidor industrial llamado MATRICON.

5.2.8 Puesta a punto del sistema

Fase dedicada a realizar correcciones a los problemas encontrados en la aplicación en la fase de pruebas.

5.2.9 Implementación del sistema

Fase dedicada a la implantación del sistema en la empresa MABE Colombia.

6. DESCRIPCIÓN DEL DESARROLLO

Este proceso se inicio con el desarrollo de una aplicación basada en WAP que permitiera interactuar con el servidor, donde se almacenan los datos de las variables físicas, pero en el transcurso del desarrollo de este aplicativo se tuvieron inconvenientes en la inserción de registros a la base de datos, este problema se presento debido la incompatibilidad de WML con el apache Tomcat en el manejo de las variables globales; al realizar una captura de estas variables no se obtenían los datos digitados por el usuarios sino que se almacenaban el nombre de las variables enviadas ($variable).

Debido a esto se decidió utilizar la herramienta de desarrollo J2ME ya que este posee características adecuadas que cuenta con librerías más especializadas para un mejor manejo de la comunicación con la base de datos, es más confiable y posee una mejor documentación.

Al finalizar la aplicación móvil desarrollada en J2ME, se realizaron varias visitas a la empresa MABE Colombia para obtener información de cómo se controlaban el proceso de mezcla R22 + poliol, a través de esto se pudo conocer una tecnología industrial para controlar y monitorear las variables físicas de un proceso, esto a través de un protocolo de comunicaciones creado por la OPC Fundation que permite obtener datos en tiempo real sin tener necesidad de someterse a un controlador especifico en cuanto al hardware se refiere (PLC).

Se desarrollo cliente Java basado en las especificaciones de la OPC Fundation que permite obtener datos en tiempo real de los dispositivos de control y almacenarlos en una base de datos para poder a su vez realizar labores como el bloqueo de mezcla, monitorear el estado de la mezcla, peso de los tanques de reactivos, niveles de los tanques, temperaturas y fallas del proceso.

Al realizar completamente el cliente, se integro con el aplicativo móvil y la base de datos obteniendo resultados satisfactorios ya que se pudo implementar en los procesos de mezcla de la empresa MABE Colombia.

7. RESULTADOS

Se desarrolló una aplicación que realiza labores de control de procesos, a través de una interfaz grafica desarrollada para dispositivos móviles, este aplicativo cuenta con 3 fases de funcionamiento que consisten en:

- **Aplicativo móvil:** este aplicativo permite realizar consultas y acciones sobre dispositivos de control como PLC que estos a su vez controlan el funcionamiento del proceso.

- **Cliente OPC:** este aplicativo permite interpretar los datos obtenidos del servidor OPC el cual está directamente conectado con los dispositivos que controlan y monitorean el proceso (PLC).

- **Base de Datos:** Esta permite obtener los datos de las variables físicas capturadas por el cliente OPC.

En la carpeta GPIAMOPC se encuentra el desarrollo de cliente OPC, en la carpeta GPIAM está depositada la Aplicativo Móvil y los Servlets correspondientes a esta aplicación se encuentran en la carpeta GPI

En el **anexo F** se encontrara los resultados de la aplicación móvil interactuando con los datos del servidor.

8. CONCLUSIONES

- El proyecto gestión de procesos industriales a través de móviles permite tener control de variables físicas como temperaturas, niveles, estados de proceso y cualquier otra variable que se quiera administrar en la automatización de proceso.

- GPIAM se ha desarrollado con herramientas como JAVA, J2ME las cuales permiten portabilidad, seguridad, robustez y confiabilidad.

- La necesidad de estar monitoreando los procesos, y la poca movilidad que se tiene, hace que esta propuesta sea viable al garantizar el acceso a los procesos industriales en cualquier lugar y tiempo.

- Al desarrollar aplicaciones como estas, es satisfactorio ver como a través de las tecnologías de la información se mejora el progreso de una industria e inclusive de un país.

- Se puede concluir que el desarrollo de un aplicativo móvil para el control de procesos industriales es complejo ya que los dispositivos de control no poseen una comunicación directa lo cual lleva a utilizar varias tecnologías.

9. RECOMENDACIONES

- La aplicación da un mejor rendimiento con un computador con 2 Gigas de memoria, un procesador Intel Core Duo 1.73 Mhz, además de esto, si se tienen celulares con buenas características (Tecnología GSM, 5 MB de memoria, J2ME versión CLDC 1.0, MIDP 2.0) el desempeño será mucho mejor.

- Se pretende que esta aplicación sea dinámica a la hora de agregar nuevos procesos industriales.

- Este sistema se puede actualizar o ser escalable de acuerdo a las nuevas versiones de software con el que fue implementado debido a que este tiene esta característica para un mejor desempeño en cuanto a funcionamiento y confiabilidad.

- Para un mejor desempeño de la aplicación se debe adquirir un buen servicio de datos, el cual es prestado por las empresas de celulares.

- No se tiene la posibilidad de personalizar las interfaces de acuerdo al usuario (colores, fondos, logotipos).

- Este sistema cuenta con una seguridad básica, se recomienda implementar un mejor algoritmo de seguridad en las comunicaciones.

BIBLIOGRAFIA

ALCÁNTARA Segura Sergio. Proyecto de Sistema de Control Domótico. Universidad Lasal le Bonanova de Barcelona. 2000. [Pagina Web en línea]. Disponible en: http://www.imarketing.es/pdf/sistema_domotico.pdf. [consulta: 19 marzo 2006].

BARRIOS Juan, MORENO José. Java Servlets. [En línea]. Chile. Fecha de publicación: 30 de Noviembre del 2001. Disponible en: http://www.dcc.uchile.cl/~jbarrios/servlets/index.html. [Consulta: 26 julio 2007].

GRACIA, Joaquin. HTML fácil. [En línea]. Madrid. Fecha de publicación: Agosto 2006. HTML / Contenido del manual / conceptos básicos. Disponible en: http://www.webestilo.com/html/cap1a.phtml

NUMPAQUE Pablo. Aplicación de Telemetría. DATUMCOMCEL. 2004. [Pagina Web en línea]. Disponible en: http://www.datumcomcel.com/. [Consulta: 19 marzo 2006].

MERELOS, Juan. Programando con JSps. [en linea]. Granada. Fecha de publicación: Octubre 2004. Tutoriales diversos / Tutorial de java server pages, JSPs. Disponible en: http://geneura.ugr.es/~jmerelo/JSP/

LÓPEZ R. Sonia, PÉREZ O. Joaquín. Prototipo de Comercio Electrónico Orientado A Dispositivos Móviles en México. Centro Nacional de Investigación y Desarrollo Tecnológico de México. 2005. [Pagina Web en línea]. Disponible en: www.cenidet.edu.mx/subaca/web-dcc/web-sd/Pazos/CIICC05-ICI05/Art035.pdf. [Consulta: 24 abril 2006].

RIFÁ Pous Helena. Diseño e implementación de una pasarela de signatura digital con tecnología WAP. Universidad Politécnica de Barcelona. 2001. [Pagina Web en línea]. Disponible en: www.coit.es/pub/ficheros/philips_f68cd55b.pdf. [Consulta: 24 abril 2006].

DOMÍNGUEZ, Octavio. Conceptos Básicos. [en línea]. Las Palmas. Fecha de publicación: Marzo 2002. Guías / Cursos / Tutorial de HTML. Disponible en: http://gias720.dis.ulpgc.es/Gias/Cursos/Tutorial_html/indice.htm

HOSPEDAJE Y DOMINIOS S.L. Apache: el comienzo. [en línea]. Murcia. Fecha n.htm de publicación: Julio 2005. Apache / Manual de Apache / Lista de características de Apache. Disponible en: http://www.hospedajeydominios.com/mambo/documentacion-manual_apache-pagina-45.html

WALES, Jimmy. Servidor Apache Tomcat.[en línea]. Fecha de publicación: Octubre 2005. Artículo / Servidor Apache Tomcat. Disponible en: http://es.wikipedia.org/wiki/Jakarta_Tomcat

GÁLVEZ R Sergio, ORTEGA D Lucas. Java a Tope: J2ME (Java 2 Microedition). [En línea]. Fecha de publicación: 2003. Universidad de Málaga. Disponible en: http://www.lcc.uma.es/~galvez/J2ME.html.

MICROSOFT Corporation. SQL Server Fundamentation. [En línea]. Fecha de publicación: 2007. Disponible en: http://msdn2.microsoft.com/es-co/library/ms166352.aspx.

FERNÁNDEZ Felipe, MUÑOZ Yerko.JDBC. [en linea]. Disponible en: http://www.dcc.uchile.cl/~lmateu/CC60H/Trabajos/jfernand/.

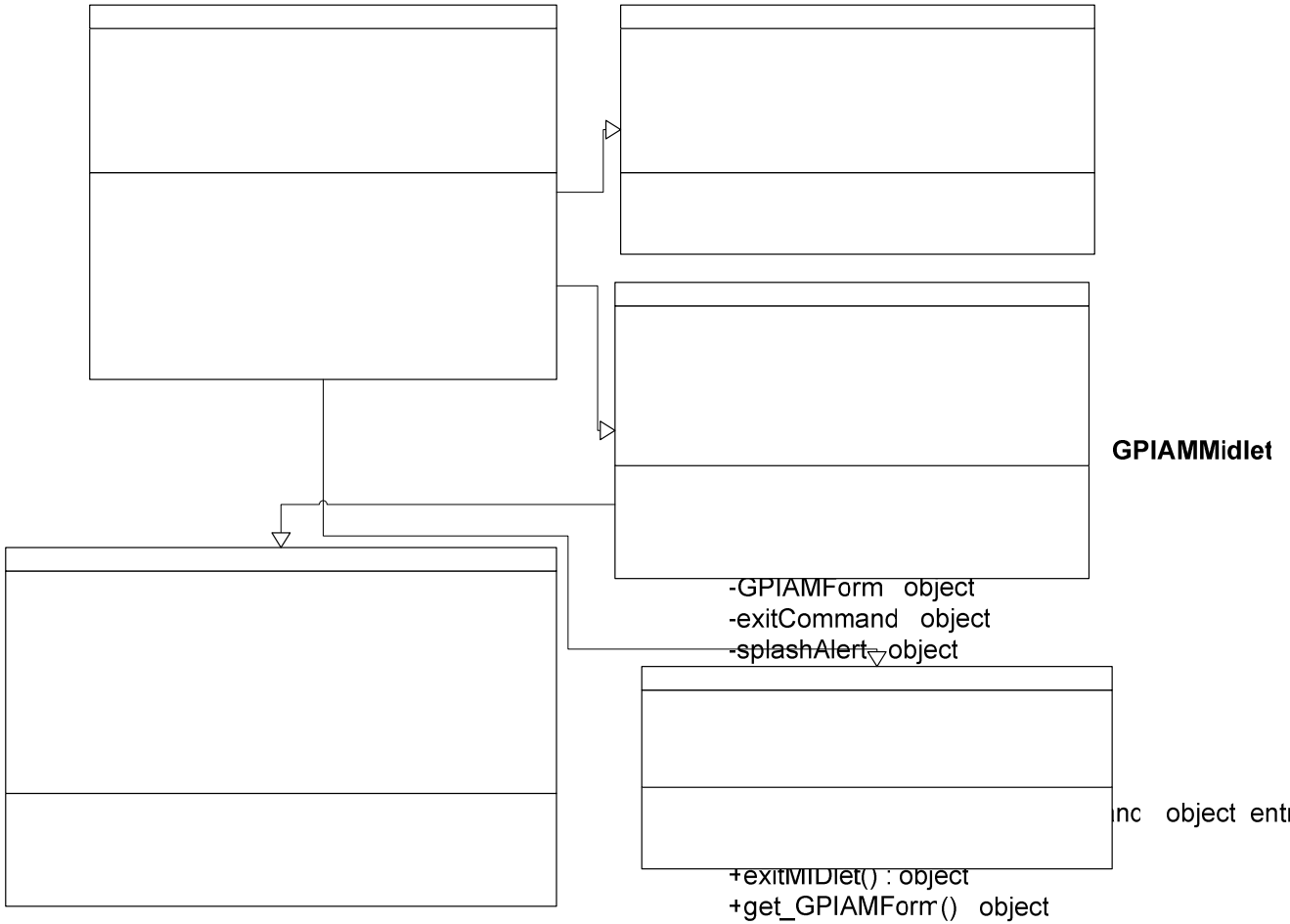
AUTOMATAS INDUSTRIALES. OPC OLE for Process Control. [En línea]. Fecha de publicación: 2006. Disponible en: http://www.automatas.org/redes/opc.htm. Consulta: 31 Enero 2008].

ANEXOS

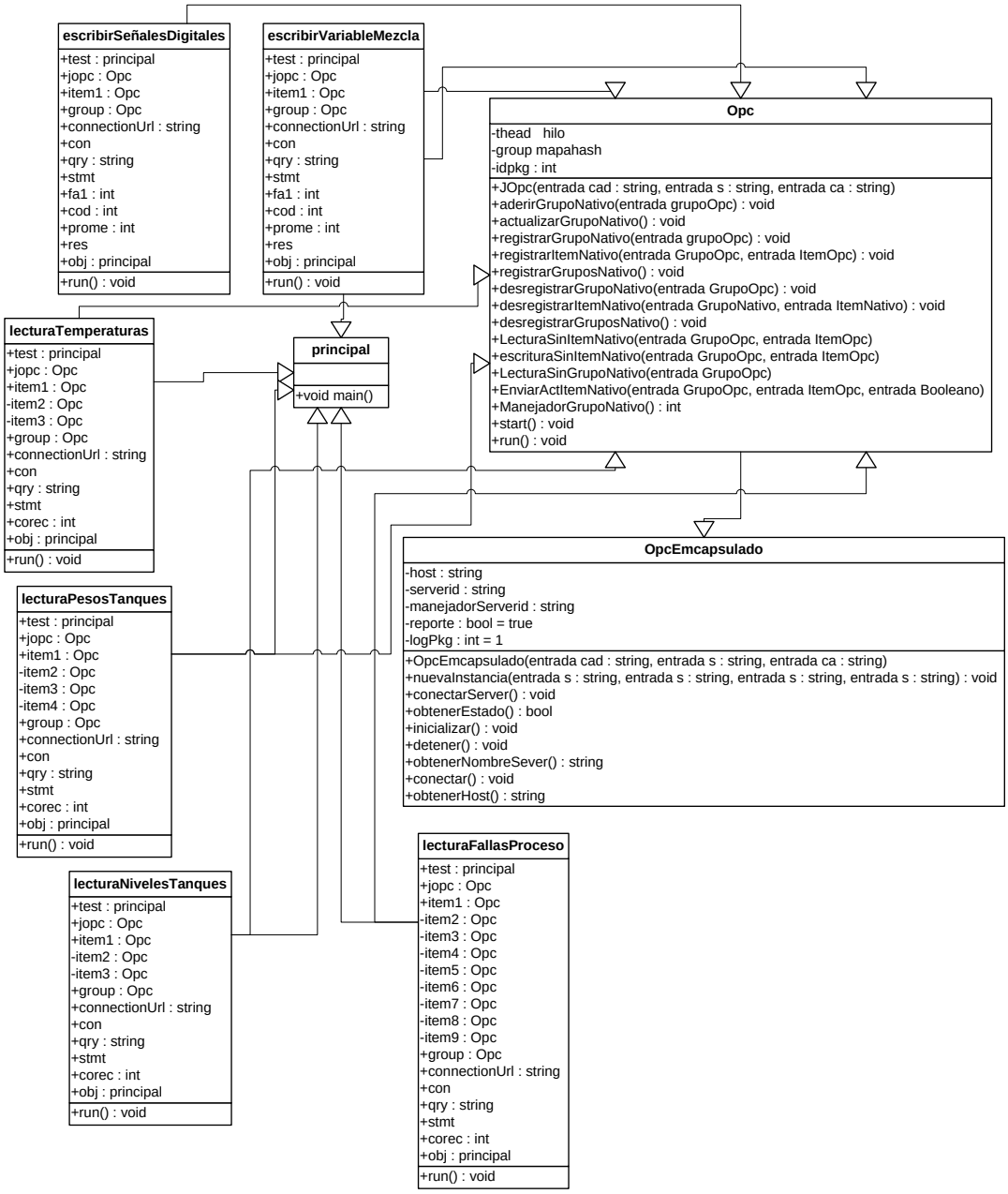
10. ANEXO A

10.1 DIAGRAMAS DE CLASES

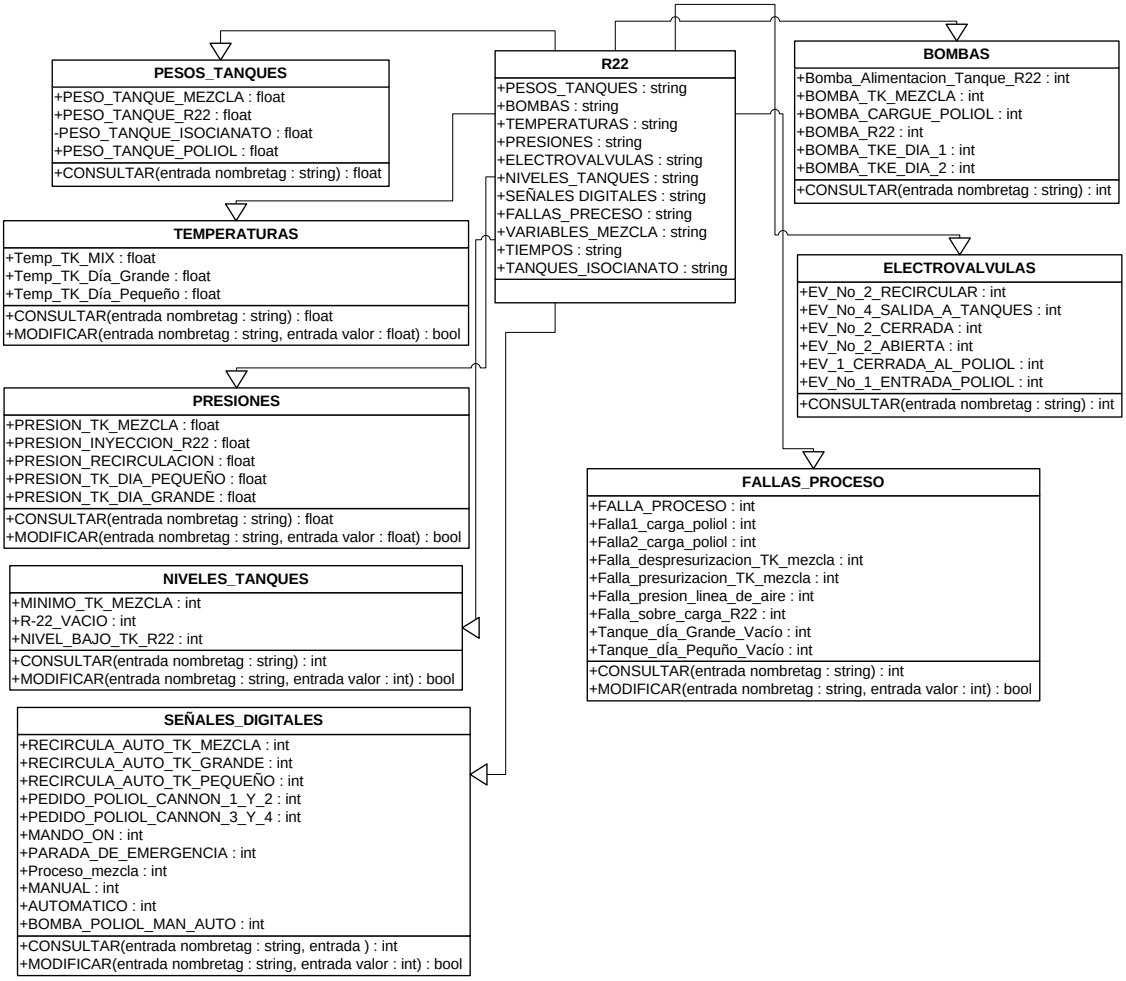
10.1.2 DIAGRAMA DE CLASES APLICACIÓN MÓVIL GPIAM



10.1.2 DIAGRAMA DE CLASES OPC

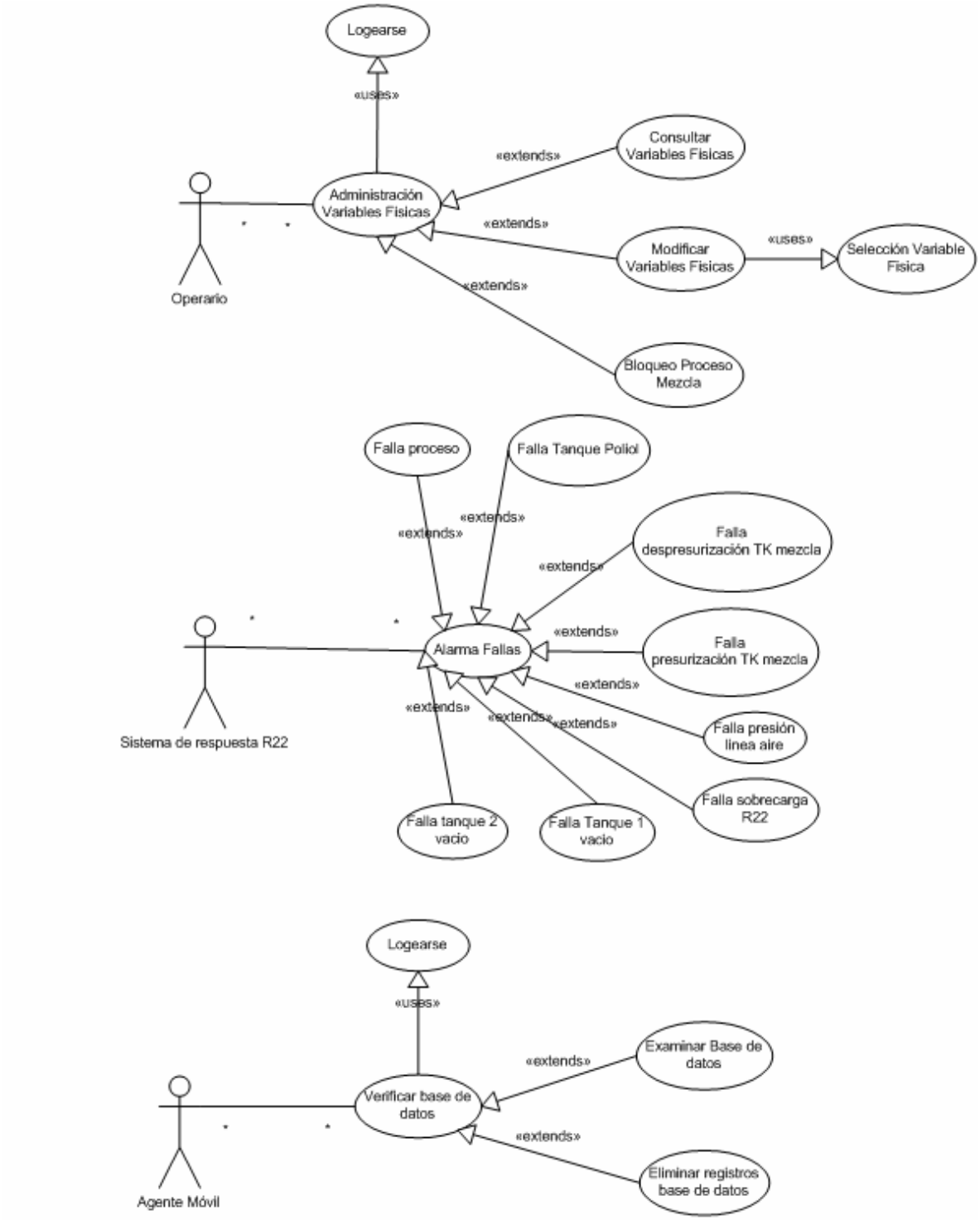


10.2.3 DIAGRAMA DE CLASES DE MEZCLA, POLIOL + R22

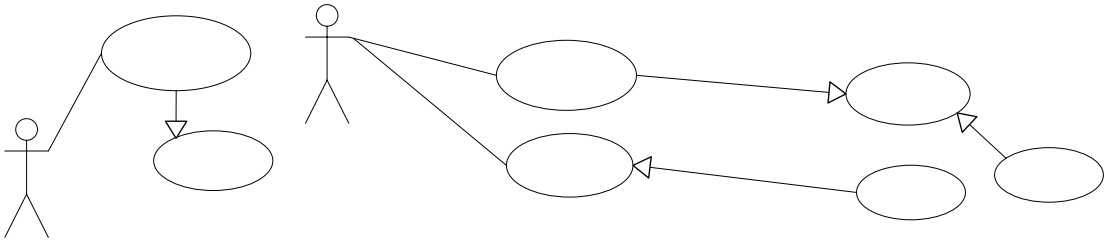


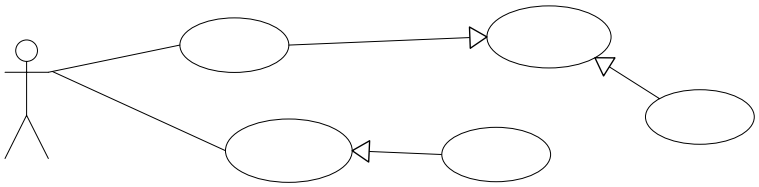
11. ANEXO B

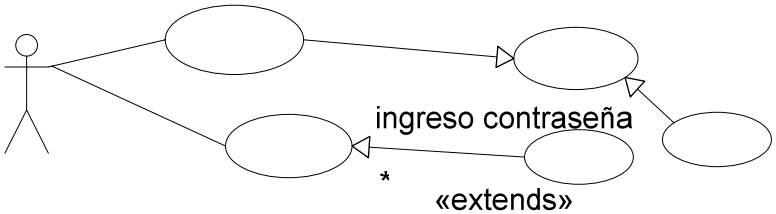
11.1 DIAGRAMAS DE CASOS DE USO

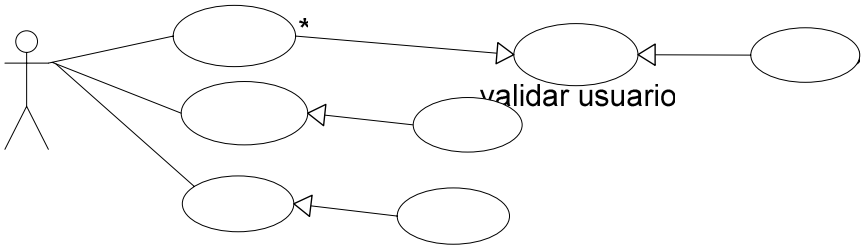


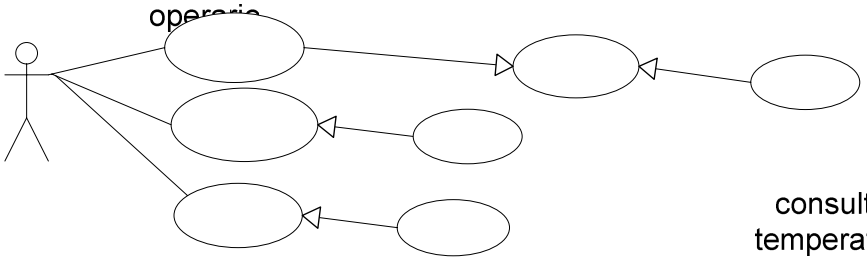
11.2 DIAGRAMAS DE CASOS DE USO DETALLADO

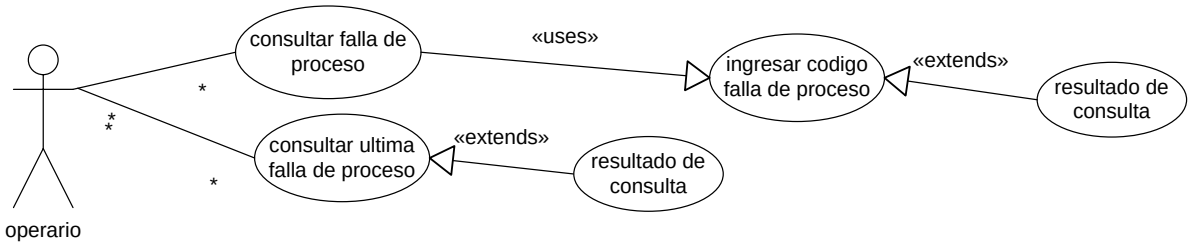






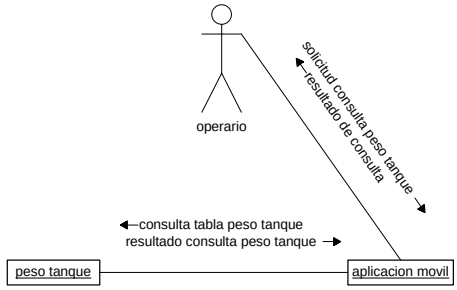


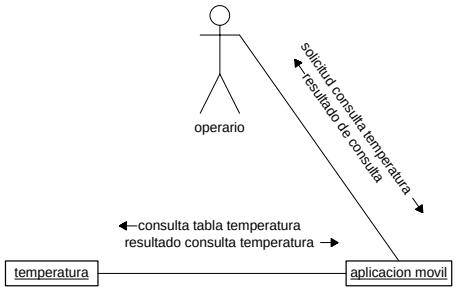


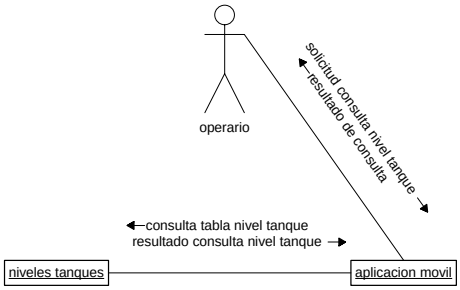

consultar falla de proceso
«uses»
ingresar codigo falla de proceso
«extends»
resultado de consulta
consultar ultima falla de proceso
«extends»
resultado de consulta
operario
*
*
*
*

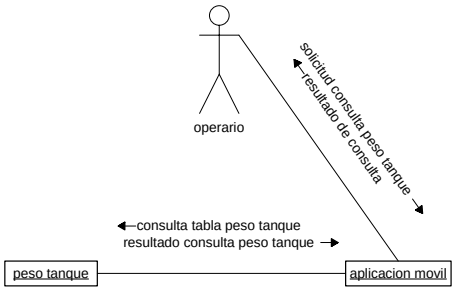
12. ANEXO C

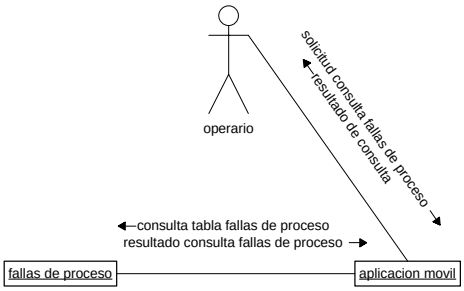
12.1 DIAGRAMAS DE COLABORACION

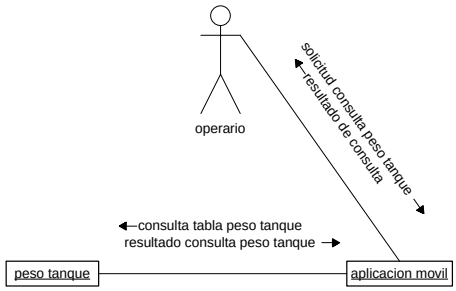

operario
solicitud consulta peso tanque
resultado de consulta
consulta tabla peso tanque
resultado consulta peso tanque
peso tanque
aplicacion movil

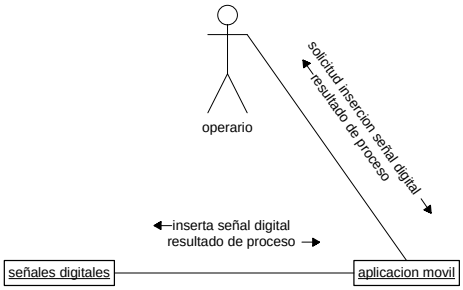

operario
solicitud consulta temperatura
resultado de consulta
consulta tabla temperatura
resultado consulta temperatura
temperatura
aplicacion movil

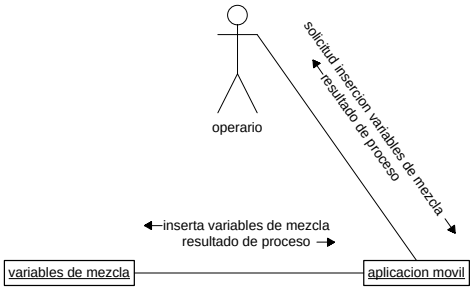

operario
solicitud consulta nivel tanque
resultado de consulta
consulta tabla nivel tanque
resultado consulta nivel tanque
niveles tanques
aplicacion movil


operario
solicitud consulta peso tanque
resultado de consulta
consulta tabla peso tanque
resultado consulta peso tanque
peso tanque
aplicacion movil


operario
solicitud consulta fallas de proceso
resultado de consulta
consulta tabla fallas de proceso
resultado consulta fallas de proceso
fallas de proceso
aplicacion movil


operario
solicitud consulta peso tanque
resultado de consulta
consulta tabla peso tanque
resultado consulta peso tanque
peso tanque
aplicacion movil

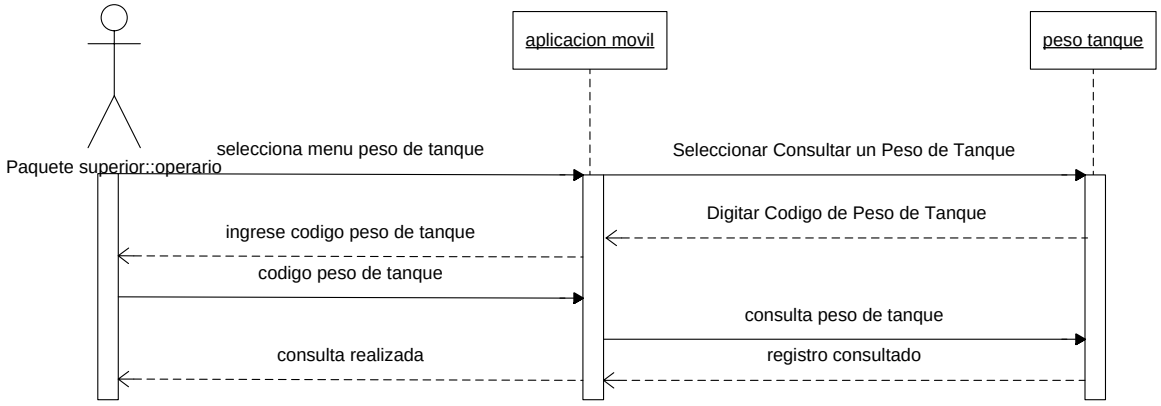

operario
solicitud insercion señal digital
resultado de proceso
inserta señal digital
resultado de proceso
señales digitales
aplicacion movil

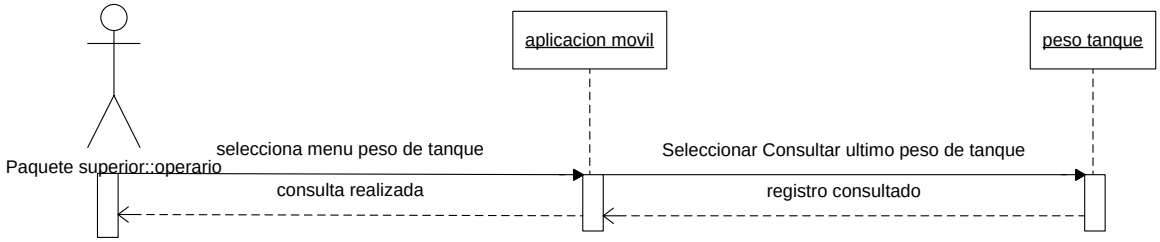

operario
solicitud insercion variables de mezcla
resultado de proceso
inserta variables de mezcla
resultado de proceso
variables de mezcla
aplicacion movil

