



Universidad
Carlos III de Madrid

Departamento de Tecnología Electrónica

PROYECTO FIN DE CARRERA

Ingeniería de Telecomunicación

Sistema de Monitorización Continua de Descargas Parciales para Trasformadores de Alta Potencia

Autor: Daniel Lastra Rodríguez

Tutor: Jose Antonio García Souto

Leganés, Junio de 2012

Título: ***Sistema de Monitorización Continua de Descargas Parciales para Transformadores de Alta Potencia***

Autor: ***Daniel Lastra Rodríguez***

Director: ***Jose Antonio García Souto***

EL TRIBUNAL

Presidente: _____

Vocal: _____

Secretario: _____

Realizado el acto de defensa y lectura del Proyecto Fin de Carrera el día ___ de _____ de 20__ en Leganés, en la Escuela Politécnica Superior de la Universidad Carlos III de Madrid, acuerda otorgarle la CALIFICACIÓN de

VOCAL

SECRETARIO

PRESIDENTE

Agradecimientos

A Raquel.

A mi padre, mi madre y mis hermanos.

A familiares y amigos.

Me habéis convertido en lo que soy así que esto también es vuestro.

Resumen

La monitorización de transformadores de alta potencia se presenta como una tarea crucial dentro del mantenimiento de las subestaciones eléctricas. Además de pérdidas de suministro, los fallos en los transformadores de alta potencia llevan asociados una serie de costes que en ocasiones pueden llegar a dispararse.

Mientras se encuentra en servicio, un transformador de alta potencia proporciona gran variedad de información entre la que podemos destacar la facilitada por las descargas parciales producidas en el propio aislante del transformador. Estas se traducen en emisiones acústicas que con el sistema de adquisición adecuado se pueden capturar para posteriores análisis. Uno de esos sistemas de adquisición es la Tarjeta de Adquisición PDA14 de Signatec.

Esta tarea de monitorización se convierte en monitorización continua ofreciendo la posibilidad de interactuar de forma remota con el sistema de monitorización. La interacción remota, ya sea en entornos experimentales o en la monitorización continua de transformadores de potencia es posible gracias a avances tecnológicos como Java y en particular Java Enterprise Edition, que además proporciona los mecanismos necesarios para solucionar algunos de los inconvenientes derivados de los sistemas distribuidos, como por ejemplo, la seguridad.

De esta forma, se ha desarrollado un Sistema de Monitorización Continua de Descargas Parciales para Transformadores de Alta Potencia que, gracias a sus funcionalidades, permite avanzar en la prevención y detección de fallos. Entre estas funcionalidades conviene destacar la capacidad de configurar la tarjeta de adquisición PDA14 de Signatec para almacenar, visualizar y descargar las emisiones acústicas generadas por las descargas parciales producidas en el aislamiento de un transformador en funcionamiento.

Palabras clave: Monitorización Continua, Transformadores de Alta Potencia, Descargas Parciales, Emisiones Acústicas, PDA14, Interacción Remota, Seguridad Web, Servlets, Java Enterprise Edition

Abstract

The monitoring of high power transformers is presented as a crucial task in the maintenance of electric substations. In addition to loss of supply, failures of high power transformers are associated with a series of costs that can sometimes skyrocket.

While it is in service, a high power transformer provides a variety of information from which we can highlight the information provided by partial discharges produced in the transformer's insulation itself. These are translated into acoustic emissions that with the appropriate procurement system can be captured for later analysis. One of these acquisition systems is Signatec PDA14.

This monitoring task turns into continuous monitoring offering the possibility to interact remotely with the monitoring system. Web based remote control, either in experimental environments or in the continuous monitoring of power transformers is made possible by technological advances among which we find the Java technology and above the Java Platform Enterprise Edition, which also provides the mechanisms needed to provide the security of distributed systems.

In this way, a system has been developed for continuous monitoring of partial discharges for High Power Transformers that, thanks to its functionality, allows us to make progress in the prevention and detection of failures. Among these features it is important to emphasize the ability to configure Signatec PDA14 to store, view and download the acoustic emissions generated by partial discharges produced in the insulation of a transformer in operation.

Keywords: Online Monitoring, High Power Transformers, Partial Discharges, Acoustic Emissions, PDA14, Remote Control, Web Security, Servlets, Java Enterprise Edition

x

Índice de Contenidos

<u>CAPÍTULO I. INTRODUCCIÓN Y OBJETIVOS</u>	7
I.1. INTRODUCCIÓN	7
I.2. OBJETIVOS	7
I.3. FASES DEL DESARROLLO	7
I.4. MEDIOS EMPLEADOS	7
I.5. ESTRUCTURA DE LA MEMORIA	7
<u>CAPÍTULO II. MONITORIZACIÓN DE TRANSFORMADORES DE ALTA POTENCIA</u>	7
II.1. INTRODUCCIÓN	7
II.2. TRANSFORMADORES DE POTENCIA	7
II.3. FALLOS EN TRANSFORMADORES DE POTENCIA	7
II.3.1. FALLOS EXTERNOS	7
II.3.2. FALLOS INTERNOS	7
II.4. MONITORIZACIÓN Y DIAGNÓSTICO	7
II.5. PARÁMETROS DE MEDIDA EN MONITORIZACIÓN CONTINUA	7
II.5.1. MEDIDAS DE CARGA Y CONDICIONES DE OPERACIÓN	7
II.5.2. ANÁLISIS DE LA HUMEDAD	7
II.5.3. MEDIDA DE LA TEMPERATURA	7
II.5.4. ANÁLISIS DE LOS GASES DISUELTOS EN EL ACEITE AISLANTE	7
II.5.5. MEDICIÓN CONTINUA DE DESCARGAS PARCIALES Y ANÁLISIS DE LA FUNCIÓN DE TRANSFERENCIA	7
II.6. CONCLUSIONES	7
<u>CAPÍTULO III. DETECCIÓN DE EMISIONES ACÚSTICAS DE DESCARGAS PARCIALES EN TRANSFORMADORES</u>	7
III.1. INTRODUCCIÓN	7
III.2. DETECCIÓN Y MEDIDA DE DESCARGAS PARCIALES	7
III.3. EMISIONES ACÚSTICAS PROCEDENTES DE DESCARGAS PARCIALES	7

III.4. SISTEMAS ACÚSTICOS PARA DETECCIÓN DE DESCARGAS PARCIALES	7
<i>III.4.1. SISTEMAS COMPLETAMENTE ACÚSTICOS</i>	7
<i>III.4.2. SISTEMAS ACÚSTICOS CON DISPARO ELÉCTRICO</i>	7
III.5. INSTRUMENTACIÓN PARA DETECCIÓN DE EMISIONES ACÚSTICAS	7
<i>III.5.1. SENSORES EXTERNOS</i>	7
<i>III.5.2. SENSORES INTERNOS</i>	7
<i>III.5.3. SISTEMAS DE 3 SENSORES</i>	7
<i>III.5.4. FILTROS PASO BANDA</i>	7
III.6. SISTEMAS DE MONITORIZACIÓN CONTINUA	7
III.7. CONCLUSIONES	7
<u>CAPÍTULO IV. MEDIDA DE EMISIONES ACÚSTICAS CON LA TARJETA DE ADQUISICIÓN PDA14</u>	<u>7</u>
IV.1. INTRODUCCIÓN	7
IV.2. SISTEMAS DE ADQUISICIÓN DE DATOS	7
IV.3. LA TARJETA DE ADQUISICIÓN PDA14	7
<i>IV.3.1. DESCRIPCIÓN DEL HARDWARE</i>	7
<i>IV.3.2. DESCRIPCIÓN DEL SOFTWARE</i>	7
IV.4. DISCUSIÓN Y CONCLUSIONES	7
<u>CAPÍTULO V. INTERACCIÓN REMOTA CON ENTORNOS EXPERIMENTALES</u>	<u>7</u>
V.1. INTRODUCCIÓN	7
V.2. ENTORNOS EXPERIMENTALES EN LA ACTUALIDAD	7
<i>V.2.1. REQUISITOS</i>	7
<i>V.2.2. ACCESO REMOTO MEDIANTE APLICACIONES WEB</i>	7
<i>V.2.3. VENTAJAS DE LAS APLICACIONES WEB</i>	7
<i>V.2.4. APLICACIONES WEB ENRIQUECIDAS (RIA)</i>	7
<i>V.2.5. SEGURIDAD EN APLICACIONES WEB</i>	7
V.3. TECNOLOGÍAS WEB	7
<i>V.3.1. CGI</i>	7
<i>V.3.2. SERVLETS</i>	7
<i>V.3.3. SEGURIDAD EN SERVLETS</i>	7
V.4. JAVA PLATFORM, ENTERPRISE EDITION (JAVA EE)	7
<i>V.4.1. COMPONENTES DE LA ARQUITECTURA JAVA EE</i>	7
<i>V.4.2. CONTENEDORES JAVA EE</i>	7
<i>V.4.3. TECNOLOGÍAS JAVA EE</i>	7
<i>V.4.4. SERVIDORES DE APLICACIONES JAVA EE</i>	7
<u>CAPÍTULO VI. IMPLEMENTACIÓN Y PRUEBAS DEL SISTEMA</u>	<u>7</u>
VI.1. INTRODUCCIÓN	7
VI.2. FUNCIONALIDADES	7
VI.3. CASOS DE USO	7
<i>VI.3.1. DIAGRAMAS</i>	7
<i>VI.3.2. DESCRIPCIONES</i>	7
VI.4. ARQUITECTURA	7
VI.5. SOFTWARE DE ADQUISICIÓN	7

VI.5.1. JNI	7
VI.5.2. IMPLEMENTACIÓN CON JNI	7
VI.5.3. API JAVA	7
VI.6. HERRAMIENTA DE ADMINISTRACIÓN LOCAL	7
VI.7. INTERFAZ DE ACCESO REMOTO	7
VI.8. PRUEBAS Y RESULTADOS	7
<u>CAPÍTULO VII. CONCLUSIONES</u>	<u>7</u>
<u>CAPÍTULO VIII. PRESUPUESTO</u>	<u>7</u>
VIII.1. MATERIAL	7
VIII.2. PERSONAL	7
VIII.3. TOTAL	7
<u>ANEXO A. MANUAL DEL SISTEMA</u>	<u>7</u>
A.1. INTRODUCCIÓN	7
A.2. REQUISITOS MÍNIMOS	7
A.3. INSTALACIÓN Y EJECUCIÓN	7
A.4. GUÍA DE USUARIO	7
A.4.1. INICIO DE SESIÓN	7
A.4.2. ESCRITORIO	7
A.4.3. MENÚ DE INICIO	7
A.4.4. ADMINISTRACIÓN DE USUARIOS	7
A.4.5. CONFIGURACIÓN	7
A.4.6. MONITORIZACIÓN	7
A.4.7. ADQUISICIONES	7
<u>ANEXO B. MANUAL DE DESARROLLO</u>	<u>7</u>
B.1. DESARROLLO DE APLICACIONES PARA LA PDA14 CON MICROSOFT VISUAL C++ 2008 EXPRESS EDITION	7
B.1.1. CONFIGURACIÓN DE UN NUEVO PROYECTO	7
B.1.2. IMPLEMENTACIÓN DE LA APLICACIÓN	7
B.1.3. GENERACIÓN Y EJECUCIÓN DE LA APLICACIÓN	7
B.2. DESARROLLO DE APLICACIONES JAVA PARA LA PDA14 CON ECLIPSE IDE	7
B.2.1. CONFIGURACIÓN DE UN NUEVO PROYECTO	7
B.2.2. IMPLEMENTACIÓN DE LA APLICACIÓN	7
B.2.3. EJECUCIÓN DE LA APLICACIÓN	7
<u>BIBLIOGRAFÍA</u>	<u>7</u>
<u>HOJAS DE CARACTERÍSTICAS</u>	<u>7</u>

Índice de Figuras

FIGURA I.1 FASES DEL DESARROLLO DEL PROYECTO	7
FIGURA II.1 PROCEDIMIENTO GENERAL DE DISTRIBUCIÓN DE ENERGÍA.....	7
FIGURA II.2 GRÁFICA DE FALLOS EN EEUU ENTRE 1983 Y 1988.....	7
FIGURA II.3 COSTES ASOCIADOS A LOS FALLOS EN TRANSFORMADORES DE POTENCIA	7
FIGURA III.1 SEÑAL ACÚSTICA PROCEDENTE DE UNA DESCARGA PARCIAL [9]	7
FIGURA III.2 DETALLE DE SEÑAL ACÚSTICA PROCEDENTE DE UNA DESCARGA PARCIAL [9]	7
FIGURA III.3 ESQUEMA EXPERIMENTAL DEL SISTEMA INTERFEROMÉTRICO DE FIBRA ÓPTICA PARA LA DETECCIÓN DE DESCARGAS PARCIALES.....	7
FIGURA III.4 DIAGRAMA DE BLOQUES DE UN SISTEMA DE MONITORIZACIÓN CONTINUA.....	7
FIGURA IV.1 DIAGRAMA DE BLOQUES DE UN SISTEMA DE ADQUISICIÓN	7
FIGURA IV.2 TARJETA DE ADQUISICIÓN PDA14.....	7
FIGURA IV.3 DIAGRAMA FUNCIONAL DE TARJETA DE ADQUISICIÓN PDA14	7
FIGURA IV.4 VERSIÓN DEL SOFTWARE DE TARJETA DE ADQUISICIÓN PDA14	7
FIGURA V.1 ARQUITECTURA CLIENTE - SERVIDOR	7
FIGURA V.2 ARQUITECTURA MULTINIVEL PROPORCIONADA POR JAVA EE	7
FIGURA V.3 MODELO DE CONTENEDORES PROPORCIONADO POR JAVA EE.....	7
FIGURA VI.1 DIAGRAMA DE CASOS DE USO DEL USUARIO ADMINISTRADOR	7
FIGURA VI.2 DIAGRAMA DE CASOS DE USO DEL USUARIO OPERADOR.....	7
FIGURA VI.3 DIAGRAMA DE CASOS DE USO DEL USUARIO VISITANTE.....	7
FIGURA VI.4 ARQUITECTURA DEL SISTEMA.....	7
FIGURA VI.5 EXTRACTO DE LA CLASE PDA14	7
FIGURA VI.6 EXTRACTO DEL FICHERO DE ENCABEZADO PARA LA CLASE PDA14	7
FIGURA VI.7 EXTRACTO DEL FICHERO PDA14.C.....	7
FIGURA VI.8 INTEFAZ DE LA HERRAMIENTA DE ADMINISTRACIÓN LOCAL.....	7
FIGURA VI.9 EXPORTACIÓN DE ADQUISICIONES DESDE LA HERRAMIENTA DE ADMINISTRACIÓN LOCAL	7
FIGURA VI.10 INTEFAZ DE ACCESO REMOTO.....	7
FIGURA VI.11 SIMULACIÓN DE EMISIONES ACÚSTICAS	7
FIGURA VI.12 VISTA DEL SISTEMA DESDE EL PUESTO DE UN OPERADOR	7
FIGURA VI.13 ADQUISICIÓN DE PRUEBA CON LA SEÑAL DE DISPARO SOBRE EL CANAL 1	7
FIGURA VI.14 ADQUISICIÓN DE PRUEBA CON LA SEÑAL DE DISPARO SOBRE EL CANAL 2	7
FIGURA VI.15 ADQUISICIÓN DE PRUEBA DISMINUYENDO EL TAMAÑO DE LOS SEGMENTOS.....	7

FIGURA A.1 DIRECTORIO DE INSTALACIÓN DE LA APLICACIÓN.....	7
FIGURA A.2 EJECUCIÓN DE LA APLICACIÓN	7
FIGURA A.3 VISTA DE LA APLICACIÓN DESDE UN NAVEGADOR WEB	7
FIGURA A.4 INICIO DE SESIÓN EN LA APLICACIÓN	7
FIGURA A.5 ESCRITORIO DE LA APLICACIÓN	7
FIGURA A.6 MENÚ DE INICIO DE LOS USUARIOS CON PERFIL DE ADMINISTRADOR.....	7
FIGURA A.7 MENÚ DE INICIO DE LOS USUARIOS CON PERFIL DE OPERADOR	7
FIGURA A.8 MENÚ DE INICIO DE LOS USUARIOS CON PERFIL DE VISITANTE	7
FIGURA A.9 VENTANA DE ADMINISTRACIÓN DE USUARIOS	7
FIGURA A.10 VENTANA DE CONFIGURACIÓN.....	7
FIGURA A.11 VENTANA DE MONITORIZACIÓN.....	7
FIGURA A.12 VENTANA DE ADQUISICIONES	7
FIGURA A.13 VISUALIZACIÓN DE UNA SEÑAL ADQUIRIDA	7
FIGURA A.14 VISTA COMPLETA DE LA APLICACIÓN	7
FIGURA B.1 ENTORNO DE DESARROLLO MICROSOFT VISUAL C++ 2008 EXPRESS EDITION.....	7
FIGURA B.2 CREACIÓN DE UN NUEVO PROYECTO.....	7
FIGURA B.3 SELECCIÓN DE PROPIEDADES DEL PROYECTO	7
FIGURA B.4 ASISTENTE PARA APLICACIONES WIN32	7
FIGURA B.5 CONFIGURACIÓN DE LA APLICACIÓN	7
FIGURA B.6 NUEVO PROYECTO EN EL EXPLORADOR DE SOLUCIONES	7
FIGURA B.7 AGREGACIÓN DE ARCHIVO DE ENCABEZADO.....	7
FIGURA B.8 AGREGACIÓN DE ARCHIVOS DE RECURSOS	7
FIGURA B.9 AGREGACIÓN DE ARCHIVOS DE CÓDIGO FUENTE	7
FIGURA B.10 ASISTENTE PARA AGREGAR NUEVO ELEMENTO	7
FIGURA B.11 CODIFICACIÓN DEL NUEVO ELEMENTO.....	7
FIGURA B.12 GENERACIÓN DE LA SOLUCIÓN	7
FIGURA B.13 SOLUCIÓN GENERADA EN EL SISTEMA DE ARCHIVOS	7
FIGURA B.14 EJEMPLO DE EJECUCIÓN.....	7
FIGURA B.15 ESCRITORIO DE ECLIPSE IDE	7
FIGURA B.16 CREACIÓN DE NUEVO PROYECTO DESDE ECLIPSE IDE.....	7
FIGURA B.17 ASISTENTE PARA CREACIÓN DE PROYECTOS JAVA.....	7
FIGURA B.18 EXPLORADOR DE PAQUETES DEL ECLIPSE IDE.....	7
FIGURA B.19 NUEVA CARPETA EN EL PROYECTO	7
FIGURA B.20 ASISTENTE DE CREACIÓN DE CARPETAS.....	7
FIGURA B.21 EXPLORADOR DE PAQUETES CON LA CARPETA DE DEPENDENCIAS	7
FIGURA B.22 AÑADIR DEPENDENCIAS AL PROYECTO	7
FIGURA B.23 EXPLORADOR DE PAQUETES CON LAS DEPENDENCIAS AÑADIDAS.....	7
FIGURA B.24 AÑADIR NUEVA CLASE AL PROYECTO JAVA	7
FIGURA B.25 ASISTENTE DE CREACIÓN DE NUEVA CLASE.....	7
FIGURA B.26 EDITOR DEL ELIPSE IDE	7
FIGURA B.27 APLICACIÓN JAVA DE ADQUISICIÓN CON LA PDA14.....	7
FIGURA B.28 ARRANCAR ASISTENTES DE CONFIGURACIÓN DE EJECUCIONES	7
FIGURA B.29 AÑADIR NUEVA APLICACIÓN JAVA.....	7
FIGURA B.30 CONFIGURACIÓN DE LA EJECUCIÓN DE LA APLICACIÓN JAVA.....	7
FIGURA B.31 ARGUMENTOS DE LA EJECUCIÓN.....	7
FIGURA B.32 EJECUCIÓN DE LA APLICACIÓN	7
FIGURA B.33 EJECUCIÓN INCORRECTA DE LA APLICACIÓN	7