13. ANEXO D

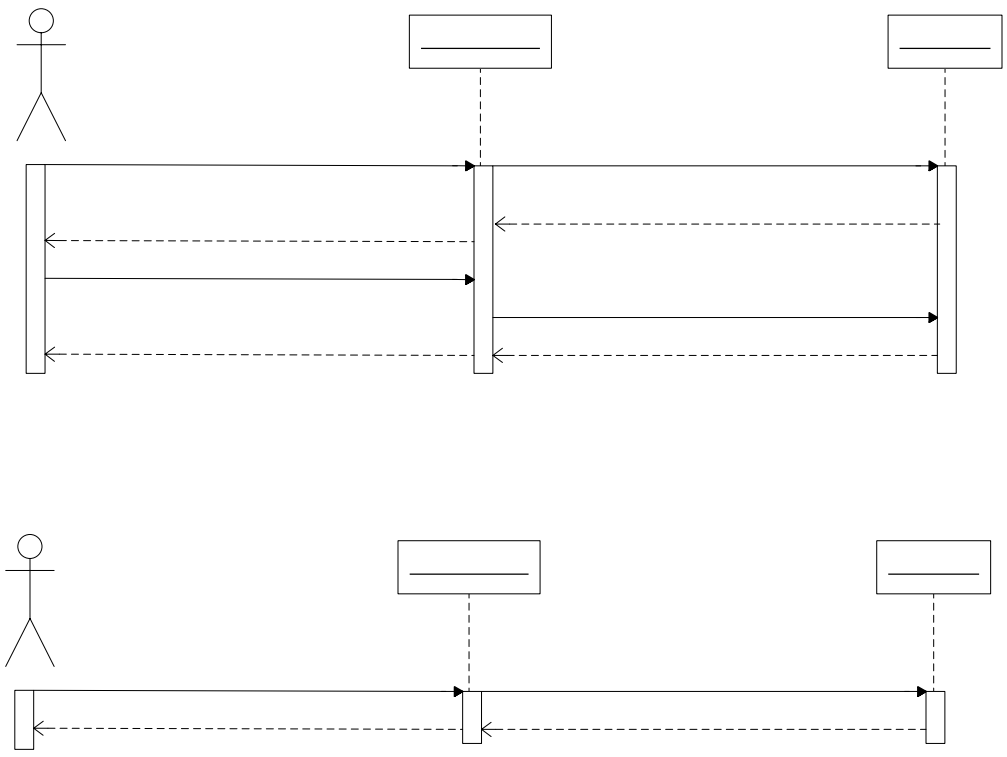
13.1 DIAGRAMAS DE SECUENCIA

13.1.1 Diagrama de secuencia consulta peso tanque

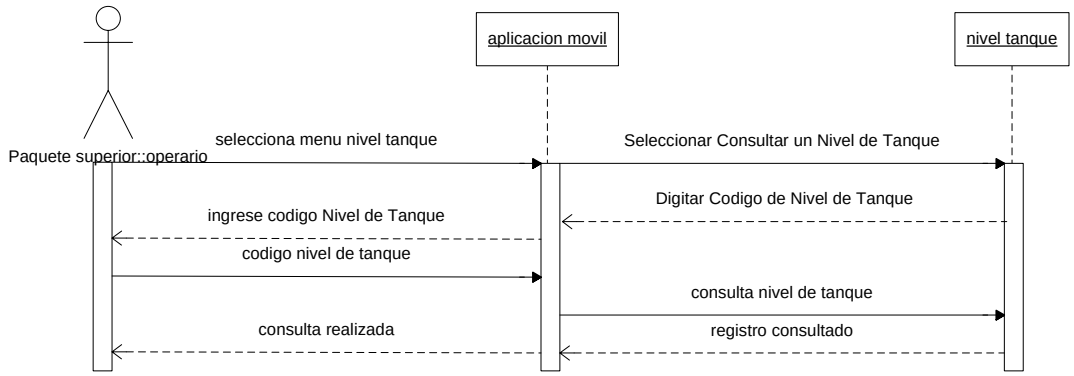


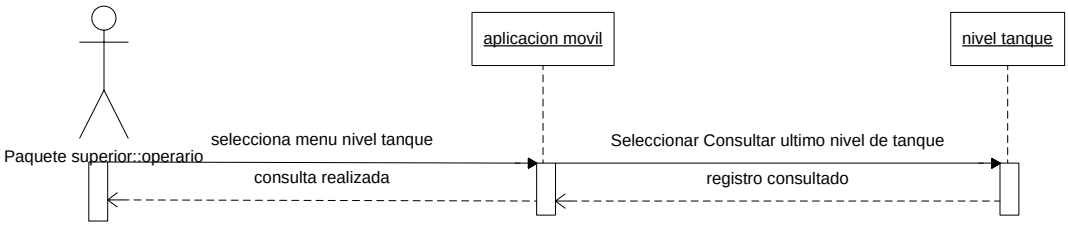


13.1.2 Diagrama de secuencia consultas temperaturas

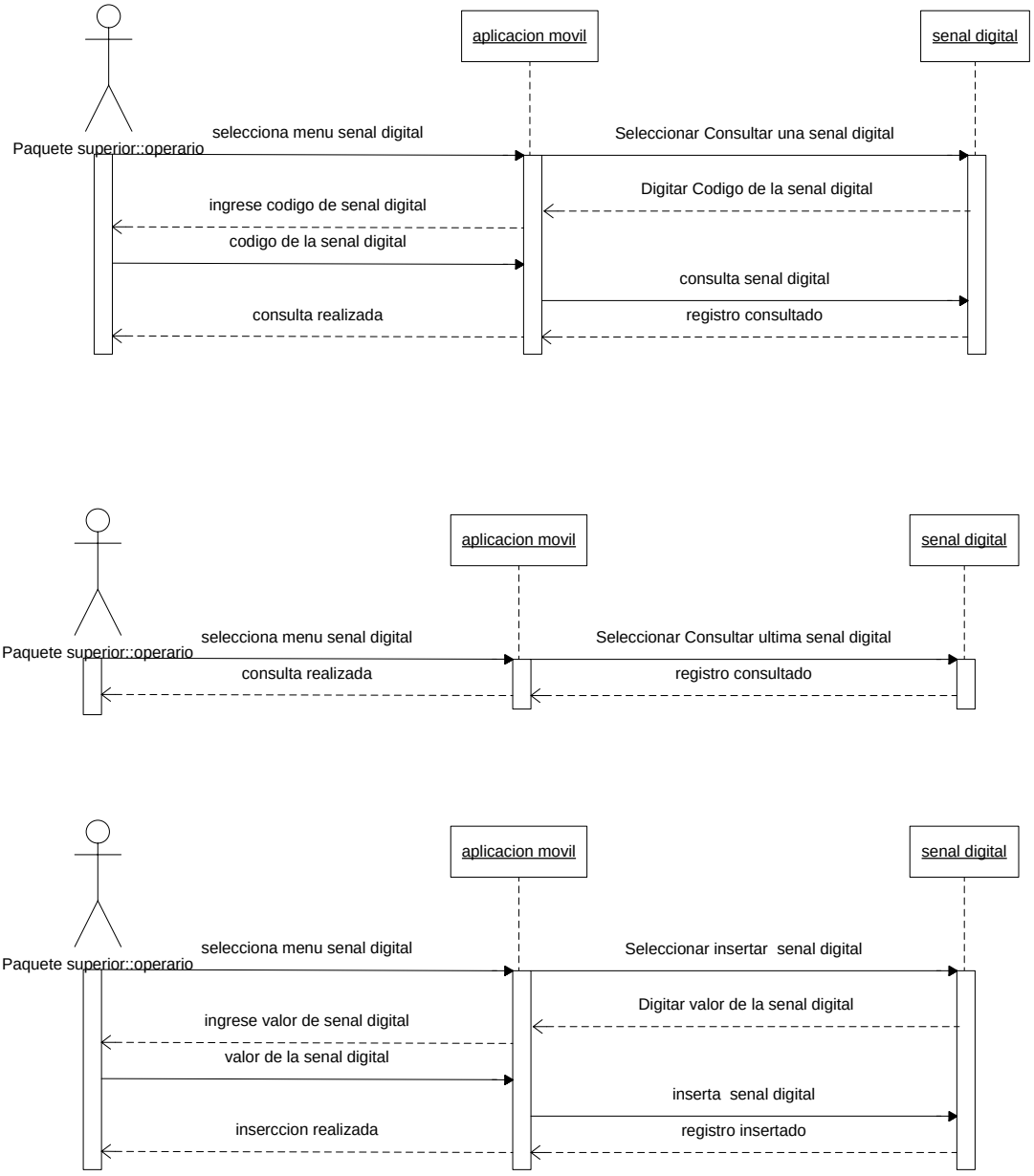


13.1.3 Diagrama de secuencia consultas niveles de tanques

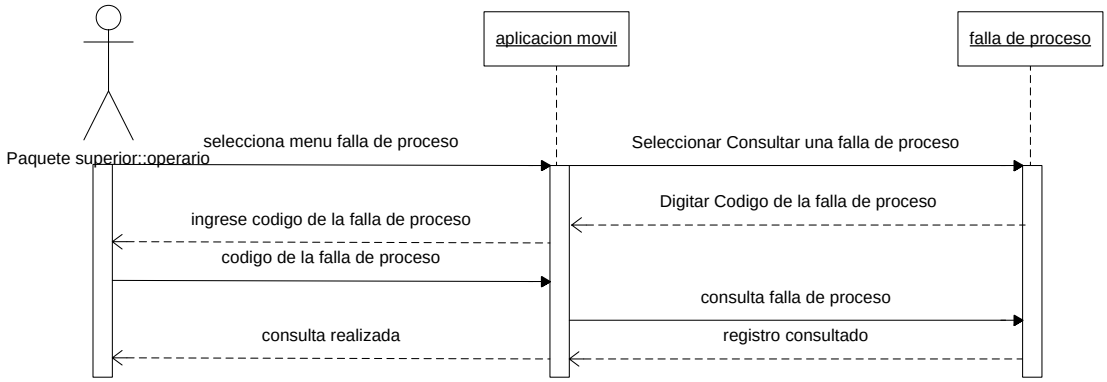


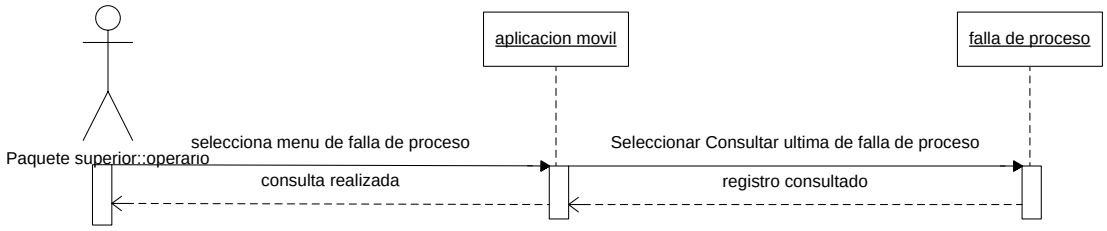


13.1.4 Diagrama de secuencia consultar señales digitales

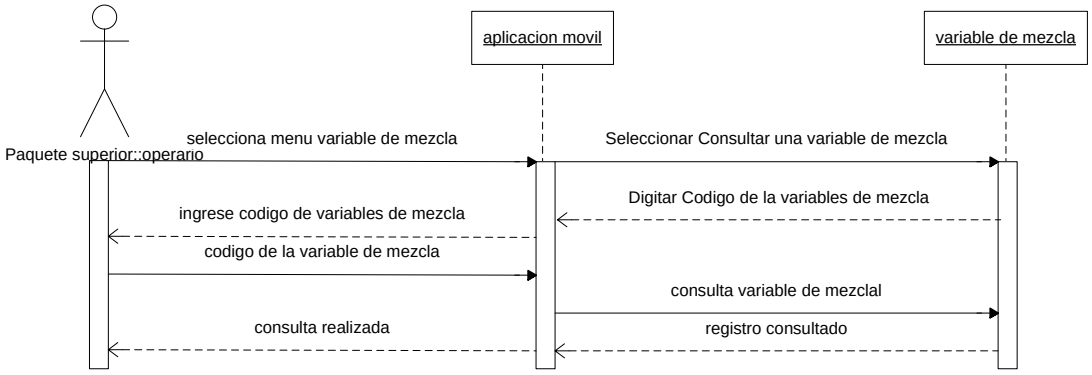


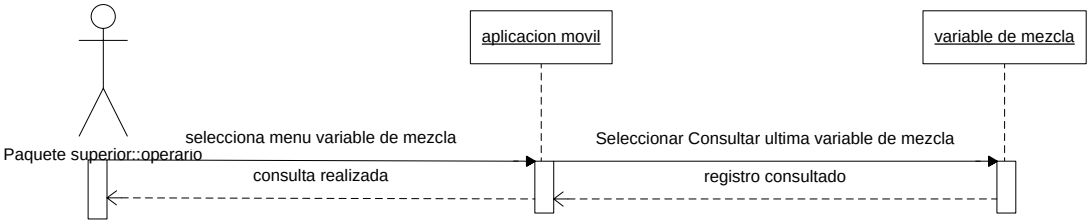
13.1.5 Diagrama de secuencia consultar fallas proceso

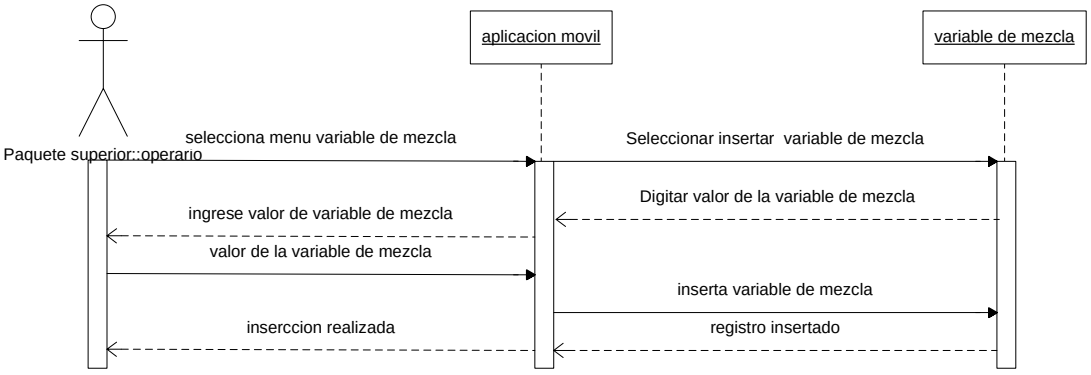




13.1.6 Diagrama de secuencia consultar variables de mezcla







14. MANUAL TÉCNICO

14.1 INSTALAR TOMCAT 6.0

Tomcat es un servidor de aplicaciones que funciona como un contenedor de servlets desarrollado bajo el proyecto Jakarta en la Apache Software Foundation. Tomcat implementa las especificaciones de los servlets y de JavaServer Pages (JSP) de Sun Microsystems.

Tomcat 6.0 necesita para su adecuado funcionamiento los siguientes componentes:

J2SE versión 1.6 del (J2SE) del JDK, *Java Development Kit* (JDK), Standard Development Kit (SDK) y Java 2 Standard Edition (J2SE), todo estos desarrollando por la SUN MICROSYSTEM.

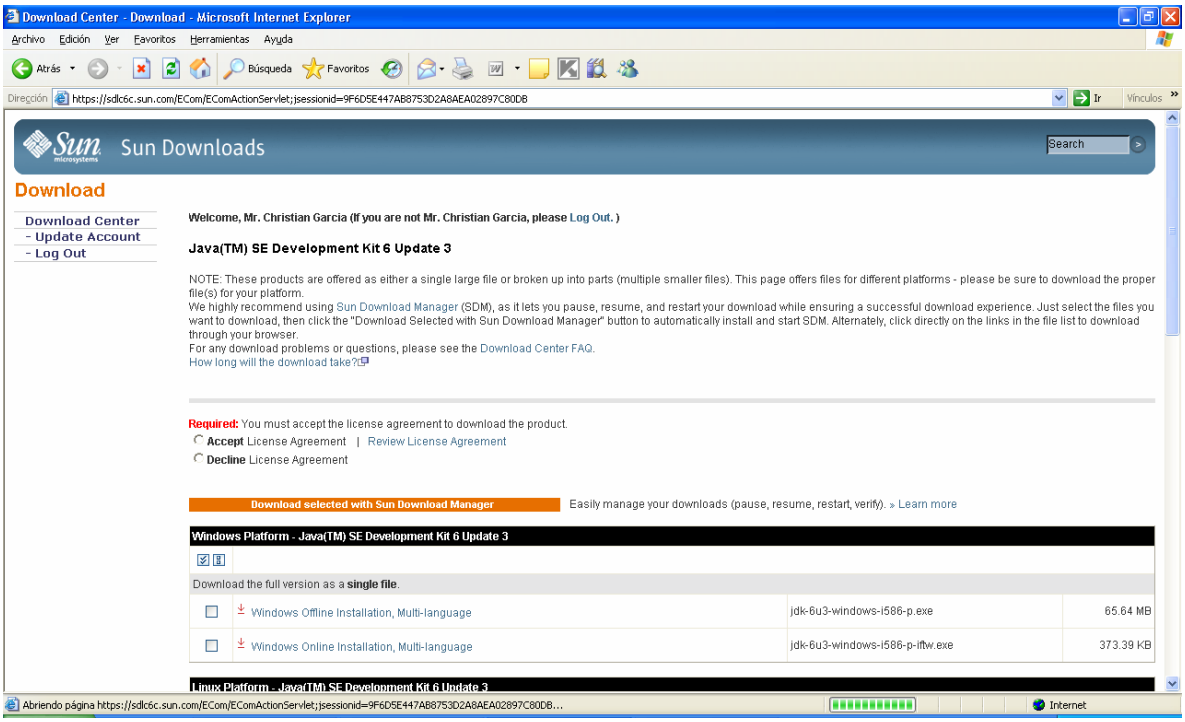
Para realizar una instalación adecuada de este software se deben realizar los siguientes pasos:

1. Es necesario poder instalar el JDK de java para el debido funcionamiento de Tomcat 6.0, el JDK se puede descargar del siguiente link: https://sdlc1b.sun.com/ECom/EComActionServlet;jsessionid=A0FB16E670121405C32C584E95A94411, se debe instalar el paquete de software.

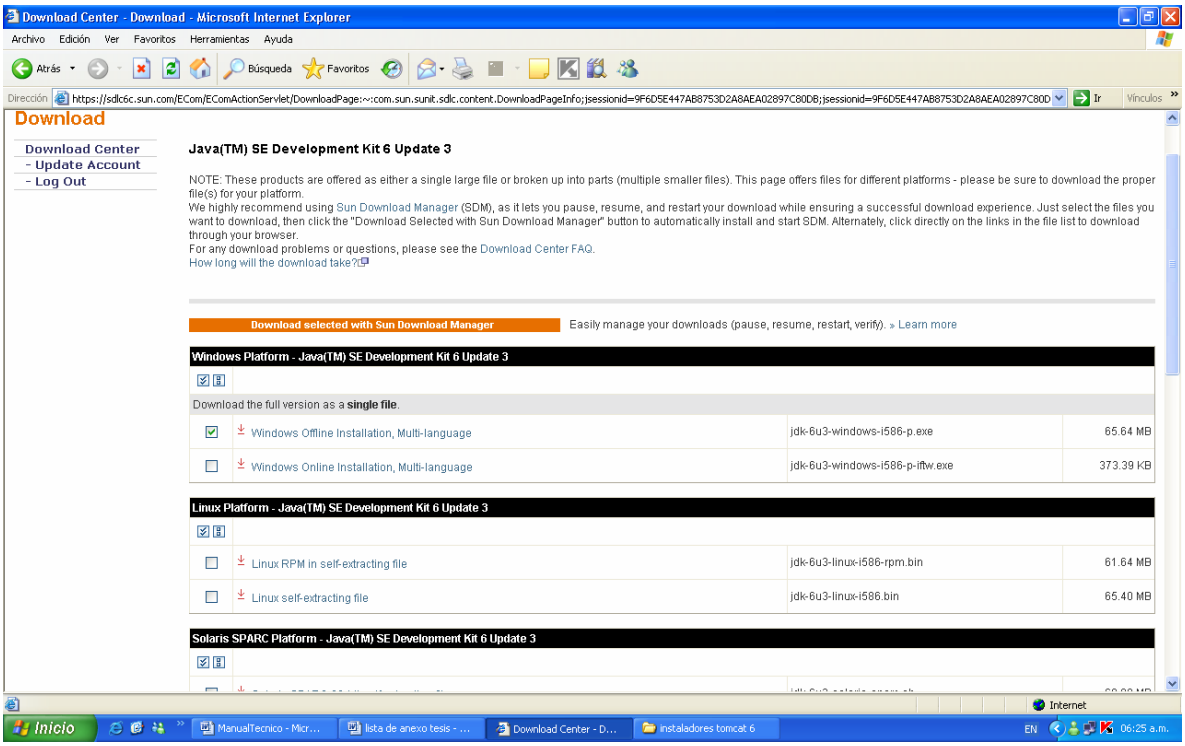
14.1.1 Instalando el SDK de java

Descarga del JDK 1.6.

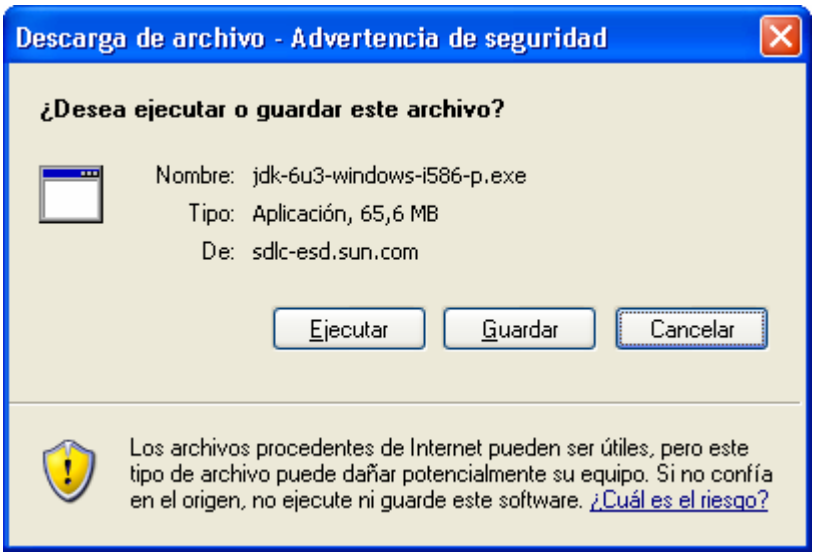
Se vá a la siguiente dirección Web: https://sdlc1b.sun.com/ECom/EComActionServlet;jsessionid=A0FB16E670121405C32C584E95A94411 se acepta los compromisos de licencia como se muestra en la siguiente figura.



Luego se debe proceder a elegir en que sistema operativo se desea bajar el JDK, en este caso se elije Windows y se pulsa sobre el hipervínculo.

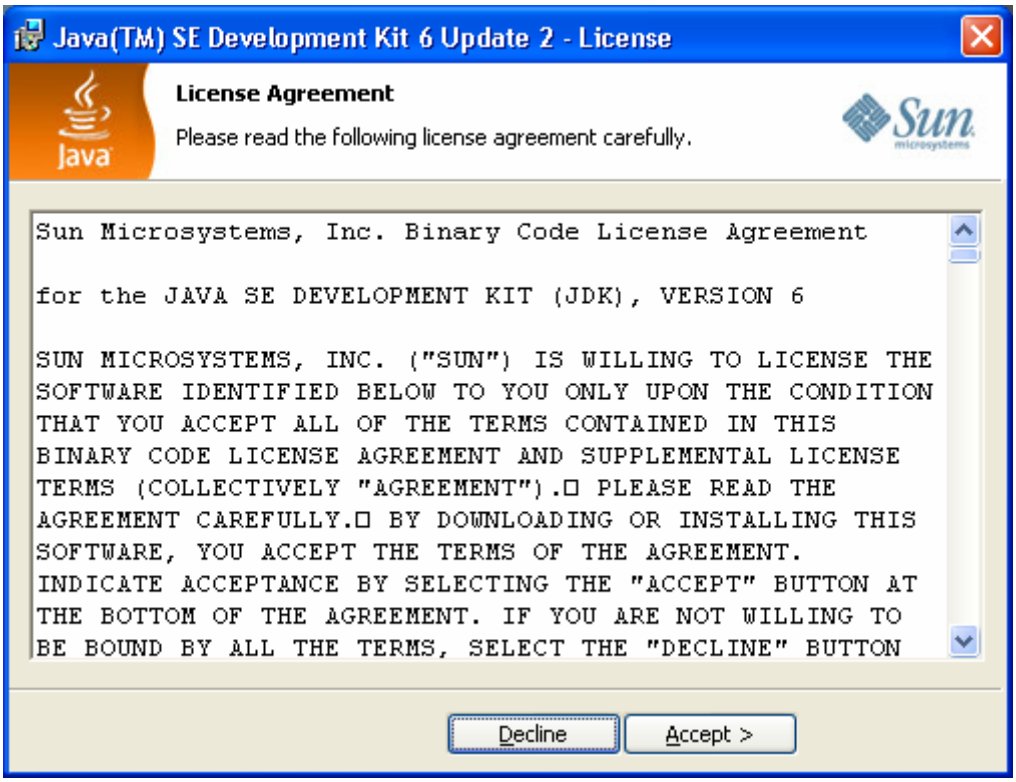


Al darle clic en el vínculo aparecerá una ventana para realizar la descarga del software.

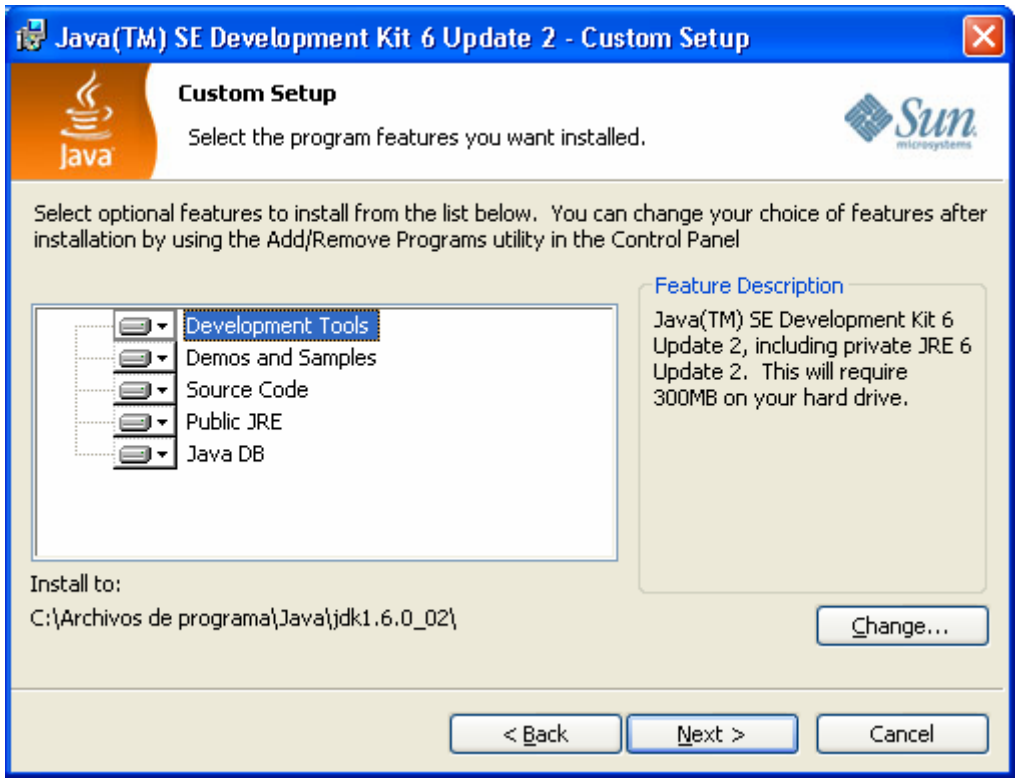


INSTALACIÓN.

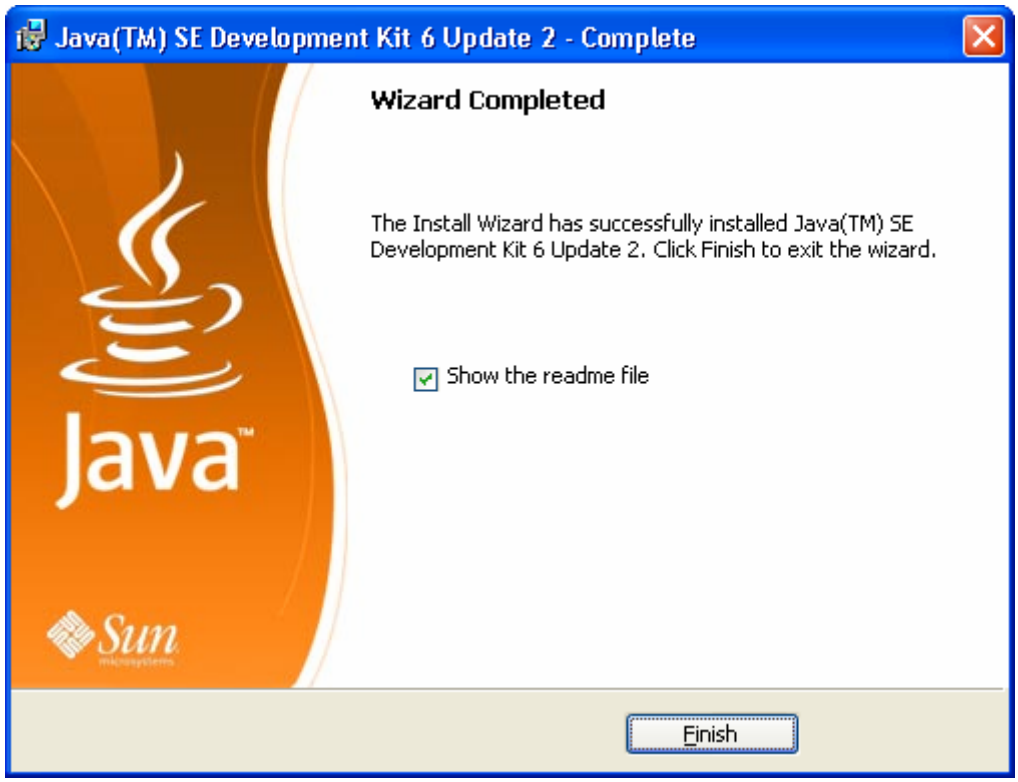
Se dirige al directorio donde se encuentra el instalador del JDK, se ejecuta el programa donde aparecerá una interfaz para realizar la instalación



Se debe aceptar la licencia de uso del programa, darle click en siguiente para seguir con la instalación.



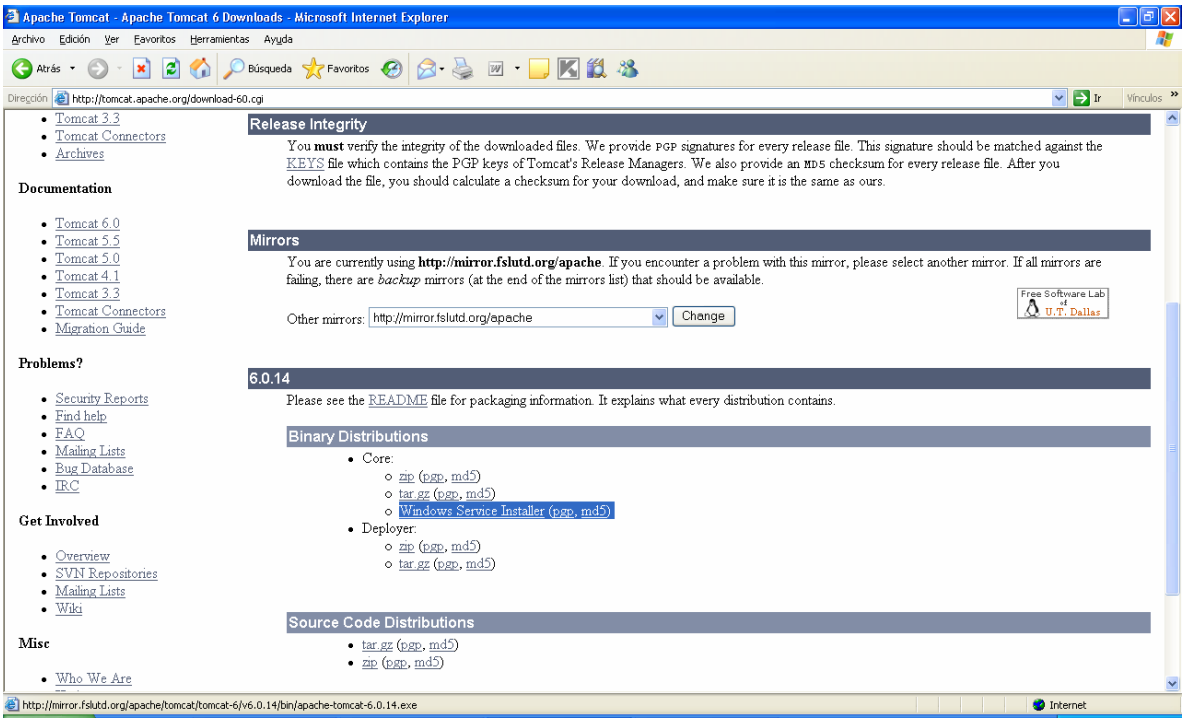
Se realiza el proceso de instalación y debe mostrar una ventana donde nos muestra si la instalación del producto fue exitosa.



Descargar Tomcat

Para realizar la descarga del producto se debe de ingresar en el siguiente link:

http://tomcat.apache.org/download-60.cgi, se da click en el hipervínculo que se muestra marcado en la siguiente imagen:

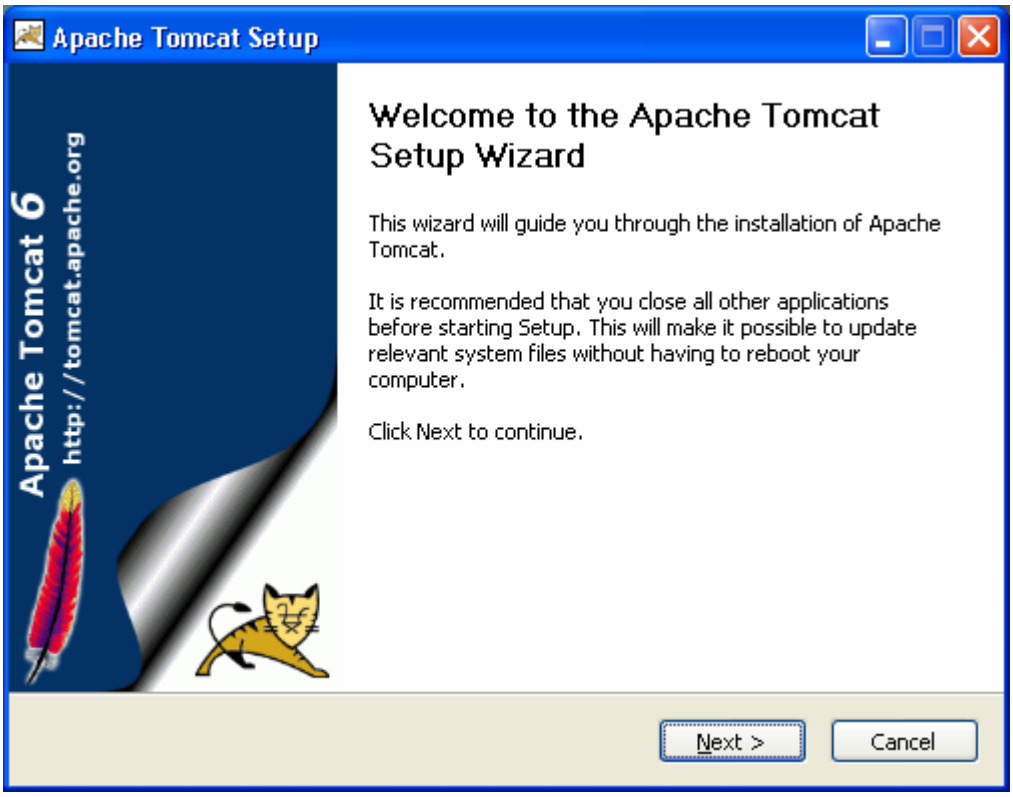


Esto también esta disponible en el contenido del CD de instalación de proyecto.