Índice de Tablas

TABLA II.1 ESTADÍSTICA DE FALLOS EN SUBESTACIONES EN EEUU ENTRE 1983 Y 1988.....	7
TABLA II.2 MONITORIZACIÓN PERIÓDICA Y MONITORIZACIÓN CONTINUA	7
TABLA III.1 VELOCIDAD DE PROPAGACIÓN DEL SONIDO EN EL ACEITE	7
TABLA IV.1 LIBRERÍA DE FUNCIONES DE LA PDA14 AGRUPADAS POR FUNCIONALIDAD.....	7
TABLA VI.1 DESCRIPCIÓN CASO DE USO: INICIAR SESIÓN	7
TABLA VI.2 DESCRIPCIÓN CASO DE USO: ALTA DE USUARIO	7
TABLA VI.3 DESCRIPCIÓN CASO DE USO: CONSULTA DE USUARIO	7
TABLA VI.4 DESCRIPCIÓN CASO DE USO: BAJA DE USUARIO	7
TABLA VI.5 DESCRIPCIÓN CASO DE USO: INICIAR ADQUISICIÓN.....	7
TABLA VI.6 DESCRIPCIÓN CASO DE USO: CANCELAR ADQUISICIÓN	7
TABLA VI.7 DESCRIPCIÓN CASO DE USO: MONITORIZAR EL SISTEMA.....	7
TABLA VI.8 DESCRIPCIÓN CASO DE USO: VISUALIZAR ADQUISICIÓN	7
TABLA VI.9 DESCRIPCIÓN CASO DE USO: EXPORTAR ADQUISICIÓN	7
TABLA VI.10 DESCRIPCIÓN CASO DE USO: CERRAR SESIÓN	7
TABLA VI.11 API DE LA LIBRERÍA JAVA PARA LA PDA14.....	7
TABLA VIII.1 COSTES DE MATERIAL	7
TABLA VIII.2 COSTES DE PERSONAL	7
TABLA VIII.3 TOTAL DE COSTES	7

Capítulo I

Introducción y Objetivos

I.1. Introducción

Muchas empresas están aprovechándose de la existencia de Internet para exponer sus operaciones a sus clientes. Actualmente, millones de personas utilizan diferentes tipos de sistemas de información basados en tecnologías Web por la facilidad de acceso que ésta ofrece. La gran mayoría de las empresas tienen ya presencia online. Por ejemplo, cualquier entidad bancaria permite realizar cualquier tipo de transacción monetaria, pago de servicios y otras operaciones más. ¿Por qué no aprovechar estos servicios de información para dar acceso a los entornos de experimentación y diagnóstico?

Sin embargo, estas aplicaciones de comercio electrónico no podrían haber tenido tanto éxito si no se hubieran añadido los correspondientes sistemas de seguridad que permiten realizar cualquier operación de forma segura.

En el caso de entornos experimentales, no sólo debemos resolver los problemas relacionados con la seguridad, además se presentan otros requisitos que dependen de la tarea de investigación que se quiera llevar a cabo. Aunque nos podemos encontrar con escenarios muy distintos, una característica muy común de los entornos de experimentación es la necesidad de monitorizar y registrar el comportamiento de diferentes variables dentro del proceso de investigación. Por ello, es necesario proporcionar los sistemas de adquisición de datos adecuados a cada caso, y las funcionalidades que permitan la configuración de dichos sistemas de adquisición y diferentes operaciones relacionadas con los datos adquiridos.

Existen en el mercado gran variedad de tarjetas de adquisición. Una de ellas, disponible en el Departamento de Tecnología Electrónica de la Universidad Carlos III de Madrid, es la Tarjeta de Adquisición PDA14 de Signatec, que ofrece una serie de funcionalidades que hacen

de esta tarjeta de adquisición la adecuada en determinados escenarios. Éste es el caso de la detección de descargas parciales producidas en el aislante de los transformadores de potencia y sus emisiones acústicas, cuyas ondas poseen un ancho de banda por encima de 1 MHz y que son detectadas en el rango de 20 kHz a 500 kHz.

Los transformadores de potencia se encuentran por lo general sumergidos en aceite, debido a sus buenas propiedades dieléctricas, de forma que se mantiene una temperatura de operación suficiente baja y se previenen los “puntos calientes” en cualquier parte del transformador. La degradación del aislante produce estos fenómenos denominados descargas parciales.

Además, la implementación de software adicional que amplíe el proporcionado por el fabricante permitiría aumentar las capacidades propias de la tarjeta. De este modo, aplicando diversas tecnologías Web podemos proporcionar la capacidad de interacción remota con el sistema de adquisición. Los Servidores de Aplicaciones son otra parte fundamental de este tipo de sistemas, proporcionando además la capacidad de resolver los problemas de vulnerabilidad en los sistemas de seguridad que se pueden llegar a producir.

I.2. Objetivos

El objetivo fundamental de este Proyecto Fin de Carrera es desarrollar un Sistema de Monitorización Continua de Descargas Parciales para Transformadores de Alta Potencia aprovechando y ampliando las características que proporciona la Tarjeta de Adquisición PDA14 de Signatec. Sobre la base de este objetivo principal se pretende conseguir una serie de objetivos parciales.

El primero de estos objetivos parciales es el de estudiar las características de las señales acústicas derivadas de las descargas parciales producidas en el aislante de los transformadores de alta potencia y argumentar la necesidad de sistemas de adquisición con ciertas características como las que proporciona la Tarjeta de Adquisición PDA14 de Signatec.

Por otro lado, la aparición de las tecnologías Web ha dado lugar a la aparición de diferentes modalidades de entornos de experimentación, como los entornos de experimentación remotos. Se presenta como otro objetivo parcial de este Proyecto Fin de Carrera la necesidad de ampliar el software de adquisición proporcionado por la Tarjeta de Adquisición PDA14 de Signatec para facilitar el acceso de forma remota a cada una de las funcionalidades que ofrece dicho dispositivo.

Por último, se pretende proporcionar una herramienta de trabajo que facilite el estudio y la detección de las descargas parciales generadas en el aislamiento de los transformadores de alta potencia, permitiendo mejorar la eficiencia con la cual se realiza actualmente la monitorización de estos equipos. El uso de esta herramienta permitirá reducir las tasas de fallos de los transformadores de alta potencia y alargar la vida útil de los mismos, ya que

permitirá pasar del tradicional **mantenimiento correctivo** a un **mantenimiento predictivo** cuya política está basada en el **diagnóstico**.

A la finalización del proyecto se pretende obtener un producto basado en diferentes componentes hardware y, sobre todo, software que, una vez instalado, permitan llevar a cabo la tarea de monitorizar los transformadores de alta potencia, ya se encuentren en la laboratorios de investigación o en subestaciones eléctricas, y sea cual sea la localización del individuo que pretenda llevar a cabo la tarea.

1.3. Fases del desarrollo

El desarrollo de la solución completa se divide en diferentes fases que se corresponden con las metodologías ofrecidas por la Ingeniería del Software [1].

La Tarjeta de Adquisición PDA14 de Signatec, proporcionada por el Departamento de Tecnología Electrónica de la Universidad Carlos III de Madrid, es el principal componente sobre el que gira este proyecto. Por ello, tras un **Estudio de Viabilidad** basado en una serie de Pruebas de Concepto sobre este dispositivo y sobre el software proporcionado por el fabricante para controlarlo, antes de comenzar con la implementación, fue necesario un profundo **Análisis de los Requisitos**. Para ello, se realizó un estudio acerca de los Transformadores de Alta Potencia en el que se pretendía conocer la importancia de este tipo de equipos eléctricos, las ventajas de la monitorización de los mismos y qué indicadores se pueden utilizar para conocer su estado.

Asimismo, en el Análisis de Requisitos se ha tenido en cuenta la localización habitual de este tipo de equipos eléctricos, añadiendo a los requisitos propios de la tarea que se pretende llevar a cabo con el sistema, otros requisitos relacionados con la deslocalización entre los equipos que llevamos a estudio y los equipos de supervisión de los sistemas.

Tras el Análisis de Requisitos se elaboró el **Diseño Funcional** con el que se estableció la Arquitectura general de la solución con una serie de componentes que representan cada uno de los requisitos obtenidos en la fase anterior.

En paralelo con el Diseño Funcional, se inició la fase de **Implementación del Sistema**, teniendo en cuenta los lenguajes de programación utilizados, así como los primeros bocetos del diseño que existían en esa fase.

Las primeras implementaciones dieron lugar a las primeras versiones del sistema que con diferentes simuladores concluyeron en las primeras **Pruebas** del sistema.

La última fase de este desarrollo es la **Documentación** concretada en la realización de esta Memoria.

En muchas ocasiones se habla de este proceso de desarrollo como un proceso secuencial. Aunque se han seguido cada una de estas fases con rigor, merece la pena

destacar como en la realización de este proyecto, al haber sido una única persona la que ha asumido los roles de analista funcional y programador, no se ha tratado exactamente de un proceso secuencial. Como se puede apreciar en la **Figura I.1**, varias fases se han ido solapando a lo largo del desarrollo de la solución.

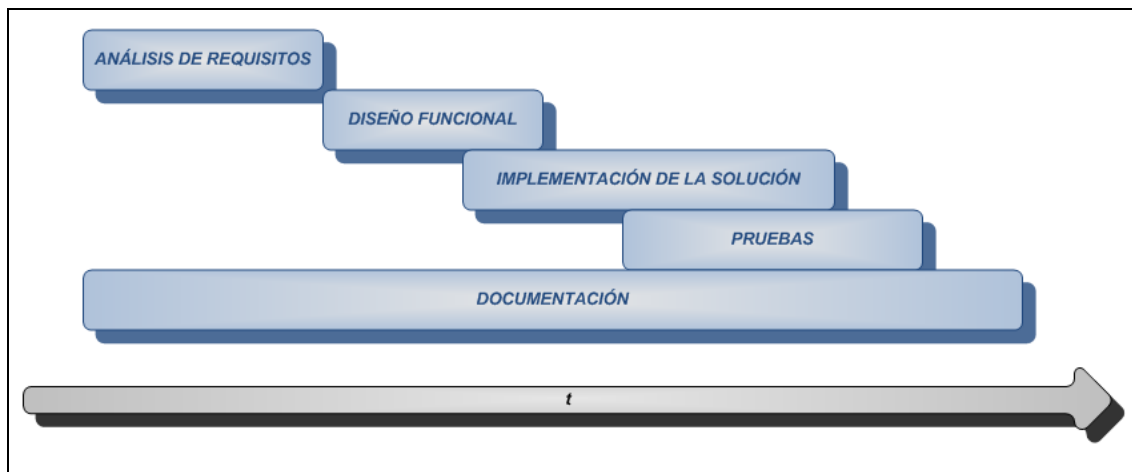


Figura I.1 Fases del Desarrollo del Proyecto

I.4. Medios empleados

La implementación de esta solución ha consistido, en su mayor parte, en el desarrollo de un software que permita ofrecer una nueva herramienta de monitorización, y por tanto, la mayoría de los medios que se han empleado tienen que ver con herramientas utilizadas típicamente en el desarrollo de software.

La herramienta de trabajo más importante de las que componen el conjunto de utilidades empleadas en el desarrollo de software es el Entorno de Desarrollo Integrado (IDE). Por definición, un IDE debe proporcionar un editor de texto, un compilador, un intérprete, un depurador, un sistema de control de versiones y debe facilitar la construcción de interfaces gráficas de usuario. Este es el caso de **Eclipse 3.4.0 (Ganymede)** que es uno de los IDE con los que se ha trabajado durante todo el desarrollo. Por haber trabajado con diferentes lenguajes de programación, ha sido necesario emplear otros IDE. Uno de ellos ha sido **Microsoft Visual C++ 2008 Express Edition**. El otro IDE que se ha empleado y que ha permitido completar el sistema con una interfaz de usuario mucho más elegante en cuanto a funcionalidad y estética se refiere, ha sido **Adobe Flash Builder 4.5**, que es una herramienta de desarrollo basada en Eclipse muy productiva y que facilita el desarrollo de aplicaciones Web, móviles y de escritorio.

Por otro lado, y para completar el entorno de desarrollo, se ha empleado un sistema de control de versiones, basado en **Subversion [2]**. Se trata de software libre bajo una licencia de tipo Apache/BSD y se le conoce también como **SVN**.

Una característica importante de *Subversion* es que, a diferencia de otros sistemas de control de versiones anteriores como **CVS [3]**, los archivos versionados no tienen cada uno un número de revisión independiente, si no que todo el repositorio tiene un único número de versión que identifica un estado común de todos los archivos del repositorio en un instante determinado.

Subversion permite el acceso al repositorio a través de redes, lo que facilita su utilización a personas que se encuentran trabajando en distintos equipos. A cierto nivel, aunque no es nuestro caso, la posibilidad de que varias personas puedan modificar y administrar el mismo conjunto de datos desde sus respectivas ubicaciones fomenta la colaboración. En nuestro caso, solo ha habido una persona realizando el desarrollo. Sin embargo, éste se ha llevado a cabo desde diferentes localizaciones, una de ellas, el laboratorio donde se encuentra un equipo en el que está instalada la Tarjeta de Adquisición PDA14 de Signatec y donde se han realizado las pruebas. De esta forma, se ha podido progresar más rápidamente. Además, el trabajo se encuentra bajo un control de versiones que permite deshacer con facilidad los cambios incorrectos.

Existen en el mercado diferentes repositorios a disposición de los usuarios cuyo coste puede variar en función de las funcionalidades proporcionadas. En nuestro caso, hemos empleado **RiouxSVN** que ofrece un tipo de licencia gratuita cuyas características de espacio y número de usuarios y proyectos son suficientes para nuestro proyecto.

Asimismo, se ha planteado un entorno de ensayos acústicos con simulación a escala de las emisiones acústicas debidas a las descargas parciales producidas en el aislamiento de los transformadores de potencia y un sistema de adquisición de datos con el que procesar las señales producidas por dichas emisiones acústicas.

1.5. Estructura de la Memoria

A continuación un breve resumen de cada uno de los capítulos que la componen la memoria.

En este primer capítulo de **INTRODUCCIÓN Y OBJETIVOS** podemos encontrar una breve explicación de la idea de la que se ha partido para desarrollar de este Proyecto Fin de Carrera, los objetivos fundamentales que se pretende alcanzar a la finalización del mismo, cada una de las fases en las que se ha dividido y los medios que se han utilizado a lo largo del desarrollo.

Como punto de partida, en el segundo capítulo se evaluará la importancia de la **MONITORIZACIÓN DE TRANSFORMADORES DE POTENCIA**. La idea es presentar la labor de los transformadores de potencia dentro del sistema eléctrico, cuáles son los fallos más importantes que se pueden producir, y sobre todo, que consecuencias pueden acarrear estos fallos. De este modo, la monitorización de ciertas magnitudes proporcionadas por los

transformadores de potencia se convierte en una tarea crucial en el mantenimiento de este tipo de equipos eléctricos.

El tercer capítulo, denominado **DETECCIÓN DE EMISIONES ACÚSTICAS DE DESCARGAS PARCIALES EN TRANSFORMADORES** se centrará en una de las magnitudes que permite conocer el estado de un transformador de potencia. Las descargas parciales producen unas emisiones acústicas cuyas características de amplitud y frecuencia son particulares. Veremos qué sistemas de adquisición son los más adecuados para su detección y localización. El estudio de este tipo de señales permitirá comprender que no todos los sistemas de adquisición son aptos para este tipo de aplicaciones.

El cuarto capítulo, denominado **MEDIDA DE EMISIONES ACÚSTICAS CON LA TARJETA DE ADQUISICIÓN PDA14**, mostrará las características más importantes de la Tarjeta de Adquisición PDA14 de Signatec. Tras una breve introducción a los Sistemas de Adquisición, se proporcionará información muy detallada de las características principales de esta tarjeta y que la convierten en una herramienta muy eficaz para la detección de las emisiones acústicas proporcionadas por las descargas parciales.

En el quinto capítulo se introduce el paradigma de la **INTERACCIÓN REMOTA CON ENTORNOS EXPERIMENTALES**. En este capítulo se mostrarán algunas de las tecnologías que permiten llevar esta idea a la práctica y entre las que destaca Java y sobre todo la plataforma empresarial de Java (Java Platform Enterprise Edition).

En el sexto capítulo, **IMPLEMENTACIÓN Y PRUEBAS DEL SISTEMA**, se presentan las funcionalidades, la arquitectura planteada y otros aspectos relacionados con el desarrollo del Sistema de Monitorización Continua de Descargas Parciales para Transformadores de Alta Potencia.

Para completar la memoria se han añadido dos anexos. El primero de ellos es el **MANUAL DEL SISTEMA**, con el que se pretende facilitar el aprendizaje a los posibles usuarios de la aplicación. El segundo anexo presenta un **MANUAL DE DESARROLLO** dirigido a posibles desarrolladores que en el futuro deseen añadir funcionalidades al sistema o simplemente deseen crear aplicaciones nuevas.

Capítulo II

Monitorización de Transformadores de Alta Potencia

II.1. Introducción

Hace más de un siglo que se inventaron los transformadores de potencia, haciendo posible la distribución de energía eléctrica a todos los hogares, industrias, etc. En definitiva, los transformadores son máquinas eléctricas estáticas que permiten modificar los valores de tensión y corriente con el fin de que éstos tomen los valores más adecuados para el transporte y distribución de la energía eléctrica, minimizando las pérdidas. Si no fuera por el transformador, la distancia que separa a los generadores de electricidad de los consumidores tendría que acortarse.

Los transformadores se encuentran entre los equipos eléctricos más importantes de todos los existentes en los sistemas eléctricos de potencia. Las fronteras de generación, transporte y distribución no podrían delimitarse sin la existencia de estas máquinas. Por este motivo, su mantenimiento resulta fundamental, no sólo para su buen funcionamiento, sino para la de todo el sistema eléctrico.

Aunque los transformadores de potencia están diseñados para tener un ciclo de vida útil determinado, su explotación suele prolongarse, de hecho gran parte del parque de operación eléctrica e industrial viene operando con máquinas muy fiables que ya han superado el límite previsto. Por eso, lo realmente importante es conocer el estado y evolución del transformador para estar en condiciones de poder operar con la máxima seguridad y saber si es apropiado continuar su uso, conocer la capacidad de sobrecarga, limitar la potencia, reacondicionarlo o, en su caso, retirarlo del servicio activo. Existen una serie de técnicas de mantenimiento que, desde el punto de vista eléctrico y a través de determinados ensayos de campo, van a permitir

seguir el estado del transformador y para, en caso de fallo, detectar con agilidad el problema acaecido y ejecutar las acciones oportunas.