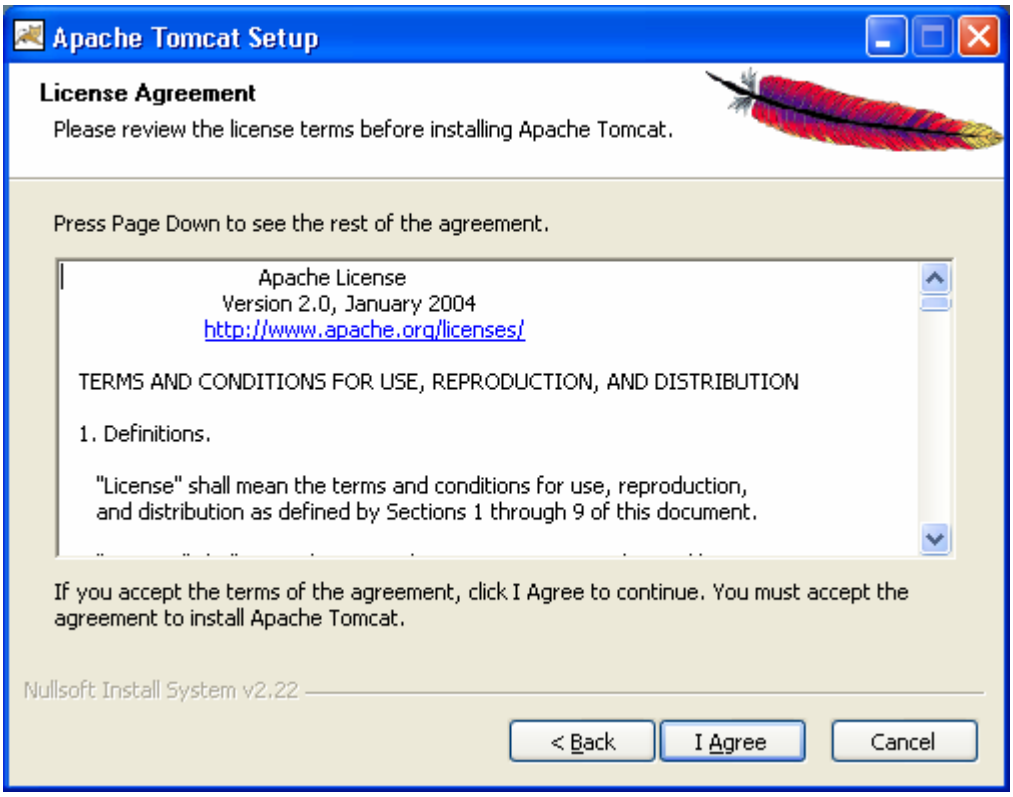
14.2.2. Instalación de Tomcat 6.0

NOTA: Antes de empezar la instalación de este producto se tener una conexión a Internet para que el software pueda descargar unas librerías que son necesarias para su instalación.

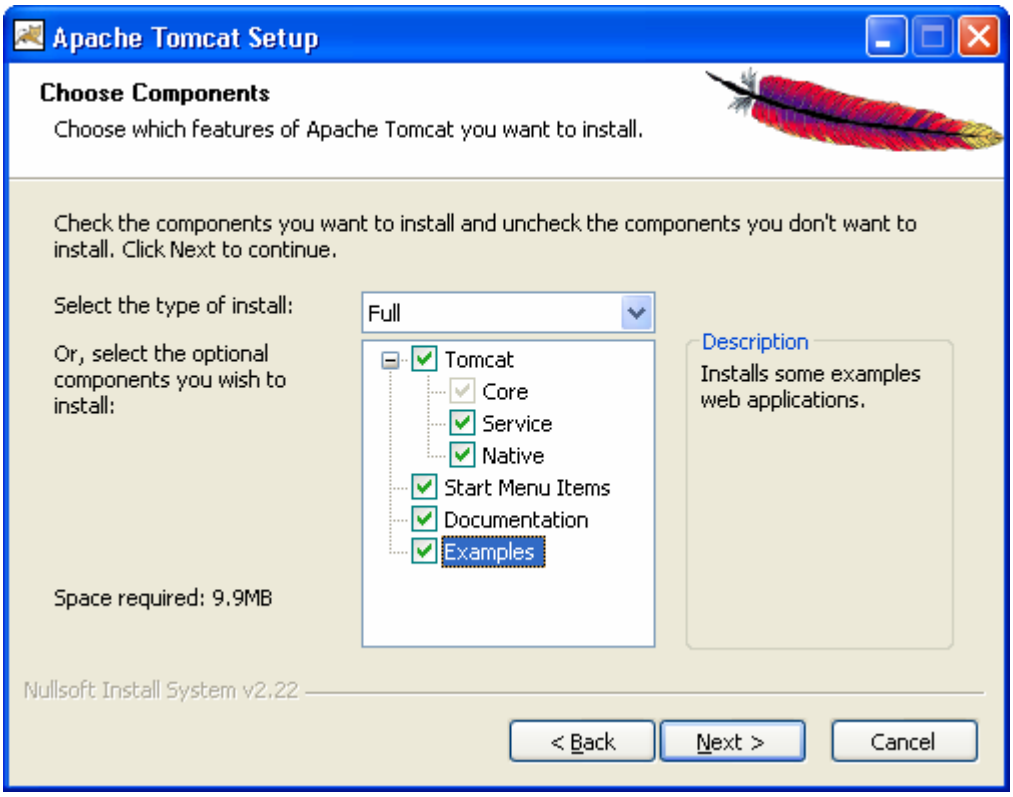
Se debe ejecutar el programa de instalación del Tomcat 6.0 que viene en el cd de instalación del proyecto, al ejecutar el programa este mostrara una ventana para iniciar el proceso.



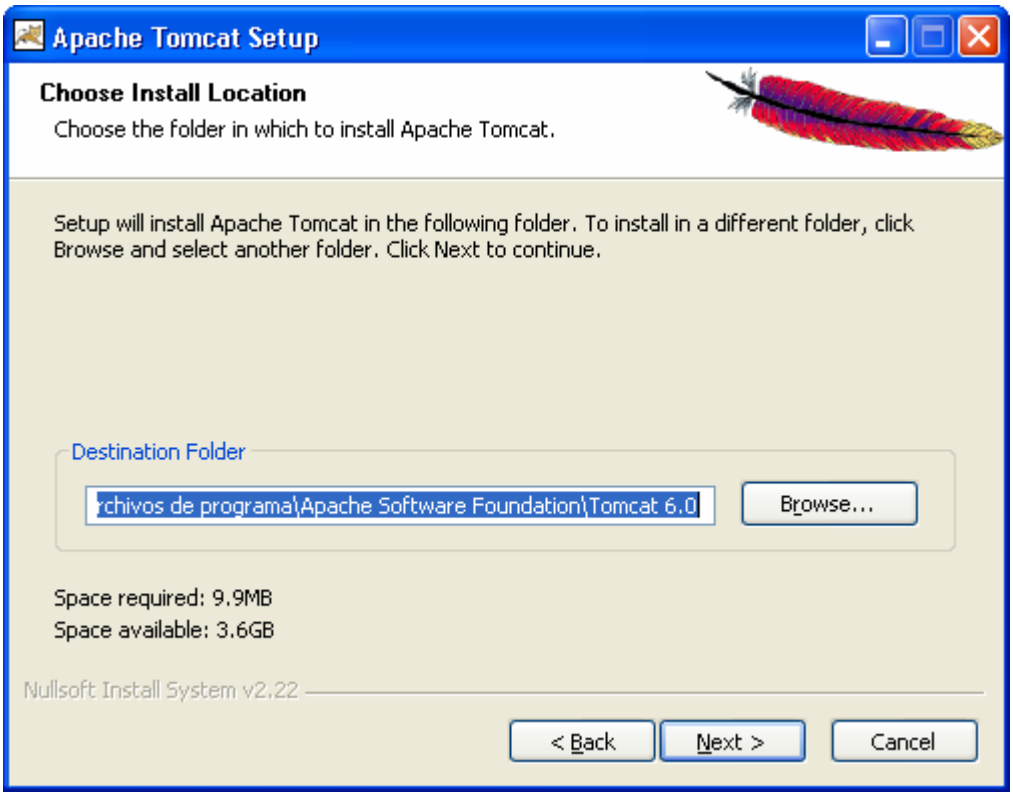
Se da click siguiente para continuar con la instalación del producto, se acepta la licencia de uso del programa.



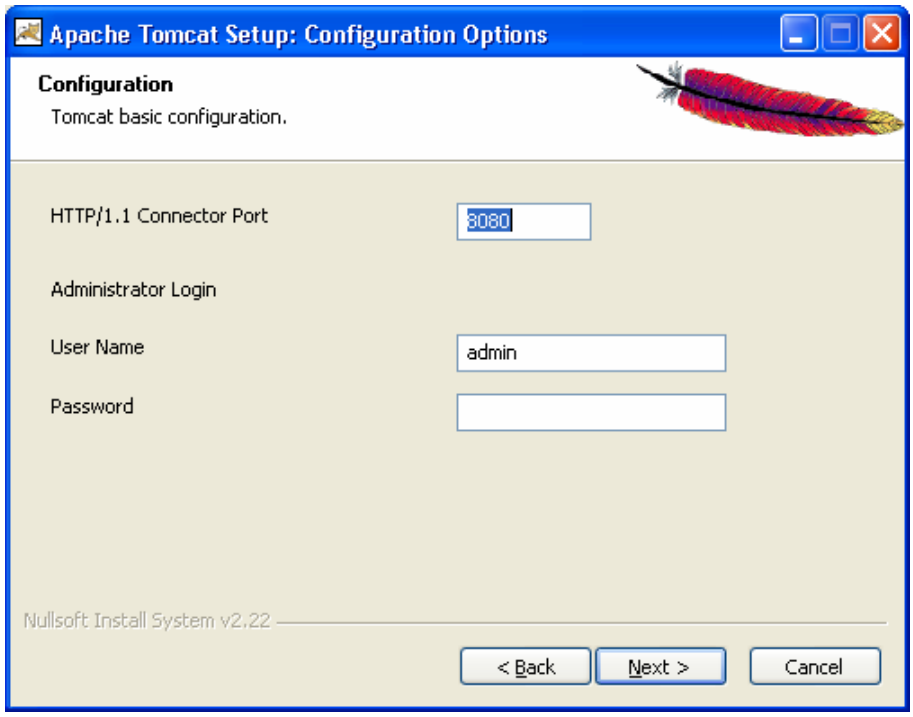
Se deben elegir los siguientes componentes para el adecuado funcionamiento del software.



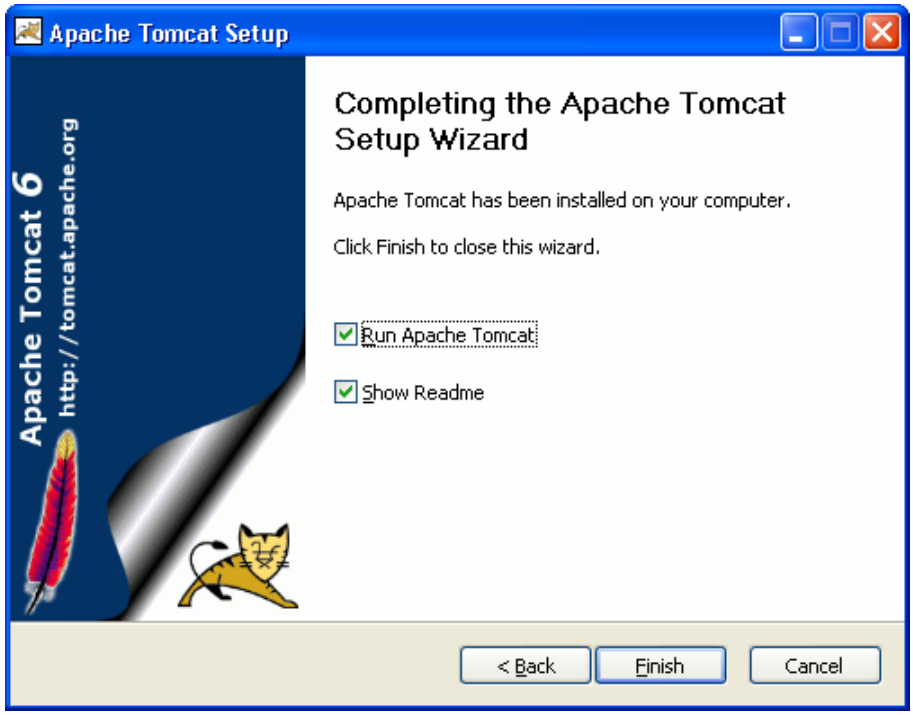
Se elige un directorio de instalación, este puede dejarse por defecto.



Se realiza la configuración respectiva al puerto y usuario en que va a funcionar el Tomcat, estos se dejan por defecto.



Al completar el proceso mostrar una ventana donde se confirmara si la instalación del producto fue exitosa



14.2 SQL SERVER 2000

Microsoft SQL Server es un sistema de gestión de bases de datos relacionales (SGBD) basada en el lenguaje Transact-SQL[22], capaz de poner a disposición de muchos usuarios grandes cantidades de datos de manera simultánea.

Para iniciar la instalación del SQL Server primero se debe primero realizar la descarga del programa de instalación.

14.2.1 Descargar SQL SERVER 2000

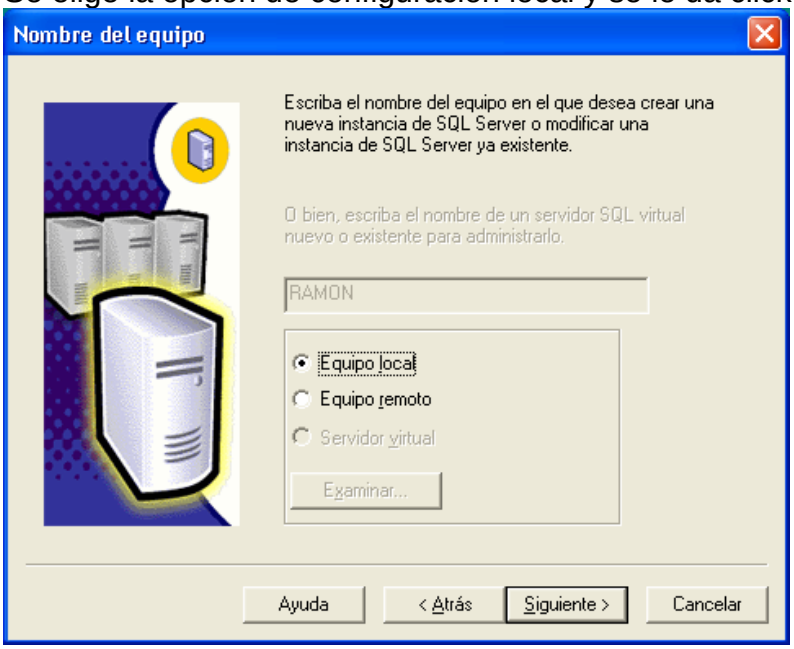
Para poder descargar el trial de SQL SERVER 2000 se debe ingresar al link: http://www.microsoft.com/downloads/details.aspx?FamilyID=D20BA6E1-F44C-4781-A6BB-F60E02DC1335&displaylang=en

14.2.2 Instalación SQL SERVER 2000

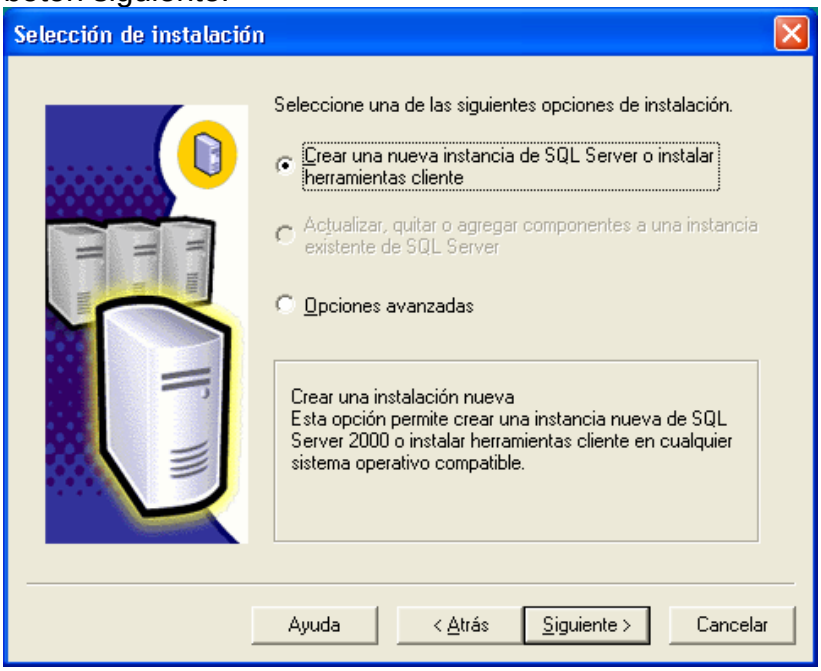
Se debe de ejecutar el programa de instalación, este muestra un ventana asistente para la configuración del SQL SERVER como se muestra en la siguiente imagen:

[22] (T-SQL). Transact-SQL es una extensión del lenguaje SQL, propiedad de Microsoft y Sybase. La implementación de Microsoft funciona en los productos Microsoft SQL Server. En tanto, Sybase utiliza el lenguaje en su Adaptative Server Enterprise, el sucesor de Sybase SQL Server.

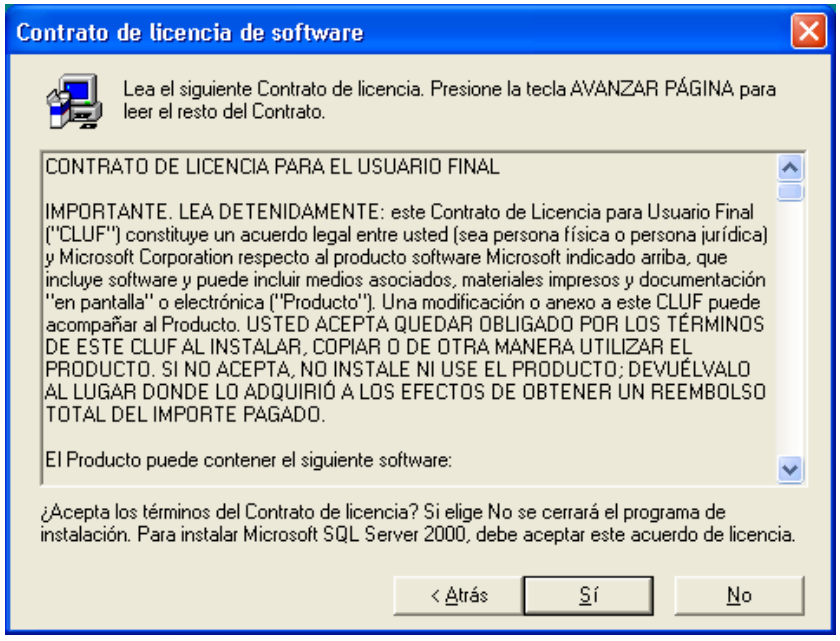
Se elige la opción de configuración local y se le da click al botón siguiente.



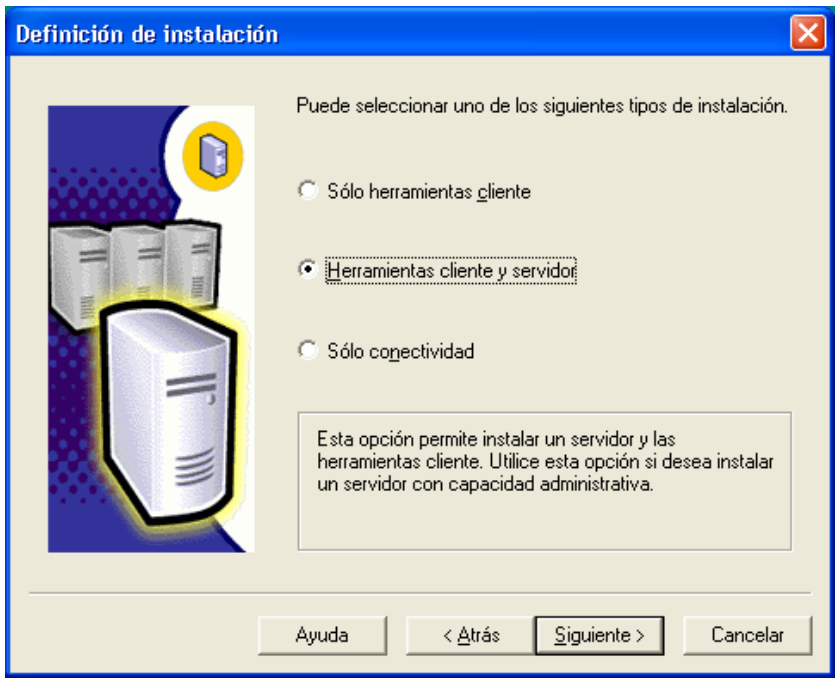
Se elige la opción de crear una nueva instancia SQL Server y se le da click al botón siguiente.



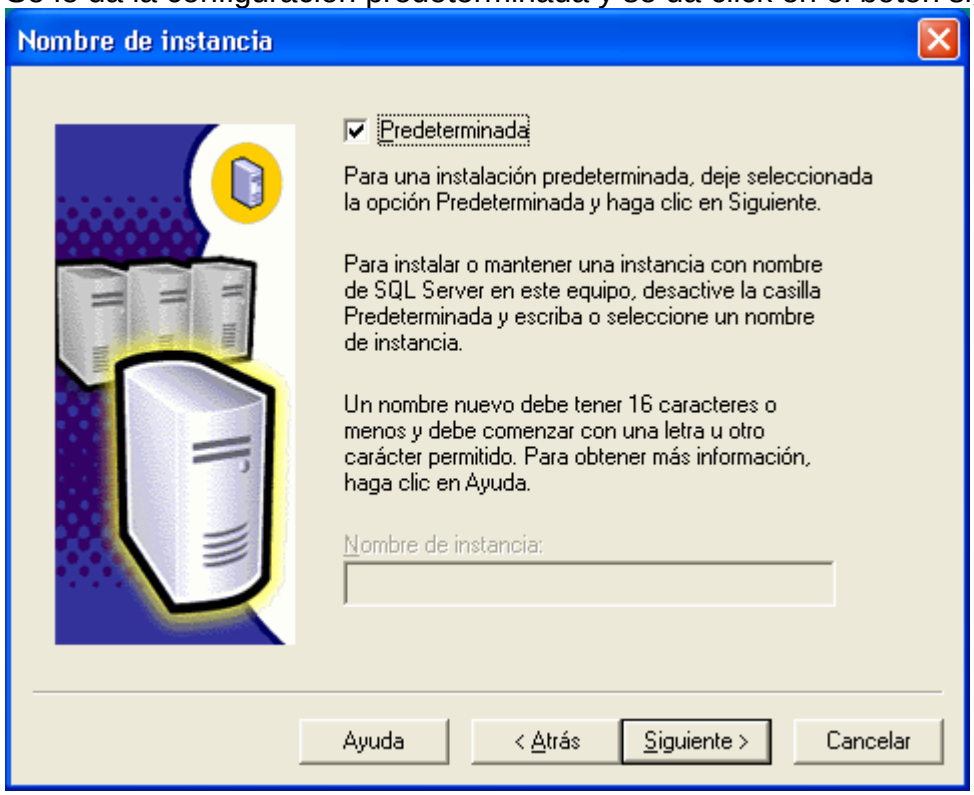
Se acepta la licencia de uso.



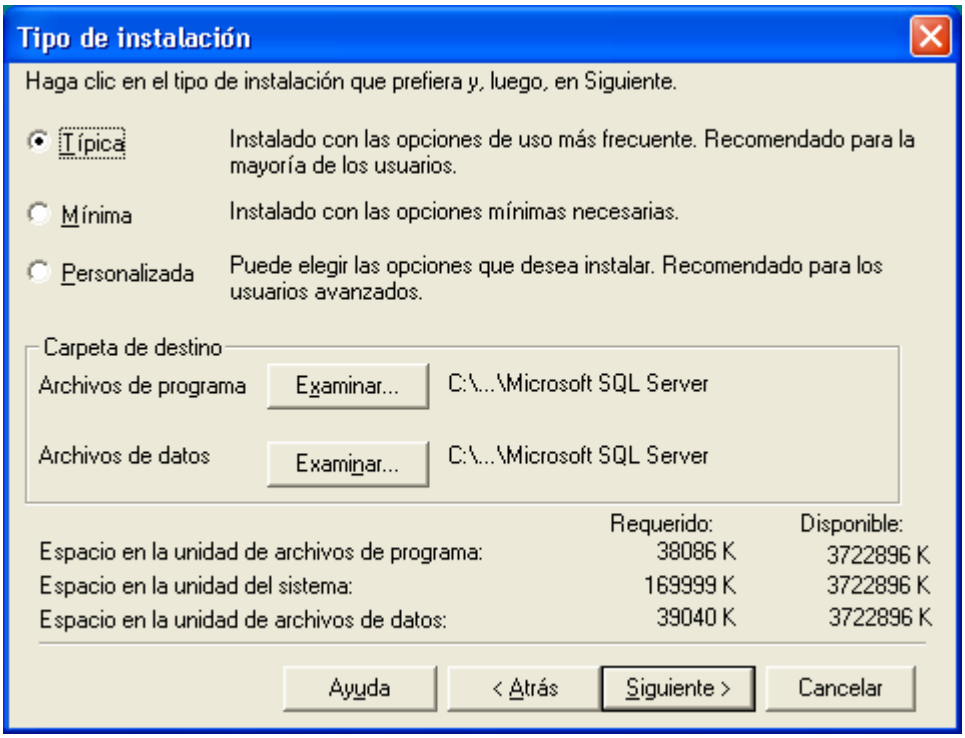
Se instala la herramienta cliente/servidor y se le da click en el botón siguiente.



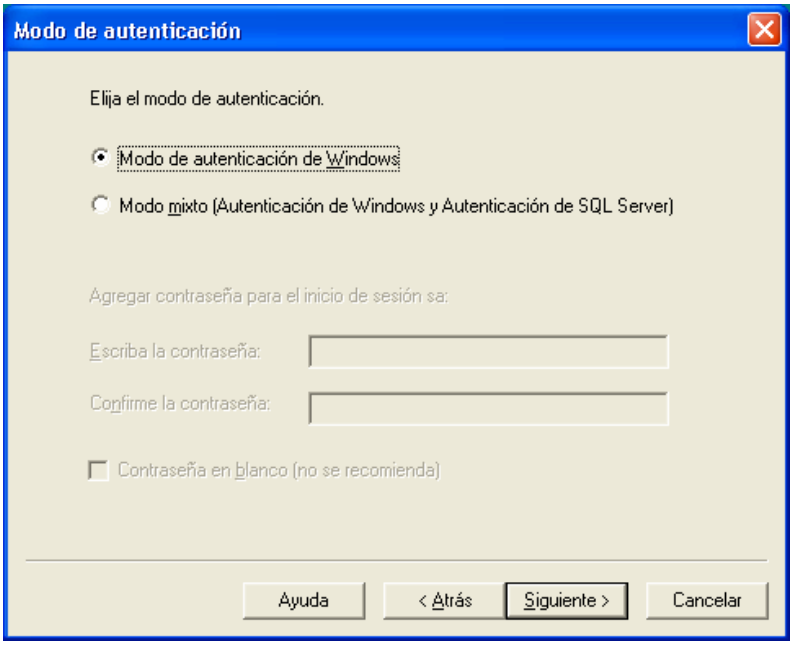
Se le da la configuración predeterminada y se da click en el botón siguiente.



Se elige la instalación típica y se da click en el botón siguiente.

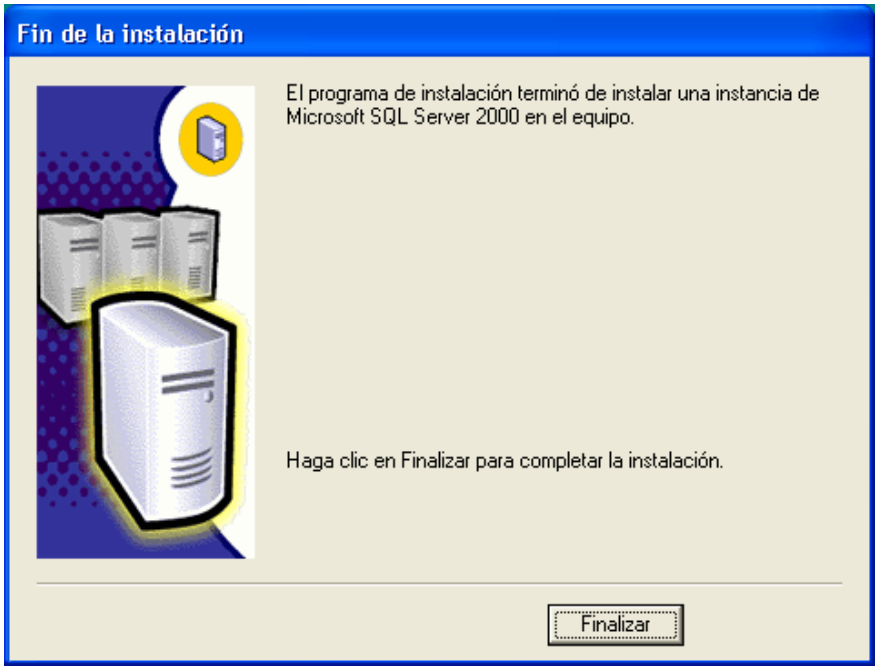


Se elige el modo de autenticación de Windows y se le da click al botón siguiente.



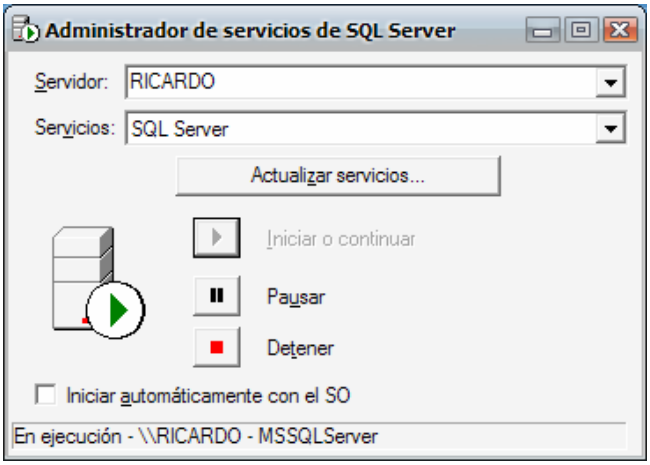
Se inicia la copia de archivos al PC presionando el botón siguiente.

Al final del proceso de instalación una ventana nos muestra si fue exitoso o no.

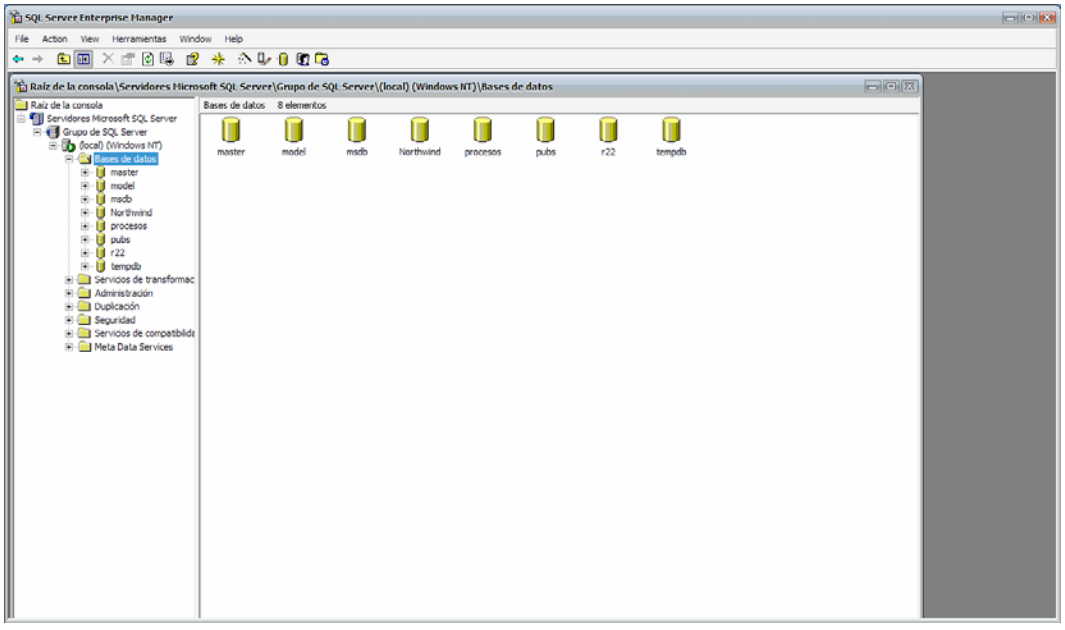


14.2.3. Configuración SQL Server

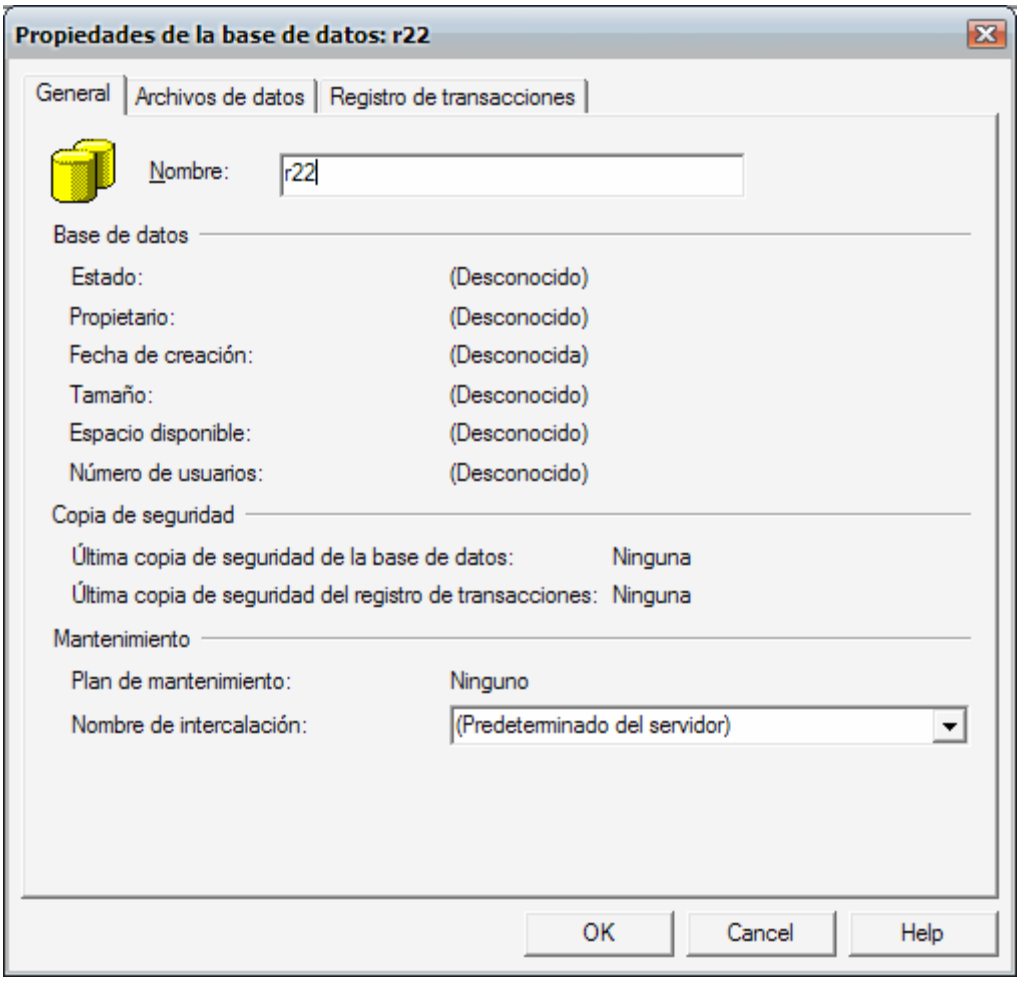
Para realizar la configuración de la base de datos se debe tener en cuenta si el servicio de SQL Server esta funcionando correctamente, para ello se debe revisar si este servicio esta corriendo en el PC, se ingresa al administrador del servicio que se encuentra en inicio/ todos los programas/ Microsoft SQL Server/administrador de servicio SQL Server y debe aparecer como se muestra en la grafica.