Por otro lado, las condiciones de operación de los transformadores de potencia también han cambiado, se intenta sacar el máximo partido a cada equipo, funcionando a los máximos regímenes posibles y en ocasiones por encima del nominal. Esta forma de trabajar tiende a envejecer prematuramente el parque de máquinas y, si éstas no son objeto de un mínimo programa de mantenimiento que detecte situaciones de riesgo o de limitación de uso, la situación resultante conducirá a un comportamiento irregular de los equipos provocando fallos, paradas no programadas e interrupciones de suministro.

Debido al envejecimiento de los parques de transformadores de potencia, las empresas eléctricas están intentando minimizar las pérdidas de ingresos y los costes de reparación asociados con los fallos en estos equipos. De hecho, los transformadores están empezando a ser uno de los componentes más críticos en la red de transmisión eléctrica, y el estado del aislamiento líquido y sólido de los mismos una de las variables a tener muy en cuenta para reducir los fallos.

Por lo tanto, en subestaciones eléctricas sería conveniente monitorizar los elementos más críticos, entre los que se encuentran los transformadores de potencia y los interruptores.

II.2. Transformadores de Potencia

Estrictamente, un transformador es un equipo eléctrico que permite aumentar o disminuir la tensión en un circuito eléctrico de corriente alterna, manteniendo la frecuencia. La potencia que ingresa al equipo, en el caso de un transformador ideal (es decir, sin pérdidas), es igual a la que se obtiene a la salida. Las máquinas reales presentan un pequeño porcentaje de pérdidas, dependiendo de su diseño, tamaño y otros factores.

Al permanecer constante la potencia y aumentar la tensión, disminuye la intensidad. Este hecho es empleado para transportar la electricidad a grandes distancias reduciendo las pérdidas por efecto Joule. En una central eléctrica, el generador está conectado al primario de un transformador de elevación de tensión, mientras que las líneas de transporte de electricidad están conectadas al secundario. En el primario hay una intensidad alta, con un valor moderado de la tensión. En el secundario, la tensión se eleva y por consiguiente, la corriente en el secundario se reduce en la misma proporción. Como las pérdidas por efecto Joule son proporcionales al cuadrado de la intensidad, al disminuir la intensidad en el secundario se reducen las pérdidas por calentamiento.

La utilidad de los transformadores eléctricos se debe además a la reducción de costes que se obtiene al efectuar el transporte de la energía eléctrica a altas tensiones. Recordemos que, para la misma potencia, a mayor tensión menor corriente circulará por el conductor y el calibre de este será menor. Y un conductor de menor calibre es más barato. La sección o área

transversal del conductor necesaria en una línea de transmisión es inversamente proporcional al cuadrado del valor de la tensión que se haya adoptado para el transporte de la electricidad, explicando la conveniencia del empleo de altas tensiones en el transporte de la energía eléctrica.

Así como los transformadores se utilizan para elevar la tensión y permitir el transporte de la corriente a largas distancias, los transformadores también se utilizan para la reducción de la tensión a niveles aceptables para uso doméstico e industrial. En el otro extremo de la línea, debe utilizarse un transformador capaz de disminuir la tensión, de forma que podamos usar la electricidad de forma más cómoda y segura, y podamos disponer de una corriente de mayor intensidad.

Un **transformador de potencia** maneja grandes magnitudes de voltio amperios VA, los cuales se expresan en KVA [kilo voltio amperios] o en MVA [mega voltio amperios]. Usualmente se considera un transformador de potencia cuando su capacidad es de un valor a partir de: 500 KVA, 750 KVA, 1000 KVA, 1250 KVA o 1.25 MVA, hasta potencias del orden de 500 MVA monofásicos y de 650 MVA trifásicos, 900 MVA. Estos últimos operan en niveles de tensión de 500 KV, 525 KV y superiores.

Generalmente estos transformadores están instalados en subestaciones para la distribución de la energía eléctrica, efectuando la tarea intermediadora entre las grandes centrales de generación y, por ejemplo, industrias. Esta tarea consiste en reducir los altos niveles de tensión, con los que es transmitida la energía, a magnitudes de tensión inferiores, que permiten derivar circuitos a los usuarios en medias o bajas tensiones.

Por otro lado, los **transformadores de distribución** son, generalmente, los transformadores de potencias iguales o inferiores a 500 kVA y de tensiones iguales o inferiores a 67000 V, tanto monofásicos como trifásicos. Las aplicaciones típicas son para alimentar residencias, edificios o almacenes públicos, talleres y centros comerciales.

En la **Figura II.1** se muestra el procedimiento general de **distribución de energía** desde su generación hasta la entrega final de esta en la industria o para uso doméstico.

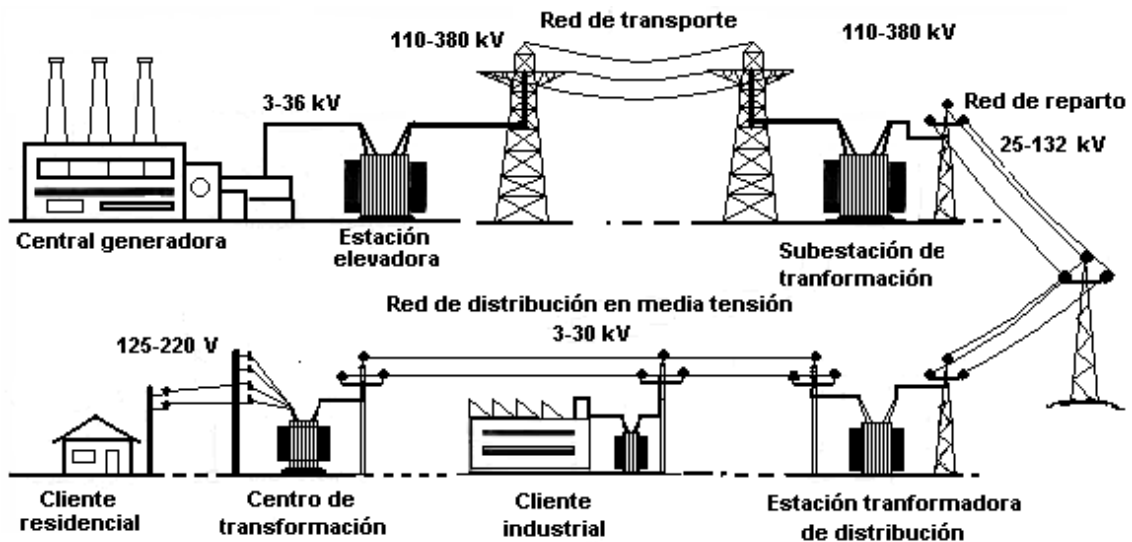


Figura II.1 Procedimiento General de Distribución de Energía

Los transformadores son elementos bastante fiables dentro de una red eléctrica. Su tasa de fallos aunque está aumentando en los últimos tiempos, sigue siendo baja. No obstante, algunos de sus fallos son muy costosos y en algunos casos catastróficos con muy serias consecuencias, por ejemplo, los incendios.

Por lo general, un transformador de potencia es un dispositivo muy confiable que está diseñado para alcanzar una vida útil entre 20 y 35 años y una vida mínima de 25 años a temperaturas de funcionamiento comprendidas entre 65 °C y 95 °C. Aunque en la práctica la vida de un transformador de potencia podría llegar a 60 años con un mantenimiento adecuado y en función de la fecha de fabricación, ya que se ha observado en transformadores producidos recientemente una edad promedio al fallo de 14,9 años en idénticas condiciones de trabajo [4].

Actualmente existen un gran número de transformadores cuyo tiempo de vida previsto ya se ha superado, por lo que se hace imprescindible conocer su estado, mediante diferentes métodos de supervisión, ya sean estos aplicados de forma continua o periódica, con el fin de extender su tiempo de vida útil o conocer con la suficiente antelación el momento en el cual es necesario sustituir el corazón de una subestación.

Uno de los componentes más importantes de los transformadores de potencia es el aislante que tiene una doble misión, refrigerante y aislante. Por eso, podemos clasificar los transformadores atendiendo al tipo de dieléctrico empleado en el aislamiento:

- ❖ **Transformadores de tipo seco**, cuyos materiales de aislamiento son sólidos o gaseosos. Normalmente estos transformadores tienen los arrollamientos embutidos en un bloque de resina epoxídica que, al tiempo que les confiere la necesaria rigidez, es ignífuga (difícilmente inflamable). El riesgo de incendio de los transformadores de tipo seco generalmente se considera bajo en comparación con los transformadores con aislamiento líquido, debido a la limitada cantidad de materiales combustibles presentes en ellos. Tienen el

inconveniente de que son más caros y que a altas tensiones y potencias, su refrigeración es bastante deficiente, por lo que no son usados en estos casos. Además, para tensiones elevadas (mayores a 66 kV), no se utilizan los transformadores en seco debido al riesgo de aparición de descargas parciales de gran intensidad.

- ❖ **Transformadores con aislamiento líquido de baja inflamabilidad**, cuyos aislantes líquidos usados son aceites aislantes de seguridad (de origen vegetal, mineral o sintético) indicados para transformadores que deben instalarse en lugares donde el riesgo de explosión y/o incendios deben minimizarse. Estos aceites deben presentar baja inflamabilidad y suelen ser principalmente de tres tipos: aceites minerales de alto punto de inflamabilidad, aceites vegetales y aceites aislantes de siliconas. El riesgo de incendio de estos transformadores es bastante más alto que el de los secos, debido a que el aislamiento líquido empleado no es ignífugo. Sin embargo, sus aceites poseen un punto de inflamación elevado comparado con los aceites minerales aislantes de uso general, aunque su capacidad de refrigeración empeora con respecto a éstos, debido a que tienen una viscosidad más alta.
- ❖ **Transformadores con aislamiento líquido inflamable**, donde los aislantes líquidos usados son aceites aislantes de uso general: Tienen origen mineral, los cuales poseen una base parafínica o nafténica, son obtenidos durante el proceso de refinación y extracción adecuado, de determinadas fracciones del petróleo natural. Estos transformadores son los que tienen mayor riesgo de incendio debido al bajo punto de inflamación del aceite, alrededor de los 140 °C. Sin embargo este tipo de aceite es el que mejor propiedades refrigerantes posee, es por ello que suele usarse en transformadores de gran potencia.

En los transformadores con aislamiento líquido, el circuito magnético y los devanados están sumergidos en el aceite dieléctrico.

II.3. Fallos en Transformadores de Potencia

Dentro de las redes eléctricas cobran vital importancia los grandes transformadores (aislados y refrigerados con aceite de origen mineral de uso general) de las centrales generadoras y subestaciones, ya que un fallo en éstos puede ocasionar importantes pérdidas económicas.

Ya se ha mostrado la importancia de los transformadores en los sistemas eléctricos de potencia. Desde el punto de vista de su explotación, la protección de estas máquinas frente a fallos resulta vital, debido al coste de los transformadores, especialmente aquellos de gran

potencia, sin perder de vista los elevados costes asociados a las pérdidas de producción debidas a la ausencia de alimentación eléctrica en la industria.

Antes de comenzar a analizar los distintos tipos de fallos que se producen en los transformadores, junto con los mecanismos de protección que pueden utilizarse para eliminar los efectos de dichos fallos sobre la máquina, conviene hacer un repaso rápido de los tipos de averías que aparecen en los transformadores, junto con su distribución estadística. En la **Tabla II.1** y en la **Figura II.2** se resumen las estadísticas de averías en transformadores de potencia, relativos a EEUU, durante el periodo comprendido entre 1983 y 1988, tal como están recogidas en la norma IEEE C37.91 - 2000 [5] dedicada a la protección de transformadores de potencia. Para la realización de dicha estadística, se analizaron 389 transformadores, agrupando sus fallos en función del elemento en el que se producía, tal como muestra la tabla.

<i>Elemento</i>	<i>Número de Fallos</i>
Bobinados	144
Cambiadores de tomas	85
Terminales	42
Conexionados	13
Circuitos Magnéticos	4
Otros	101
TOTAL	389

Tabla II.1 Estadística de Fallos en Subestaciones en EEUU entre 1983 y 1988

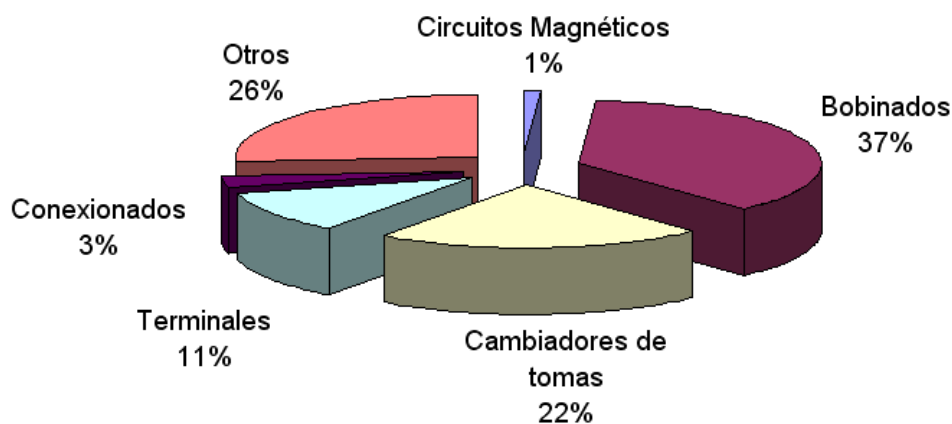


Figura II.2 Gráfica de Fallos en EEUU entre 1983 y 1988

Dado que fallos en los transformadores se originan tanto en el interior como en el exterior de la máquina, emplearemos esta diferencia para clasificar los fallos producidos en los transformadores de potencia [6].

II.3.1. Fallos externos

Los fallos externos al transformador son aquellos que se producen físicamente fuera del equipo eléctrico. Desde la perspectiva de la vida del transformador, este tipo de fallos son tan importantes como los que se producen internamente, ya que si no se despejan adecuadamente las condiciones que originan el defecto, se producirá una reducción de la vida de la máquina, que puede derivar, si éste es importante, en una avería e incluso en su destrucción. Se analizan, a continuación, los tipos de fallos externos más comunes.

⊕ Sobrecargas

La sobrecarga eléctrica es el motivo principal de envejecimiento prematuro de una máquina. Desde un punto de vista térmico, la sobrecarga se produce cuando la condición de equilibrio térmico en la máquina se establece a una temperatura tal que provoca la degradación de los dieléctricos que aíslan los conductores o las chapas que forman el núcleo magnético del transformador. La condición de sobrecarga involucra parámetros de distinta naturaleza:

1. Nivel de carga eléctrica
2. Condiciones ambientales: temperatura, humedad y altura sobre el nivel del mar.
3. Condiciones de explotación: continua, ocasional, etc.

⊕ Cortocircuitos

El cortocircuito externo al transformador es la condición de explotación más grave a la que puede someterse. Desde el punto de vista eléctrico la intensidad de cortocircuito simétrico en una red está limitada únicamente por la potencia de cortocircuito de dicha red. Si dicho cortocircuito se produce en el secundario del transformador, la potencia de cortocircuito en dicho punto se obtendrá como combinación de la potencia de cortocircuito de la red en el primario y la propia potencia de cortocircuito del transformador. Cuando se produce un cortocircuito, además del efecto térmico que produce dicha intensidad en los devanados, aparece un elevado esfuerzo electrodinámico entre conductores que puede producir daños mecánicos en la máquina.

⊕ Sobretensiones / Reducciones de frecuencia

Un circuito magnético alimentado mediante una tensión alterna genera un flujo cuyo valor eficaz es proporcional al cociente entre el valor eficaz de la tensión y la frecuencia de dicha tensión,

Cuando el valor del flujo alcanza la zona de saturación en la curva B-H del material utilizado para la construcción del circuito magnético, el consumo de intensidad aumenta, incrementando las pérdidas del transformador.

El incremento de las pérdidas debido a la saturación puede producirse tanto por una sobretensión como por una reducción de la frecuencia de la tensión de alimentación.

⊕ **Condiciones ambientales adversas**

Desde un punto de vista conceptual, la reducción de vida útil de un dieléctrico se produce por un incremento de temperatura, independientemente de que este sea provocado por una disipación de calor debida al efecto Joule en los conductores, por un incremento de la temperatura ambiente o por un fallo en el sistema de refrigeración.

⊕ **Ciclos de trabajo**

En determinadas condiciones de explotación, como emergencias o condiciones ambientales muy adversas, es necesario que el transformador opere durante un determinado periodo de tiempo por encima de sus características nominales. Esta situación puede ser admisible si se verifica un nivel medio de carga, generalmente inferior al nominal, durante un intervalo global de 24 horas.

II.3.2. Fallos Internos

Por otro lado, los fallos internos son aquellos que se producen en alguno de los elementos que componen el transformador. Los fallos internos normalmente están relacionados con el envejecimiento del aislamiento o con un mal diseño o construcción del transformador.

⊕ **Cortocircuitos entre espiras en la misma fase**

Se trata del tipo de fallo más difícil de detectar, ya que en sus fases iniciales, el fallo involucra únicamente unas pocas espiras. En los transformadores de alta tensión, el problema es aún mayor debido al elevado número de espiras.

⊕ **Cortocircuitos entre espiras en fases distintas**

Son fallos menos frecuentes, pero más dañinos y problemáticos. Son fáciles de detectar.

⊕ **Cortocircuitos monofásicos a tierra**

Los defectos de aislamiento entre fase y carcasa (puesta a tierra) debido al deterioro de los dieléctricos provoca la circulación de una intensidad de defecto hacia tierra siempre que el sistema de distribución sea de tipo neutro rígido a tierra o a través de una impedancia.

⊕ Defectos en el circuito magnético

Los defectos en el circuito magnético suelen producirse como resultado del deterioro del aislamiento y que es producido por un incremento excesivo de temperatura. Estos incrementos pueden estar producidos por una condición de sobrecarga, por la presencia de armónicos en la intensidad, por sobretensiones o disminuciones de frecuencia que den lugar a una saturación de flujo magnético. Suelen ser causantes de cortocircuitos fase-tierra, ya que normalmente el núcleo está conectado a tierra.

⊕ Defectos en el conexionado / aisladores

Este tipo de defectos son producidos por conexiones en el aislador de entrada – salida del transformador. En muchos casos, la solución consiste en la limpieza periódica de los aisladores, y su protección mediante barnices dieléctricos que repelen la suciedad. Los defectos producidos por malas conexiones, se caracterizan por provocar incrementos de temperatura.

⊕ Defectos en el sistema de refrigeración

Los defectos en el sistema de refrigeración comprenden todos aquellos fallos que afectan a la adecuada refrigeración del equipo. Estos defectos se deben a la pérdida de refrigerante en la cuba, en el depósito de expansión o en las conducciones intermedias, al deterioro de las características del refrigerante debido a procesos de envejecimiento o contaminación, a la avería de los termostatos, sondas térmicas o ventiladores, a la obstaculización de las canalizaciones de refrigeración o al cálculo inadecuado de la refrigeración o modificación de las condiciones de explotación.

En la **Figura II.3** encontramos una gráfica que relaciona el número de fallos por causa con el coste provocado por dichos fallos. Los datos fueron obtenidos en base a una población de 94 transformadores con potencias iguales o superiores a 25 MVA, situados en diferentes partes del mundo entre 1997 y 2001. La causa principal de fallo en los transformadores de potencia se debe a defectos de aislamiento siendo, además, los que mayor coste generan [4].