Ya comprobado que el servicio esta corriendo se debe ingresar en el administrador de SQL Server.



Se procede entonces a crear la base de datos para el funcionamiento del proyecto, dándole clic izquierdo en la carpeta bases de datos y pulsando la opción crear nueva base de datos, a continuación nos aparece una ventana para la creación de la base de datos.



Se debe poner el no nombre de la base de datos r22 como se muestra en la grafica anterior, para finalizar el proceso se pulsa el botón OK.

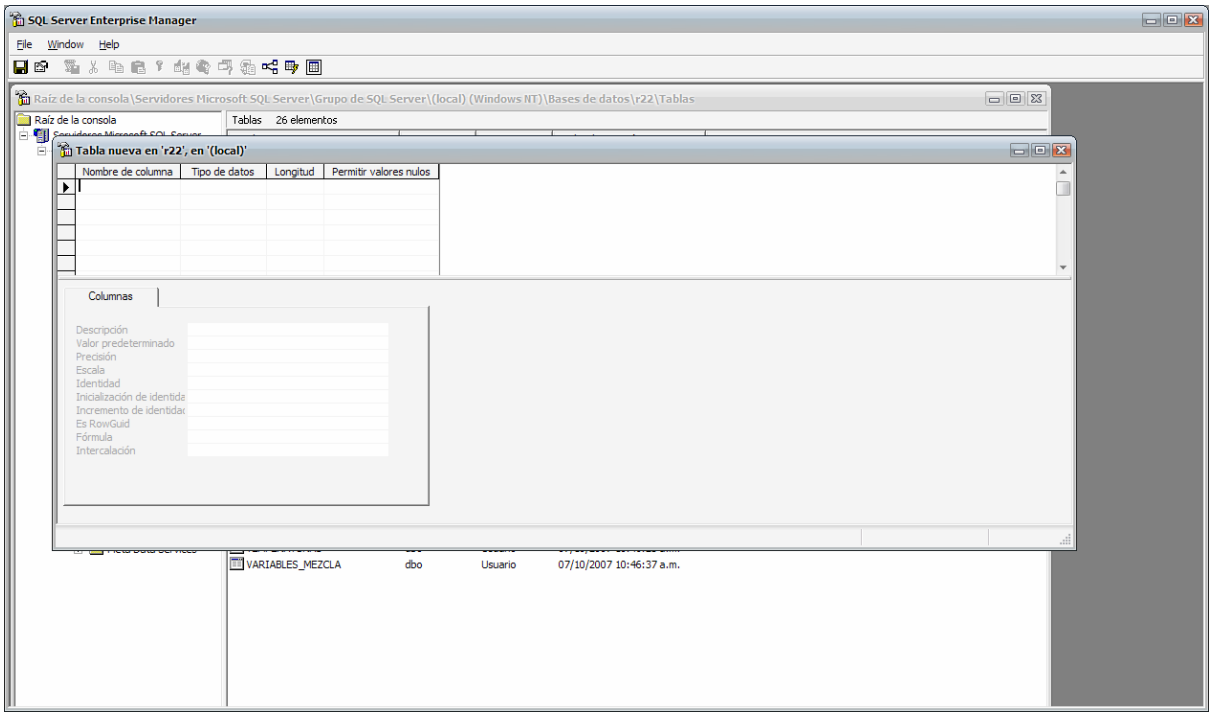
Se deben crear las siguientes tablas en la base de datos:

VARIABLE_MEZCLA
FALLAS_PROCESO
Falla1_carga_poliol
Falla2_carga_poliol

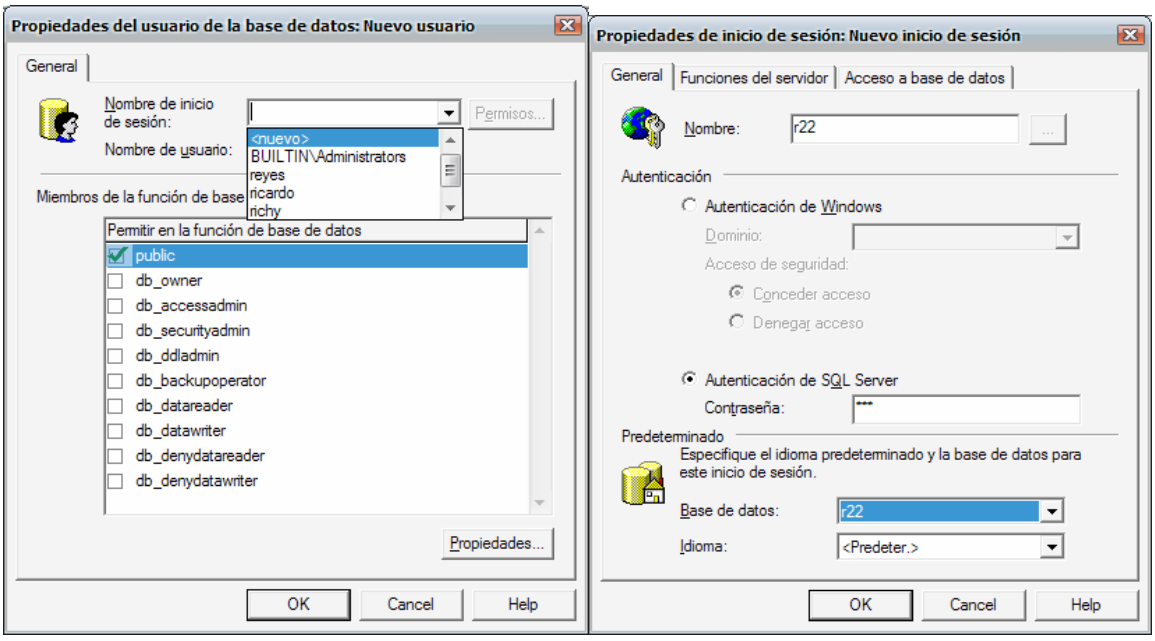
Falla_despresurizacion_TK_mezcla
Falla_presurizacion_TK_mezcla
Falla_presion_TK_mezcla
Falla_sobre_carga_R22
Tanque_Dia_Grande_Vacio
Tanque_Dia_Pequeno_Vacio
NIVELES_TANQUES
PESOS_TANQUES
TEMPERATURAS
SENALES_DIGITALES

DIAGRAMA DE CLASES CON SUS RESPECTIVOS CAMPOS Y LLAVES

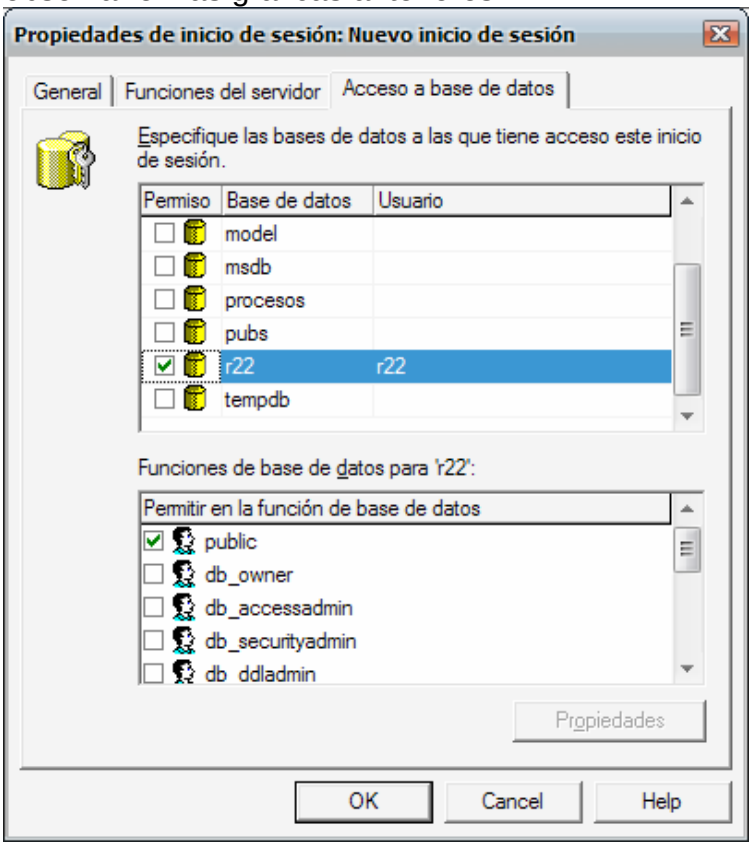
Se debe ir primero al administrador de SQL Server, luego se debe entrar en el menú con el nombre “TABLAS”, luego se da click izquierdo y se elige la opción crear nueva tabla como se muestra en siguiente figura:



Ya creadas todas las tablas se debe ingresar dentro del administrador de SQL Server en la opción usuarios, se debe dar click izquierdo y elegir la opción crear nuevo usuario en la base de datos.

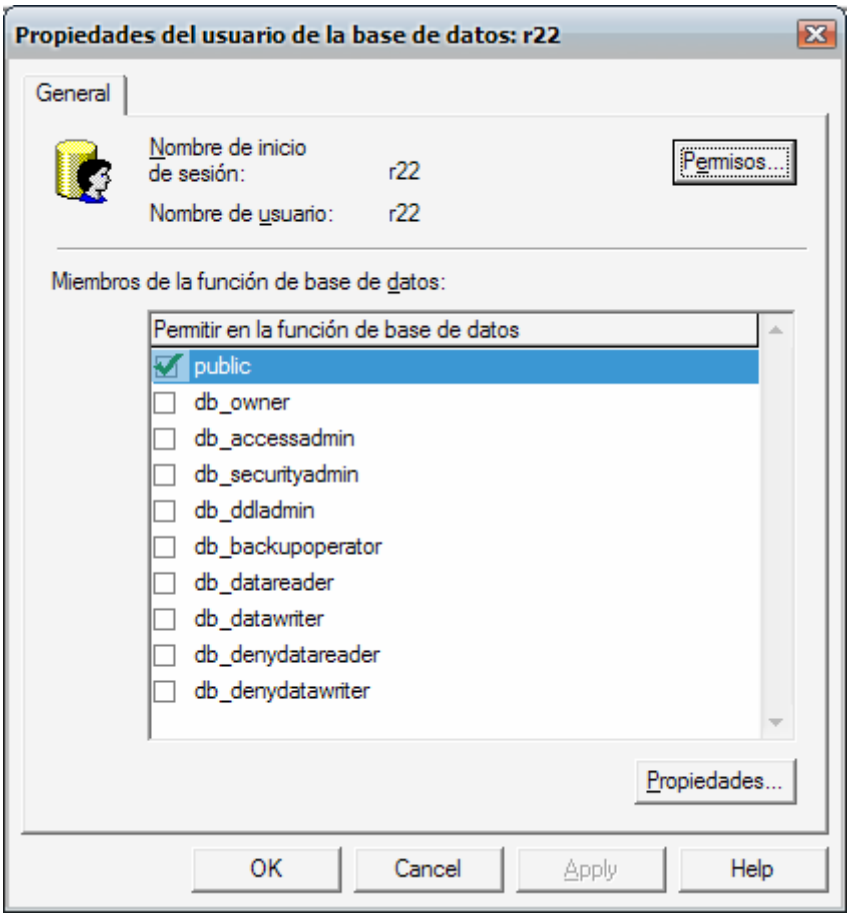


Se elige una sesion nueva, con el nombre de r22 y password r22 como se puede observar en las graficas anteriores.

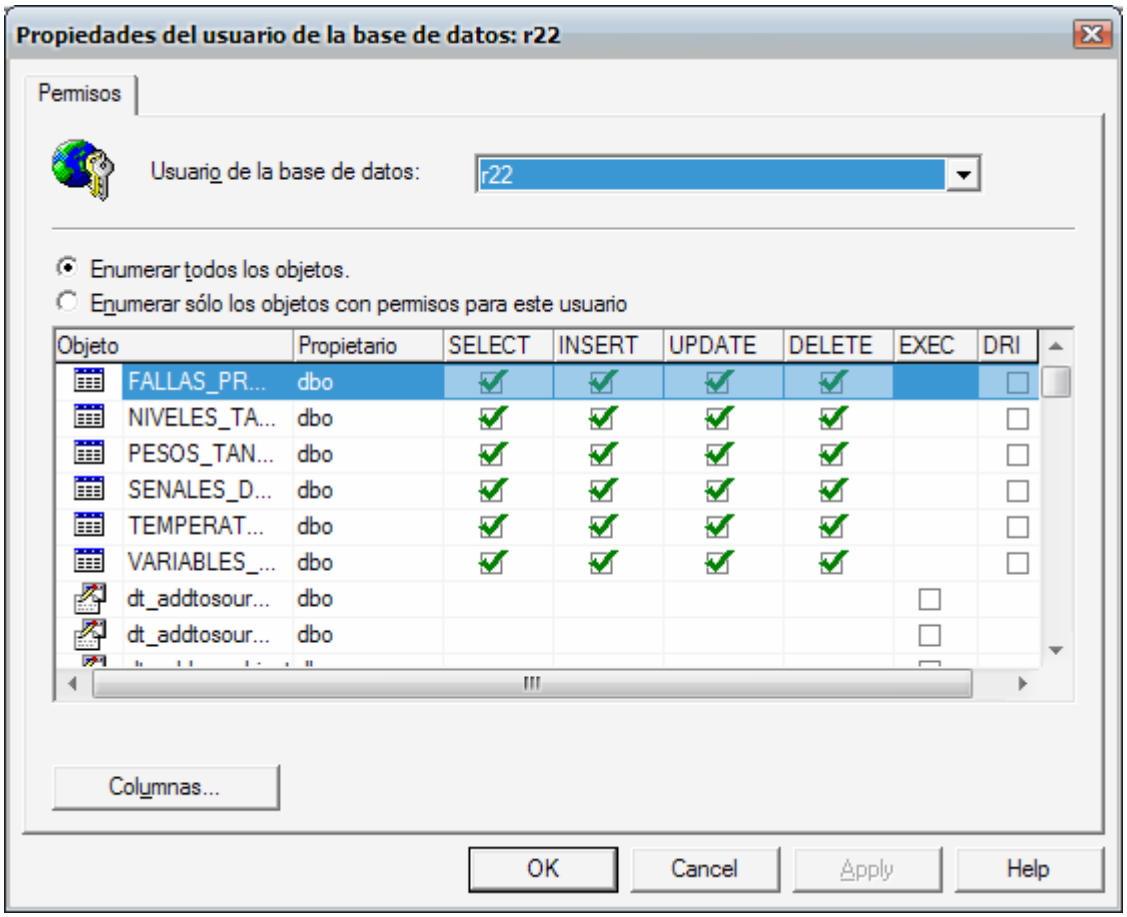


Luego de se debe ir a la pestaña acceso a base de datos y se debe elegir la base de datos r22 como se muestra en la grafica anterior.

Ya creado el usuario en la base de datos procedemos a darle permisos para poder interactuar con la base de datos, se debe ir al administrador de SQL Server en la opción usuarios y luego dar click izquierdo sobre el usuario r22, elegir la opción propiedades.



Se da click en el botón permisos, y elegimos las tablas creadas anteriormente dándole permisos de SELECT, INSERT, UPDATE, DELETE, como se muestra en la siguiente grafica.



14.3 INSTALACION DE CONTROLADOR JDBC

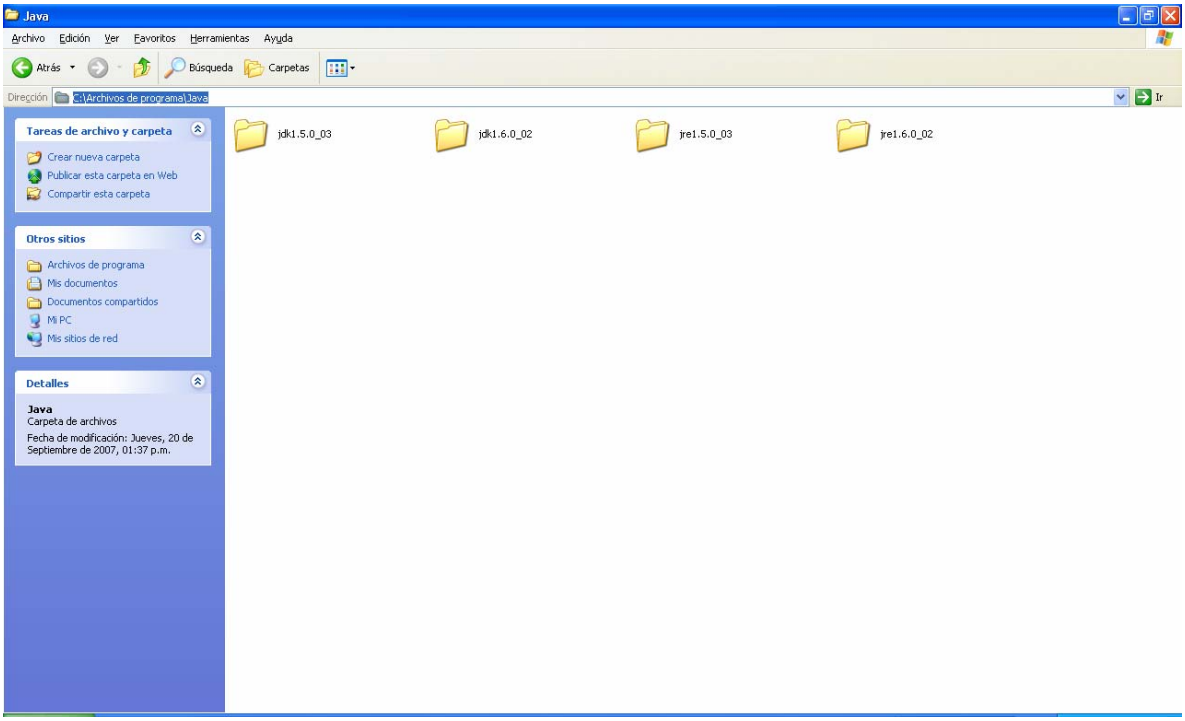
14.3.1 Descarga controlador JDBC

Para descargar este controlador se debe ingresar al siguiente link:

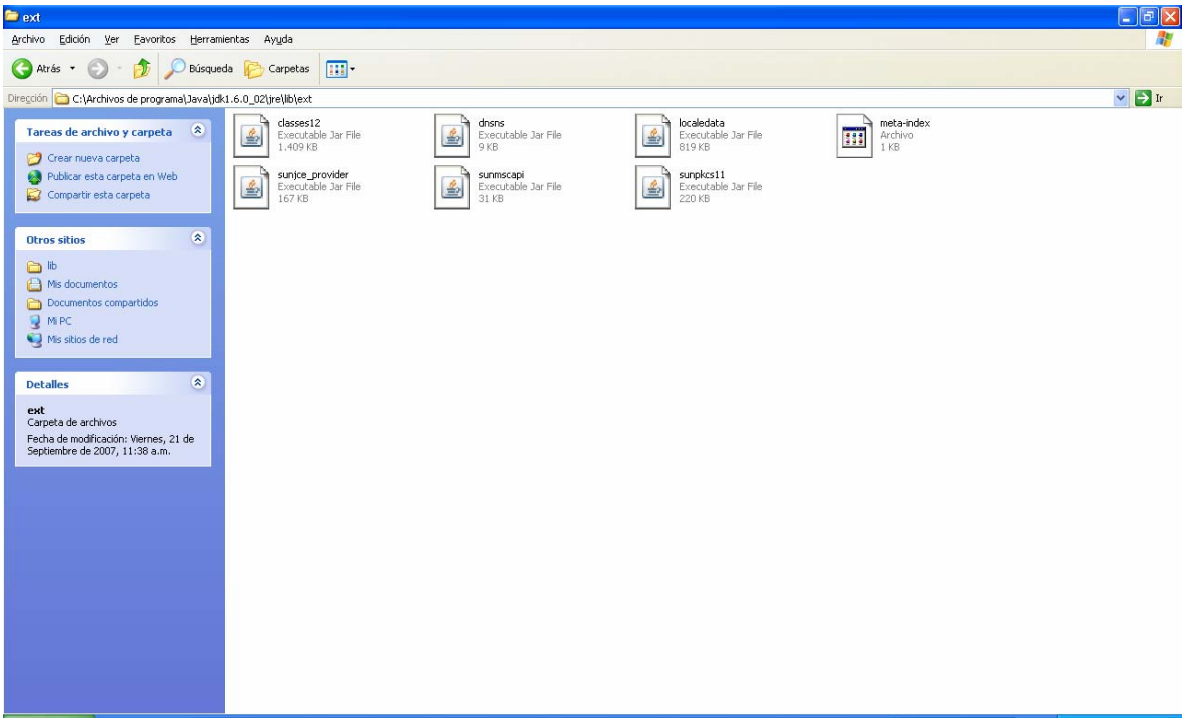
O a través del contenido del CD del proyecto

14.3.2. Instalación controlador JDBC

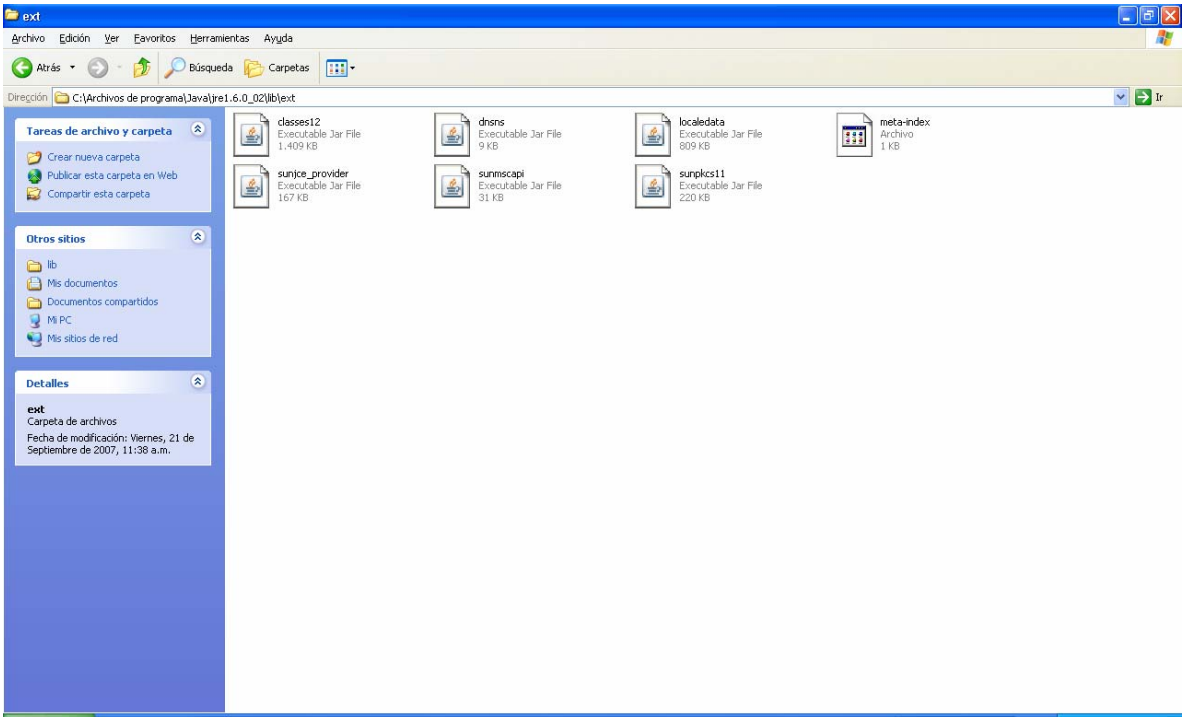
Se debe de ingresar al siguiente directorio: C:\Archivos de programa\Java



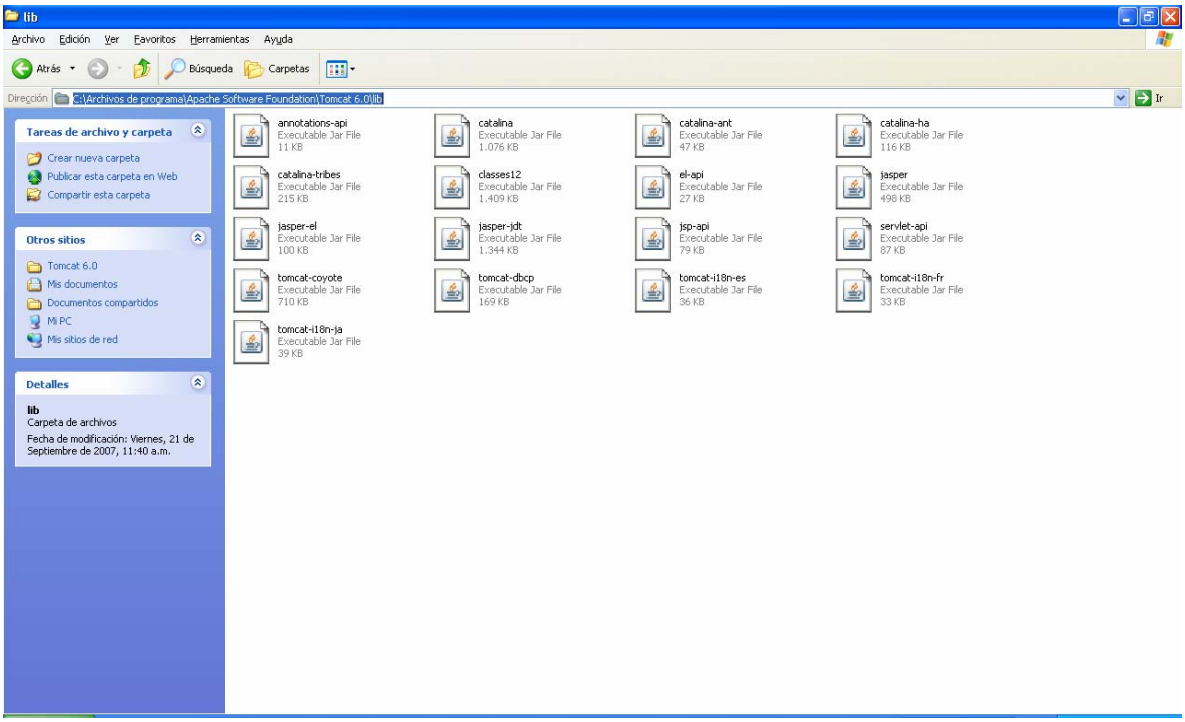
Copia el controlador JDBC que esta comprimido, se ingresa al directorio C:\Archivos de programa\Java\jdk1.6.0_02 y se pega en la siguiente ruta: C:\Archivos de programa\Java\jdk1.6.0_02\jre\lib\ext



Luego se realiza la misma operación para este directorio C:\Archivos de programa\Java\jre1.6.0_02\lib\ext



Para instalar el JDBC en el Tomcat se debe ingresar al siguiente directorio: C:\Archivos de programa\Apache Software Foundation\Tomcat 6.0\lib



Luego se debe copiar el JDBC y pegarlo para que los servlets de la aplicación móvil puedan funcionar adecuadamente.

14.4 INSTALACION SERVIDOR DE OPC

14.4.1 Descargar de servidor OPC

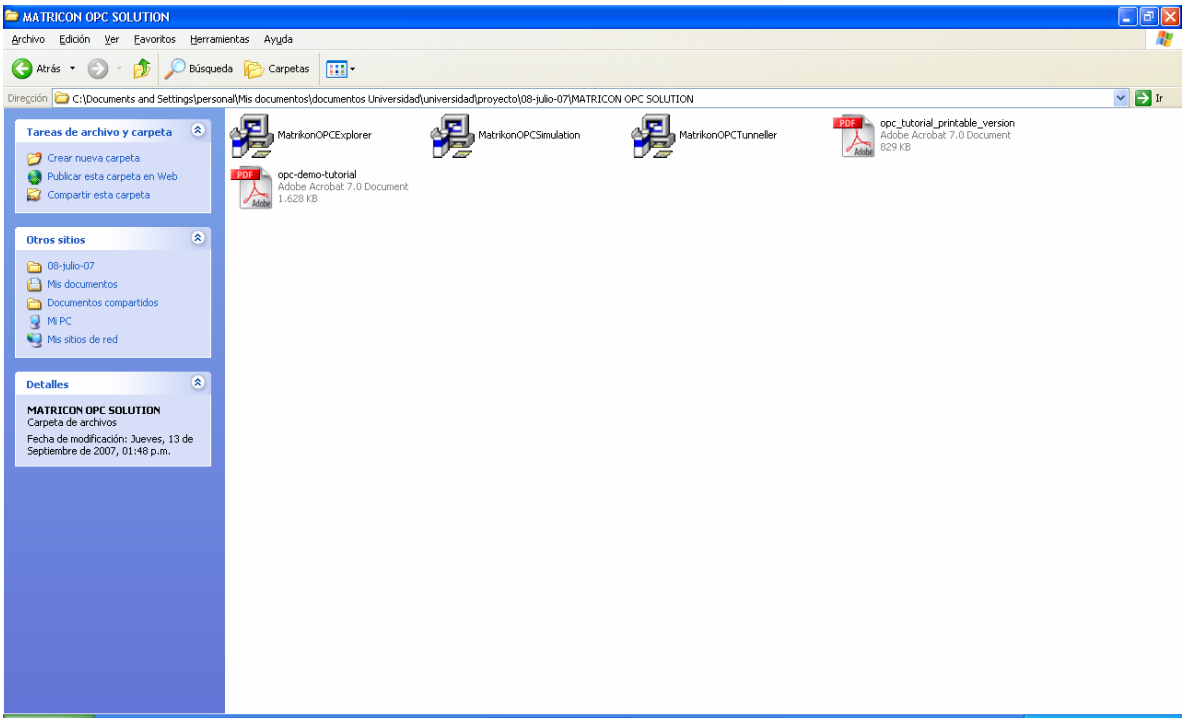
Para descargar el servidor OPC se debe ingresar al siguiente link: http://www.matrikonopc.com/downloads/178/index.aspx

Este instalador está disponible en el CD del proyecto.

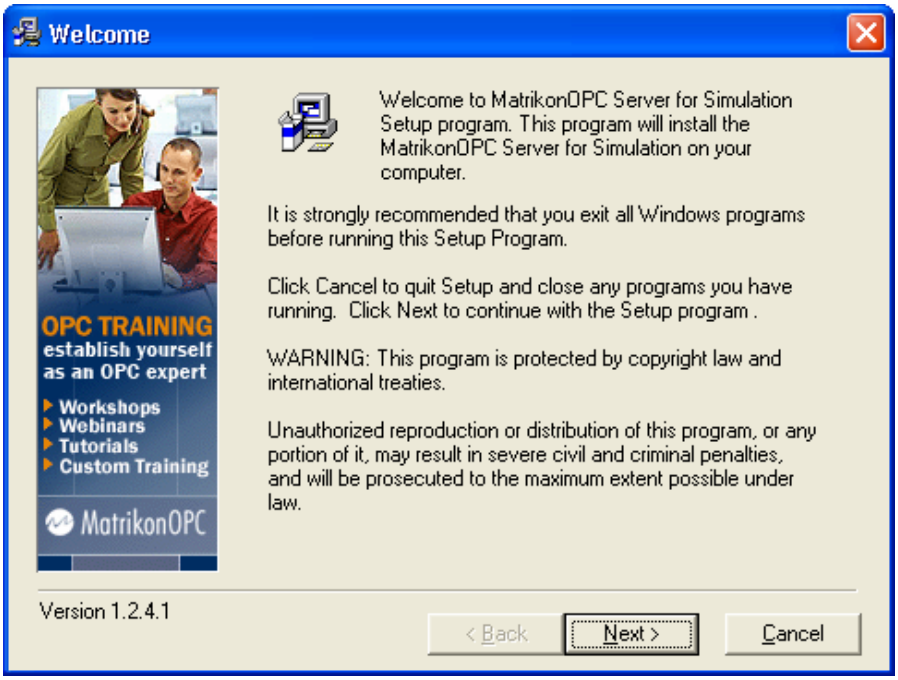
14.4.2 Instalación del servidor OPC

Procedemos a revisar el CD donde se encuentra los instaladores del servidor OPC, para realizar la instalación se deben tener los siguientes archivos:

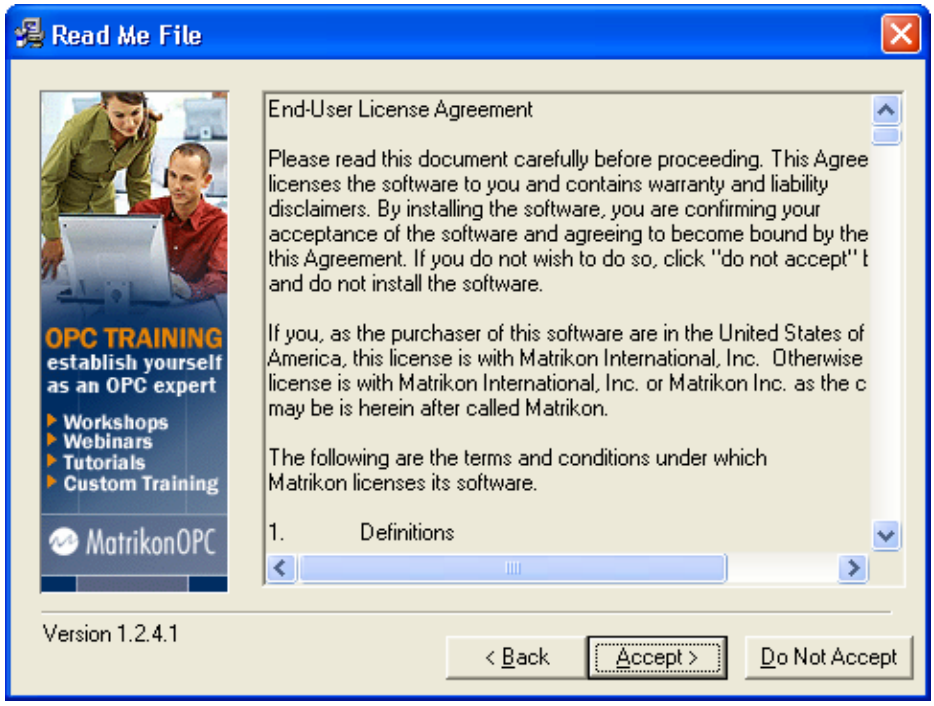
MatrikonOPCSimulation, MatrikonOPCExplorer



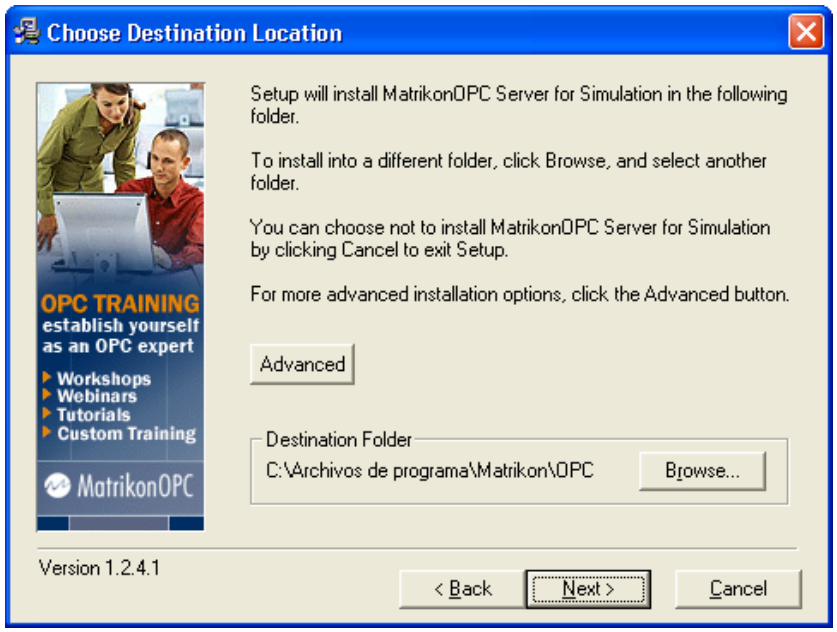
Primero se debe instalar el MatrikonOPCSimulation, se ejecuta y aparece una ventana asistente para la instalación del producto.