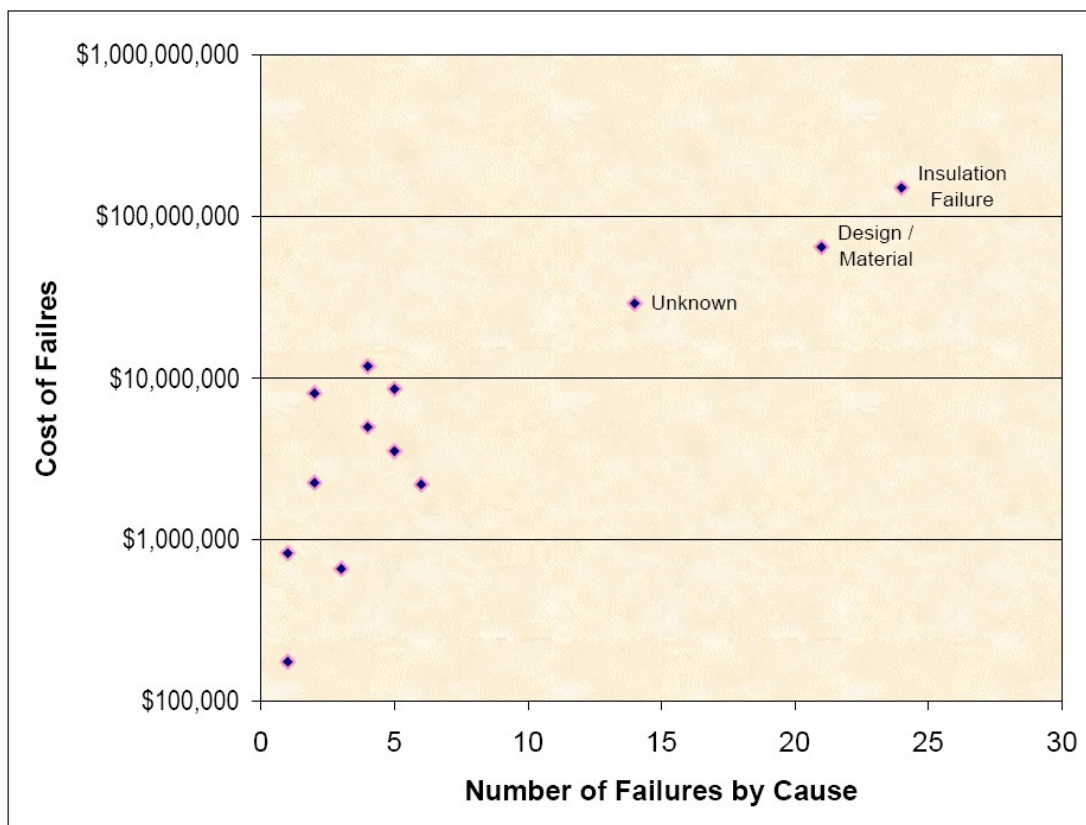


Figura II.3 Costes asociados a los Fallos en Transformadores de Potencia

II.4. Monitorización y Diagnóstico

Existe una diferencia importante entre estos dos términos. Monitorización se refiere principalmente al proceso de adquisición de datos, despliegue de sensores e implementación de métodos de acondicionamiento de las medidas de los transformadores de potencia. El diagnóstico es un paso posterior a la monitorización. En este sentido, el diagnóstico consiste en la interpretación tanto posterior como continua de los datos adquiridos. La monitorización es fuentes de información para realizar el diagnóstico.

Cada vez son más las instituciones que se preocupan por desarrollar nuevos métodos de monitorización. El análisis y el intercambio de resultados es un requisito imprescindible para asegurar el rápido avance de los sistemas de monitorización y así alcanzar los objetivos previstos.

Para hacer frente a las expectativas generales, el sistema de monitorización de un transformador debe satisfacer una serie de requisitos. El sistema de monitorización debe estar diseñado para permanecer operativo durante un largo periodo de tiempo y debe ser muy fiable con respecto al tiempo de vida de un transformador de potencia que puede alcanzar los 30 años o más. La relación entre los costes de operación e instalación y los beneficios proporcionados por el sistema de monitorización de un transformador debe ser razonable. El sistema de monitorización debe proporcionar información que permita realizar una estimación

lo más fiable posible del tiempo de vida que le queda a un transformador y evitar descargos inesperados. En la creación de sistemas de monitorización de transformadores, los parámetros deben fijarse de forma que las medidas permitan obtener la información deseada. El almacenamiento de la información debe estar organizado convenientemente para que sea accesible para el usuario. Los datos deben estar disponibles transcurridos varios años. Una de las más importantes funcionalidades de los sistemas de monitorización de transformadores es una moderna estrategia de comunicación de los datos, como por ejemplo a través de Internet.

Centrándonos en el concepto de monitorización, se debe distinguir entre la monitorización continua (*online*) y la monitorización periódica (*offline*). En la **Tabla II.2** se encuentra un breve análisis comparativo de esos dos conceptos.

	Monitorización periódica (offline)	Monitorización continua (online)
Definición	Se basa en la comprobación periódica de parámetros previamente definidos	Se efectúa de forma permanente durante la vida útil del transformador
Ventajas	El equipo de diagnóstico se conecta únicamente a unos sensores o equipos durante el análisis, se obtiene un valor y si excede los límites normales, se realiza el correspondiente mantenimiento.	Se obtiene una supervisión continua de los equipos, permitiendo un análisis de la tendencia de su estado y favoreciendo el reconocimiento de desviaciones en el funcionamiento del transformador en una fase temprana.
Inconvenientes	Se complica fijar el límite a partir del cual se considera necesario efectuar el mantenimiento y detectar una situación grave. Si la monitorización se realiza con el transformador fuera de servicio, no es posible medir ciertas magnitudes	Se precisan sensores y equipos instalados permanentemente en el transformador, complicándose en instalaciones existentes. El volumen de datos disponibles para el análisis puede ser muy alto si la configuración de los parámetros de la adquisición no se hace correctamente

Tabla II.2 Monitorización Periódica y Monitorización Continua

En el caso de la monitorización periódica, es necesario que para obtener los datos relevantes, por razones técnicas, el transformador esté apagado (descargo), provocando una pérdida en el suministro continuo de energía y los correspondientes costes asociados a dicha parada. Por el contrario, en la monitorización *online*, los datos son adquiridos mientras el transformador esta operativo. Debido a la relevancia que en la actualidad está tomando la monitorización continua será en este tipo de monitorización en la que nos centremos a partir de ahora.

II.5. Parámetros de Medida en Monitorización Continua

Todo sistema de monitorización continua para transformadores de alta potencia consta principalmente de sensores, sistemas de adquisición de datos, herramientas para el manejo de información y métodos para la evaluación de la condición operativa de los transformadores de potencia [7].

Durante el tiempo de vida de un transformador de potencia, éste está sometido a numerosas condiciones extremas que degradan su sistema de aislamiento. Las causas principales de esta degradación son: temperaturas excesivas y la presencia de oxígeno y humedad que combinada con los esfuerzos eléctricos aceleran el proceso. Otras causas que contribuyen a la degradación del sistema de aislamiento son los esfuerzos mecánicos y los productos de descomposición del aceite. Esta degradación se traduce en una reducción de las propiedades del aislamiento.

El proceso de degradación evoluciona gradualmente hasta presentarse un fallo o una avería, que en ocasiones puede acaba en catástrofe. La detección oportuna de una degradación puede ser la diferencia entre someter el transformador a una reparación de enormes dimensiones o simplemente sustituir una pieza dañada. La detección de cierto tipo de averías puede efectuarse mediante la monitorización continua de parámetros clave que sirven para diagnosticar la condición del aislamiento.

El fenómeno de degradación debe emplearse para obtener información sobre las condiciones de medida. Estas cantidades físicas deben derivar en la estimación del tiempo de vida restante y deben ser útiles para aplicar de forma temprana los sistemas de alerta. En este sentido, los parámetros más importantes a monitorizar son: la carga, las condiciones de operación del transformador, el análisis de la humedad, la medida de la temperatura y el análisis los gases disueltos en el aceite aislante.

II.5.1. Medidas de Carga y Condiciones de Operación

Tradicionalmente, las medidas de carga y de las condiciones de operación se obtienen a partir de los valores de tensiones y corrientes a la temperatura y frecuencia de operación. Para las medidas de tensión se pueden usar los dispositivos de potencial y para las medidas de corriente de carga se utilizan los transformadores de corriente instalados en las boquillas o mediante dispositivos ubicados en otros puntos de la subestación. Las medidas de tensión y corriente junto con la posición del selector de relación de un transformador (Selector de Tensión de Salida) permite el cálculo de la potencia transportada a través de un transformador. Hasta hace unos años, esta era básicamente la información disponible de un transformador durante su servicio.

II.5.2. Análisis de la Humedad

Los procesos de degradación del aislamiento producen moléculas de agua. Por este motivo la medida de la humedad es un elemento que permite evaluar el estado del aislamiento (aceite o cualquier otro material). Existen desarrollos basados en la tecnología de semiconductores o en medidas de corrientes de polarización para monitorización continua.

El análisis de la humedad guarda relación con el estado de los pasos de enfriamiento, que se puede medir mediante dispositivos que indican el estado en que estos se encuentran (encendidos o apagados), permitiendo relacionar la actividad de los pasos de enfriamiento con la carga en función del tiempo. La medición de corrientes de bombas, permite detectar problemas mecánicos y eléctricos en estos accesorios. Cualquier incremento de la corriente demandada por ellos es una indicación de la existencia de problemas.

II.5.3. Medida de la Temperatura

El envejecimiento del aislamiento, normalmente aceite o papel, se determina principalmente por las condiciones térmicas dentro del transformador. Para evaluar las condiciones en las que se encuentra el sistema aislante debido a los esfuerzos térmicos es importante monitorizar la temperatura en diferentes puntos del transformador. Las temperaturas demasiado elevadas pueden deberse a sobretensiones o a recalentamientos locales. Incluso en bajas condiciones de carga, el transformador puede presentar recalentamientos debido a una reducción de la eficiencia del sistema de refrigeración. De acuerdo con un modelo relativamente simple de envejecimiento térmico, se puede definir una tasa de envejecimiento. Esta tasa de envejecimiento se duplica con cada incremento de 6 K de temperatura con respecto a la temperatura máxima [7].

Básicamente, existen tres posibilidades para medida de temperatura:

- **Medida de temperatura empleando dispositivos PT100** (basados en la variación de resistencia a cambios de temperatura), que es el método más

sencillo y proporciona información de la temperatura del aceite a la parte superior e inferencia de la unidad de refrigeración.

- **Sistemas de fibra óptica para mediciones de temperatura a lo largo de las bobinas**, que han sido probados para aplicaciones dentro de grandes transformadores de potencia. Sin embargo, existen una serie de desventajas claras. Se trata de sistemas excesivamente caros, lo que se contradice con la idea de sistemas de monitorización de bajo coste. Además, la fibra óptica es un elemento mucho más delicado y su manejo debe ser más cuidadoso. En el caso de medidas distribuidas a lo largo de las bobinas, la fibra óptica debe estar junto con el conductor, por lo que estará sometida a altas presiones. Por eso, la aplicación de los sistemas de fibra óptica son sólo sensibles para experimentos en laboratorios de investigación y para ciertos diseños, pero no para monitorización continua.

II.5.4. Análisis de los gases disueltos en el aceite aislante

La medición de los gases disueltos en el aceite aislante es el parámetro más importante que debe monitorizar cualquier sistema de monitorización continua para transformadores de potencia.

El aceite se considera uno de los mejores medios de refrigeración que tiene además buenas propiedades dieléctricas y que cumple con las siguientes funciones:

- ❖ Actúa como aislante eléctrico.
- ❖ Actúa como refrigerante.
- ❖ Protege a los aisladores sólidos contra la humedad y el aire.

El análisis de los gases disueltos en aceite se convierte en la principal baza de los sistemas de monitorización continua para evaluar las condiciones en las que se encuentra un transformador de potencia. Debido al sobrecalentamiento, descargas parciales o roturas locales dentro del transformador en servicio, en el aceite se producen y disuelven una gran cantidad de gases. La naturaleza, la cantidad y las proporciones de estos gases, dependen del material aislante presente, de la naturaleza del fenómeno responsable de la descomposición y de los niveles de energía involucrados en el proceso. Si se excede de una cierta tasa de generación, comienzan a aparecer burbujas de gas. Estas burbujas de gas pueden causar una rotura local si provienen de regiones internas de los sistemas de aislamiento con grandes campos eléctricos. Los gases más importantes producidos en un transformador en caso de fallos son hidrógeno, monóxido de carbono, dióxido de carbono, metano, acetileno, etano y etileno. El contenido de los gases individualmente depende de la clase del fallo. Se han desarrollado diferentes métodos que permiten la indicación de fallos por la evaluación de la composición de los gases producidos durante un fallo interno en un transformador [8].

II.5.5. Medición Continua de Descargas Parciales y Análisis de la Función de Transferencia

Los cortocircuitos que se pueden producir en la red de potencia cerca del transformador o simplemente, el sometimiento a altas tensiones, puede producir rupturas dialécticas en el aislamiento de la bobina o en el aislamiento principal. Existen dos métodos para detectar este tipo de fallos, la medición continua de descargas parciales y el análisis de la función de transferencia.

Aplicar el **análisis de la función de transferencia** como un método periódico, significa comparar funciones de transferencia tomadas antes y después de un evento particular, por ejemplo, un cortocircuito en la red de potencia. Las diferencias entre una y otra indican cambios en el interior del transformador, por ejemplo, el desplazamiento de las bobinas. Además, las dos funciones de transferencia proporcionan información acerca de la integridad de las bobinas instaladas en el transformador.

Por otro lado, para la **medición de descargas parciales** suelen emplearse técnicas de ultrasonidos, basándose en el hecho de que las descargas parciales producen ondas de presión que se propagan, a través del medio aislante, desde el sitio de la descarga hasta la pared del tanque del transformador, donde pueden detectarse por medio de sensores ultrasónicos. El análisis de descargas parciales es un enfoque proactivo del diagnóstico que utiliza las mediciones de las descargas parciales para evaluar la integridad de los equipos.

II.6. Conclusiones

- ✓ Las transformadores de potencia son uno de los elementos más críticos dentro del sistema de generación y distribución de energía eléctrica.
- ✓ Entre las averías más frecuentes, destacan las relacionadas con el aislamiento y el envejecimiento del mismo. Además, este tipo de fallos suele suponer un elevado coste.
- ✓ Dentro de un transformador de potencia existen una serie de variables que se pueden medir y que permiten conocer, entre otras cosas, el estado de su aislamiento.
- ✓ La monitorización continua de los Transformadores de Alta Potencia permite conocer el estado de los mismos sin necesidad de realizar un descargo en la subestación.
- ✓ Además el Sistema de Monitorización Continua adecuado podría permanecer instalado en la subestación, a diferencia de los sistemas habituales de medida acústica que, aunque no requieren el descargo de la subestación, no están pensados para permanecer instalados.

Capítulo III

Detección de Emisiones Acústicas de Descargas Parciales en Transformadores

III.1. Introducción

Uno de los problemas más significativos de los equipos eléctricos en funcionamiento es la degradación que se origina en los mismos al combinarse negativamente efectos eléctricos, térmicos y mecánicos que limitan el tiempo de funcionamiento de los mismos. En transformadores de potencia este proceso produce la degradación del aislamiento dieléctrico, en la mayoría de los casos aceite mineral, que puede inferir la aparición de fallos graves en su operación. Como resultado se produce un fenómeno de degradación muy típico en transformadores que es el de descargas parciales. Este fenómeno es simultáneamente síntoma y causa de la degradación del aislamiento y su detección es fundamental para evitar posibles fallos catastróficos de los transformadores.

Durante la segunda mitad del siglo XX ha habido un creciente interés en la medición de descargas parciales, ya que ha demostrado ser un buen indicador del estado de degradación de los aislantes eléctricos. La medida de las descargas parciales es fundamental para analizar el estado de transformadores de potencia tanto aislados con aceite como secos y ofrece la posibilidad de detectar la presencia de defectos en el aislamiento del transformador. Esto significa que un conocimiento adecuado del comportamiento de las descargas parciales en un aislante es muy ventajoso a la hora de determinar su vida útil y por ello, se han desarrollado e implementado rápidamente nuevas tecnologías para la detección de descargas parciales con técnicas digitales.

Una descarga parcial es un fenómeno de ruptura eléctrica que está confinado y localizado en la región de un medio aislante, entre dos conductores que se encuentran a

diferente potencial. En el interior de los dieléctricos, aparecen en forma de pequeñas descargas como manifestación de la degradación de los mismos. Estas descargas dan señales de fenómenos de degradación térmica y con el tiempo pueden contribuir a la destrucción del transformador de potencia.

Para determinar el nivel de gravedad de las descargas parciales es preciso tener en cuenta es el tipo de aislamiento en el que se están produciendo. Las descargas parciales en aceite no son exageradamente peligrosas debido a la naturaleza de regeneración del aceite mineral. Sin embargo, las descargas parciales producidas en el papel provocan degradaciones muy rápidas del material y, por lo tanto, el riesgo de fallo en un corto periodo de tiempo se incrementa. Por otro lado, en transformadores secos en resina, la aparición de descargas parciales puede indicar una mala impregnación o la existencia de procesos locales de degradación. Una combinación de monitorización de descargas parciales, junto con análisis de gas, humedad y temperatura permiten un control del estado del transformador en tiempo real.

III.2. Detección y Medida de Descargas Parciales

Los fenómenos de las descargas parciales, que son del orden de picocolumbios, no pueden ser medidos directamente. Sin embargo, la energía producida por estos fenómenos produce una serie de transitorios eléctricos, cambios químicos, emisiones electromagnéticas, vibraciones, sonido, luz y calor que si que pueden medirse. La identificación de descargas parciales basada en la detección de cada una de estas alteraciones tiene sus ventajas e inconvenientes.

Por tanto, para medir las descargas parciales se ha de recurrir a la detección de los fenómenos derivados de la energía que contienen. Entre otros, una descarga parcial presenta un pulso eléctrico con un rápido transitorio y una emisión acústica. Dependiendo de la localización de la descarga parcial y del camino recorrido entre fuente y detector puede usarse tanto la señal eléctrica como la señal acústica para detectar la descarga parcial. Ambos métodos tienen distintos medios de detección y sensibilidades para las señales no deseadas (ruido), por lo que pueden ser redundantes en algunos tipos de descargas parciales o complementarias en la detección y caracterización.

24

El lugar concreto donde se produce la descarga puede ser la consecuencia de un aumento del campo eléctrico es relativamente pequeño, comparado con las dimensiones del medio aislante. El aumento del campo puede deberse a cambios bruscos en la naturaleza del aislante, que pueden provocarse, por ejemplo, por espacios de gas entre las superficies de un aislante con un conductor o con otro aislante.

El proceso de descargas parciales es característicamente pulsante y se manifiesta como pulsos de corriente en un circuito externo. Este proceso tiene un tiempo de duración de entre 3 y 300 nanosegundos y está catalogado como estocástico porque sus propiedades son descritas en función de variables aleatorias dependientes del tiempo.

Dependiendo del mecanismo de medida empleado puede hacerse una clasificación [9]:

- Métodos basados en radio frecuencia
- Métodos basados en el acoplamiento capacitivo, sensores de luz ultravioleta, gases disueltos
- Métodos basados en transductores de ultrasonidos, sensibles a las emisiones acústicas generadas por una fuente de descargas parciales.

Asimismo, estas técnicas suelen combinarse para obtener mejores resultados y más fiables. Así, por ejemplo, a partir de una detección de posibles descargas parciales en transformadores, con el Análisis de Gases Disueltos (DGA), se puede recurrir a una medida acústica sin descargo con el fin de confirmar el daño y acotar su procedencia.

No obstante, de entre las distintas técnicas de detección, son de mayor relevancia los sistemas acústicos, basados en las ondas ultrasónicas generadas, pues ésta no está influida por los fuertes campos y además aporta información adicional para la localización de su origen si se utilizan técnicas de triangulación mediante la medida de los retardos de propagación que se producen en un transformador.

El método acústico de detección ofrece la ventaja fundamental de que las ondas acústicas generadas por las descargas parciales son inmunes al ruido electromagnético, que es la principal fuente de problemas en cuanto a la detección eléctrica se refiere.

El diagnóstico obtenido a partir del análisis de las descargas parciales es un sistema mucho más efectivo que el mantenimiento basado en intervalos de tiempo (mantenimiento periódico), ya que hace innecesario la realización de operaciones costosas y minimiza el riesgo de fallos que pueden ocurrir entre dos intervalos de mantenimiento. De esta forma se permite a los responsables tomar decisiones con mayor información y optimizar sus recursos con el consiguiente ahorro de tiempo y dinero.

III.3. Emisiones Acústicas procedentes de Descargas Parciales

Las fuentes de descargas parciales activas producen emisiones acústicas que se propagan lejos de la fuente en todas direcciones. Las señales acústicas viajan a través del aislante para finalmente llegar a la pared del transformador. La distancia recorrida en un medio particular depende del tiempo que tarda la señal acústica en completar su recorrido como se muestra en la siguiente ecuación:

$$\text{Distancia recorrida} = \text{Velocidad de la onda acústica en el medio} \times \text{Tiempo}$$

Debemos tener en cuenta que la energía de las señales acústicas que podemos medir en un transformador de potencia en funcionamiento procede de descargas parciales y de otras fuentes mecánicas y térmicas que se encuentran en el interior y en el exterior del transformador.

La **Figura III.1** muestra la forma de onda detectada por un sensor acústico instalado en el exterior de un transformador cuando se produce una descarga parcial. El fuerte aumento del frente de onda indica que una onda directa ha afectado a la pared interior del transformador dentro del ángulo crítico para una onda de la presión.

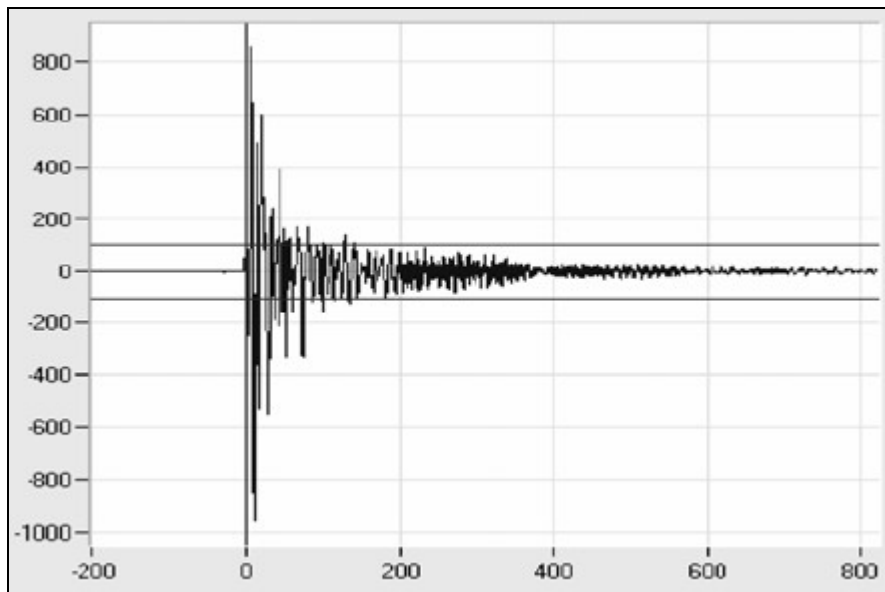


Figura III.1 Señal Acústica procedente de una Descarga Parcial [9]

El eje horizontal muestra el tiempo en microsegundos. El impulso tiene una duración de 144 μ s desde que cruza el umbral establecido por primera vez hasta la última. El eje vertical es adimensional e indica amplitud. El número de impulsos acústicos por unidad de tiempo nos proporcionaría una medida de la actividad acústica.

A continuación, en la **Figura III.2** podemos encontrar de nuevo esta misma señal pero con algo más de detalle, permitiéndonos contabilizar el número de oscilaciones que se producen a través del umbral establecido de ± 100 .

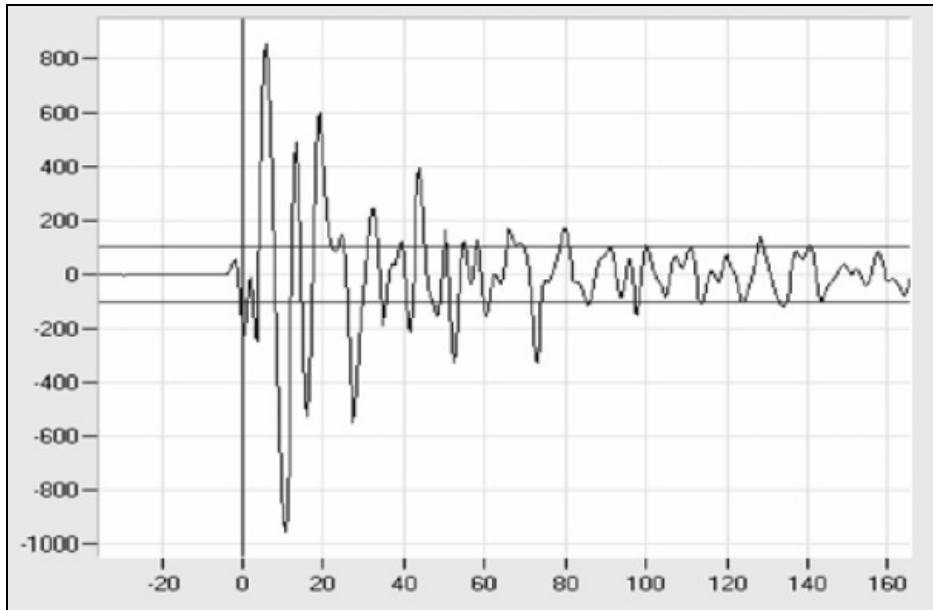


Figura III.2 Detalle de Señal Acústica procedente de una Descarga Parcial [9]

De este modo, un sistema básico de detección de emisiones acústicas podría ser aquel capaz de contabilizar el número de impulsos que se suceden y el número de oscilaciones producidas en cada una de ellas. No obstante, se debería tener en cuenta que si las descargas parciales son más rápidas que el tiempo de relajación del transductor o que la ventana usada para la clasificación de impulsos y en la distribución de los mismos, se pueden suceder errores en el cálculo de oscilaciones y en la longitud del impulso.

Para calcular la distancia entre la fuente de la descarga parcial y el sensor se suele emplear la velocidad de **1413 m/s a 20 °C**. Esta velocidad es mayor a medida que la temperatura del medio aumenta. Sin embargo, no se suele tener en cuenta su variación ya que la incertidumbre debida a la propagación en el material es normalmente mucho mayor.

La siguiente tabla muestra la estimación de diferentes velocidades de propagación del sonido en aceite a otras temperaturas.

Temperatura del Aceite [°C]	Velocidad de Propagación del Sonido [m/s]
20	1413
50	1300
80	1200
110	1100

Tabla III.1 Velocidad de Propagación del Sonido en el Aceite

En la práctica, la velocidad también depende de la existencia de otros materiales que se pueden interponer en el camino de la onda y puede variar en función de las propiedades del aceite.

Para caracterizar si una señal acústica es debida a una descarga parcial se suelen considerar diversos factores, como la longitud de la ráfaga y el tiempo de subida de la primera oscilación.

Las descargas parciales obtenidas por métodos convencionales eléctricos emplean un umbral alto para predecir una actividad intensa. Estos niveles generalmente están entre 300 pC y 500 pC. Debido a las variaciones en las señales acústicas causadas por la distancia y por la intervención de barreras estructurales, dicho umbral es distinto para cada sistema acústico. Además, las diferencias en ajustes de ganancia del amplificador causan diferencias de amplitud. Por lo general, se consideran dos tipos de actividad de descargas parciales: continua y esporádica.

Cuando la actividad es continua, los impulsos generados por descargas parciales están presentes todo el tiempo, aunque tendrán una amplitud variable. Este tipo de señal es el típico producido por una fuente energética de descargas parciales.

Cuando la actividad es esporádica puede subdividirse en los siguientes tipos:

- a. Emisiones acústicas esporádicas procedentes de una fuente continua de descargas parciales. Se caracterizan por tener una actividad durante todo el tiempo pero con cortos periodos de reposo. Este tipo de señales pueden producirse por fuentes de exposición en conductores y conectores, defectos en aislamientos, circuitos sin conexiones a tierra.
- b. Descargas parciales esporádicas con largos periodos de reposo (incluso minutos) seguidos de periodos cortos de una actividad muy elevada. Este tipo de señal se asocia a descargas estáticas y malos contactos. Los arcos de corta duración producen señales con mucha energía durante los periodos activos.

Por otro lado, es importante verificar la señal acústica se debe a la existencia de descargas parciales internas o a otras fuentes como el ruido mecánico. Estas verificaciones requieren de experiencia y de otras evidencias tales como la presencia de indicadores de gases disueltos o de descargas parciales eléctricas. El procesamiento de datos también puede ser de gran ayuda en esta tarea, a la hora de clasificarlas basándose en las particularidades de cada fuente individual (amplitud, duración, energía, número de cuenta, etc.).

En general, una fuente de descargas parciales más intensa producirá una amplitud y una frecuencia de oscilaciones de la emisión acústica mayor que una fuente débil. Esto es porque en el lugar de una descarga intensa puede haber múltiples localizaciones o perturbaciones que están produciendo mayor energía. No obstante, la medida acústica por sí sola no proporciona un diagnóstico.

III.4. Sistemas acústicos para detección de descargas parciales

En un escenario sencillo, un sistema acústico para detección de descargas parciales consiste en uno o más transductores de ultrasonidos que son sensibles a las emisiones acústicas generadas por una fuente de descargas parciales. Estas ondas, debido a las características de propagación del medio aislante (aceite) y a la estructura mecánica interna del equipo, abarcan un amplio rango espectral ultrasónico, entre 20 kHz y 500 kHz, con un ancho de banda que se sitúa por encima de 1 MHz. Por este motivo, se han utilizado sensores piezoeléctricos, PZT, colocados en el exterior de los transformadores para detectar las ondas acústicas ultrasónicas que poseen una frecuencia de resonancia típica de 150 kHz.

En la mayoría de los casos [10], los sistemas acústicos de descargas parciales se utilizan cuando una descarga parcial ya ha sido identificada como resultado de otros procedimientos de diagnóstico, como por ejemplo el análisis de los gases disueltos o las medidas eléctricas de las descargas parciales.

Los sistemas de detección de señales acústicas se pueden clasificar en dos categorías generales: **sistemas completamente acústicos** y **sistemas con disparo eléctrico**. Además, podemos encontrarnos con los sistemas acústicos de monitorización continua que se emplean para detectar las emisiones en primera instancia y comprobar la tendencia de las mismas.

En los siguientes apartados se encuentran las implementaciones más típicas de estos tipos de sistemas, con las ventajas y limitaciones de cada una de ellas.

III.4.1. Sistemas completamente acústicos

Los sistemas completamente acústicos se componen simplemente de uno o más transductores de ultrasonidos sensibles a las emisiones acústicas producidas por las descargas parciales. La detección y localización se obtiene gracias a los sensores instalados en el exterior del transformador, en diferentes posiciones con respecto a la posible fuente de descargas parciales. Si se desea obtener una localización más precisa de la fuente de descargas parciales, podríamos basarnos en los tiempos de llegada relativos de cada una de las señales acústicas a cada uno de los sensores.

En este tipo de sistemas no es necesario realizar ningún tipo de lectura de tensión o de corriente sobre el transformador. Se trata por tanto de una herramienta muy útil para monitorizar equipos que se encuentran en funcionamiento.

La instalación de los transductores de ultrasonidos en el exterior del transformador, facilita la tarea de reconfigurarlos tantas veces como sea necesario hasta obtener una señal acústica nítida. Además, ofrece gran flexibilidad para mover el sistema de localización y detección de emisiones acústicas a otros equipos. El problema es que el sistema es más sensible a fuentes de ruido externo.

Por otro lado, las ventajas de realizar la instalación de los sensores en el interior del equipo produce señales más claras y por tanto, señales cuya relación señal a ruido (SNR) es mejor. Sin embargo, este tipo de instalaciones es más invasiva en el equipo, su reconfiguración es más compleja y su eliminación e instalación en otro equipo no es tan sencilla.

III.4.2. Sistemas acústicos con disparo eléctrico

Otro tipo de sistemas acústicos son aquellos que se ayudan de un disparo eléctrico para la detección de descargas parciales. Consisten en un conjunto de sensores acústicos como los descritos en el apartado anterior con una corriente o tensión de medida sobre el dispositivo que detecta eléctricamente la descarga parcial. Esta señal eléctrica se considera que es detectada instantáneamente. Asumiendo que esto es cierto, el tiempo de llegada de la señal eléctrica es usado como el instante de inicio del evento de descarga parcial. La diferencia entre los tiempos de llegada de la señal eléctrica y las señales acústicas es el tiempo de propagación entre la fuente de descargas parciales y la localización del sensor. La localización de la descarga parcial se basa en el tiempo absoluto de llegada a cada sensor, al contrario que en los sistemas completamente acústicos descritos en el apartado anterior donde se emplea la diferencia en los tiempos de llegada de las señales a cada uno de los sensores.

El supuesto de detección instantánea de la señal eléctrica no es correcto en todos los casos. En la mayoría de los casos, cualquier retardo se encuentra en el rango de unos pocos microsegundos por lo que puede despreciarse. Sin embargo, no debemos olvidar que en este tipo de sistemas, para proporcionar el disparo, es necesario procesar las señales eléctricas obtenidas por lo que puede producir un retardo mayor que se puede cuantificar según el tipo de detector y que por tanto, se debe tener en cuenta.

Una de las ventajas de estos sistemas combinados es que la medida eléctrica proporciona la certeza de que los sensores acústicos han detectado una descarga parcial y no una fuente de ruido. Además, la detección de la señal eléctrica es una buena fuente de disparo para iniciar la adquisición de datos en los sensores acústicos.

La mayor desventaja es que es muy complicado obtener una medida eléctrica clara de una descarga parcial en un equipo en funcionamiento debido al ruido eléctrico y otros factores que podemos encontrarnos.

III.5. Instrumentación para detección de emisiones acústicas

Existen diferentes sistemas de instrumentación capaces de detectar emisiones acústicas. En éste apartado mostraremos algunos de ellos que se han demostrado muy efectivos para llevar a cabo esta tarea. Sin embargo, otros sistemas distintos a los expuestos, pueden ser iguales o incluso más efectivos dependiendo de los parámetros físicos y de la localización de la posible fuente de descargas parciales.

Un sistema típico de detección de emisiones acústicas está formado por los siguientes componentes:

- a) Un sistema de detección digital, esto es, uno o más osciloscopios digitales estándar, de 4 canales, con una tasa de muestreo por canal de al menos 1 megamuestra por segundo y una memoria cuya capacidad sea de al menos 5000 muestras. Lógicamente, otras capacidades como detección de pico, zoom, transformaciones... pueden ser muy útiles.
- b) Sensores
- c) Cableado y fuentes de alimentación para los sensores

Existen otro tipo de sistemas que además de sensores acústicos incluyen un transformador de corriente de alta frecuencia instalado en la conexión a tierra o un transductor especial de radiofrecuencia instalado en el interior del transformador. Este tipo de sistemas utilizan la concurrencia de un evento acústico y un evento eléctrico o una señal de radio frecuencia para confirmar que la señal acústica proviene de una descarga parcial. En algunas ocasiones, la diferencia de tiempos es usada para obtener una medida de la localización de la descarga parcial.

III.5.1. Sensores externos

Los primeros sensores desarrollados para la detección de emisiones acústicas son acelerómetros o sensores electro-acústicos (piezoeléctricos) ubicados sobre las paredes del transformador, selectivos a frecuencias ultrasónicas (típicamente entorno a 150 kHz).

Teniendo en cuenta que la frecuencia principal de una descarga parcial de 150 pC es de 100 kHz, podríamos escoger sensores con frecuencias de resonancia (para ondas longitudinales) de 60 kHz o 150 kHz. Para descargas mayores la frecuencia debería disminuir. Además, la atenuación afecta a las altas frecuencias en mayor medida que a las bajas. Estos factores favorecen el uso del sensor de 60 kHz en fábrica. Sin embargo, en trabajos de campo, el ruido es mayor y sus armónicos están en frecuencias de 20 kHz a 60 kHz. En este caso, los sensores de 60 kHz serían sensibles a las ondas de presión. Estos ruidos provocarían falsas lecturas y es por eso por lo que en muchos casos se emplean sensores a la frecuencia de resonancia de 150 kHz.

Por ser dispositivos piezoeléctricos, los sensores también responderán a la variación de campos electromagnéticos. Para minimizar este efecto, el transductor puede ser tanto de tipo diferencial (utilizando dos cristales con la fase invertida para reducir el ruido) como apantallado con un solo cristal y un circuito de preamplificación integrado. Este último es el más utilizado debido a su mayor sensibilidad y a su baja impedancia de salida que es menos susceptible a la degradación debida al ruido flicker ($1/f$) en las conexiones.

Antes de comenzar a medir, debería aplicarse en la cara del transductor o en la pieza de unión el gel o lubricante acoplador acústico. Se suelen utilizar los geles o sólidos que mantienen una alta viscosidad en la pared a la temperatura de funcionamiento para transferir la señal apropiadamente. Son apropiados los acopladores para propósitos de test no destructivos con ultrasonidos. Los geles de glicerina y de grasa de silicona son particularmente más eficientes y más recomendables, aunque la grasa de silicona puede ser difícil de eliminar de la superficie del transformador.

III.5.2. Sensores Internos

Los sensores externos suelen utilizarse en entornos favorables como el de un laboratorio. No obstante, en condiciones no tan favorables, si sólo empleamos estas técnicas, las señales adquiridas presentan problemas debidos a la atenuación de las señales en el aceite y en las paredes del transformador, por la presencia de barreras mecánicas internas, y por la superposición e interferencia del resultado de varias trayectorias de las señales acústicas. La tendencia actual es ir hacia la utilización de sensores internos que además facilitan la localización en las descargas parciales.