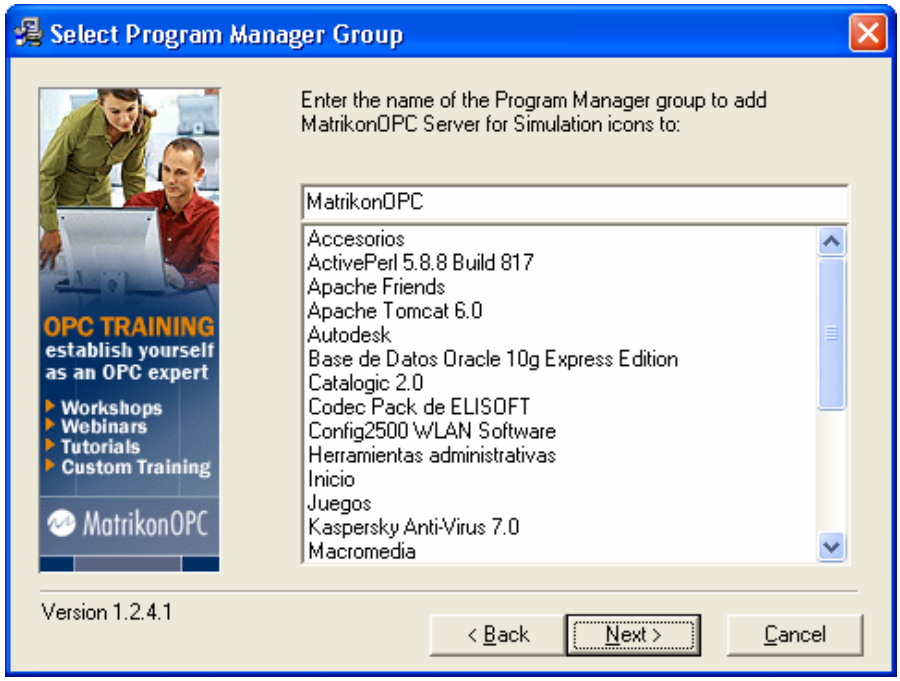
Se pulsa en el botón siguiente y aceptamos la licencia de uso del producto.



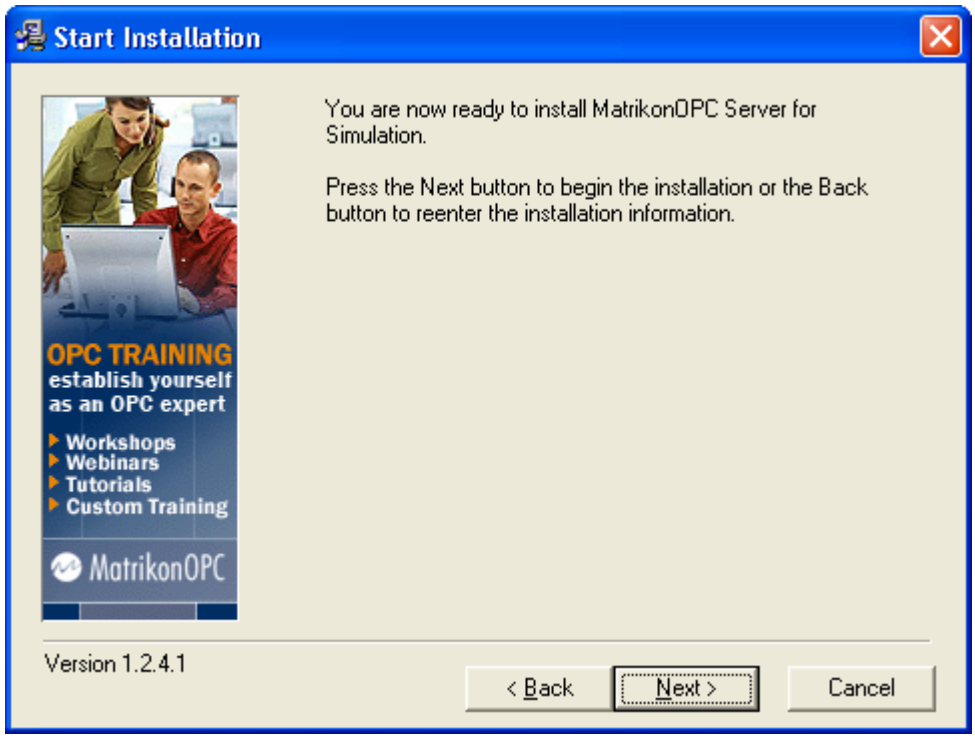
Se elige la ruta donde se instala el servidor OPC.



Se da clic en el botón siguiente.



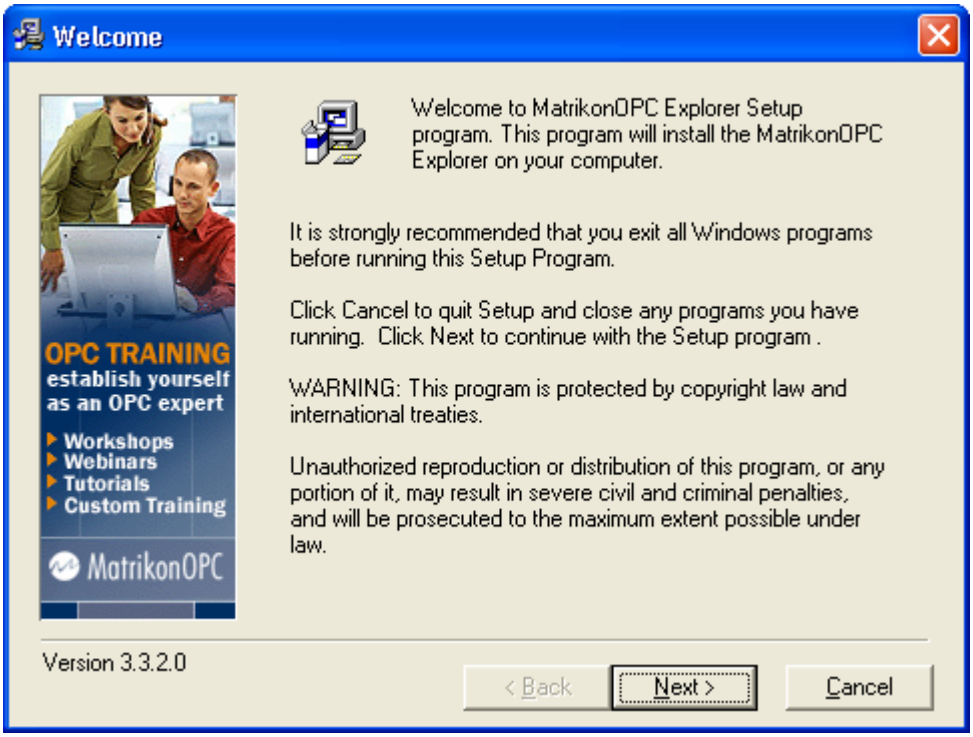
Se inicia el proceso de instalación del servidor.



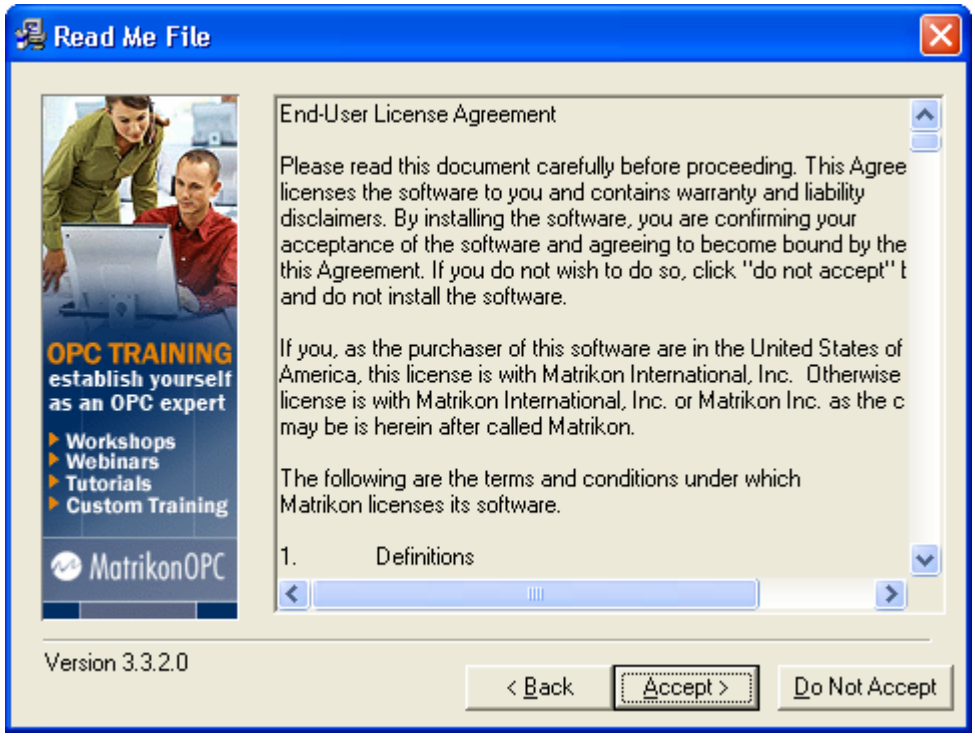
Luego de instalar el servidor OPC se debe instalar MatrikonOPCExplorer para poder administrar y configurar el servidor OPC.

Se debe ejecutar el instalador de MatrikonOPCExplorer, este muestra un ventana de asistencia para la instalación del OPC Explorer.

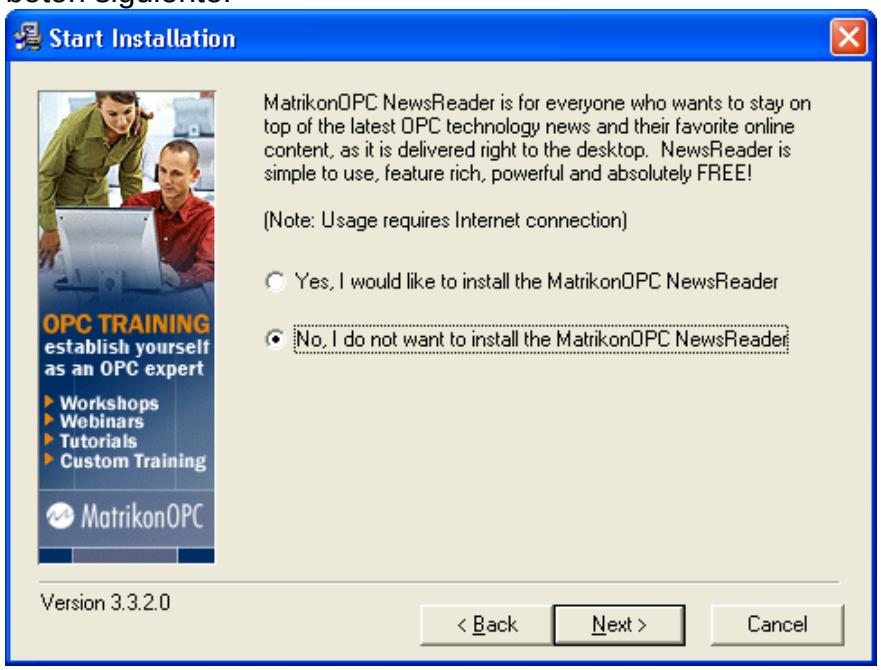
Se da click al botón siguiente.



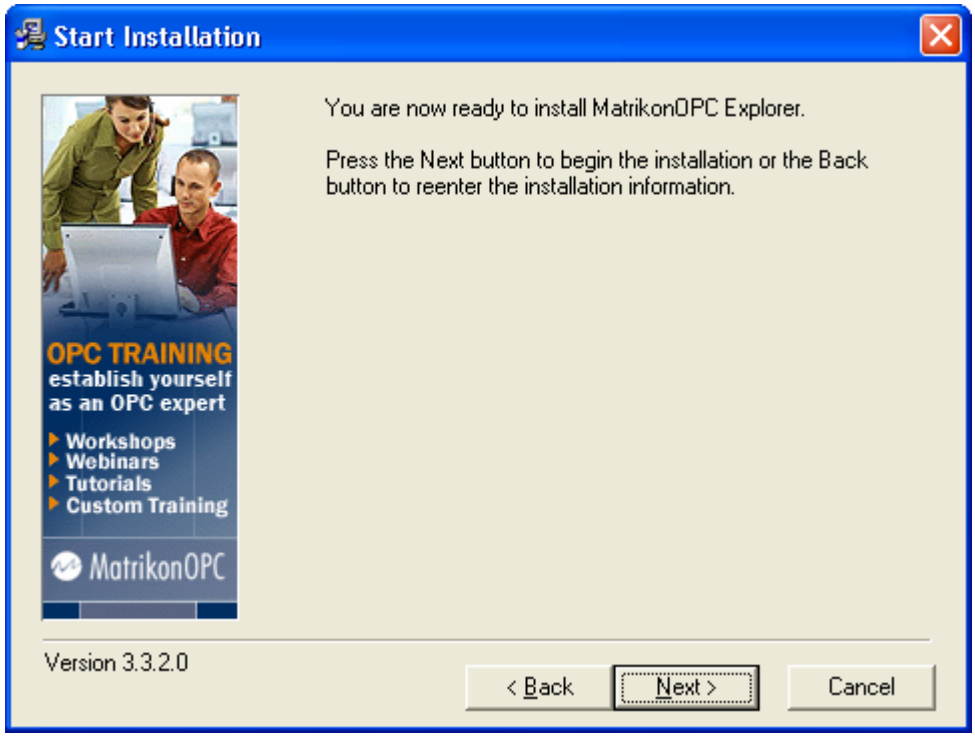
Se acepta la licencia de uso del OPC Explorer.



Se selecciona la opción que se muestra en la siguiente grafica y se da click al botón siguiente.



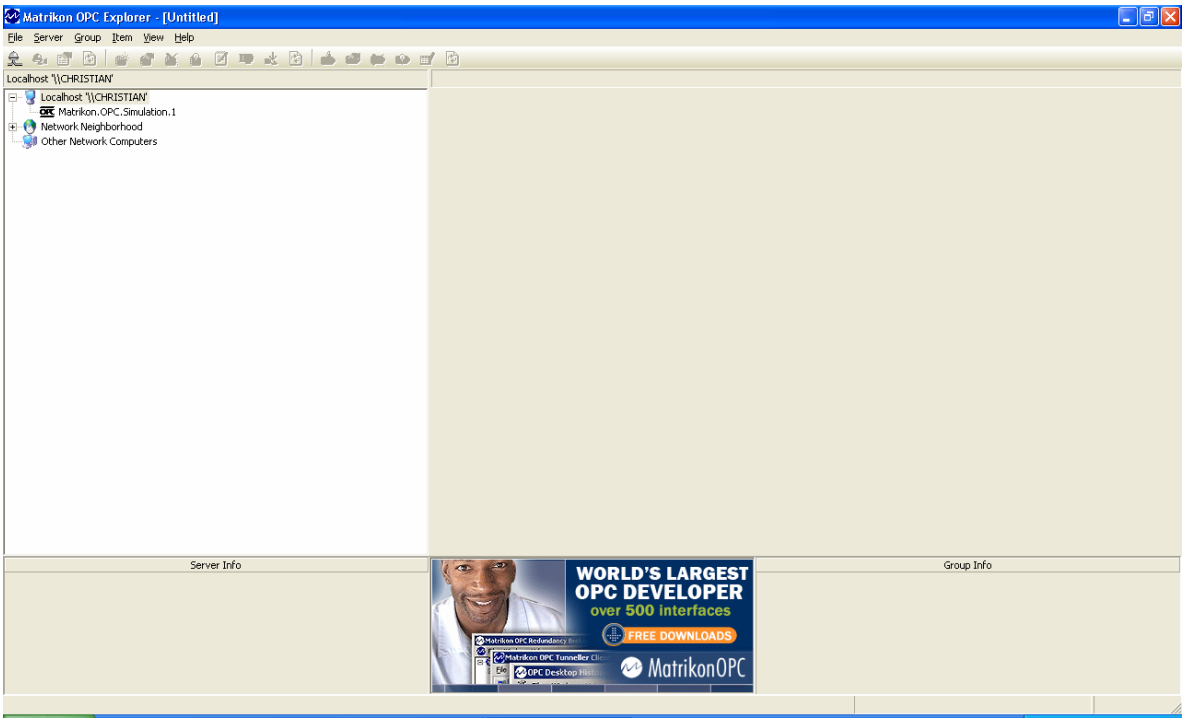
Se da click en el botón siguiente para iniciar el proceso de instalación.



14.4.3 Configuración Servidor OPC

Se debe crear los tags que manejan la información de los PLCs, esto se realiza a través del OPC Explorer.

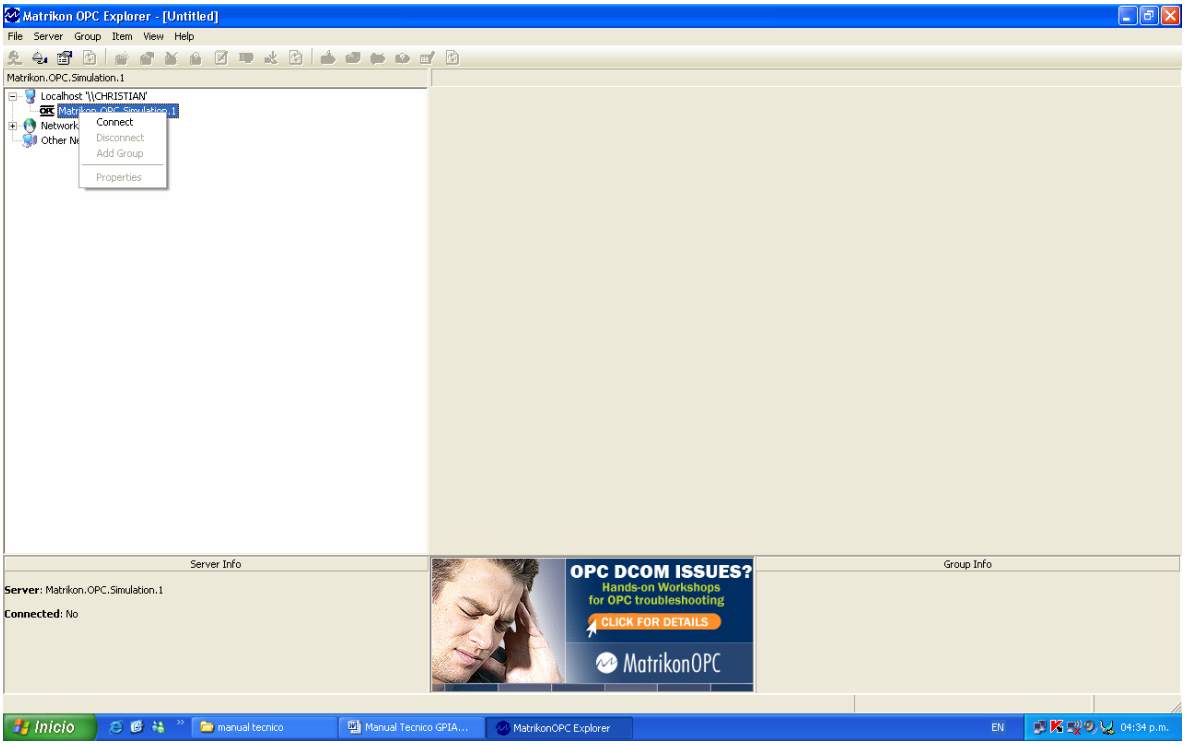
Se ejecuta y aparece la siguiente interfaz.



El proceso de configuración es realizador por un xml que se encuentra en la carpeta del CD del proyecto, este contiene los tags con sus respectivos grupos e ítems, este se carga a través de el menú file/open y se elige el archivo xml con el nombre de proceso_mezcla_R22.xml al cargarlo se inicia el servidor automáticamente, sin embargo en este manual se mostrara como crean los grupos e ítems para el funcionamiento del servidor OPC.

Los pasos son los siguientes:

Se da click derecho sobre el servidor OPC y se elige la opción conectar.



La estructura que se debe crear en el servidor es la siguiente:

Se deben crear los siguientes grupos

PESOS_TANQUES
TEMPERATURAS
NIVELES_TANQUES
FALLAS_PROCESO
SENALES_DIGITALES
VARIABLES_MEZCLA

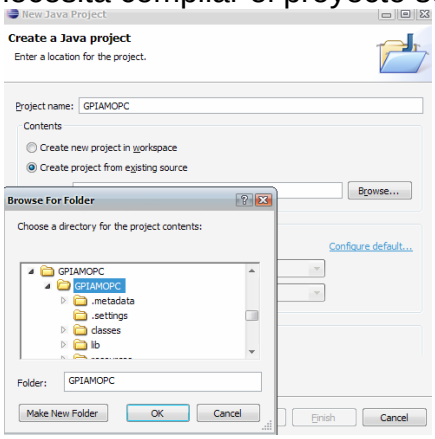
Los ítems de cada grupo son:

PESOS_TANQUES

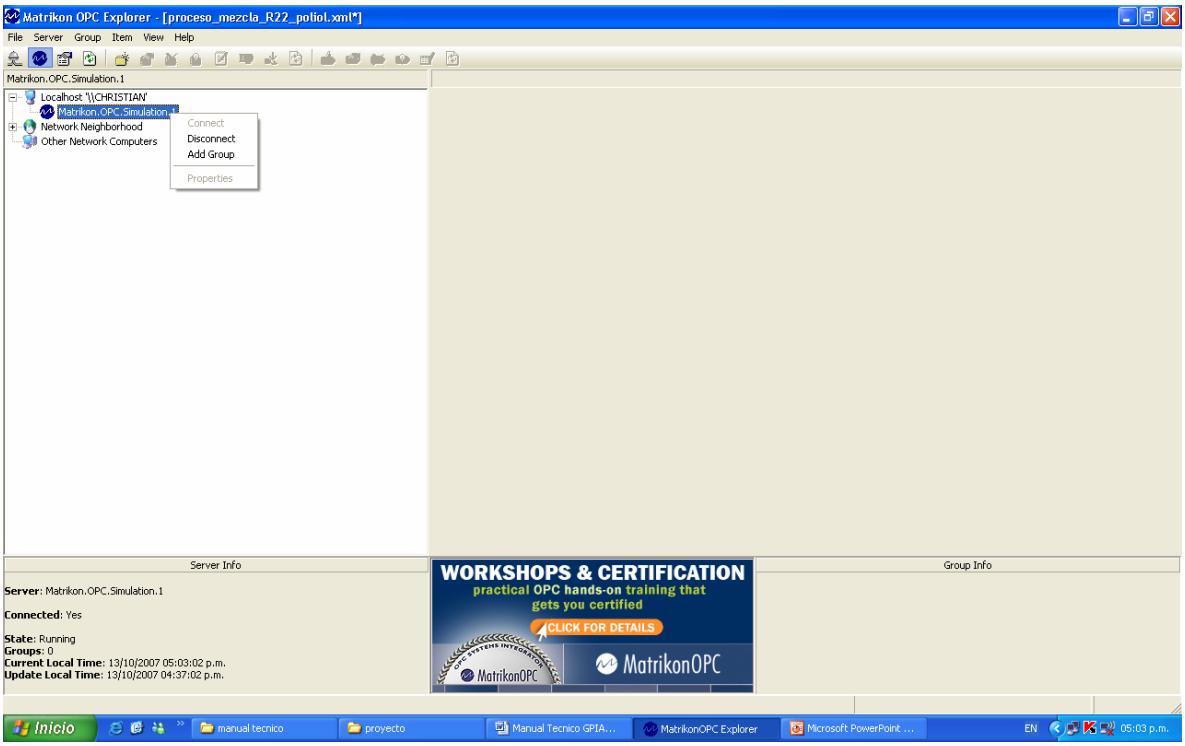
Ítems:	PESO_TANQUE_MEZCLA	single float
	PESO_TANQUE_R22	single float
	PESO_TANQUE_ISOCIANATO	single float

	PESO_TANQUE_POLIOL	single float
TEMPERATURAS		
Ítems:	Temp_TK_Mix	single float
	Temp_TK_Dia_Grande	single float
	Temp_TK_Dia_Pequeno	single float
NIVELES_TANQUES		
Ítems:	MINIMO_TK_MEZCLA	single float
	R22_VACIO	single float
	NIVEL_BAJO_TK_R22	single float
FALLAS_PROCESO		
Ítems:	FALLA_PROCESO	Integer
	Falla1_carga_poliol	Integer
	Falla2_carga_poliol	Integer
	Falla_despresurizacion_TK_mezcla	Integer
	Falla_presurizacion_TK_mezcla	Integer
	Falla_presion_linea_aire	Integer
	Falla_sobre_carga_R22	Integer
	Tanque_Dia_Grande_Vacio	Integer
	Tanque_Dia_Pequeno_Vacio	Integer
SENALES_DIGITALES		
Ítems:	Proceso_mezcla	Integer
VARIABLES_MEZCLA		
Ítems:	Bloqueo_mezcla	Integer

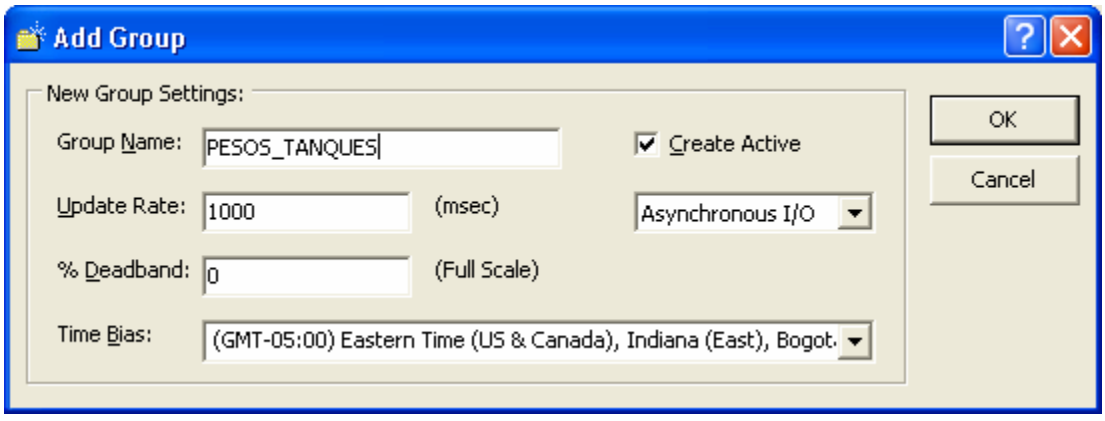
Antes de crear los grupos y los ítems procedemos a instalar el eclipse o descargarlo en la página oficial www.eclipse.org luego instalado se ejecuta el eclipse y se remite a la siguiente instrucción:
Archivo – nuevo – proyecto java y allí se escribe el nombre del proyecto y se debe escoger la opción crear un proyecto desde una fuente existente y buscamos la carpeta GPIAMOPC que se encuentra en el CD. Luego de esto se procede a ejecutar la aplicación (no necesita compilar el proyecto solo se debe ejecutar).



Para crear un grupo en el servidor se debe hacer click derecho y se elige la opción add group como se muestra en la grafica.

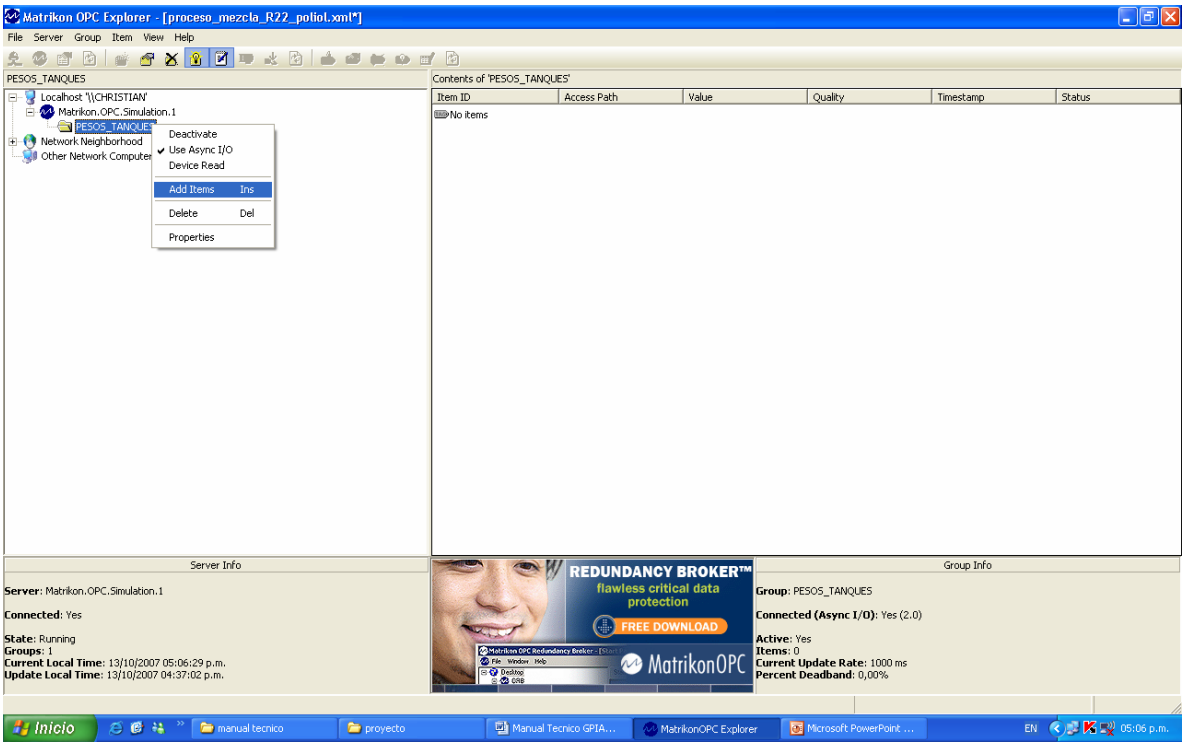


Se debe ingresar el nombre del tag, se introduce el primer grupo.

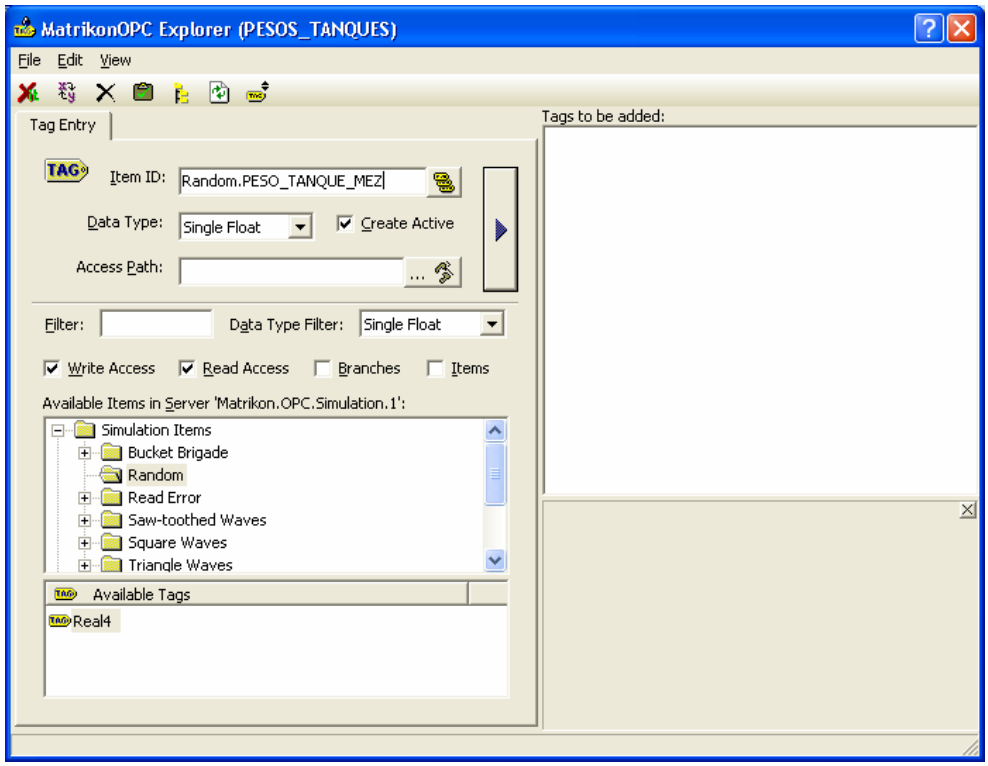


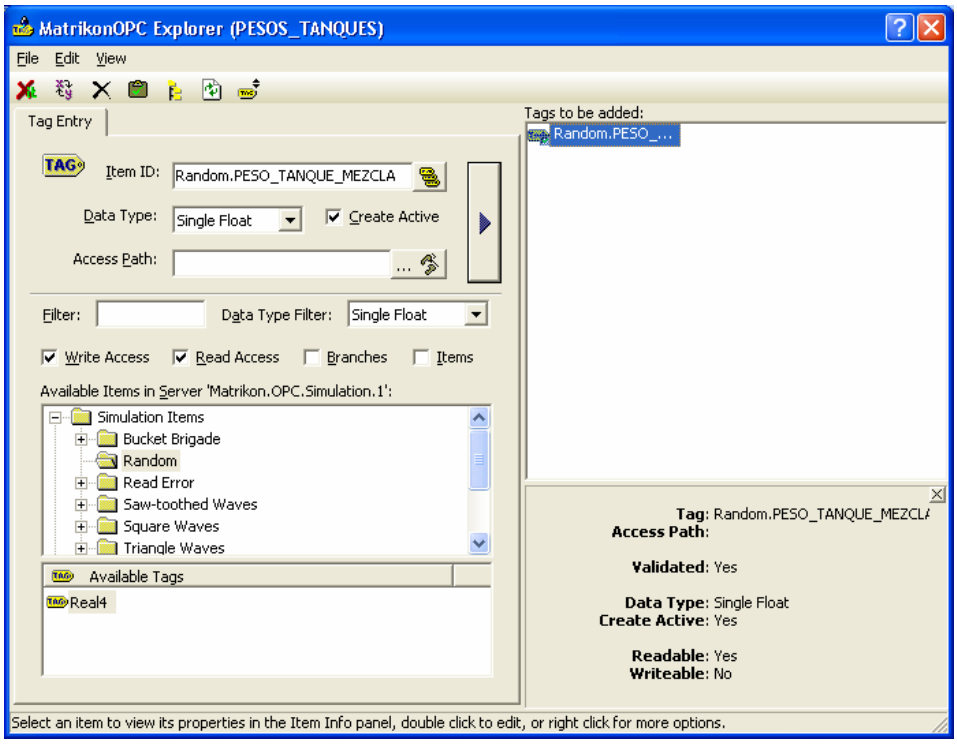
Se da click al botón OK, ya creado el grupo procedemos a crear los ítems del grupo PESOS_TANQUES.