Los sensores internos para emisiones acústicas o guías de onda son dispositivos que se sumergen en el aceite para acoplar la energía producida por una descarga parcial desde el aceite a sensor externo. Un ejemplo de este tipo de sensores sería una varilla de fibra de vidrio sólida insertada en el aislante del transformador con un sensor instalado en el extremo externo de la varilla. Se debe prestar atención para mantener las características y dimensiones dieléctricas. La guía de onda es menos direccional que los sensores situados en la pared exterior del transformador porque la impedancia de la pared del transformador de acero no es un factor relevante. Es menos sensible al ruido externo como, por ejemplo, el provocado por las tormentas.

Por otro lado, existen otras técnicas basadas en fibra óptica [11]. Este tipo de sensores son muy atractivos para medir un amplio rango de parámetros físicos debido a las ventajas inherentes de la fibra óptica, incluyendo ligereza, pequeño tamaño, flexibilidad geométrica, gran ancho de banda e inmunidad a campos electromagnéticos incluso con en contacto con elementos del transformador con altas caídas de tensión. Estas ventajas hacen de los sensores de fibra óptica unos candidatos muy atractivos para la medida de descargas parciales en el interior de transformadores.

Un posible esquema experimental se presenta en la **Figura III.3**, donde se detallan con claridad el sistema de generación de descargas parciales y el sistema de medida, basado en interferometría de fibra óptica para la medida de variaciones de fase a través de un transductor de fibra óptica intrínseco

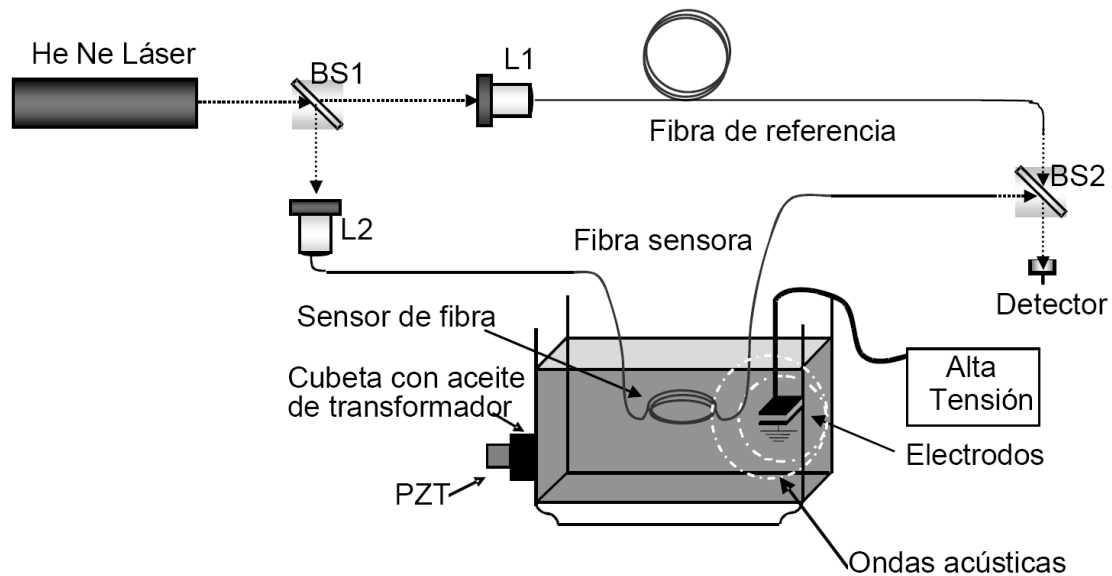


Figura III.3 Esquema experimental del sistema interferométrico de fibra óptica para la detección de descargas parciales

Este sistema interferométrico está constituido por un interferómetro Mach-Zehnder implementando los brazos de referencia y sensor con fibra óptica. Ambos brazos poseen la misma longitud de fibra y han sido construidos con arrollamientos idénticos. La fibra sensora tiene el arrollamiento sensor sumergido en una cubeta de ensayo, rellena de aceite de transformador, expuesto a las perturbaciones generadas por ondas acústicas procedentes de la fuente de descargas parciales. La fibra de referencia no se encuentra expuesta a las perturbaciones de las descargas parciales y se emplea como referencia para el camino óptico de la luz que se propaga a través de del sistema. Las medidas se obtienen a partir del desfase óptico de la luz que se produce en el interferómetro permitiendo la determinación precisa de caminos ópticos por estar ponderada la medida a la longitud de onda que sirve como patrón. Así, al estar en relación con la longitud de onda aplicada, se tienen órdenes de magnitud inferiores a $1 \mu\text{m}$ para un periodo. Además, la sensibilidad que se puede alcanzar es muy elevada ya que ésta es función de la longitud de fibra óptica expuesta al parámetro a medir, pudiéndose emplear sensores con más de 10 metros de fibra sensora. Se consiguen sensores de varios órdenes de magnitud más sensibles que los clásicos sensores eléctricos PZT.

III.5.3. Sistemas de 3 sensores

Este tipo de sistemas consiste en un sistema de transductores cuyo objetivo principal es el de optimizar la sensibilidad y rechazar las señales que no proceden de descargas parciales.

Consiste en tres sensores acústicos instalados en una estructura rígida. Un detector eléctrico es usado para marcar el origen de la descarga parcial, permitiéndonos determinar la duración de la señal. Este sistema está pensado para ser operado por personal con experiencia. En la búsqueda de posibles señales generadas por descargas parciales y su origen, la estructura de detección es movida seleccionando diferentes posiciones alrededor del equipo a examen. En cada posición, un conjunto de diferentes disparos y configuraciones son estudiados por la unidad de evaluación, por ejemplo un osciloscopio digital. Aplicando un conjunto de algoritmos proporcionados por la unidad de evaluación se podría determinar la dirección y la distancia a la posible fuente de la descarga,

III.5.4. Filtros paso banda

El empleo de filtros paso banda en los sistemas de detección de emisiones acústicas es opcional. Su propósito es reducir al máximo los efectos producidos por otras señales que podemos encontrar en el entorno y que no tienen nada que ver con las descargas parciales. Esto incluye vibraciones causadas por la acción magnetoestrictiva del núcleo (ruido Barkhausen), bombas y refrigeradores. La mayoría de ellos están por debajo de los 30 kHz; sin embargo, el ruido Barkhausen en ocasiones se encuentra en entorno a los 50 kHz. Por lo tanto, se necesita una franja paso alto con frecuencia de corte 100 kHz y con una transición rápida en frecuencia. Un filtro paso banda razonablemente generoso (200 kHz) permite variar entre los diferentes transductores en cuanto a su frecuencia de resonancia.

Por tanto, el filtro que nos permite mejorar la señal proporcionada por los transductores es de tipo paso banda con sus respectivas frecuencias de corte inferior y superior. Estas frecuencias son tales que la respuesta a una tensión de entrada sinusoidal constante caiga 3 dB desde el valor máximo. Cuando usamos un sensor de 150 kHz, la frecuencia de corte inferior debería ser 100 kHz y frecuencia de corte superior debería ser aproximadamente 300 kHz. Las características de rizado del filtro deberían ser como mínimo de 48 dB/octava (240 dB/decada para la sección paso alto. Esto significa que, con respecto a la señal de interés (150 kHz), una señal de frecuencia 50 kHz debería atenuarse 48 dB. Para la sección paso bajo del filtro, la caída del filtro debería ser de al menos 24 dB/octava (120 dB/decada), por lo que una señal a la frecuencia de 600 kHz, debería atenuarse 24 dB. Este tipo de comportamiento en frecuencia es posible conseguir mediante filtro digitales.

⊕ Ventajas de los Filtros Digitales frente a los Analógicos

Un filtro analógico tiene una señal analógica tanto a su entrada como a su salida. Ambas son funciones de una variable continua, t , y puede tomar un infinito número de valores. El diseño de filtros analógicos es sencillo, existen diseños de filtros muy probados y se encuentran fácilmente en la bibliografía. Sin embargo, este tipo de diseño de filtros está a menudo limitado por la precisión de los componentes a utilizar y a la pérdida de flexibilidad en el diseño y modificación del filtro. Su complejidad y estas limitaciones se hacen más evidentes para filtros de orden superior a 2 - 4 polos.

La adquisición de muestras y las herramientas de procesamiento digital de la señal han hecho posible reemplazar los filtros analógicos con filtros digitales en aplicaciones que requieren flexibilidad y programabilidad. Las ventajas de los filtros digitales sobre los filtros analógicos son:

1. Un filtro digital es programable, es decir, su funcionamiento está determinado por un programa almacenado en la memoria contigua al procesador. Esto significa que puede variarse fácilmente sin afectar al hardware, mientras que la única manera de variar un filtro analógico es alterando el circuito.
2. Los filtros digitales pueden ser fácilmente diseñados, probados e implementados en un ordenador. Los analógicos pueden ser simulados, pero siempre hay que implementarlos a través de componentes discretos para ver su funcionamiento real.
3. Las características de los filtros analógicos, particularmente los que contienen componentes activos, están sujetos a alteraciones y dependen de la temperatura. Los filtros digitales no sufren estos problemas y son extremadamente estables ante factores externos.
4. A diferencia de los filtros analógicos, los digitales pueden manejar con mucha precisión las bajas frecuencias. Como la tecnología de los DSP va mejorando, el aumento de su velocidad permite que también sean aplicados en el campo de la radio frecuencia (muy altas frecuencias), la cual en el pasado era exclusivamente dominio de la tecnología analógica.
5. Los filtros digitales son mucho más versátiles a la hora de manipular la señal, que pueden llegar a variarla y tratarla radicalmente cambiando sus características.
6. Los procesadores DSP más rápidos pueden manejar combinaciones complejas de filtros en paralelo o en serie (en cascada), haciendo los requerimientos de hardware relativamente simples y compactos en comparación con la circuitería analógica equivalente.

III.6. Sistemas de Monitorización Continua

El propósito principal de los sistemas de monitorización continua es la identificación rápida de un fallo incipiente en una localización remota. Este tipo de sistemas consisten normalmente en múltiples sensores, amplificadores y módulos de adquisición y procesamiento de señales y se caracterizan porque, generalmente, el personal encargado de la recolección de datos no está presente mientras el transformador está en funcionamiento.

Los sensores suelen situarse en posiciones donde se prevén fallos. Estas posiciones son generalmente seleccionadas gracias a experiencias pasadas o porque son posiciones donde existe una alta probabilidad de que ocurran este tipo de incidencias.

Los sistemas de adquisición y procesamiento de datos son los que nos permiten almacenar y enviar los datos recuperados de forma que es posible lanzar alertas en los sistemas de mantenimiento y actuar en consecuencia. Estos sistemas suelen tener menor capacidad de procesado que los sistemas consistentes en estaciones de trabajo DSP (*Digital Signal Processor*). La información transmitida normalmente es limitada.

Para una monitorización continua, tanto los sensores como el detector de descargas parciales se instalan de forma permanente. La señal producida por las descargas parciales es adquirida por los sensores que componen el detector de descargas parciales. Éste se conecta a un MODEM, a una red Ethernet o cualquier otro sistema que permita la adquisición y supervisión remota. En la **Figura III.4** se presenta el diagrama de bloques de un sistema de monitorización continua.

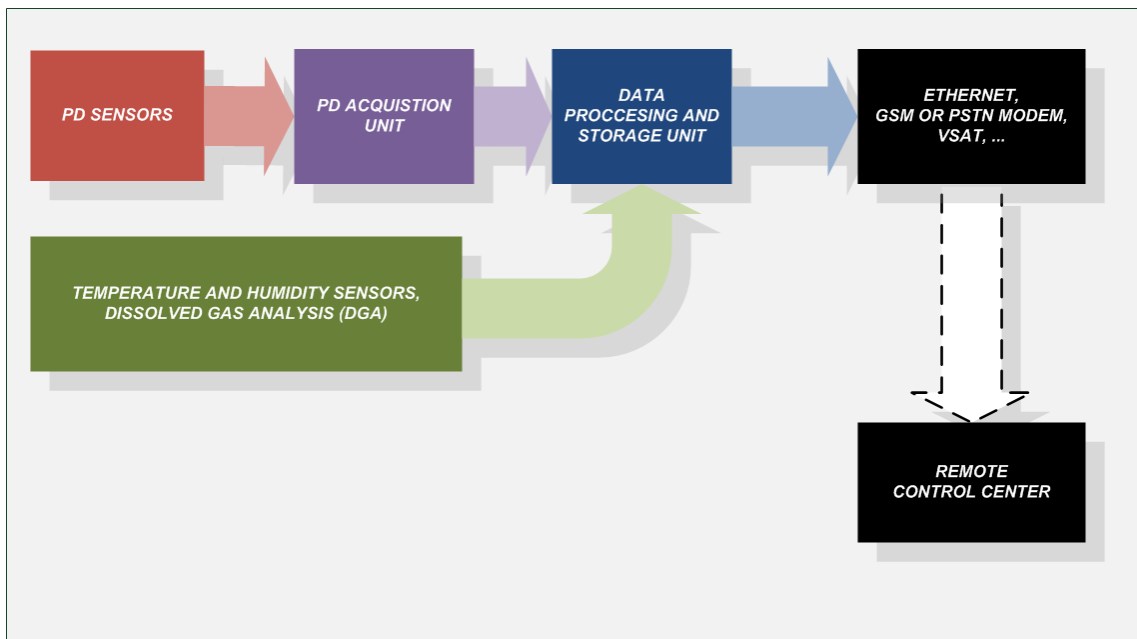


Figura III.4 Diagrama de Bloques de un Sistema de Monitorización Continua

Las primeras pruebas para establecer la base de la actividad en el transformador se realizan durante el proceso de instalación. Esta prueba proporciona la información para establecer los mejores ajustes del soporte físico y del software para la supervisión permanente. Es aconsejable que los ajustes de ganancia sean los mismos en los transformadores adyacentes para comparar lecturas si las señales son probablemente el resultado de los disturbios de la red. El montaje permanente de sensores es deseable para las instalaciones de largo plazo.

III.7. Conclusiones

- ✓ La medida de descargas parciales ha demostrado ser un buen indicador del estado de degradación de los aislantes eléctricos
- ✓ En el caso de transformadores de potencia, tanto aislados con aceite como secos, la medida de dichas descargas parciales es fundamental para analizar su estado y detectar la presencia de defectos en el aislamiento.
- ✓ Los fenómenos de las descargas parciales no pueden medirse directamente. Sin embargo, la energía producida por estos fenómenos produce una serie de transitorios eléctricos, cambios químicos, emisiones electromagnéticas, vibraciones, sonido, luz y calor que sí pueden medirse. La identificación de descargas parciales basada en la detección de cada una de estas alteraciones tiene sus ventajas e inconvenientes.
- ✓ Las ondas acústicas generadas por las descargas parciales son inmunes a ruido electromagnético, que es la principal fuente de problemas en cuanto a la detección eléctrica se refiere, por lo que la detección y localización de descargas parciales empleando las técnicas basadas en las Emisiones Acústicas es una opción muy válida.
- ✓ Los primeros sensores desarrollados para la detección de emisiones acústicas son acelerómetros o sensores electro-acústicos (piezoeléctricos) ubicados sobre las paredes del transformador, selectivos a frecuencias ultrasónicas (típicamente entorno a 150 kHz).
- ✓ Aunque los sistemas de detección de este tipo de señales acústicas se suele combinar con otros métodos de detección, lo que realmente interesa son los sistemas acústicos de monitorización continua que se encuentran permanentemente instalados y cuyo objetivo es la identificación rápida de un fallo incipiente en una localización remota.
- ✓ Un sistema básico de detección de emisiones acústicas podría ser aquel capaz de contabilizar el número de impulsos que se suceden y el número de oscilaciones producidas en cada una de ellas.
- ✓ Si las descargas parciales son más rápidas que el tiempo de relajación del transductor o la ventana usada para la clasificación de impulsos y en la distribución de los mismos, se pueden suceder errores en el cálculo de oscilaciones y en la longitud del impulso.
- ✓ El Sistema de Monitorización Continua debe permitir registrar este tipo de señales, cuyas características de frecuencia y amplitud ya conocemos, y sobre todo, impedir la pérdida de eventos debido al tiempo de rearme.

Capítulo IV

Medida de Emisiones Acústicas con la Tarjeta de Adquisición PDA14

IV.1. Introducción

Existen en el mercado gran variedad de productos que permiten capturar señales. Cada uno de estos sistemas, según sus especificaciones, puede ser útil para determinadas aplicaciones, pero pueden no ser los más apropiados en otros entornos de experimentación.

Tras haber profundizado en la naturaleza de las señales acústicas generadas por las descargas parciales, en este capítulo se presenta la Tarjeta de Adquisición PDA14 de Signatec, cuyas especificaciones la convierte en una candidata muy adecuada para la detección de este tipo de señales. Además, el fabricante, junto con la tarjeta de adquisición, suministra una librería de funciones en C que permiten ampliar las capacidades de la tarjeta.

En el mercado existen otros fabricantes con productos similares a la PDA14 y que están especializados en la medida de emisiones acústicas. Algunos ejemplos son AMSY-6 de Vallen y PCI - 8 de Mistras, que se caracterizan sobre todo por tratarse de sistemas multicanal.

IV.2. Sistemas de Adquisición de Datos

Un Sistema de Adquisición de Datos se usa cuando se está interesado en medir y analizar algún fenómeno físico. De una manera muy sencilla, un Sistema de Adquisición de Datos se define como un equipo electrónico que permite tomar señales físicas del entorno y convertirlas en datos que posteriormente podremos procesar y presentar al usuario.

En definitiva, un Sistema de Adquisición de Datos es una colección de herramientas hardware y software que nos permite interactuar con el mundo físico. En él se integran

diferentes componentes: transductores de diferentes tipos y naturaleza, multiplexores, amplificadores, circuitos Sample and Hold, conversores A/D y D/A, y además un microcontrolador con todas sus prestaciones: interrupciones, temporizadores, comunicación serie y la capacidad de hacer uso de memorias y puertos externos. Todo ello compone un sistema que se puede encargar de una aplicación específica como es chequear variables (PH, humedad relativa, temperatura, iluminación, concentración,...) para una posterior utilización de la misma ya sea con fines docentes, científicos, de almacenamiento o control y utilización de la misma.

La adquisición de datos se inicia con el fenómeno físico o la propiedad física de un objeto (objeto de la investigación) que se desea medir. Esta propiedad física o fenómeno podría ser el cambio de temperatura o la temperatura de una habitación, la intensidad o intensidad del cambio de una fuente de luz, la presión dentro de una cámara, la fuerza aplicada a un objeto, las emisiones acústicas,... Un eficaz sistema de adquisición de datos puede medir todas estas diferentes propiedades o fenómenos.

A grandes rasgos, los sistemas de adquisición de datos están compuestos por una serie de sensores adecuados que convierten cualquier parámetro de medición a una señal eléctrica que es recuperada por el hardware de adquisición de datos. Los datos adquiridos se visualizan, analizan y almacenan, ya sea utilizando el proveedor de software suministrado u otro software. Los controles y visualizaciones se pueden desarrollar utilizando varios lenguajes de programación de propósito general como Visual BASIC, C++, Fortran, Java, Lisp, Pascal. Los lenguajes especializados de programación utilizados para la adquisición de datos incluyen EPICS, utilizada en la construcción de grandes sistemas de adquisición de datos, LabVIEW, que ofrece un entorno gráfico de programación optimizado para la adquisición de datos, y MATLAB. Estos entornos de adquisición proporcionan un lenguaje de programación además de bibliotecas y herramientas para la adquisición de datos y posterior análisis.