Para crear los ítems se debe dar click derecho sobre la carpeta del grupo y se elige la opción add ítems.



Al crear los ítems se debe elegir el tipo de dato en el data type y en el campo data type filter luego se elige el ítem de la carpeta simulation ítems luego en la carpeta ramdom se le da doble click al ítem disponible que se muestra en la parte inferior de la ventana y se pone el nombre del tag después de la palabra “ramdom.” Este proceso se muestra en la siguiente ventana.


MatrikonOPC Explorer (PESOS_TANQUES)
File Edit View
Tag Entry
Item ID: Random.PESO_TANQUE_MEZ
Data Type: Single Float
Create Active
Access Path:
Filter:
Data Type Filter: Single Float
Write Access
Read Access
Branches
Items
Available Items in Server 'Matrikon.OPC.Simulation.1':
Simulation Items
Bucket Brigade
Random
Read Error
Saw-toothed Waves
Square Waves
Triangle Waves
Available Tags
Real4
Tags to be added:

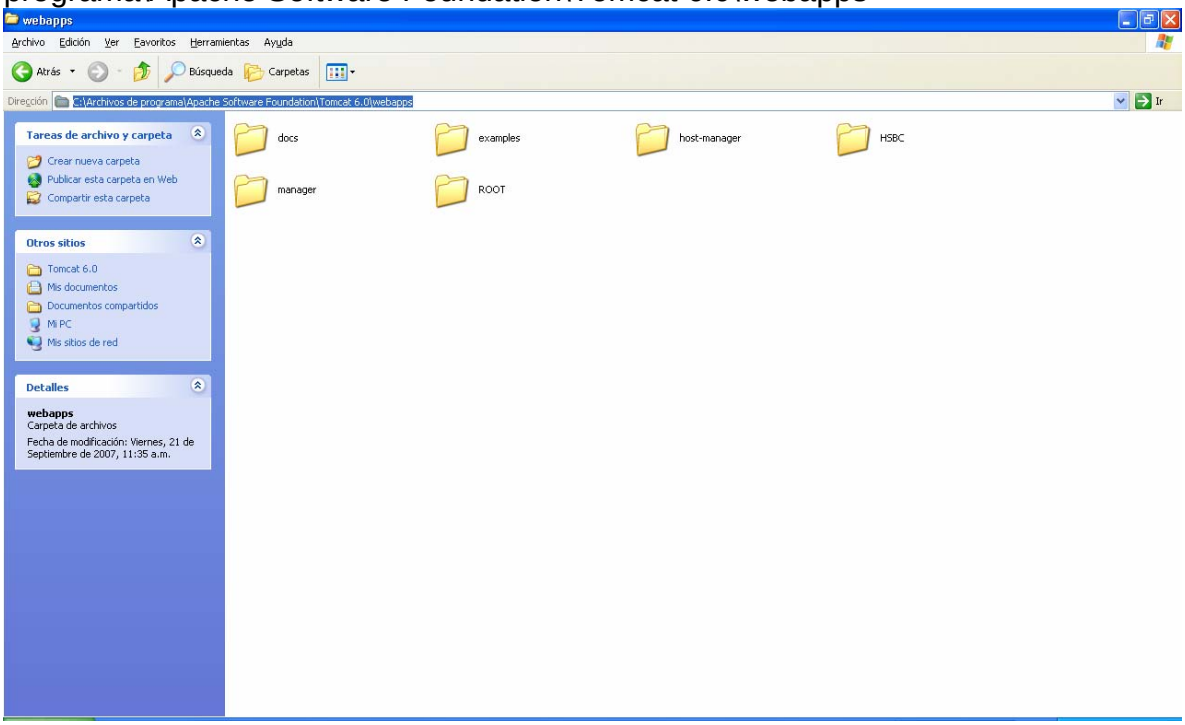

MatrikonOPC Explorer (PESOS_TANQUES)
File Edit View
Tag Entry
Item ID: Random.PESO_TANQUE_MEZCLA
Data Type: Single Float
Create Active
Access Path:
Filter:
Data Type Filter: Single Float
Write Access
Read Access
Branches
Items
Available Items in Server 'Matrikon.OPC.Simulation.1':
Simulation Items
Bucket Brigade
Random
Read Error
Saw-toothed Waves
Square Waves
Triangle Waves
Available Tags
Real4
Tags to be added:
Random.PESO_...
Tag: Random.PESO_TANQUE_MEZCLA
Access Path:
Validated: Yes
Data Type: Single Float
Create Active: Yes
Readable: Yes
Writeable: No
Select an item to view its properties in the Item Info panel, double click to edit, or right click for more options.

14.5 SUBIR ARCHIVOS

14.5.1 Subir archivos de aplicación móvil

Para que la aplicación móvil funcione de forma adecuada se debe instalar los archivos que se encuentran en la carpeta GPI que contiene el CD del proyecto.

La carpeta GPI debe copiarse en el siguiente directorio: C:\Archivos de programa\Apache Software Foundation\Tomcat 6.0\webapps



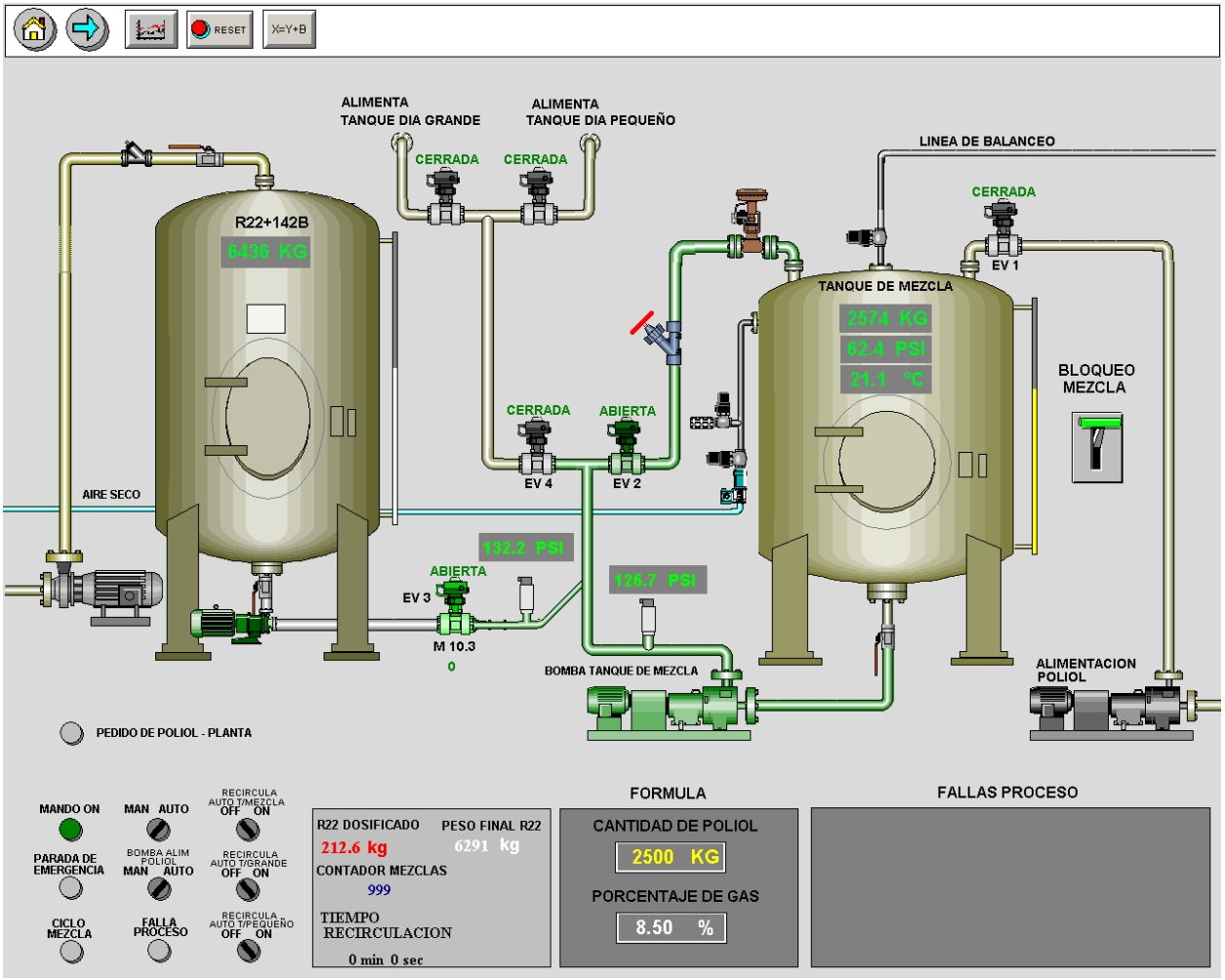
Nota: para poder realizar la ejecución de la aplicación móvil que se encuentra en la carpeta GPIAM debe instalar el paquete netbeans con el mobility pack que se encuentran en el CD o descargarlos de la página oficial http://www.netbeans.org/products/index.html y el J2ME Wireless tool kit que también está contenido en este o descargarlo de la página http://developers.sun.com/downloads/ para poder emular el proyecto.

Para poder ejecutar el proyecto en netbeans se debe abrir el proyecto con la opción archivo, abrir proyecto y vincularla con la carpeta GPIAM que contiene todas las fuentes de la aplicación móvil.

Si se desea subir la aplicación a un dispositivo móvil se debe tener en cuenta dos archivos que se encuentran en la carpeta dist del proyecto netbeans, los cuales son GPIAM.jad y GPIAM.jar

15. MANUAL DE USUARIO

GESTIÓN DE PROCESOS INDUSTRIALES A TRAVÉS DE MÓVILES
PROCESO DE MEZCLA (POLIOL + R22)



Para las pruebas que se realizan en este manual se utilizo el emulador por defecto de celulares proporcionado por NetBeans MobilityPack:

19.1 Pantalla principal

La opción que aparece es la de Launch para empezar con la aplicación. Dar clic en Launch

19.2 Menú principal

Luego aparece el menú principal de la aplicación móvil. R-22 el cual es el donde se contiene todos los procesos que se van a controlar

Menú R-22: Aquí se escoge el grupo que se quiera controlar.

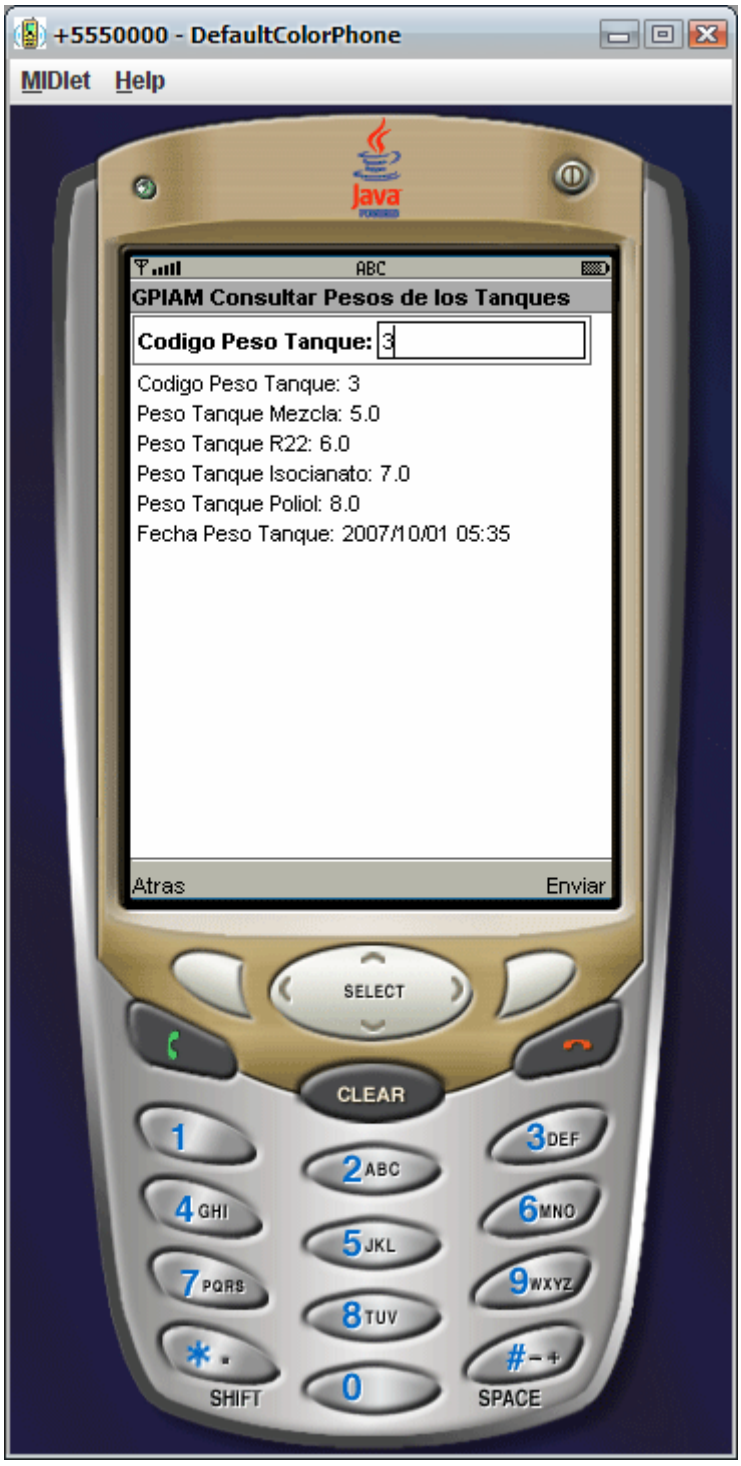
19.3 PESOS DE LOS TANQUES

En este menú se puede consultar un peso de un tanque o el último peso de los tanques registrado por el sistema.

19.3.1 Consultar un peso de los tanques

En esta pantalla se debe digitar el código del peso del tanque que se quiere consultar. Luego se da enviar.

Resultado de la consulta: Si se quiere consultar otro registro se digita de nuevo el nuevo código y se da enviar; si no, se escoge la opción atrás para regresar al menú principal.



19.3.2 Consultar ultimo peso de los tanques

Seleccionando esta opción se presentara en pantalla un mensaje de confirmación. Dar Ok

Pulsar Ok para continuar

Resultado de la consulta: Para volver al menú principal escoger la opción atrás.

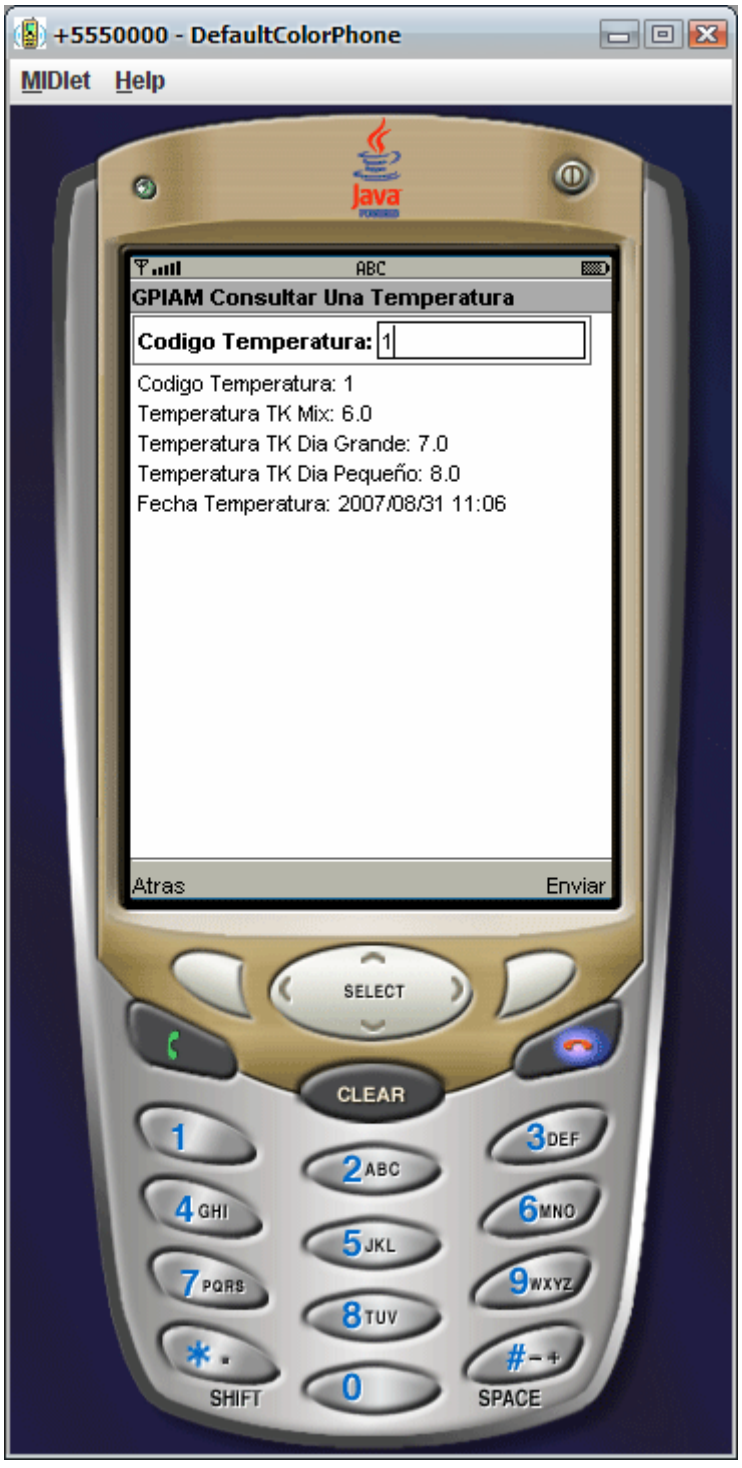
19.4 TEMPERATURAS

En este menú se puede consultar una temperatura o las ultimas temperaturas registradas por el sistema.

19.4.1 Consultar una temperatura

Se digita el código para la temperatura que se quiera consultar.

Resultado de la consulta: Si se quiere consultar otra temperatura se digita el codigo de la temperatura, se da la opcion de regresar al menu principal seleccionando atrás.



19.4.2 Consultar últimas temperaturas

Seleccionando esta opción se presentara en pantalla un mensaje de confirmación. Dar Ok

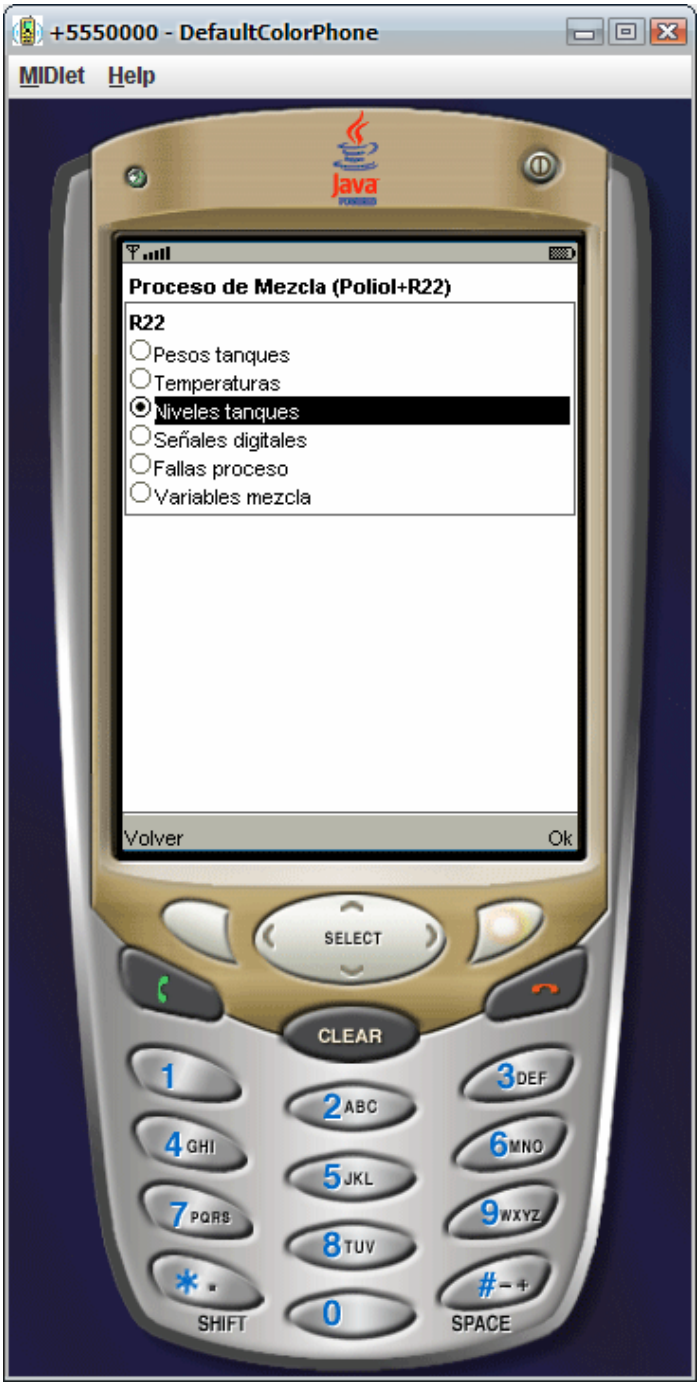
Pulse Ok para continuar


+5550000 - DefaultColorPhone
MIDlet
Help
Java
GPIAM Temperaturas
Pulse Ok para continuar
Atras
Ok
SELECT
CLEAR
1
2 ABC
3 DEF
4 GHI
5 JKL
6 MNO
7 PQRS
8 TUV
9 WXYZ
* .
SHIFT
0
- +
SPACE

Resultado de la consulta

15.5 NIVELES DE LOS TANQUES

En este menú se puede consultar un nivel de los tanques o las últimos niveles registrados por el sistema



15.5.1 Consultar niveles de los tanques

Escoger la opción uno y luego en el menú desplegable Ok

Luego digitar el código del nivel del tanque a consultar

Resultado de la consulta

15.5.2 Consultar el último nivel de los tanques

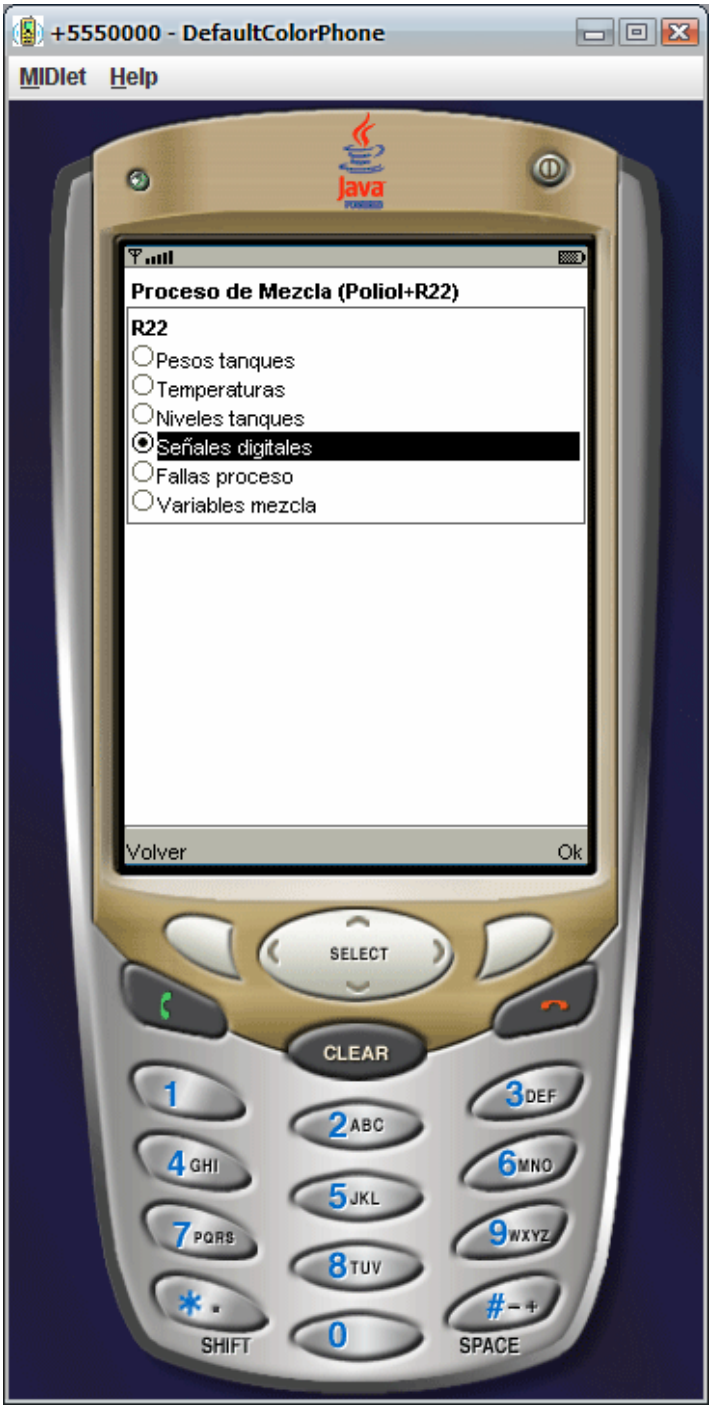
Con esta opción se observara el último nivel de los tanques en el sistema.

Pulsar Ok para continuar

Resultado de la consulta: para regresar al menú principal pulsar atrás.

15.6 SEÑALES DIGITALES

Es este menú se podrá consultar una señal digital o la última señal digital registrada por el sistema. Pulsamos Ok para continuar.

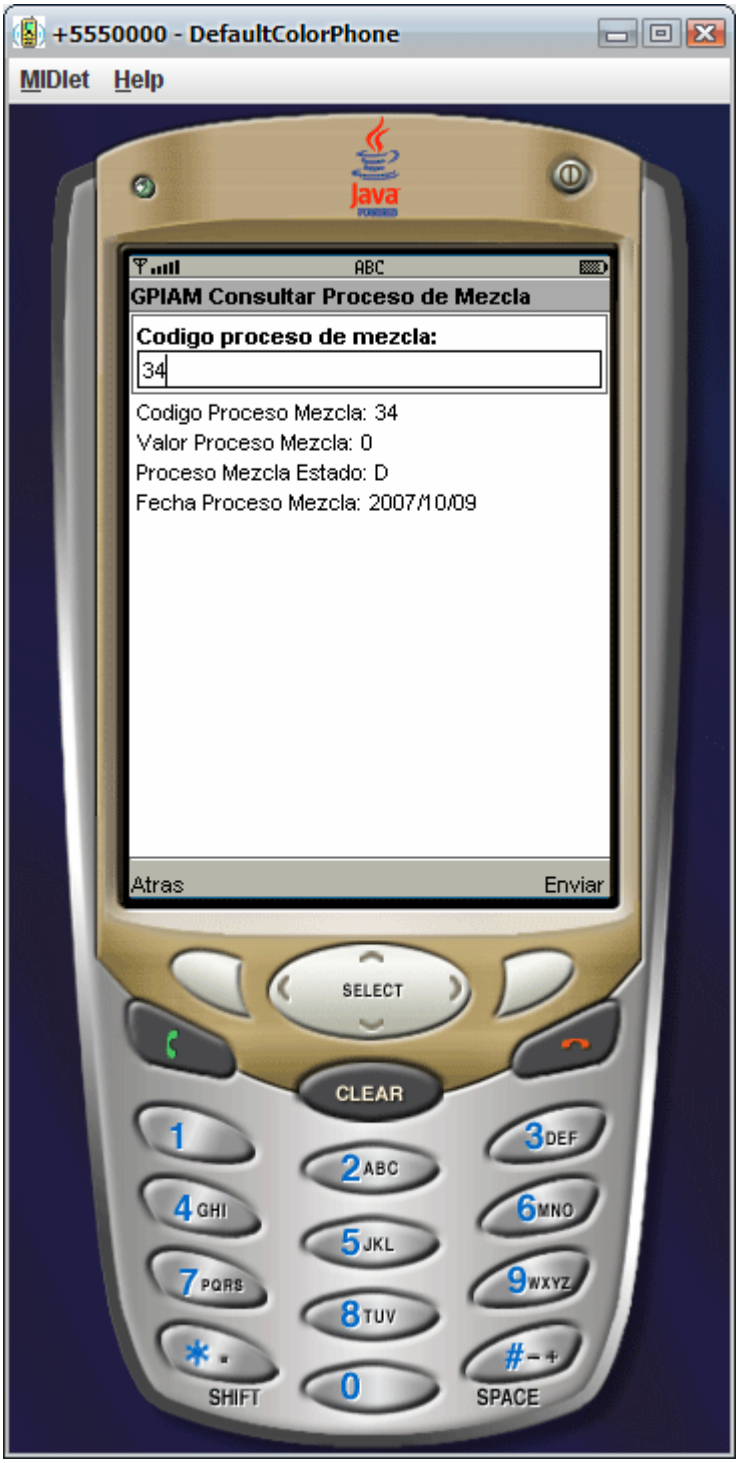


Menú de señales digitales

15.6.1 Consultar un proceso de mezcla

Esta variable permite consultar un proceso de la mezcla en especial, solo hay que digitar el código que se desea ver y luego pulsar ok.

Resultado de la consulta

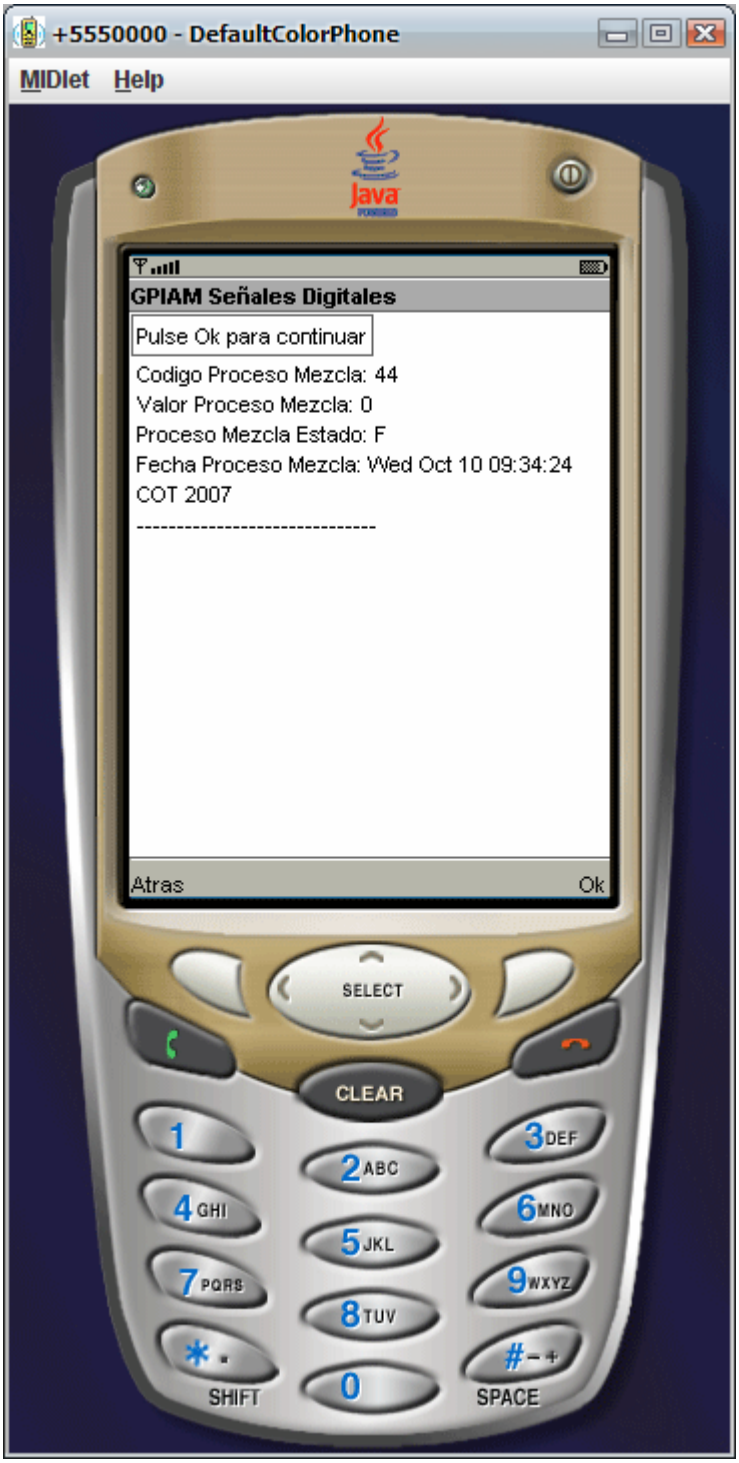


15.6.2 Consultar último proceso de mezcla

Con esta opción se observara el último proceso de mezcla que está haciendo el sistema.

Pulse Ok para continuar

Resultado de la consulta


+5550000 - DefaultColorPhone
MIDlet
Help
GPIAM Señales Digitales
Pulse Ok para continuar
Codigo Proceso Mezcla: 44
Valor Proceso Mezcla: 0
Proceso Mezcla Estado: F
Fecha Proceso Mezcla: Wed Oct 10 09:34:24
COT 2007

Atras
Ok
SELECT
CLEAR
1
2 ABC
3 DEF
4 GHI
5 JKL
6 MNO
7 PQRS
8 TUV
9 WXYZ
* .
SHIFT
0
- +
SPACE

15.7 FALLAS PROCESO

En este menú se podrá consultar que fallas del proceso están activas o desactivas. Se podrá observar una en especial digitando el código o las ultimas fallas registradas por el sistema.