En la **Figura IV.1** se muestra el diagrama de bloques de un sistema de adquisición. En él las señales de entradas son adquiridas por un sensor, se acondicionan, se digitalizan (convierten en 'bits') para que un ordenador las pueda leer y son manipuladas o analizadas para extraer información relevante. Los datos desde el ordenador son transformados en señales analógicas y son enviados hacia el 'exterior' por medio de un actuador.

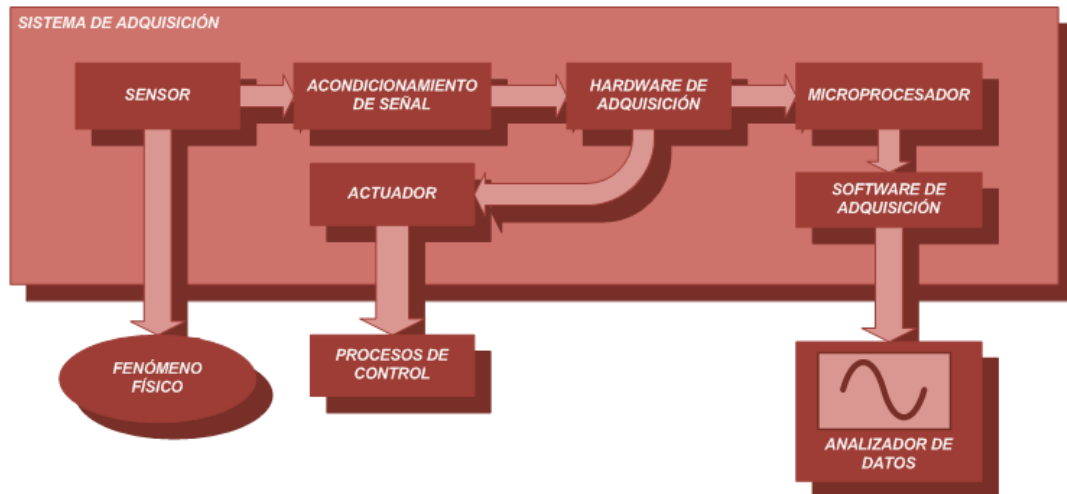


Figura IV.1 Diagrama de Bloques de un Sistema de Adquisición

El sensor es el dispositivo que convierte la propiedad física o fenómeno en una señal eléctrica medible, tal como tensión, corriente, cambio en los valores de resistencia o condensador, etc. La capacidad de un sistema de adquisición de datos para medir los distintos fenómenos depende de la capacidad de los transductores para convertir las señales derivadas de los fenómenos físicos. Los transductores son sinónimo de sensores en los sistemas de adquisición de datos. Hay transductores específicos para diferentes aplicaciones, como la medición de la temperatura, la presión o flujo de fluidos. Nos encontramos con sensores digitales (switches, encoders,...) y con sensores analógicos (acelerómetros, micrófonos, medidores de presión, medidores de temperatura,...). Existen dos características importantes que hay que tener en cuenta a la hora de escoger el sensor más adecuado:

1. Tipo de salida del sensor
 - i. Salidas digitales o analógicas
 - ii. Salidas de corriente: 4 – 20 mA, implica el uso de una resistencias de precisión para convertir a tensión.
 - iii. Salidas de tensión: caracterizadas por Amplitud, Frecuencia y Duración
2. Ancho de Banda del sensor: Establece las frecuencias presentes en la señal que está siendo medida.

Los transductores generalmente son incompatibles con el hardware de adquisición. Para resolver esta incompatibilidad, las señales adquiridas se deben acondicionar de alguna forma. Los sistemas de adquisición de datos también utilizan diversas técnicas de acondicionamiento de señales para modificar adecuadamente diferentes señales eléctricas en tensión, que luego pueden digitalizarse. La señal puede amplificarse o desamplificarse, o bien, puede requerir de un filtrado o una demodulación. Otros ejemplos de acondicionamiento de señales podría ser el circuito que proporciona la corriente o tensión de excitación al sensor, aislamiento,

linealización, etc. Este pretratamiento de la señal normalmente lo realiza un pequeño módulo acoplado al transductor.

El hardware de adquisición de datos comprende por lo general las interfaces entre la señal y un PC. Podría ser en forma de módulos que pueden conectarse a la computadora de los puertos (paralelo, serie, USB,...) o ranuras de las tarjetas conectadas a (PCI, ISA) en la placa madre. Las tarjetas de adquisición a menudo contienen múltiples componentes (multiplexores, ADC, DAC, TTL-IO, temporizadores de alta velocidad, memoria RAM) que son accesibles a través de un bus por un microcontrolador que puede ejecutar pequeños programas.

El hardware de adquisición puede presentarse de dos maneras:

- interna e instalada directamente en un ranura de expansión dentro de un PC
- externa que se conecta a la computadora a través de un cable externo

El PC proporciona el microprocesador, un reloj, un bus de datos, memoria y espacio en disco para el almacenamiento de datos. Por otro lado, el software de adquisición de datos normalmente viene suministrado junto con el hardware de adquisición de y permite que el sistema operativo pueda reconocer el hardware de adquisición de datos y dar así a los programas acceso a las señales capturadas por el hardware de adquisición. Permite la interacción entre el ordenador y el hardware: configurar la tasa de muestreo de la tarjeta, adquirir una predeterminada cantidad de datos y visualizar. Llegados a este punto, podemos establecer la diferencia entre dos tipos de software:

- controladores software: permite el acceso y controla las capacidades del hardware
- librerías software: funciones de alto nivel para realizar aplicaciones

El objetivo no es simplemente crear un sistema de adquisición de datos, sino crear un sistema capaz de monitorizar de forma continua un transformador con el fin de capturar cualquier descarga parcial que se pueda producir en su interior. Sin embargo, parte de la solución está compuesta por un sistema de adquisición de datos cuyos componentes deben ser los más adecuados para permitir capturar las señales deseadas.

En el capítulo anterior ya se mencionaron las técnicas empleadas para capturar las emisiones acústicas generadas por las descargas parciales. Una vez conocidas las características de este tipo de señales, es necesario seleccionar el hardware de adquisición de datos adecuado para nuestro propósito. Para ello, debemos tener en cuenta el software de adquisición asociado a este hardware de forma que habremos completado el Sistema de Adquisición de datos. No sólo se debe prestar atención al hardware de adquisición, que debe cumplir los requisitos necesarios para adquirir el tipo de señales con el que vamos a trabajar, también debemos hacer hincapié en el software de adquisición que se proporciona con el

hardware de adquisición, ya que será éste el que nos permite implementar el Sistema de Monitorización Continua de Descargas Parciales.

En el laboratorio de electrónica de la Universidad Carlos III de Madrid ya se dispone de una Tarjeta de Adquisición que reúne algunas de las características necesarias para adquirir las señales acústicas derivadas de Descargas Parciales. Además, esta Tarjeta de Adquisición proporciona ciertas funcionalidades necesarias para formar parte de un Sistema de Adquisición de Datos completo. A continuación, se presenta un Sistema de Adquisición basado en la Tarjeta de Adquisición PDA14 de Signatec que completa los bloques de hardware de adquisición y software de adquisición de un Sistema de Adquisición de Datos.

IV.3. La Tarjeta de Adquisición PDA14

La Tarjeta de Adquisición PDA14 [12] es un producto de Signatec que proporciona la capacidad de capturar señales combinando alta velocidad y alta resolución con una capacidad de memoria extremadamente amplia.

Una de sus características principales es su memoria de 512 MB que se emplea como una gigantesca FIFO para la adquisición de datos directamente del bus PCI o del SAB (Signatec Auxiliary Bus). Esta FIFO es la que previene la pérdida de datos proporcionando un flujo continuo de datos a través del bus PCI.

En cuanto a la adquisición de datos, es importante destacar que se dispone de tres modos de disparo:

- **Post-Trigger:** en el que la detección de la señal de disparo, provoca el inicio la adquisición de datos. Todos los datos, o al menos la mayoría, son capturados después de la detección de la señal de disparo. Una vez iniciada la adquisición, ésta continúa hasta que se llena toda la memoria activa.
- **Segmentado (Segmented):** en el que la señal de disparo provoca la adquisición de una porción fija (segmento) del total de la memoria activa reservada. Se requiere una señal de disparo para llenar sucesivamente cada uno de los segmentos de la memoria. La adquisición finaliza cuando se llena toda la memoria activa. Lógicamente, la longitud de cada segmento debe ser menor que la memoria activa que ha sido reservada para la adquisición
- **Pre-Trigger:** en el que la señal de disparo provoca que el fin de la adquisición de datos. Una vez armada para la adquisición, la memoria activa se va llenando con los datos proporcionados por el canal o los canales de entrada. Cuando se detecta la señal de disparo, la adquisición de datos finaliza. A diferencia del modo post-trigger, todos los datos, o al menos la mayoría, son capturados antes de la detección de la señal de disparo. Una vez detectada la señal de disparo y antes de que finalice la adquisición, se capturarán el número

de muestras que se haya configurado. El modo pre-trigger es interesante cuando se desea considerar la información de la señal antes y después de la señal de disparo.

La funcionalidad proporcionada por el **modo segmentado** al que se acaba de hacer referencia es otra de las características diferenciadoras de esta tarjeta con respecto a otras que se encuentran en el mercado. En el modo segmentado, el tiempo empleado por la tarjeta para rearmarse tras la adquisición de un segmento es de 150 nanosegundos. Asimismo, en el modo segmentado, además de los segmentos, se almacenan las marcas de tiempo correspondientes a cada uno de los segmentos permitiéndonos obtener una relación temporal entre cada uno de los segmentos capturados. Esto es especialmente importante a la hora de establecer una relación temporal entre segmentos cuando el intervalo de tiempo entre uno y otro no es uniforme o no está bien definido. Las marcas de tiempo son contadores de 32 bits con una resolución del reloj de 7.5 nanosegundos. Se pueden almacenar hasta 2048 marcas de tiempo en una memoria diferente de la memoria dedicada a la adquisición de señales y son accesibles una vez concluida la adquisición.

Existen una serie de parámetros de ajuste de la tarjeta que afectan a la cantidad y al método de adquisición de las muestras y que se enumeran a continuación:

- **Rate:** Indica la tasa de muestreo y aunque se puede utilizar una frecuencia de reloj externa, cuando se utiliza el reloj interno se pueden seleccionar hasta 22 tasas de muestreo:
 - desde los 100 MHz hasta 97.7 kHz, cuando el reloj interno funciona en 100MHz
 - desde 62.5 MHz hasta 61.0 kHz, cuando el reloj interno funciona en 62.5 MHz
- **Active Memory Size:** Este parámetro indica el número de muestras que serán recogidas después de que la memoria se llene y se haya terminado la adquisición. Cuando se detecta que la memoria está completa, se fija un flag que puede leerse a través del PC o del software seleccionado y así causar una interrupción o enviar una interrupción a través del SAB. La cantidad de memoria que está reservada para la adquisición de datos se puede fijar a partir de 8 bytes hasta los 512 megabytes completos en pasos de 8 bytes.
- **Segment Size:** En el modo segmentado, éste parámetro indica el número de muestras serán capturadas cada vez que se detecte una señal de disparo válida.
- **Pre-Trigger Samples:** En los modos post-trigger y segmentado, este parámetro indica el número de muestras que serán almacenadas en la RAM antes de que se produzca la señal de disparo.

- **Delayed Trigger:** Este parámetro fija un retardo entre la señal de disparo y la señal de disparo efectiva en la tarjeta. Este retardo toma valores desde 0 a 64k ciclos de reloj. En el modo de pre-trigger, este parámetro establece el número de muestras tras el disparo que serán almacenadas.

IV.3.1. Descripción del Hardware

Como se puede observar en la **Figura IV.2** la Tarjeta de Adquisición PDA14 es una tarjeta PCI de 64 bits, equipada con las funcionalidades 'Plug and Play', comunes en los sistemas PCI y que por tanto, podemos conectar fácilmente a la placa base de un ordenador.

La Tarjeta de Adquisición PDA14 es capaz de conseguir tasas de transferencia 250 MB/s llegando incluso hasta 266 MB/s. También puede funcionar en ranuras de expansión de la placa base PCI de 32 bits, pero en este caso las tasas de transferencia se limitan a unos 125 MB/s. Además, incorpora un avanzado bus auxiliar propio denominado SAB (Signatec Auxiliary Bus) que permite transferencias de datos de hasta 500 MB/s, permitiendo la transferencia de alta velocidad con otros periféricos tales como PMP8A de Signatec que es un DSP (Digital Signal Processor) muy útil cuando se necesiten aplicaciones que requieran operaciones numéricas a muy alta velocidad.

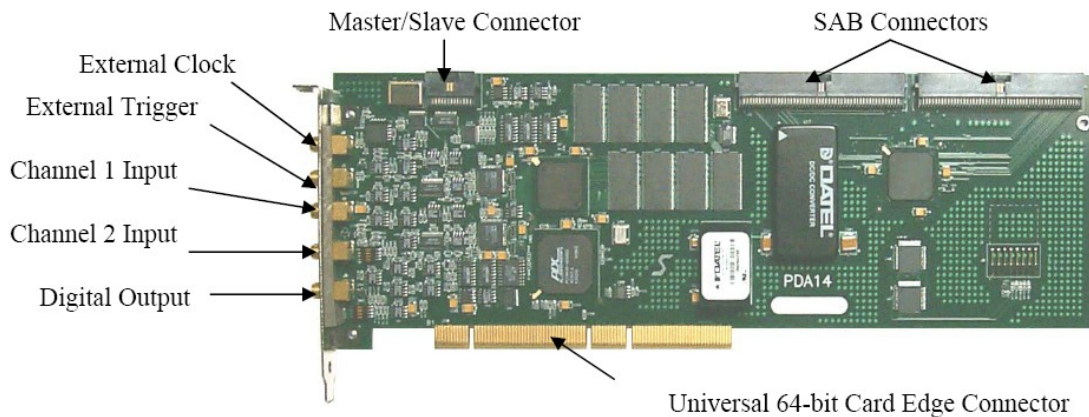


Figura IV.2 Tarjeta de Adquisición PDA14

La Tarjeta de Adquisición PDA14 está equipada de un puerto de interconexión para permitir la interconexión de múltiples tarjetas en una configuración maestro/esclavo. Hasta tres tarjetas auxiliares pueden funcionar con un solo maestro. En esta configuración el reloj y las señales de los disparadores de la tarjeta operando como maestro son las que actúan sobre cada uno de los esclavos de modo que ocurra el muestreo de los datos en todas las tarjetas simultáneamente.

Además de los conectores SMA para las señales de entrada, la PDA14 también proporciona los conectores SMA para una entrada de reloj, una señal de disparo y una señal de salida digital. La entrada de reloj se puede seleccionar como la frecuencia de muestreo del ADC. El disparador externo se puede utilizar para sincronizar el comienzo de la adquisición de

datos con un evento externo. Los parámetros del disparador tales como nivel de disparo, pendiente, etc. pueden programarse por el usuario. La salida digital es una señal seleccionable por el usuario útil para los propósitos de la sincronización.

La **Figura IV.3** siguiente muestra un diagrama funcional simplificado de la Tarjeta de Adquisición PDA14.

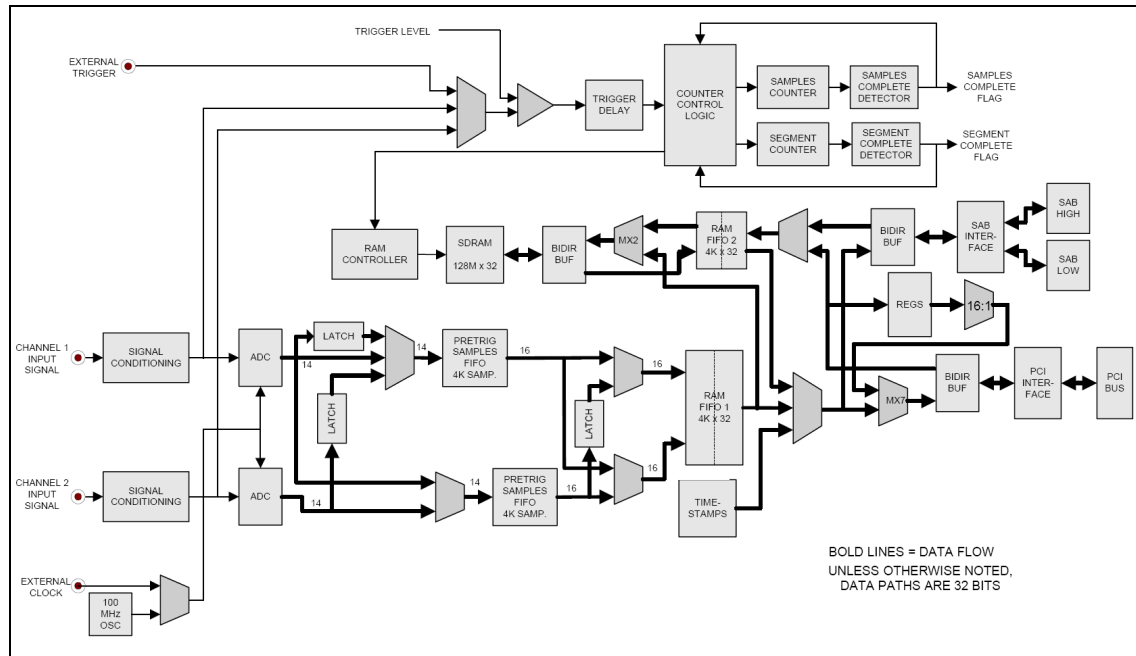


Figura IV.3 Diagrama Funcional de Tarjeta de Adquisición PDA14

Las señales de entrada se pueden fijar para el acoplador de CC o de CA. El bloque de acondicionamiento de la señal proporciona seis rangos de tensión desde 200 mV_{pp} hasta 3 V_{pp}. El filtro paso bajo fija la anchura de banda del canal a 50 MHz.

Los datos de entrada a la RAM FIFO 1 se pueden considerar como dos flujos de datos de 16 bits separados. Los interruptores y los multiplexores en la salida de los ADC permiten capturar datos de ambos canales o del canal 1 solamente o del canal 2 solamente.

Los registros de desplazamiento de las muestras de pre-trigger son programables en longitud hasta las 4k muestras. Pueden utilizarse para capturar muestras anteriores al disparo en el modo de post-trigger o en el modo segmentado. Antes de que se reciba el disparo, los datos se escriben en los registros de cambio pero no van más lejos. Después de recibir un disparo, las muestras de los datos comienzan a almacenarse en la RAM FIFO1.

Los convertidores A/D funcionan siempre en 100 MHz o 62.5 MHz cuando se utiliza el reloj interno. Sin embargo, también se puede utilizar una frecuencia de reloj externa. El funcionamiento a frecuencias de muestreo más reducidas se consigue eliminando los octetos de datos apropiados del flujo de datos. Así, la tasa de muestreo efectiva se divide a partir el 2 hasta 1024 en factores de 2.

Los datos se escriben en la SDRAM vía FIFO1 y se leen desde la RAM vía FIFO2. La RAM funciona a una frecuencia de reloj de 133 MHz así que tiene una capacidad de ancho de banda levemente mayor de 500 MB/s. Cuando los datos se están adquiriendo a una frecuencia de 250 MB/s o menos (62.5 MHz en 2 canales o 100 MHz en el solo canal) es posible operar la tarjeta en el modo Buffered Acquisition. En este modo la RAM funciona como una FIFO muy grande para adquirir datos directamente al bus **PCI** o al **SAB**.