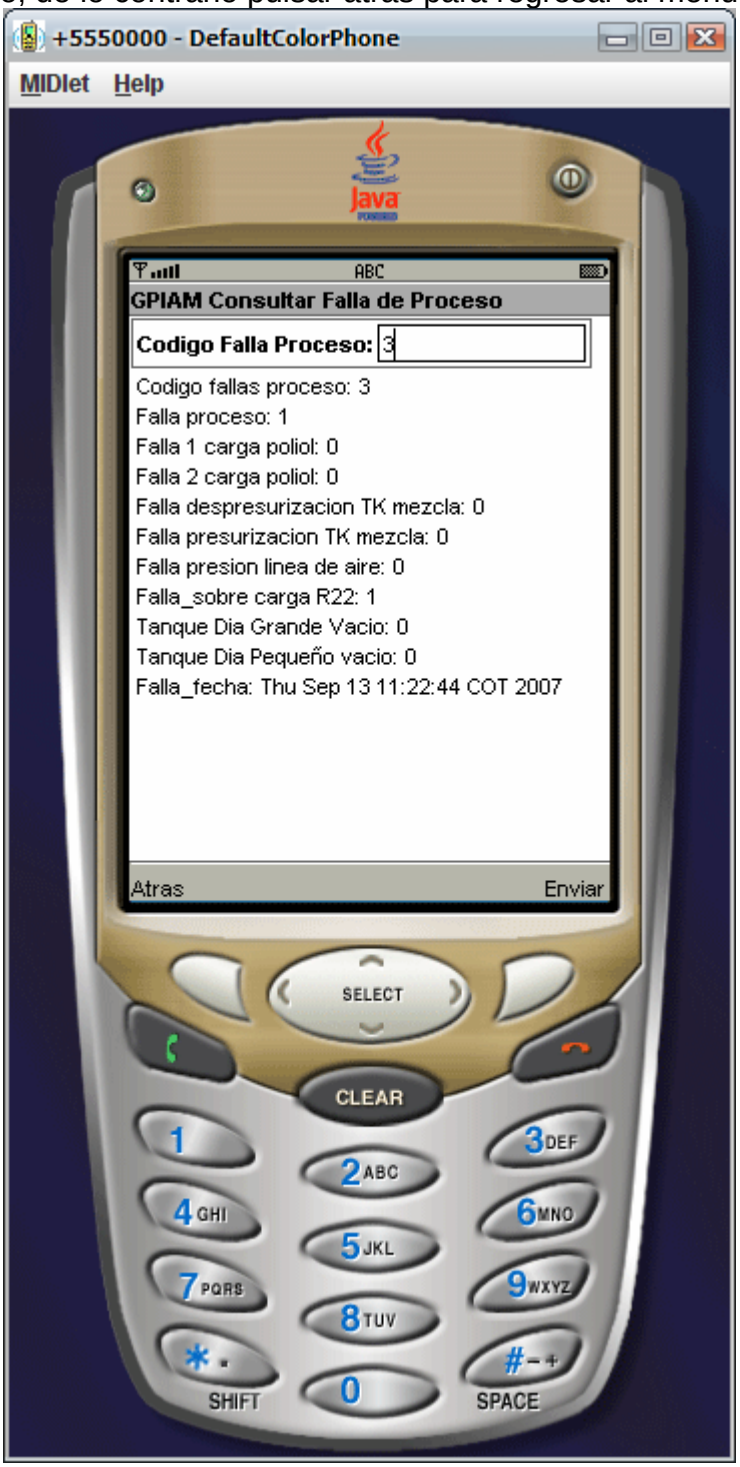
Menú fallas de procesos


+5550000 - DefaultColorPhone
MIDlet
Help
Java
Fallas de Proceso
Tags
Consultar una falla de proceso
Consultar ultima falla de proceso
Volver
Menu
SELECT
CLEAR
1
2 ABC
3 DEF
4 GHI
5 JKL
6 MNO
7 PQRS
8 TUV
9 WXYZ
* .
SHIFT
0
- +
SPACE

15.7.1 Consultar una falla de proceso

Para consultar la falla de proceso digitar el código correspondiente, luego pulsar ok.

Resultado de la consulta: Si se desea observar otra falla digitar de el nuevo código, de lo contrario pulsar atrás para regresar al menú principal.



15.7.2 Consultar últimas fallas de proceso

Aquí se podrá consultar las últimas fallas reportadas por el sistema.

Pulse Ok para continuar

Resultado de la consulta


+5550000 - DefaultColorPhone
MIDlet
Help
GPIAM Fallas de Proceso
Pulse Ok para continuar
Codigo fallas proceso: 154
Falla proceso: 1
Falla 1 carga poliol: 0
Falla 2 carga poliol: 0
Falla despresurizacion TK mezcla: 0
Falla presurizacion TK mezcla: 0
Falla presion linea de aire: 0
Falla_sobre carga R22: 0
Tanque Dia Grande Vacio: 0
Tanque Dia Pequeño vacio: 0
Falla_fecha: Wed Oct 10 11:30:19 COT 2007

Atras
Ok
SELECT
CLEAR
1
2 ABC
3 DEF
4 GHI
5 JKL
6 MNO
7 PQRS
8 TUV
9 WXYZ
* .
SHIFT
0
- +
SPACE

15.8 VARIABLES DE MEZCLA

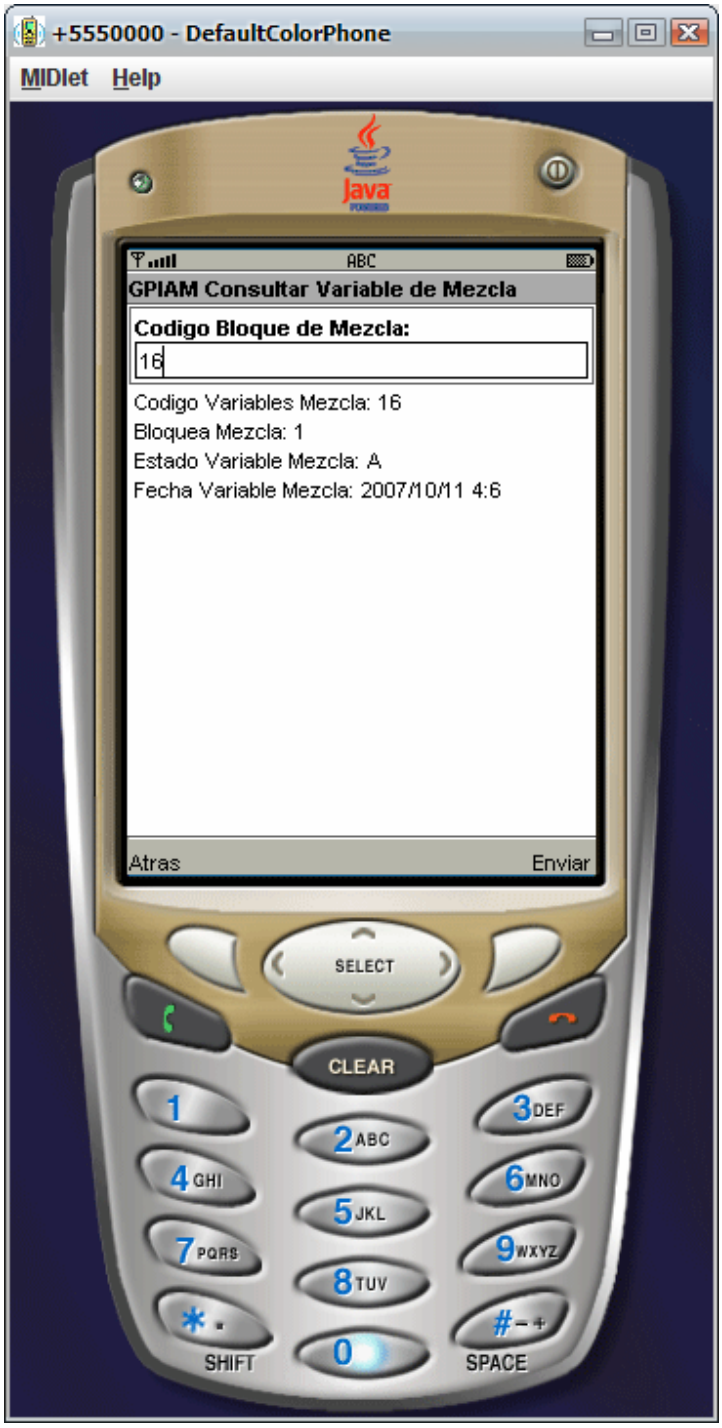
En esta opción se podrá consultar las variables de mezcla que hay en el sistema, como también se podrá insertar un nuevo valor para parar o iniciar la mezcla de R22 y Poliol.

Menú Variables de mezcla

15.8.1 Consultar una variable de mezcla

Para esto se debe ingresar el código de la variable de mezcla y luego pulsar el botón ok.

Resultado de la consulta: Si se desea consultar otra variable digitar el nuevo código y luego pulsar enviar, de lo contrario pulsar atrás para regresar al menú principal.

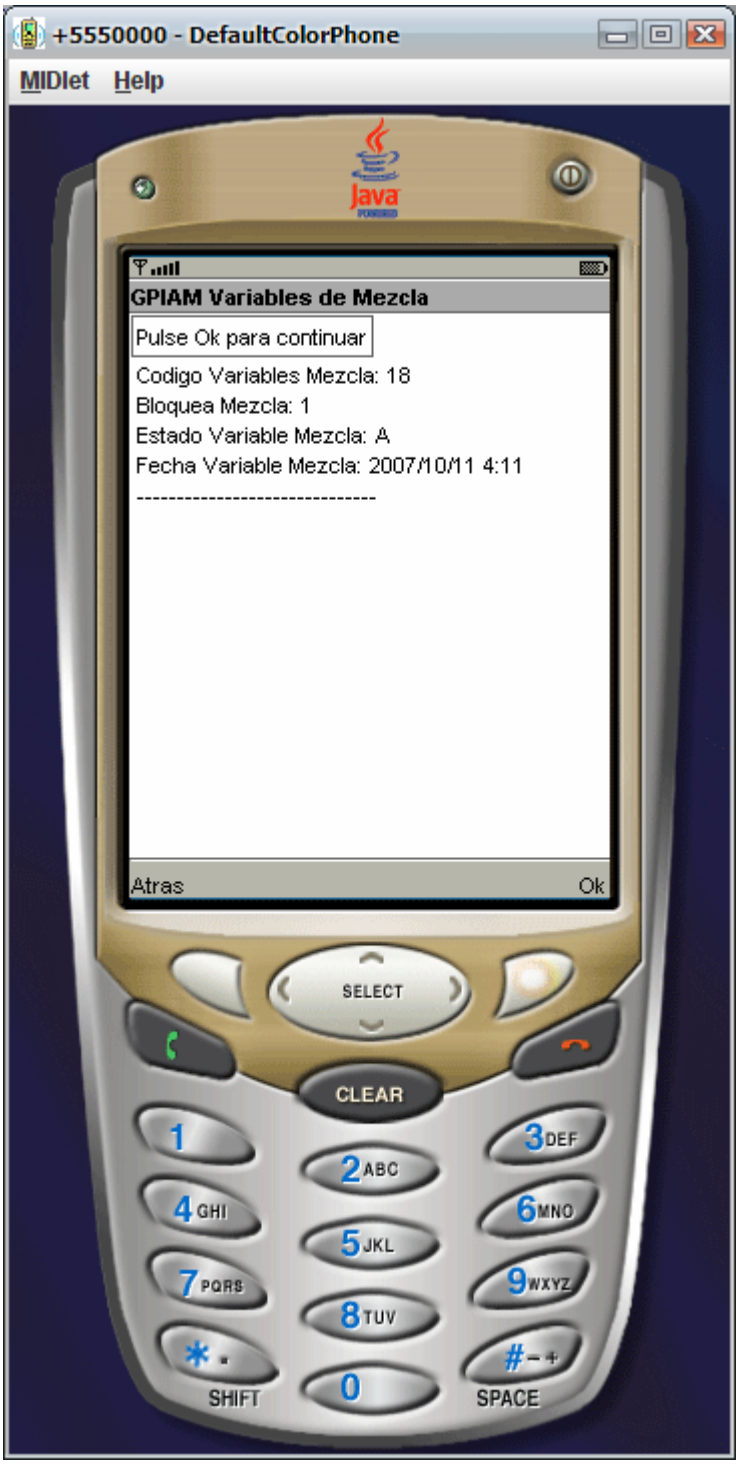


15.8.2 Consultar ultima variable de mezcla

Con esta opción se observara el estado del último registro de la variable de mezcla.

Pulse Ok para continuar

Resultado de la Consulta



15.8.3 Insertar Variable de mezcla

Esta opción permite cambiar el estado de la mezcla, enviando un 0 si se quiere para la mezcla o 1 si se quiere activar.


+5550000 - DefaultColorPhone
MIDlet Help
ABC
GPIAM Insertar Bloqueo Mezcla
Valor Bloquar mezcla 0-1:
1
Valor 1
El registro ha sido agregado
Atras
Enviar
SELECT
CLEAR
1
2 ABC
3 DEF
4 GHI
5 JKL
6 MNO
7 PQRS
8 TUV
9 WXYZ
* .
SHIFT
0
- +
SPACE

16. ANEXO H

16.1 DICCIONARIO DE MÉTODOS

1. METODO commandAction
   CLASE GPIAMMidlet
   PARAMETROS (Command command, Displayable displayable)
   DESCRIPCION DEL METODO Método que es invocado por el sistema que indica que un comando en particular ha sido llamado.
   METODOS QUE LLAMA exitMIDlet, getSelectedIndex, getDisplay().setCurrent(objeto);
   ORDEN DEL ALGORITMO O(k)

2. METODO getDisplay()
   CLASE GPIAMMidlet
   PARAMETROS No tiene parámetros.
   DESCRIPCION DEL METODO Este método debe retornar una instancia de la pantalla
   METODOS QUE LLAMA Display.getDisplay(this)
   ORDEN DEL ALGORITMO O(k)

3. METODO exitMIDlet()
   CLASE GPIAMMidlet
   PARAMETROS No tiene parámetros.
   DESCRIPCION DEL METODO Este método hace que la aplicación finalice
   METODOS QUE LLAMA getDisplay().setCurrent(null); destroyApp(true); notifyDestroyed();
   ORDEN DEL ALGORITMO O(k)

4. METODO get_GPIAMForm()
   CLASE GPIAMMidlet
   PARAMETROS No tiene parámetros.
   DESCRIPCION DEL METODO Este método retorna una instancia del componente GPIAForm o sea del formulario principal.
   METODOS QUE LLAMA get_helloStringItem(), get_choiceGroup1(), addCommand(get_exitCommand()); addCommand(get_okCommand1()); setCommandListener(this);
   ORDEN DEL ALGORITMO O(k)

5. METODO get_helloStringItem()
   CLASE GPIAMMidlet

PARAMETROS No tiene parámetros.
DESCRIPCION DEL METODO Este método retorna una instancia del componente helloStringItem, retorna el texto que aparece en la primera pantalla de bienvenida.
METODOS QUE LLAMA StringItem("Bienvenido a", "GPIAM");
ORDEN DEL ALGORITMO O(k)

6. METODO get_exitCommand()
   CLASE GPIAMMidlet
   PARAMETROS No tiene parámetros.
   DESCRIPCION DEL METODO Este método retorna una instancia del componente exitCommand, comando para finalizar la aplicación.
   METODOS QUE LLAMA Command("Men\u00FA Ppal", Command.EXIT, 1); ORDEN DEL ALGORITMO O(k)

7. METODO get_splashAlert()
   CLASE GPIAMMidlet
   PARAMETROS No tiene parámetros.
   DESCRIPCION DEL METODO Este método retorna una instancia del componente splashAlert, mustra una imagen de bienvenida durante 2 segundos .
   METODOS QUE LLAMA Alert("GPIAM", "Gesti\u00F3n de procesos industriales a trav\u00E9s de m\u00F3viles. !Bienvenido!", get_image1(), null);
   setTimeout(2000);
   ORDEN DEL ALGORITMO O(k)

8. METODO get_choiceGroup1()
   CLASE GPIAMMidlet
   PARAMETROS No tiene parámetros.
   DESCRIPCION DEL METODO Este método retorna una instancia del componente choiceGroup1, para seleccionar una opción del menú 1
   METODOS QUE LLAMA ChoiceGroup("Men\u00FA Principal", Choice.EXCLUSIVE, new String[] {"R-22"}, new Image[] {null});
   setSelectedFlags(new boolean[] {false});
   ORDEN DEL ALGORITMO O(k)

9. METODO get_okCommand1()
   CLASE GPIAMMidlet
   PARAMETROS No tiene parámetros.
   DESCRIPCION DEL METODO Este método retorna una instancia del componente okCommand1, ejecuta la siguiente acción.
   METODOS QUE LLAMA Command("Ok", Command.OK, 1);

ORDEN DEL ALGORITMO O(k)

10. METODO get_image1()
    CLASE GPIAMMidlet
    PARAMETROS No tiene parámetros.
    DESCRIPCION DEL METODO Este método retorna una instancia del componente image1.
    METODOS QUE LLAMA Image.createImage("/gpiam/figura1.png");
    ORDEN DEL ALGORITMO O(k)

11. METODO get_R22()
    CLASE GPIAMMidlet
    PARAMETROS No tiene parámetros.
    DESCRIPCION DEL METODO Este método retorna una instancia del componente R22, devuelve la pantalla del formulario R22.
    METODOS QUE LLAMA get_stringItem1(), get_choiceGroup2(),
    addCommand(get_okCommand1());
    addCommand(get_backCommand1());
    setCommandListener(this);
    ORDEN DEL ALGORITMO O(k)

12. METODO get_stringItem1()
    CLASE GPIAMMidlet
    PARAMETROS No tiene parámetros.
    DESCRIPCION DEL METODO Este método retorna una instancia del componente stringItem1, retorna el texto que aparece en pantalla del formulario R22.
    METODOS QUE LLAMA StringItem("Proceso de Mezcla (Poliol+R22)", "");
    ORDEN DEL ALGORITMO O(k)

13. METODO get_choiceGroup2()
    CLASE GPIAMMidlet
    PARAMETROS No tiene parámetros.
    DESCRIPCION DEL METODO Este método retorna una instancia del componente choiceGroup2, para seleccionar una opción del menú del formulario R22.
    METODOS QUE LLAMA ChoiceGroup("R22", Choice.EXCLUSIVE, new String[] {
        "Pesos tanques",
        "Temperaturas",
        "Niveles tanques",
        "Se\u00F1ales digitales",
        "Fallas proceso",

```
                "Variables mezcla"
            }
    setSelectedFlags(new boolean[] {
                false,
                false,
                false,
                false,
                false,
                false
            });
```
ORDEN DEL ALGORITMO O(k)

14. METODO get_backCommand1()
    CLASE GPIAMMidlet
    PARAMETROS No tiene parámetros.
    DESCRIPCION DEL METODO Este método retorna una instancia del componente backCommand1, vuelve a la pantalla anterior.
    METODOS QUE LLAMA Command("Volver", Command.BACK, 1);
    ORDEN DEL ALGORITMO O(k)

15. METODO get_okCommand2()
    CLASE GPIAMMidlet
    PARAMETROS No tiene parámetros.
    DESCRIPCION DEL METODO Este método retorna una instancia del componente okCommand2, ejecuta la siguiente acción.
    METODOS QUE LLAMA Command("Ok", Command.OK, 1);
    ORDEN DEL ALGORITMO O(k)

16. METODO get_ PesoTanques ()
    CLASE GPIAMMidlet
    PARAMETROS No tiene parámetros.
    DESCRIPCION DEL METODO Este método retorna una instancia del componente PesoTanques, devuelve la pantalla del formulario PesoTanques.
    METODOS QUE LLAMA get_stringItem3(), get_choiceGroup4(),
    addCommand(get_okCommand1());
    addCommand(get_backCommand1());
    setCommandListener(this);
    ORDEN DEL ALGORITMO O(k)

17. METODO get_stringItem3()
    CLASE GPIAMMidlet
    PARAMETROS No tiene parámetros.

DESCRIPCION DEL METODO Este método retorna una instancia del componente stringItem3, retorna el texto que aparece en pantalla del formulario PesosTanque.
METODOS QUE LLAMA StringItem("Peso Tanques", "");
ORDEN DEL ALGORITMO O(k)

18. METODO get_choiceGroup3()
CLASE GPIAMMidlet
PARAMETROS No tiene parámetros.
DESCRIPCION DEL METODO Este método retorna una instancia del componente choiceGroup3, para seleccionar una opción del menú del formulario PesosTanque.
METODOS QUE LLAMA ChoiceGroup("TAGS", Choice.EXCLUSIVE, new String[] {

```
"Consultar un  peso de los tanques",
"Consultar ultimo peso de tanques"
}
```

setSelectedFlags(new boolean[] {

```
false,
false,
});
```

ORDEN DEL ALGORITMO O(k)

19. METODO get_okCommand3()
CLASE GPIAMMidlet
PARAMETROS No tiene parámetros.
DESCRIPCION DEL METODO Este método retorna una instancia del componente okCommand3, ejecuta la siguiente acción.
METODOS QUE LLAMA Command("Ok", Command.OK, 1);
ORDEN DEL ALGORITMO O(k)

20. METODO get_Temperaturas ()
CLASE GPIAMMidlet
PARAMETROS No tiene parámetros.
DESCRIPCION DEL METODO Este método retorna una instancia del componente Temperaturas, devuelve la pantalla del formulario Temperaturas.
METODOS QUE LLAMA get_stringItem4(), get_choiceGroup4(),
addCommand(get_okCommand4());
addCommand(get_backCommand2());
addCommand(get_backCommand3());
setCommandListener(this);
ORDEN DEL ALGORITMO O(k)

21. METODO get_stringItem4()
    CLASE GPIAMMidlet
    PARAMETROS No tiene parámetros.
    DESCRIPCION DEL METODO Este método retorna una instancia del componente stringItem4, retorna el texto que aparece en pantalla del formulario Temperaturas.
    METODOS QUE LLAMA StringItem("Temperaturas", "");
    ORDEN DEL ALGORITMO O(k)

22. METODO get_choiceGroup4()
    CLASE GPIAMMidlet
    PARAMETROS No tiene parámetros.
    DESCRIPCION DEL METODO Este método retorna una instancia del componente choiceGroup4, para seleccionar una opción del menú del formulario Temperaturas.
    METODOS QUE LLAMA ChoiceGroup("Tags", Choice.EXCLUSIVE, new String[] {
                "Consultar una Temperatura",
                "Consultar ultimas temperaturas"
            }
    setSelectedFlags(new boolean[] {
                false,
                false,
            });
    ORDEN DEL ALGORITMO O(k)

23. METODO get_okCommand4()
    CLASE GPIAMMidlet
    PARAMETROS No tiene parámetros.
    DESCRIPCION DEL METODO Este método retorna una instancia del componente okCommand4, ejecuta la siguiente acción.
    METODOS QUE LLAMA Command("Ok", Command.OK, 1);
    ORDEN DEL ALGORITMO O(k)

24. METODO get_backCommand2()
    CLASE GPIAMMidlet
    PARAMETROS No tiene parámetros.
    DESCRIPCION DEL METODO Este método retorna una instancia del componente backCommand2, retorna a la pantalla anterior.
    METODOS QUE LLAMA Command("Volver", Command.BACK, 1);
    ORDEN DEL ALGORITMO O(k)

25. METODO get_backCommand3()
    CLASE GPIAMMidlet
    PARAMETROS No tiene parámetros.
    DESCRIPCION DEL METODO Este método retorna una instancia del componente backCommand3, vuelve a la pantalla anterior.
    METODOS QUE LLAMA Command("Volver", Command.BACK, 1);
    ORDEN DEL ALGORITMO O(k)

26. METODO get_ NivelesTanques()
    CLASE GPIAMMidlet
    PARAMETROS No tiene parámetros.
    DESCRIPCION DEL METODO Este método retorna una instancia del componente NivelesTanques, devuelve la pantalla del formulario NivelesTanques.
    METODOS QUE LLAMA get_stringItem5(), get_choiceGroup5(),
    addCommand(get_okCommand5());
    addCommand(get_backCommand4());
    addCommand(get_backCommand5());
    setCommandListener(this);
    ORDEN DEL ALGORITMO O(k)

27. METODO get_stringItem5()
    CLASE GPIAMMidlet
    PARAMETROS No tiene parámetros.
    DESCRIPCION DEL METODO Este método retorna una instancia del componente stringItem4, retorna el texto que aparece en pantalla del formulario NivelesTanques.
    METODOS QUE LLAMA StringItem("Niveles Tanques", "");
    ORDEN DEL ALGORITMO O(k)

28. METODO get_choiceGroup5()
    CLASE GPIAMMidlet
    PARAMETROS No tiene parámetros.
    DESCRIPCION DEL METODO Este método retorna una instancia del componente choiceGroup4, para seleccionar una opción del menú del formulario Temperaturas.
    METODOS QUE LLAMA ChoiceGroup("Tags", Choice.EXCLUSIVE, new String[] {
                "Consultar un Nivel de Tanque",
                "Consultar ultimo Nivel de Tanques"
            }
    setSelectedFlags(new boolean[] {
                false,

```
                false,
            });
```
ORDEN DEL ALGORITMO O(k)

29. METODO get_okCommand5()
    CLASE GPIAMMidlet
    PARAMETROS No tiene parámetros.
    DESCRIPCION DEL METODO Este método retorna una instancia del componente okCommand5, ejecuta la siguiente acción.
    METODOS QUE LLAMA Command("Ok", Command.OK, 1);
    ORDEN DEL ALGORITMO O(k)

30. METODO get_backCommand5()
    CLASE GPIAMMidlet
    PARAMETROS No tiene parámetros.
    DESCRIPCION DEL METODO Este método retorna una instancia del componente backCommand5, retorna a la pantalla anterior.
    METODOS QUE LLAMA Command("Volver", Command.BACK, 1);
    ORDEN DEL ALGORITMO O(k)

31. METODO get_ SeñalesDigitales()
    CLASE GPIAMMidlet
    PARAMETROS No tiene parámetros.
    DESCRIPCION DEL METODO Este método retorna una instancia del componente SeñalesDigitales, devuelve la pantalla del formulario SeñalesDigitales.
    METODOS QUE LLAMA get_stringItem6(), get_choiceGroup6(),
    addCommand(get_okCommand6());
    addCommand(get_backCommand6());
    addCommand(get_backCommand7());
    setCommandListener(this);
    ORDEN DEL ALGORITMO O(k)

32. METODO get_stringItem6()
    CLASE GPIAMMidlet
    PARAMETROS No tiene parámetros.
    DESCRIPCION DEL METODO Este método retorna una instancia del componente stringItem6, retorna el texto que aparece en pantalla del formulario Señales digitales.
    METODOS QUE LLAMA StringItem("Se\u00F1ales Digitales", "\n");
    ORDEN DEL ALGORITMO O(k)

33. METODO get_choiceGroup6()

CLASE GPIAMMidlet
PARAMETROS No tiene parámetros.
DESCRIPCION DEL METODO Este método retorna una instancia del componente choiceGroup6, para seleccionar una opción del menú del formulario SeñalesDigitales.
METODOS QUE LLAMA ChoiceGroup("Tags\n", Choice.EXCLUSIVE, new String[] {
"Consultar un proceso de mezcla",
"Consultar ultimo proceso de mezcla",
"Insertar valor "
}
setSelectedFlags(new boolean[] {
false,
false,
false,
});
ORDEN DEL ALGORITMO O(k)

34. METODO get_okCommand6()
CLASE GPIAMMidlet
PARAMETROS No tiene parámetros.
DESCRIPCION DEL METODO Este método retorna una instancia del componente okCommand6, ejecuta la siguiente acción.
METODOS QUE LLAMA Command("Ok", Command.OK, 1);
ORDEN DEL ALGORITMO O(k)

35. METODO get_okCommand7()
CLASE GPIAMMidlet
PARAMETROS No tiene parámetros.
DESCRIPCION DEL METODO Este método retorna una instancia del componente okCommand7, ejecuta la siguiente acción.
METODOS QUE LLAMA Command("Ok", Command.OK, 1);
ORDEN DEL ALGORITMO O(k)

36. METODO get_backCommand6()
CLASE GPIAMMidlet
PARAMETROS No tiene parámetros.
DESCRIPCION DEL METODO Este método retorna una instancia del componente backCommand6, retorna a la pantalla anterior.
METODOS QUE LLAMA Command("Volver", Command.BACK, 1);
ORDEN DEL ALGORITMO O(k)

37. METODO get_backCommand7()

CLASE GPIAMMidlet
PARAMETROS No tiene parámetros.
DESCRIPCION DEL METODO Este método retorna una instancia del componente backCommand7, retorna a la pantalla anterior.
METODOS QUE LLAMA Command("Volver", Command.BACK, 1);
ORDEN DEL ALGORITMO O(k)

38. METODO get_ FallasProceso()
CLASE GPIAMMidlet
PARAMETROS No tiene parámetros.
DESCRIPCION DEL METODO Este método retorna una instancia del componente FallasProceso, devuelve la pantalla del formulario FallasProceso.
METODOS QUE LLAMA get_stringItem7(), get_choiceGroup8(),
addCommand(get_okCommand8());
addCommand(get_backCommand8());
addCommand(get_backCommand9());
setCommandListener(this);
ORDEN DEL ALGORITMO O(k)

39. METODO get_stringItem7()
CLASE GPIAMMidlet
PARAMETROS No tiene parámetros.
DESCRIPCION DEL METODO Este método retorna una instancia del componente stringItem7, retorna el texto que aparece en pantalla del formulario FallasProceso.
METODOS QUE LLAMA StringItem("Fallas de proceso", "\n");
ORDEN DEL ALGORITMO O(k)

40. METODO get_choiceGroup8()
CLASE GPIAMMidlet
PARAMETROS No tiene parámetros.
DESCRIPCION DEL METODO Este método retorna una instancia del componente choiceGroup8, para seleccionar una opción del menú del formulario FallasProceso.
METODOS QUE LLAMA ChoiceGroup("Tags", Choice.EXCLUSIVE, new String[] {
"Consultar una falla de proceso",
"Consultar ultima falla de proceso"
}
setSelectedFlags(new boolean[] {
false,
false,

});
ORDEN DEL ALGORITMO O(k)

41. METODO get_okCommand8()
CLASE GPIAMMidlet
PARAMETROS No tiene parámetros.
DESCRIPCION DEL METODO Este método retorna una instancia del componente okCommand8, ejecuta la siguiente acción.
METODOS QUE LLAMA Command("Ok", Command.OK, 1);
ORDEN DEL ALGORITMO O(k)

42. METODO get_backCommand9()
CLASE GPIAMMidlet
PARAMETROS No tiene parámetros.
DESCRIPCION DEL METODO Este método retorna una instancia del componente backCommand7, retorna a la pantalla anterior.
METODOS QUE LLAMA Command("Volver", Command.BACK, 1);
ORDEN DEL ALGORITMO O(k)

43. METODO get_VariablesMezcla()
CLASE GPIAMMidlet
PARAMETROS No tiene parámetros.
DESCRIPCION DEL METODO Este método retorna una instancia del componente VariablesMezcla, devuelve la pantalla del formulario VariablesMezcla.
METODOS QUE LLAMA get_stringItem8(), get_choiceGroup9(),
addCommand(get_okCommand9());
addCommand(get_backCommand10());
addCommand(get_backCommand11());
setCommandListener(this);
ORDEN DEL ALGORITMO O(k)

44. METODO get_stringItem8()
CLASE GPIAMMidlet
PARAMETROS No tiene parámetros.
DESCRIPCION DEL METODO Este método retorna una instancia del componente stringItem8, retorna el texto que aparece en pantalla del formulario VariablesMezcla.
METODOS QUE LLAMA StringItem("Variables de mezcla", "\n");
ORDEN DEL ALGORITMO O(k)

45. METODO get_choiceGroup9()
CLASE GPIAMMidlet

PARAMETROS No tiene parámetros.
DESCRIPCION DEL METODO Este método retorna una instancia del componente choiceGroup8, para seleccionar una opción del menú del formulario VariablesMezcla.
METODOS QUE LLAMA ("Tags", Choice.EXCLUSIVE, new String[] {
"Consultar una variable de mezcla",
"Consultar ultima variable de mezcla",
"Insertar variable de mezcla"
}
setSelectedFlags(new boolean[] {
false,
false,
false,
});
ORDEN DEL ALGORITMO O(k)

46. METODO get_okCommand10()
CLASE GPIAMMidlet
PARAMETROS No tiene parámetros.
DESCRIPCION DEL METODO Este método retorna una instancia del componente okCommand9, ejecuta la siguiente acción.
METODOS QUE LLAMA Command("Ok", Command.OK, 1);
ORDEN DEL ALGORITMO O(k)

47. METODO get_backCommand11()
CLASE GPIAMMidlet
PARAMETROS No tiene parámetros.
DESCRIPCION DEL METODO Este método retorna una instancia del componente backCommand7, retorna a la pantalla anterior.
METODOS QUE LLAMA Command("Volver", Command.BACK, 1);
ORDEN DEL ALGORITMO O(k)

48. METODO startApp()
CLASE GPIAMMidlet
PARAMETROS No tiene parámetros.
DESCRIPCION DEL METODO Este método inicializa la aplicación movil
METODOS QUE LLAMA initialize();
ORDEN DEL ALGORITMO O(k)

49. METODO pauseApp()
CLASE GPIAMMidlet
PARAMETROS No tiene parámetros.

DESCRIPCION DEL METODO Se suspenden todas las acciones que se realizan en segundo plano y se liberan los recursos
METODOS QUE LLAMA No llama ningún método
ORDEN DEL ALGORITMO O(k)

50. METODO destroyApp
CLASE GPIAMMidlet
PARAMETROS (boolean unconditional)
DESCRIPCION DEL METODO Detiene todas las actividades del midlet y libera todos los recursos que haya acaparado del dispositivo, cuando el midlet llega al final de su vida útil
METODOS QUE LLAMA No llama ningún método
ORDEN DEL ALGORITMO O(k)

51. METODO commandAction
CLASE ConsultarFallasProceso
PARAMETROS (Command command, Displayable displayable)
DESCRIPCION DEL METODO Método implementado para satisfacer la interfaz CommandListener que se encarga de manejar el comando que se ha asignado al midlet para enviar los datos a través de la red o para volver al menú principal
METODOS QUE LLAMA get_FallasProceso(), getString(), run(),invokeServlet()
ORDEN DEL ALGORITMO O(k)

52. METODO invokeServlet()
CLASE ConsultarFallasProceso
PARAMETROS No recibe parámetros
DESCRIPCION DEL METODO Método que hace la conexión con el servidor Tomcat el cual ejecuta un servlet el cual hace referencia a una base de datos
METODOS QUE LLAMA (HttpConnection)Connector.open(url), setRequestMethod(HttpConnection.GET), setRequestProperty("IF-Modified-Since", "17 Dic 2006 13:39:14 GMT"); setRequestProperty("User-Agent","Profile/MIDP-2.0 Configuration/CLDC-1.0"); setRequestProperty("Content-Language", "es-ES"); setRequestProperty("Content-Type", "application/x-www-form-urlencoded"); openDataInputStream();getResponseCode(); getLength();read(); close();setRespuesta(respuesta); getResponseMessage();
ORDEN DEL ALGORITMO O(k)

53. METODO setRespuesta
CLASE ConsultarFallasProceso

PARAMETROS (String res)
DESCRIPCION DEL METODO Método que encapsula el texto enviado
METODOS QUE LLAMA setText(res); append(txt);
ORDEN DEL ALGORITMO O(k)

54. METODO getRespuesta
CLASE ConsultarFallasProceso
PARAMETROS No recibe parámetros
DESCRIPCION DEL METODO Método que retorna el texto
METODOS QUE LLAMA No llama ningún método
ORDEN DEL ALGORITMO O(k)

55. METODO commandAction
CLASE ConsultarFallasProceso
PARAMETROS (Command command, Displayable displayable)
DESCRIPCION DEL METODO Método implementado para satisfacer la interfaz CommandListener que se encarga de manejar el comando que se ha asignado al midlet para enviar los datos a través de la red o para volver al menú principal
METODOS QUE LLAMA get_FallasProceso(), getString(), run(),invokeServlet()
ORDEN DEL ALGORITMO O(k)

56. METODO invokeServlet()
CLASE ConsultarFallasProceso
PARAMETROS No recibe parámetros
DESCRIPCION DEL METODO Método que hace la conexión con el servidor Tomcat el cual ejecuta un servlet el cual hace referencia a una base de datos
METODOS QUE LLAMA (HttpConnection)Connector.open(url), setRequestMethod(HttpConnection.GET), setRequestProperty("IF-Modified-Since", "17 Dic 2006 13:39:14 GMT"); setRequestProperty("User-Agent","Profile/MIDP-2.0 Configuration/CLDC-1.0"); setRequestProperty("Content-Language", "es-ES"); setRequestProperty("Content-Type", "application/x-www-form-urlencoded"); openDataInputStream();getResponseCode(); getLength();read(); close();setRespuesta(respuesta); getResponseMessage();
ORDEN DEL ALGORITMO O(k)

57. METODO setRespuesta
CLASE ConsultarFallasProceso
PARAMETROS (String res)
DESCRIPCION DEL METODO Método que encapsula el texto enviado

METODOS QUE LLAMA setText(res); append(txt);
ORDEN DEL ALGORITMO O(k)

58. METODO getRespuesta
CLASE ConsultarFallasProceso
PARAMETROS No recibe parámetros
DESCRIPCION DEL METODO Método que retorna el texto
METODOS QUE LLAMA No llama ningún método
ORDEN DEL ALGORITMO O(k)

59. METODO commandAction
CLASE ConsultarNivelTanque
PARAMETROS (Command command, Displayable displayable)
DESCRIPCION DEL METODO Método implementado para satisfacer la interfaz CommandListener que se encarga de manejar el comando que se ha asignado al midlet para enviar los datos a través de la red o para volver al menú principal
METODOS QUE LLAMA get_NivelTanque (), getString(), run(),invokeServlet()
ORDEN DEL ALGORITMO O(k)

60. METODO invokeServlet()
CLASE ConsultarNivelTanque
PARAMETROS No recibe parámetros
DESCRIPCION DEL METODO Método que hace la conexión con el servidor Tomcat el cual ejecuta un servlet el cual hace referencia a una base de datos
METODOS QUE LLAMA (HttpConnection)Connector.open(url), setRequestMethod(HttpConnection.GET), setRequestProperty("IF-Modified-Since", "17 Dic 2006 13:39:14 GMT"); setRequestProperty("User-Agent","Profile/MIDP-2.0 Configuration/CLDC-1.0"); setRequestProperty("Content-Language", "es-ES"); setRequestProperty("Content-Type", "application/x-www-form-urlencoded"); openDataInputStream();getResponseCode(); getLength();read(); close();setRespuesta(respuesta); getResponseMessage();
ORDEN DEL ALGORITMO O(k)

61. METODO setRespuesta
CLASE ConsultarNivelTanque
PARAMETROS (String res)
DESCRIPCION DEL METODO Método que encapsula el texto enviado
METODOS QUE LLAMA setText(res); append(txt);
ORDEN DEL ALGORITMO O(k)

62. METODO getRespuesta
    CLASE ConsultarNivelTanque
    PARAMETROS No recibe parámetros
    DESCRIPCION DEL METODO Método que retorna el texto
    METODOS QUE LLAMA No llama ningún método
    ORDEN DEL ALGORITMO O(k)

63. METODO commandAction
    CLASE ConsultarPesosTanques
    PARAMETROS (Command command, Displayable displayable)
    DESCRIPCION DEL METODO Método implementado para satisfacer la interfaz CommandListener que se encarga de manejar el comando que se ha asignado al midlet para enviar los datos a través de la red o para volver al menú principal
    METODOS QUE LLAMA get_PesosTanques (), getString(), run(),invokeServlet()
    ORDEN DEL ALGORITMO O(k)

64. METODO invokeServlet()
    CLASE ConsultarPesosTanques
    PARAMETROS No recibe parámetros
    DESCRIPCION DEL METODO Método que hace la conexión con el servidor Tomcat el cual ejecuta un servlet el cual hace referencia a una base de datos
    METODOS QUE LLAMA (HttpConnection)Connector.open(url), setRequestMethod(HttpConnection.GET), setRequestProperty("IF-Modified-Since", "17 Dic 2006 13:39:14 GMT"); setRequestProperty("User-Agent","Profile/MIDP-2.0 Configuration/CLDC-1.0"); setRequestProperty("Content-Language", "es-ES"); setRequestProperty("Content-Type", "application/x-www-form-urlencoded"); openDataInputStream();getResponseCode(); getLength();read(); close();setRespuesta(respuesta); getResponseMessage();
    ORDEN DEL ALGORITMO O(k)

65. METODO setRespuesta
    CLASE ConsultarPesosTanques
    PARAMETROS (String res)
    DESCRIPCION DEL METODO Método que encapsula el texto enviado
    METODOS QUE LLAMA setText(res); append(txt);
    ORDEN DEL ALGORITMO O(k)

66. METODO getRespuesta

CLASE ConsultarPesosTanques
PARAMETROS No recibe parámetros
DESCRIPCION DEL METODO Método que retorna el texto
METODOS QUE LLAMA No llama ningún método
ORDEN DEL ALGORITMO O(k)

67. METODO commandAction
CLASE ConsultarSenalesDigitales
PARAMETROS (Command command, Displayable displayable)
DESCRIPCION DEL METODO Método implementado para satisfacer la interfaz CommandListener que se encarga de manejar el comando que se ha asignado al midlet para enviar los datos a través de la red o para volver al menú principal
METODOS QUE LLAMA get_SenalesDigitales (), getString(), run(),invokeServlet()
ORDEN DEL ALGORITMO O(k)

68. METODO invokeServlet()
CLASE ConsultarSenalesDigitales
PARAMETROS No recibe parámetros
DESCRIPCION DEL METODO Método que hace la conexión con el servidor Tomcat el cual ejecuta un servlet el cual hace referencia a una base de datos
METODOS QUE LLAMA (HttpConnection)Connector.open(url), setRequestMethod(HttpConnection.GET), setRequestProperty("IF-Modified-Since", "17 Dic 2006 13:39:14 GMT"); setRequestProperty("User-Agent","Profile/MIDP-2.0 Configuration/CLDC-1.0"); setRequestProperty("Content-Language", "es-ES"); setRequestProperty("Content-Type", "application/x-www-form-urlencoded"); openDataInputStream();getResponseCode(); getLength();read(); close();setRespuesta(respuesta); getResponseMessage();
ORDEN DEL ALGORITMO O(k)

69. METODO setRespuesta
CLASE ConsultarSenalesDigitales
PARAMETROS (String res)
DESCRIPCION DEL METODO Método que encapsula el texto enviado
METODOS QUE LLAMA setText(res); append(txt);
ORDEN DEL ALGORITMO O(k)

70. METODO getRespuesta
CLASE ConsultarSenalesDigitales
PARAMETROS No recibe parámetros

DESCRIPCION DEL METODO Método que retorna el texto
METODOS QUE LLAMA No llama ningún método
ORDEN DEL ALGORITMO O(k)

71. METODO commandAction
CLASE ConsultarUnaTemperatura
PARAMETROS (Command command, Displayable displayable)
DESCRIPCION DEL METODO Método implementado para satisfacer la interfaz CommandListener que se encarga de manejar el comando que se ha asignado al midlet para enviar los datos a través de la red o para volver al menú principal
METODOS QUE LLAMA get_UnaTemperatura (), getString(), run(),invokeServlet()
ORDEN DEL ALGORITMO O(k)

72. METODO invokeServlet()
CLASE ConsultarUnaTemperatura
PARAMETROS No recibe parámetros
DESCRIPCION DEL METODO Método que hace la conexión con el servidor Tomcat el cual ejecuta un servlet el cual hace referencia a una base de datos
METODOS QUE LLAMA (HttpConnection)Connector.open(url), setRequestMethod(HttpConnection.GET), setRequestProperty("IF-Modified-Since", "17 Dic 2006 13:39:14 GMT"); setRequestProperty("User-Agent","Profile/MIDP-2.0 Configuration/CLDC-1.0"); setRequestProperty("Content-Language", "es-ES"); setRequestProperty("Content-Type", "application/x-www-form-urlencoded"); openDataInputStream();getResponseCode(); getLength();read(); close();setRespuesta(respuesta); getResponseMessage();
ORDEN DEL ALGORITMO O(k)

73. METODO setRespuesta
CLASE ConsultarUnaTemperatura
PARAMETROS (String res)
DESCRIPCION DEL METODO Método que encapsula el texto enviado
METODOS QUE LLAMA setText(res); append(txt);
ORDEN DEL ALGORITMO O(k)

74. METODO getRespuesta
CLASE ConsultarUnaTemperatura
PARAMETROS No recibe parámetros
DESCRIPCION DEL METODO Método que retorna el texto

METODOS QUE LLAMA No llama ningún método
ORDEN DEL ALGORITMO O(k)

75. METODO commandAction
CLASE ConsultarVariablesMezcla
PARAMETROS (Command command, Displayable displayable)
DESCRIPCION DEL METODO Método implementado para satisfacer la interfaz CommandListener que se encarga de manejar el comando que se ha asignado al midlet para enviar los datos a través de la red o para volver al menú principal
METODOS QUE LLAMA get_VariablesMezcla (), getString(), run(),invokeServlet()
ORDEN DEL ALGORITMO O(k)

76. METODO invokeServlet()
CLASE ConsultarVariablesMezcla
PARAMETROS No recibe parámetros
DESCRIPCION DEL METODO Método que hace la conexión con el servidor Tomcat el cual ejecuta un servlet el cual hace referencia a una base de datos
METODOS QUE LLAMA (HttpConnection)Connector.open(url), setRequestMethod(HttpConnection.GET), setRequestProperty("IF-Modified-Since", "17 Dic 2006 13:39:14 GMT"); setRequestProperty("User-Agent","Profile/MIDP-2.0 Configuration/CLDC-1.0"); setRequestProperty("Content-Language", "es-ES"); setRequestProperty("Content-Type", "application/x-www-form-urlencoded"); openDataInputStream();getResponseCode(); getLength();read(); close();setRespuesta(respuesta); getResponseMessage();
ORDEN DEL ALGORITMO O(k)

77. METODO setRespuesta
CLASE ConsultarVariablesMezcla
PARAMETROS (String res)
DESCRIPCION DEL METODO Método que encapsula el texto enviado
METODOS QUE LLAMA setText(res); append(txt);
ORDEN DEL ALGORITMO O(k)

78. METODO getRespuesta
CLASE ConsultarVariablesMezcla
PARAMETROS No recibe parámetros
DESCRIPCION DEL METODO Método que retorna el texto
METODOS QUE LLAMA No llama ningún método
ORDEN DEL ALGORITMO O(k)

79. METODO commandAction
    CLASE InsertarVariablesMezcla
    PARAMETROS (Command command, Displayable displayable)
    DESCRIPCION DEL METODO Método implementado para satisfacer la interfaz CommandListener que se encarga de manejar el comando que se ha asignado al midlet para enviar los datos a través de la red o para volver al menú principal
    METODOS QUE LLAMA get_VariablesMezcla (), getString(), run(),invokeServlet()
    ORDEN DEL ALGORITMO O(k)

80. METODO invokeServlet()
    CLASE InsertarVariablesMezcla
    PARAMETROS No recibe parámetros
    DESCRIPCION DEL METODO Método que hace la conexión con el servidor Tomcat el cual ejecuta un servlet el cual hace referencia a una base de datos
    METODOS QUE LLAMA (HttpConnection)Connector.open(url), setRequestMethod(HttpConnection.POST), setRequestProperty("IF-Modified-Since", "17 Dic 2006 13:39:14 GMT"); setRequestProperty("User-Agent","Profile/MIDP-2.0 Configuration/CLDC-1.0"); setRequestProperty("Content-Language", "es-ES"); setRequestProperty("Content-Type", "application/x-www-form-urlencoded"); openDataInputStream();
    getResponseCode(); getLength(); read(); close();
    setRespuesta(respuesta); getResponseMessage();
    ORDEN DEL ALGORITMO O(k)

81. METODO commandAction
    CLASE MostrarFallasProceso
    PARAMETROS (Command command, Displayable displayable)
    DESCRIPCION DEL METODO Método implementado para satisfacer la interfaz CommandListener que se encarga de manejar el comando que se ha asignado al midlet para enviar los datos a través de la red o para volver al menú principal
    METODOS QUE LLAMA get_ FallasProceso(), getString(), run(), invokeServlet();start();
    ORDEN DEL ALGORITMO O(k)

82. METODO invokeServlet()
    CLASE MostrarFallasProceso

PARAMETROS No recibe parámetros
DESCRIPCION DEL METODO Método que hace la conexión con el servidor Tomcat el cual ejecuta un servlet el cual hace referencia a una base de datos
METODOS QUE LLAMA (HttpConnection)Connector.open(url), setRequestMethod(HttpConnection.GET), setRequestProperty("IF-Modified-Since", "17 Dic 2006 13:39:14 GMT"); setRequestProperty("User-Agent","Profile/MIDP-2.0 Configuration/CLDC-1.0"); setRequestProperty("Content-Language", "es-ES"); setRequestProperty("Content-Type", "application/x-www-form-urlencoded"); openDataInputStream();getResponseCode(); getLength();read(); close();setRespuesta(respuesta); getResponseMessage();
ORDEN DEL ALGORITMO O(k)

83. METODO setRespuesta
CLASE MostrarFallasProceso
PARAMETROS (String res)
DESCRIPCION DEL METODO Método que encapsula el texto enviado
METODOS QUE LLAMA setText(res); append(txt);
ORDEN DEL ALGORITMO O(k)

84. METODO getRespuesta
CLASE MostrarFallasProceso
PARAMETROS No recibe parámetros
DESCRIPCION DEL METODO Método que retorna el texto
METODOS QUE LLAMA No llama ningún método
ORDEN DEL ALGORITMO O(k)

85. METODO commandAction
CLASE MostrarNivelesTanques
PARAMETROS (Command command, Displayable displayable)
DESCRIPCION DEL METODO Método implementado para satisfacer la interfaz CommandListener que se encarga de manejar el comando que se ha asignado al midlet para enviar los datos a través de la red o para volver al menú principal
METODOS QUE LLAMA get_ NivelesTanques(), getString(), run(), invokeServlet();start();
ORDEN DEL ALGORITMO O(k)

86. METODO invokeServlet()
CLASE MostrarNivelesTanques
PARAMETROS No recibe parámetros

DESCRIPCION DEL METODO Método que hace la conexión con el servidor Tomcat el cual ejecuta un servlet el cual hace referencia a una base de datos
METODOS QUE LLAMA (HttpConnection)Connector.open(url), setRequestMethod(HttpConnection.GET), setRequestProperty("IF-Modified-Since", "17 Dic 2006 13:39:14 GMT"); setRequestProperty("User-Agent","Profile/MIDP-2.0 Configuration/CLDC-1.0"); setRequestProperty("Content-Language", "es-ES"); setRequestProperty("Content-Type", "application/x-www-form-urlencoded"); openDataInputStream();getResponseCode(); getLength();read(); close();setRespuesta(respuesta); getResponseMessage();
ORDEN DEL ALGORITMO O(k)

87. METODO setRespuesta
CLASE MostrarNivelesTanques
PARAMETROS (String res)
DESCRIPCION DEL METODO Método que encapsula el texto enviado
METODOS QUE LLAMA setText(res); append(txt);
ORDEN DEL ALGORITMO O(k)

88. METODO getRespuesta
CLASE MostrarNivelesTanques
PARAMETROS No recibe parámetros
DESCRIPCION DEL METODO Método que retorna el texto
METODOS QUE LLAMA No llama ningún método
ORDEN DEL ALGORITMO O(k)

89. METODO commandAction
CLASE MostrarPesosTanques
PARAMETROS (Command command, Displayable displayable)
DESCRIPCION DEL METODO Método implementado para satisfacer la interfaz CommandListener que se encarga de manejar el comando que se ha asignado al midlet para enviar los datos a través de la red o para volver al menú principal
METODOS QUE LLAMA get_ PesosTanques(), getString(), run(), invokeServlet();start();
ORDEN DEL ALGORITMO O(k)

90. METODO invokeServlet()
CLASE MostrarPesosTanques
PARAMETROS No recibe parámetros

DESCRIPCION DEL METODO Método que hace la conexión con el servidor Tomcat el cual ejecuta un servlet el cual hace referencia a una base de datos
METODOS QUE LLAMA (HttpConnection)Connector.open(url), setRequestMethod(HttpConnection.GET), setRequestProperty("IF-Modified-Since", "17 Dic 2006 13:39:14 GMT"); setRequestProperty("User-Agent","Profile/MIDP-2.0 Configuration/CLDC-1.0"); setRequestProperty("Content-Language", "es-ES"); setRequestProperty("Content-Type", "application/x-www-form-urlencoded"); openDataInputStream();getResponseCode(); getLength();read(); close();setRespuesta(respuesta); getResponseMessage();
ORDEN DEL ALGORITMO O(k)

91. METODO setRespuesta
CLASE MostrarPesosTanques
PARAMETROS (String res)
DESCRIPCION DEL METODO Método que encapsula el texto enviado
METODOS QUE LLAMA setText(res); append(txt);
ORDEN DEL ALGORITMO O(k)

92. METODO getRespuesta
CLASE MostrarPesosTanques
PARAMETROS No recibe parámetros
DESCRIPCION DEL METODO Método que retorna el texto
METODOS QUE LLAMA No llama ningún método
ORDEN DEL ALGORITMO O(k)

93. METODO commandAction
CLASE MostrarSenalesDigitales
PARAMETROS (Command command, Displayable displayable)
DESCRIPCION DEL METODO Método implementado para satisfacer la interfaz CommandListener que se encarga de manejar el comando que se ha asignado al midlet para enviar los datos a través de la red o para volver al menú principal
METODOS QUE LLAMA get_ SenalesDigitales(), getString(), run(), invokeServlet();start();
ORDEN DEL ALGORITMO O(k)

94. METODO invokeServlet()
CLASE MostrarSenalesDigitales
PARAMETROS No recibe parámetros

DESCRIPCION DEL METODO Método que hace la conexión con el servidor Tomcat el cual ejecuta un servlet el cual hace referencia a una base de datos
METODOS QUE LLAMA (HttpConnection)Connector.open(url), setRequestMethod(HttpConnection.GET), setRequestProperty("IF-Modified-Since", "17 Dic 2006 13:39:14 GMT"); setRequestProperty("User-Agent","Profile/MIDP-2.0 Configuration/CLDC-1.0"); setRequestProperty("Content-Language", "es-ES"); setRequestProperty("Content-Type", "application/x-www-form-urlencoded"); openDataInputStream();getResponseCode(); getLength();read(); close();setRespuesta(respuesta); getResponseMessage();
ORDEN DEL ALGORITMO O(k)

95. METODO setRespuesta
CLASE MostrarSenalesDigitales
PARAMETROS (String res)
DESCRIPCION DEL METODO Método que encapsula el texto enviado
METODOS QUE LLAMA setText(res); append(txt);
ORDEN DEL ALGORITMO O(k)

96. METODO getRespuesta
CLASE MostrarSenalesDigitales
PARAMETROS No recibe parámetros
DESCRIPCION DEL METODO Método que retorna el texto
METODOS QUE LLAMA No llama ningún método
ORDEN DEL ALGORITMO O(k)

97. METODO commandAction
CLASE MostrarTemperaturas
PARAMETROS (Command command, Displayable displayable)
DESCRIPCION DEL METODO Método implementado para satisfacer la interfaz CommandListener que se encarga de manejar el comando que se ha asignado al midlet para enviar los datos a través de la red o para volver al menú principal
METODOS QUE LLAMA get_ Temperaturas(), getString(), run(), invokeServlet();start();
ORDEN DEL ALGORITMO O(k)

98. METODO invokeServlet()
CLASE MostrarTemperaturas
PARAMETROS No recibe parámetros

DESCRIPCION DEL METODO Método que hace la conexión con el servidor Tomcat el cual ejecuta un servlet el cual hace referencia a una base de datos
METODOS QUE LLAMA (HttpConnection)Connector.open(url), setRequestMethod(HttpConnection.GET), setRequestProperty("IF-Modified-Since", "17 Dic 2006 13:39:14 GMT"); setRequestProperty("User-Agent","Profile/MIDP-2.0 Configuration/CLDC-1.0"); setRequestProperty("Content-Language", "es-ES"); setRequestProperty("Content-Type", "application/x-www-form-urlencoded"); openDataInputStream();getResponseCode(); getLength();read(); close();setRespuesta(respuesta); getResponseMessage();
ORDEN DEL ALGORITMO O(k)

99. METODO setRespuesta
CLASE MostrarTemperaturas
PARAMETROS (String res)
DESCRIPCION DEL METODO Método que encapsula el texto enviado
METODOS QUE LLAMA setText(res); append(txt);
ORDEN DEL ALGORITMO O(k)

100. METODO getRespuesta
CLASE MostrarTemperaturas
PARAMETROS No recibe parámetros
DESCRIPCION DEL METODO Método que retorna el texto
METODOS QUE LLAMA No llama ningún método
ORDEN DEL ALGORITMO O(k)

101. METODO commandAction
CLASE Mostrar VariablesMezcla
PARAMETROS (Command command, Displayable displayable)
DESCRIPCION DEL METODO Método implementado para satisfacer la interfaz CommandListener que se encarga de manejar el comando que se ha asignado al midlet para enviar los datos a través de la red o para volver al menú principal
METODOS QUE LLAMA get_ VariablesMezcla(), getString(), run(), invokeServlet();start();
ORDEN DEL ALGORITMO O(k)

102. METODO invokeServlet()
CLASE MostrarVariablesMezcla
PARAMETROS No recibe parámetros

DESCRIPCION DEL METODO Método que hace la conexión con el servidor Tomcat el cual ejecuta un servlet el cual hace referencia a una base de datos
METODOS QUE LLAMA (HttpConnection)Connector.open(url), setRequestMethod(HttpConnection.GET), setRequestProperty("IF-Modified-Since", "17 Dic 2006 13:39:14 GMT"); setRequestProperty("User-Agent","Profile/MIDP-2.0 Configuration/CLDC-1.0"); setRequestProperty("Content-Language", "es-ES"); setRequestProperty("Content-Type", "application/x-www-form-urlencoded"); openDataInputStream();getResponseCode(); getLength();read(); close();setRespuesta(respuesta); getResponseMessage();
ORDEN DEL ALGORITMO O(k)

103. METODO setRespuesta
CLASE MostrarVariablesMezcla
PARAMETROS (String res)
DESCRIPCION DEL METODO Método que encapsula el texto enviado
METODOS QUE LLAMA setText(res); append(txt);
ORDEN DEL ALGORITMO O(k)

104. METODO getRespuesta
CLASE MostrarVariablesMezcla
PARAMETROS No recibe parámetros
DESCRIPCION DEL METODO Método que retorna el texto
METODOS QUE LLAMA No llama ningún método
ORDEN DEL ALGORITMO O(k)

105. METODO init
CLASE insertarVariableMezclaMid
PARAMETROS (ServletConfig config)
DESCRIPCION DEL METODO Método inicia la conexión a la bases de dator R22
METODOS QUE LLAMA getConnection(connectionUrl);
ORDEN DEL ALGORITMO O(k)

106. METODO doPost
CLASE insertarVariableMezclaMid
PARAMETROS (HttpServletRequest request, HttpServletResponse response)
DESCRIPCION DEL METODO Método que recibe los parámetros enviados por el cliente.
METODOS QUE LLAMA getWriter();getInstance();get(Calendar.DATE); get(Calendar.MONTH); get(Calendar.YEAR); get(Calendar.HOUR);

get(Calendar.MINUTE); request.getParameter("Valor");
response.setContentLength(msg.length());response.getWriter();
println(msg); close(); conn.createStatement();stmt.executeUpdate(qry);
getString();
ORDEN DEL ALGORITMO O(k)

107. METODO init
CLASE verVariablesMezclaMid
PARAMETROS (ServletConfig config)
DESCRIPCION DEL METODO Método inicia la conexión a la bases de dator R22
METODOS QUE LLAMA getConnection(connectionUrl);
ORDEN DEL ALGORITMO O(k)

108. METODO doGet
CLASE verVariablesMezclaMid
PARAMETROS (HttpServletRequest request, HttpServletResponse response)
DESCRIPCION DEL METODO Método que recibe los parámetros enviados por el cliente.
METODOS QUE LLAMA getWriter();
response.setContentLength(msg.length()); response.getWriter();
println(msg); close(); conn.createStatement(); stmt.executeUpdate(qry);
ORDEN DEL ALGORITMO O(k)

109. METODO init
CLASE verFallasProcesoMid
PARAMETROS (ServletConfig config)
DESCRIPCION DEL METODO Método inicia la conexión a la bases de dator R22
METODOS QUE LLAMA getConnection(connectionUrl);
ORDEN DEL ALGORITMO O(k)

110. METODO doGet
CLASE verFallasProcesoMid
PARAMETROS (HttpServletRequest request, HttpServletResponse response)
DESCRIPCION DEL METODO Método que recibe los parámetros enviados por el cliente.
METODOS QUE LLAMA getWriter();
response.setContentLength(msg.length()); response.getWriter();
println(msg); close(); conn.createStatement(); stmt.executeUpdate(qry);
getString();

ORDEN DEL ALGORITMO O(k)

111. METODO init
CLASE verNivelTanqueMid
PARAMETROS (ServletConfig config)
DESCRIPCION DEL METODO Método inicia la conexión a la bases de dator R22
METODOS QUE LLAMA getConnection(connectionUrl);
ORDEN DEL ALGORITMO O(k)

112. METODO doGet
CLASE verNivelTanqueMid
PARAMETROS (HttpServletRequest request, HttpServletResponse response)
DESCRIPCION DEL METODO Método que recibe los parámetros enviados por el cliente.
METODOS QUE LLAMA getWriter();
response.setContentLength(msg.length()); response.getWriter(); println(msg); close(); conn.createStatement(); stmt.executeUpdate(qry);
ORDEN DEL ALGORITMO O(k)

113. METODO init
CLASE verPesosTanquesMid
PARAMETROS (ServletConfig config)
DESCRIPCION DEL METODO Método inicia la conexión a la bases de dator R22
METODOS QUE LLAMA getConnection(connectionUrl);
ORDEN DEL ALGORITMO O(k)

114. METODO doGet
CLASE verPesosTanquesMid
PARAMETROS (HttpServletRequest request, HttpServletResponse response)
DESCRIPCION DEL METODO Método que recibe los parámetros enviados por el cliente.
METODOS QUE LLAMA getWriter();
response.setContentLength(msg.length()); response.getWriter(); println(msg); close(); conn.createStatement(); stmt.executeUpdate(qry); getString();
ORDEN DEL ALGORITMO O(k)

115. METODO init
CLASE verSenalesDigitalesMid

PARAMETROS (ServletConfig config)
DESCRIPCION DEL METODO Método inicia la conexión a la bases de dator R22
METODOS QUE LLAMA getConnection(connectionUrl);
ORDEN DEL ALGORITMO O(k)

116. METODO doGet
CLASE verSenalesDigitalesMid
PARAMETROS (HttpServletRequest request, HttpServletResponse response)
DESCRIPCION DEL METODO Método que recibe los parámetros enviados por el cliente.
METODOS QUE LLAMA getWriter();
response.setContentLength(msg.length()); response.getWriter();
println(msg); close(); conn.createStatement(); stmt.executeUpdate(qry);
getString();
ORDEN DEL ALGORITMO O(k)

117. METODO init
CLASE verTemperaturaMid
PARAMETROS (ServletConfig config)
DESCRIPCION DEL METODO Método inicia la conexión a la bases de dator R22
METODOS QUE LLAMA getConnection(connectionUrl);
ORDEN DEL ALGORITMO O(k)

118. METODO doGet
CLASE verTemperaturaMid
PARAMETROS (HttpServletRequest request, HttpServletResponse response)
DESCRIPCION DEL METODO Método que recibe los parámetros enviados por el cliente.
METODOS QUE LLAMA getWriter();
response.setContentLength(msg.length()); response.getWriter();
println(msg); close(); conn.createStatement(); stmt.executeUpdate(qry);
getString();
ORDEN DEL ALGORITMO O(k)

119. METODO init
CLASE MostrarFallasProcesoMid
PARAMETROS (ServletConfig config)
DESCRIPCION DEL METODO Método inicia la conexión a la bases de dator R22

METODOS QUE LLAMA getConnection(connectionUrl);
ORDEN DEL ALGORITMO O(k)

120. METODO doGet
CLASE MostrarFallasProcesoMid
PARAMETROS (HttpServletRequest request, HttpServletResponse response)
DESCRIPCION DEL METODO Método que recibe los parámetros enviados por el cliente.
METODOS QUE LLAMA getWriter();
response.setContentLength(msg.length()); response.getWriter();
println(msg); close(); conn.createStatement(); stmt.executeUpdate(qry);
getString();
ORDEN DEL ALGORITMO O(k)

121. METODO init
CLASE MostrarNivelesTanquesMid
PARAMETROS (ServletConfig config)
DESCRIPCION DEL METODO Método inicia la conexión a la bases de dator R22
METODOS QUE LLAMA getConnection(connectionUrl);
ORDEN DEL ALGORITMO O(k)

122. METODO doGet
CLASE MostrarNivelesTanquesMid
PARAMETROS (HttpServletRequest request, HttpServletResponse response)
DESCRIPCION DEL METODO Método que recibe los parámetros enviados por el cliente.
METODOS QUE LLAMA getWriter();
response.setContentLength(msg.length()); response.getWriter();
println(msg); close(); conn.createStatement(); stmt.executeUpdate(qry);
getString();
ORDEN DEL ALGORITMO O(k)

123. METODO init
CLASE MostrarPesosTanquesMid
PARAMETROS (ServletConfig config)
DESCRIPCION DEL METODO Método inicia la conexión a la bases de dator R22
METODOS QUE LLAMA getConnection(connectionUrl);
ORDEN DEL ALGORITMO O(k)

124. METODO doGet
CLASE MostrarPesosTanquesMid
PARAMETROS (HttpServletRequest request, HttpServletResponse response)
DESCRIPCION DEL METODO Método que recibe los parámetros enviados por el cliente.
METODOS QUE LLAMA getWriter(); response.setContentLength(msg.length()); response.getWriter(); println(msg); close(); conn.createStatement(); stmt.executeUpdate(qry); getString();
ORDEN DEL ALGORITMO O(k)

125. METODO init
CLASE MostrarSenalesDigitalesMid
PARAMETROS (ServletConfig config)
DESCRIPCION DEL METODO Método inicia la conexión a la bases de dator R22
METODOS QUE LLAMA getConnection(connectionUrl);
ORDEN DEL ALGORITMO O(k)

126. METODO doGet
CLASE MostrarSenalesDigitalesMid
PARAMETROS (HttpServletRequest request, HttpServletResponse response)
DESCRIPCION DEL METODO Método que recibe los parámetros enviados por el cliente.
METODOS QUE LLAMA getWriter(); response.setContentLength(msg.length()); response.getWriter(); println(msg); close(); conn.createStatement(); stmt.executeUpdate(qry); getString();
ORDEN DEL ALGORITMO O(k)

127. METODO init
CLASE MostrarTemperaturasMid
PARAMETROS (ServletConfig config)
DESCRIPCION DEL METODO Método inicia la conexión a la bases de dator R22
METODOS QUE LLAMA getConnection(connectionUrl);
ORDEN DEL ALGORITMO O(k)

128. METODO doGet
CLASE MostrarTemperaturasMid

PARAMETROS (HttpServletRequest request, HttpServletResponse response)
DESCRIPCION DEL METODO Método que recibe los parámetros enviados por el cliente.
METODOS QUE LLAMA getWriter(); response.setContentLength(msg.length()); response.getWriter(); println(msg); close(); conn.createStatement(); stmt.executeUpdate(qry); getString();
ORDEN DEL ALGORITMO O(k)

129. METODO init
CLASE MostrarVariablesMezclaMid
PARAMETROS (ServletConfig config)
DESCRIPCION DEL METODO Método inicia la conexión a la bases de dator R22
METODOS QUE LLAMA getConnection(connectionUrl);
ORDEN DEL ALGORITMO O(k)

130. METODO doGet
CLASE MostrarVariablesMezclaMid
PARAMETROS (HttpServletRequest request, HttpServletResponse response)
DESCRIPCION DEL METODO Método que recibe los parámetros enviados por el cliente.
METODOS QUE LLAMA getWriter(); response.setContentLength(msg.length()); response.getWriter(); println(msg); close(); conn.createStatement(); stmt.executeUpdate(qry); getString();
ORDEN DEL ALGORITMO O(k)

131. METODO Public void run()
CLASE lecturaTemperaturas
PARAMETROS (vacio)
DESCRIPCION DEL METODO Método que es utilizador para hacer las acciones del hilo para realizar una lectura.
METODOS QUE LLAMA opcgroup(groupname, active, updaterate, percentdeadband), opcitem(itemname, active, accesspath), coInitialize(), additem(opc item), synchReadItem(opcgroup, opcitem)
ORDEN DEL ALGORITMO Método Run del hilo O(n) donde n es el numero de variables a examinar, y este se tiende a hacia el infinito

132. METODO Public void run()
CLASE lecturaPesosTanques
PARAMETROS (vacio)
DESCRIPCION DEL METODO Método que es utilizador para hacer las acciones del hilo para realizar una lectura.
METODOS QUE LLAMA opcgroup(groupname, active, updaterate, percentdeadband), opcitem(itemname, active, accesspath), coInitialize(), additem(opc item), synchReadItem(opcgroup, opcitem)
ORDEN DEL ALGORITMO Método Run del hilo O(n) donde n es el numero de variables a examinar, y este se tiende a hacia el infinito

133. METODO Public void run()
CLASE lecturaNivelesTanques
PARAMETROS (vacio)
DESCRIPCION DEL METODO Método que es utilizador para hacer las acciones del hilo para realizar una lectura.
METODOS QUE LLAMA opcgroup(groupname, active, updaterate, percentdeadband), opcitem(itemname, active, accesspath), coInitialize(), additem(opc item), synchReadItem(opcgroup, opcitem)
ORDEN DEL ALGORITMO Método Run del hilo O(n) donde n es el numero de variables a examinar, y este se tiende a hacia el infinito.

134. METODO Public void run()
CLASE lecturaFallasProceso
PARAMETROS (vacio)
DESCRIPCION DEL METODO Método que es utilizador para hacer las acciones del hilo para realizar una lectura.
METODOS QUE LLAMA opcgroup(groupname, active, updaterate, percentdeadband), opcitem(itemname, active, accesspath), coInitialize(), additem(opc item), synchReadItem(opcgroup, opcitem)
ORDEN DEL ALGORITMO Método Run del hilo O(n) donde n es el numero de variables a examinar, y este se tiende a hacia el infinito

135. METODO Public void run()
CLASE escribirVariableMezcla
PARAMETROS (vacio)
DESCRIPCION DEL METODO Método que es utilizador para hacer las acciones del hilo para realizar una escritura.
METODOS QUE LLAMA opcgroup(groupname, active, updaterate, percentdeadband), opcitem(itemname, active, accesspath), coInitialize(),

additem(opc item), addgroup(opc group), synchReadItem(opcgroup, opcitem), registerGroup(opc group), synchWriteItem(opc group, opc item)
ORDEN DEL ALGORITMO Método Run del hilo O(n) donde n es el numero de variables a examinar, y este se tiende a hacia el infinito

136. METODO Public void run()
CLASE escribirSeñalesDigitales
PARAMETROS (vacio)
DESCRIPCION DEL METODO Método que es utilizador para hacer las acciones del hilo para realizar una escritura.
METODOS QUE LLAMA opcgroup(groupname, active, updaterate, percentdeadband), opcitem(itemname, active, accesspath), coInitialize(), additem(opc item), addgroup(opc group), synchReadItem(opcgroup, opcitem), registerGroup(opc group), synchWriteItem(opc group, opc item)
ORDEN DEL ALGORITMO Método Run del hilo O(n) donde n es el numero de variables a examinar, y este se tiende a hacia el infinito

16.2 ESTADÍSTICA GENERAL

KLDC
3647/1000= 3.647

KLDD
1405/1000= 1.405

CLASES
35

METODOS
136

KLDC/HORA
3.647/ 175 = 0.02084

KLDD/HORA
1.405/35 = 0.04014286

ERR/CLASE
1/35=0.0286

KLDC/DEV

3.647/2=1.8235

KLDD/DEV
1.405/2= 0.7025

PAG/KLDC

97/3.647=26.6

16.3 PUNTOS DE FUNCIÓN

1. Número de entradas (Formularios de Inserccion) = 4.
2. Número de salidas (Consultas y Reportes) = 26
3. Número de peticiones (Update, delete, procesos) = 7
4. Número de Archivos = 37
5. Número de interfaces externas = 0

16.3.1 FACTOR PONDERADO

SIMPLE	
4 x 3	12
26 x 4	104
7 x 3	21
37 x 3	111
0 x 7	0
TOTAL	**248**

PF = Cantidad total * [0.65+ (0.01x6 ($\sum fi$))]

PF = 248 x [0.65+ (0.01x 6x34)]
PF = 667.12
PF = 400 – 600 = simple
600 – 800 = medio
> 800 = complejo

KLDC/PF = 3.647/667.12 = 0,00546678

PF/HORA = 667.12/175 = 3,12388E-05

PF/DEV = 667.12/2 = 333,56