Así, la PDA14 puede configurar el SAB para realizar transferencias de datos a 64 bits o a 32 bits sobre los puertos altos (SABH) o puertos bajos del bus (SABL). Esto proporciona flexibilidad cuando se incorporan múltiples tarjetas en un sistema. En 64 bits la tasa de transferencia máxima es 500 MB/s

La PDA14 es capaz de mantener durante un largo periodo una transferencia de datos sobre el bus PCI, de 250 MB/s cuando está instalado en una ranura PCI de 64 bits. Puede también instalarse en las ranuras tradicionales de 32 bits, aunque en este caso la tasa máxima de transferencia es aproximadamente de 125 MB/s.

De esta forma, durante su funcionamiento la Tarjeta de Adquisición PDA14 utiliza una serie de modos de operación en función del tipo tarea que en ese momento éste llevando a cabo. La lista de modos de operación se muestra a continuación:

- *Standby*
- *Acquisition to RAM*
- *Acquisition to PCI Bus*
- *Buffered Acquisition to PCI bus (RAM as FIFO)*
- *Acquisition to SAB*
- *Buffered Acquisition to SAB (RAM as FIFO)*
- *Data Transfer, RAM to PCI bus*
- *Data Transfer, RAM to SAB*
- *Write RAM (from PCI bus)*
- *Write RAM (from SAB)*
- *Read Time Stamps (from PCI bus)*
- *Read Time Stamps (from SAB)*

En el propio manual de operación de la PDA14 [12] se puede encontrar más información a cerca de cada uno de estos modos de operación.

IV.3.2. Descripción del Software

La PDA14 se suministra con una biblioteca de funciones escritas en el lenguaje de programación C y que se puede utilizar en programas para simplificar el trabajo de crear aplicaciones para utilizar la PDA14. Además, proporciona el código fuente para poder utilizarse con otros compiladores del Lenguaje C y para que puedan modificarse para adaptarse a las necesidades del usuario. Aunque Signatec da al usuario esta opción o flexibilidad adicional de modificar la biblioteca de software PDA14, realizar tales cambios es raramente necesario y la gran mayoría de usuarios utiliza simplemente las bibliotecas pre-construidas proporcionadas por Signatec. No obstante, es posible que sea necesaria alguna modificación para su uso con otros compiladores.

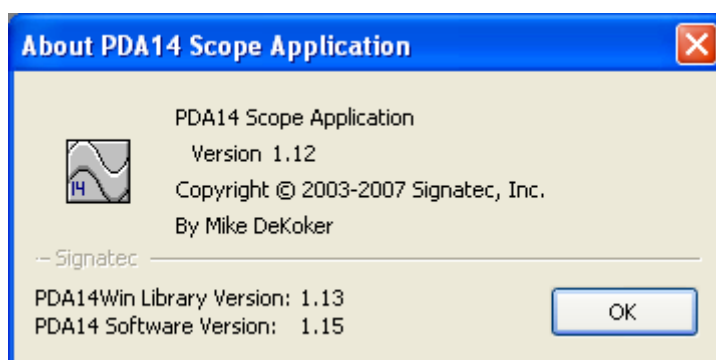


Figura IV.4 Versión del Software de Tarjeta de Adquisición PDA14

Las funciones de la biblioteca PDA14 han sido probadas y compiladas por el fabricante usando Microsoft's Visual C++ 6.0 y Microsoft's Visual Studio .NET.

La librería de funciones permite realizar gran variedad de operaciones sobre la PDA14 y que se pueden agrupar por funcionalidad tal y como se muestra en la **Tabla IV.1**:

<i>Data Acquisition Routines</i>		
AcquireToPciP14	AcquireToBoardP14	AcquireAndTransferP14
AcquireToSabP14	GetPciAcqDataP14	
<i>FFT Routines</i>		
FftStatsP14	RealFftP14	RealFftMagP14
<i>File Routines</i>		
BoardToDiskP14	DiskToBoardP14	DiskToPcP14
ReadSetupP14	WriteSetupP14	

Hardware Settings Routines		
IssueSoftTriggerP14	NullOffsetP14	
SetAddrCntAutoRstOvrP14 GetAddrCntAutoRstOvrP14	SetClockDividerP14 GetClockDividerP14	SetClockP14 GetClockP14
SetModeP14 GetModeP14	SetOffsetCh1P14 GetOffsetCh1P14	SetOffsetCh2P14 GetOffsetCh2P14
SetPreSamplesP14 GetPreSamplesP14	SetFreeRunModeP14 GetFreeRunModeP14	GetEffectiveClockRateP14
SetChannelsP14 GetChannelsP14	GetTimeStampDataP14	SetSegmentSizeP14 GetSegmentSizeP14
SetStartSampleP14 GetStartSampleP14	SetTotalSamplesP14 GetTotalSamplesP14	TrigInitP14
SetTrigDelayP14 GetTrigDelayP14	SetTrigLevelP14 GetTrigLevelP14	SetTrigModeP14 GetTrigModeP14
SetTrigSlopeP14 GetTrigSlopeP14	SetTrigSourceP14 GetTrigSourceP14	ReadHwRegP14
WriteHwRegP14	WriteAllHwRegsP14	SabInitP14
SetSabClockP14 GetSabClockP14	SetSabBoardNumP14 GetSabBoardNumP14	SetSabModeP14 GetSabModeP14
SetSabConfigP14 GetSabConfigP14	SetVoltRangeCh1P14 GetVoltRangeCh1P14	SetVoltRangeCh2P14 GetVoltRangeCh2P14
Hardware Monitor Routines		
GetFifoFullP14	GetSegmentDetectP14	GetSamplesCompleteP14
GetCurRamAddrP14		
Interrupt Routines		
CancelSampCompIntP14	EnableSampCompIntP14	WaitForSampCompIntP14
Memory Manipulation Routines		
BoardToPcP14	PcToBoardP14	GetCurRamAddrP14

<i>PCI DMA Buffer Routines</i>		
AllocateDmaBufferP14	FreeDmaBufferP14	FreeAllDmaBuffersP14
<i>Windows Device Driver Routines</i>		
InitDeviceP14	CloseDeviceP14	InitVirtualDeviceP14
<i>Miscellaneous Routines</i>		
SetCancelableFlagP14	GetSetupCommentSizeP14	ReadEEPROMP14
WriteEEPROMP14		
<i>PDA14 Configuration Routines</i>		
GetMasterSlaveConfigP14	GetLogicVersionP14	GetHardwareRevisionP14

Tabla IV.1 Librería de Funciones de la PDA14 agrupadas por funcionalidad

Además, juntos con éste software, el fabricante proporciona diversos ejemplos y algunas aplicaciones como un Osciloscopio Digital con el que nos facilita el aprendizaje.

IV.4. Discusión y conclusiones

Dadas las características de hardware, la Tarjeta de Adquisición PDA14 nos ofrece una solución al problema de detectar las señales acústicas procedentes de las descargas parciales. Asimismo, el software proporcionado por el fabricante facilita la implementación de un Sistema de Monitorización Continua de Descargas Parciales.

El primer objetivo es utilizar un solo canal de escucha desde el que detectar y almacenar las descargas parciales. Sin embargo, al poseer dos canales, en experimentos del laboratorio se puede incluir un canal de medida eléctrica, ya que permitiría comparar la actividad de descargas para ambas medidas. La posibilidad de introducir una señal de disparo externa permite la implementación de sistemas acústicos con disparo eléctrico, dejando libres los dos canales para la adquisición de emisiones acústicas.

No obstante, si fuera necesario implementar sistemas más complejos que nos permitieran no sólo detectar una descarga parcial, sino también conocer la ubicación de la fuente, dos canales no serían suficientes. Una posible solución podría consistir en ampliar la adquisición desde varios canales gracias a las configuraciones maestro/esclavo que ofrece la tarjeta de adquisición. Sin embargo, gracias a que la Tarjeta de Adquisición PDA14 tiene mucho más ancho de banda del que haría falta para un sensor de ultrasonidos como los usados habitualmente y que es 150 kHz, podemos obtener más canales con una sola tarjeta, simplemente multiplexando a la entrada [13] demultiplexando por software la señal capturada. La PDA14 obliga a seleccionar el canal desde el que se detectará la señal de disparo. Si se

tienen instalados más de un sensor interesa no perder información y esto puede ocurrir si la emisión acústica llega antes al sensor conectado al canal que no está seleccionado para detectar la señal de disparo. Con la multiplexación de canales a la entrada, este problema se soluciona, al llegar todas las señales a un único canal de la PDA14 que sería el que se seleccionaría para detectar la señal de disparo.

Al buscar una monitorización continua de la emisión acústica, lo que interesa es detectar cada ráfaga de emisión acústica, generalmente seleccionando el nivel de disparo adecuado, y almacenarla para su posterior tratamiento estadístico. Ese mismo procedimiento se sigue para medidas eléctricas de descargas parciales, para las cuales si es necesario el ancho de banda de un canal de PDA14.

Por otro lado, son importantes los instantes en que suceden las descargas parciales y sobre todo es importante la gran profundidad de memoria que le permite no perder eventos y almacenarlos en largos periodos de monitorización. Con las marcas de tiempo y las señales almacenadas está toda la información necesaria para caracterizar cada canal de escucha y la relación entre los mismos.

El modo segmentado proporcionado por la PDA14 impide la pérdida de eventos cuando el intervalo de tiempo entre uno y otro es superior a 150 nanosegundos. Así, seleccionando el máximo tamaño de adquisición 268435456 (2^{28}) muestras, adquiriendo sólo por uno de los dos canales, con la máxima frecuencia de muestreo que es 100 MHz, **podríamos adquirir de forma continua durante 2,68435456 segundos**. Según se va reduciendo la frecuencia de muestreo, la duración de la adquisición será mayor

La duración del fenómeno eléctrico de una descarga parcial no es superior a los 300 nanosegundos. Sin embargo, el fenómeno acústico de una descarga parcial tiene una duración de aproximadamente 150 microsegundos. De esta forma, empleando la máxima frecuencia de muestreo, para capturar una descarga parcial completa bastaría con capturar 15000 muestras. Así, gracias al modo segmentado, empleando segmentos 16384 (2^{14}) muestras, y empleando toda la memoria disponible, **podríamos capturar ráfagas de 16384 descargas parciales**.

Capítulo V

Interacción Remota con Entornos Experimentales

V.1. Introducción

En los últimos tiempos, las tecnologías Web no sólo nos han permitido la posibilidad de realizar operaciones más o menos sencillas a través de Internet. Además, están ofreciendo la posibilidad llevar a cabo tareas más complejas como por ejemplo las relacionadas con el mundo laboral. No es difícil encontrarse con personas que son capaces de realizar su actividad laboral desde casa o simplemente desde un sitio donde se dispone de un punto acceso a Internet. Sin embargo, estos nuevos servicios no serían posibles si no se hubieran solucionado los problemas derivados de la seguridad.

La aplicación de las tecnologías Web al mundo laboral se puede extrapolar a los entornos experimentales, donde también se deben resolver los problemas relacionados con la seguridad. Además, se presentan otros requisitos que se centran en la configuración de los sistemas de adquisición y otra serie de operaciones que nos permitan manipular los datos recuperados.

La protección de las aplicaciones distribuidas plantea un problema difícil de resolver sin recurrir a medidas que disminuyen el rendimiento. La autenticación mutua y el posterior cifrado del contenido es la única forma que permite estar razonablemente seguro de que la comunicación se produce entre entidades de confianza. Como consecuencia, se debe evaluar los requisitos de seguridad de la aplicación sin perder de vista las exigencias de rendimiento.

V.2. Entornos Experimentales en la Actualidad

En diferentes campos de investigación, se tiende cada vez más a la colaboración entre diferentes laboratorios. Este es el caso del Proyecto TJ-II en el CIEMAT [14]. Esta colaboración es debida no sólo a propósitos científicos, sino también a que las máquinas diseñadas cada vez son más complejas y caras de construir. Esto hace que en su diseño, construcción y explotación se reúnan esfuerzos de laboratorios de diferentes asociaciones para repartir el gasto entre las entidades colaboradoras. El diseño puede hacerse desde diferentes localizaciones geográficas pero para su construcción es necesario elegir un emplazamiento. Para que los científicos de los diferentes países que han cooperado en su diseño y construcción puedan participar en la explotación de resultados se han de desarrollar y probar herramientas que faciliten la interacción remota.

Otro ejemplo es el caso del proyecto ITER (International Thermonuclear Experimental Reactor) [15] que se está construyendo en Cadarache (Francia). Se trata de un proyecto de gran complejidad, ideado en 1986 para demostrar la factibilidad científica y tecnológica de la fusión nuclear y en el que actualmente participan la Unión Europea, Rusia, Estados Unidos, Corea del Sur, China, India y Japón.

La situación geográfica de un laboratorio de investigación no debe suponer un impedimento para que científicos situados en cualquier parte del mundo, puedan realizar sus tareas. Ésta es una de las premisas de las que parte cualquier gran instalación de investigación con participación multinacional.

V.2.1. Requisitos

Normalmente, cada instalación de investigación se desarrolla para unos determinados propósitos, centrándose en un determinado campo de la ciencia. Por tanto, es difícil generalizar sobre las características que debe cumplir un sistema que permita interactuar con este tipo de instalaciones.

Si nos centramos en un caso sencillo, como es la Monitorización de un Transformador de Alta Potencia, bastará con un sistema que ofrezca la posibilidad de:

- ⊕ Configurar los dispositivos de adquisición instalados en la subestación.
- ⊕ Monitorizar el estado de los sistemas.
- ⊕ Acceder a los datos obtenidos.

Sin embargo, como ya se ha mencionado, en la actualidad se tiende a la construcción de grandes infraestructuras a las que deben acceder investigadores que pueden estar localizados en cualquier parte del mundo. Incluso en nuestro caso, en la Monitorización de Transformadores de Alta Potencia, ya sea en un entorno de investigación o durante su actividad, cuando éstos están instalados en zonas de difícil acceso o simplemente se requiere la monitorización de

varias instalaciones a la vez, es necesario proporcionar acceso remoto a los sistemas de adquisición.

Un sistema que proporcione interacción remota con los sistemas de adquisición o simplemente con cualquier otro sistema sea cual sea su cometido, debe cumplir con una serie de requisitos típicos de las aplicaciones Web. Pensando en una aplicación Web no como un programa informático, sino como una plataforma de integración de servicios, estos son los requisitos que debe cumplir:

- ⊕ **Seguridad:** Autenticación y Autorización
- ⊕ **Alta disponibilidad:** Tolerancia a fallos.
- ⊕ **Escalabilidad:** Capacidad de crecer junto con las necesidades de sus usuarios y los requerimientos del mercado.
- ⊕ **Portabilidad:** Compatibilidad con los múltiples sistemas operativos y plataformas existentes en el mercado.

V.2.2. Acceso Remoto mediante Aplicaciones Web

Las aplicaciones Web suponen un importante cambio de enfoque con respecto al desarrollo de aplicaciones tradicionales. Su principal característica consiste en que la comunicación con el usuario se establece utilizando páginas Web, que se pueden visualizar desde un navegador que se esté ejecutando en cualquier ordenador conectado a la red. Otra característica importante consiste en que el código de la aplicación se puede ejecutar en el cliente, en el servidor o distribuirse entre ambos en función de las necesidades y limitaciones que podamos encontrarlos.

Otra característica básica de este tipo de sistemas es que, debido al gran volumen de información que pueden llegar a manejar, suelen utilizar una o varias Bases de Datos para organizar y facilitar el acceso a la información. Una aplicación Web de este tipo interrogará la Base de Datos, para de esta manera visualizar de nuevo los resultados en un navegador Web. Éste es el caso de las típicas aplicaciones que mediante formularios capturan los datos de entrada, se comunican con un Sistema Gestor de Bases de Datos (SGBD) y formatean la correspondiente información de salida para que pueda visualizarse en un navegador Web.

V.2.3. Ventajas de las Aplicaciones Web

Las aplicaciones Web son un tipo de aplicaciones que se caracterizan porque los usuarios pueden utilizarlas simplemente accediendo a un servidor Web a través de Internet o de una intranet mediante un navegador Web. Las aplicaciones Web son populares debido a lo práctico del navegador Web como cliente ligero, a la independencia del sistema operativo, así como a la facilidad para actualizar y mantener aplicaciones Web sin necesidad de distribuir e

instalar software a miles de usuarios potenciales. Por lo tanto, las aplicaciones Web proporcionan una serie de ventajas que pasamos a enumerar a continuación:

- ✦ **Portabilidad.** Las aplicaciones Web tienen un camino mucho más sencillo para la compatibilidad multiplataforma que las aplicaciones de software descargables. Varias tecnologías incluyendo PHP, Java, Flash, ASP y Ajax permiten un desarrollo efectivo de programas soportando todos los sistemas operativos principales.
- ✦ **Actualización.** Las aplicaciones basadas en tecnologías Web están siempre actualizadas con el último despliegue sin requerir que el usuario tome acciones proactivas y sin necesitar llamar la atención del usuario o interferir con sus hábitos de trabajo con la esperanza de que va a iniciar nuevas descargas y procedimientos de instalación (algunas veces imposible cuando se está trabajando dentro de grandes organizaciones).
- ✦ **Acceso Inmediato.** Las aplicaciones basadas en tecnologías Web no necesitan ser descargadas, instaladas y configuradas. Cada uno de los servicios proporcionados por una aplicación Web están listos para ser utilizados sin importar cuál es su configuración o su hardware.
- ✦ **Eficiencia.** Menos requerimientos de memoria. Al residir y ejecutarse en servidores, las aplicaciones Web liberan las CPU de los PCs de los usuarios, permitiendo la ejecución al mismo tiempo de otras tareas en dichos equipos, sin incurrir en deterioros de rendimientos.
- ✦ **Menos Errores (Bugs).** Las aplicaciones Web deberían ser menos propensas a colgarse y crear problemas técnicos debido a software o conflictos de hardware con otras aplicaciones existentes, protocolos o software personal interno. Con aplicaciones basadas en tecnologías Web todos utilizan la misma versión y todos los errores pueden corregirse tan pronto como son descubiertos. Esta es la razón por la cual las aplicaciones Web deberían tener muchos menos errores que el software de escritorio al que estamos acostumbrados.
- ✦ **Precio.** Las aplicaciones Web no requieren la infraestructura de distribución, soporte técnico y marketing requerido por el software de escritorio. Esto permite reducir los costes derivados de la distribución de nuevas versiones.
- ✦ **Accesibilidad.** A nadie le gusta no poder acceder a su cuenta de correo electrónico cuando está de viaje o no poder recuperar un documento particular cuando se conecta desde un ciber café a 15.000 kilómetros de su oficina. Con las aplicaciones Web este problema se resuelve, ya que se puede acceder a la información desde cualquier parte del mundo. Las aplicaciones Web son accesibles desde cualquier punto de la red, lo cuál significa que un usuario autorizado puede utilizarla si dispone de una conexión a Internet. Los navegadores Web se han desarrollado para todo tipo

de máquinas, por lo que cualquier usuario de Internet dispone de la herramienta básica para lanzar la aplicación.

- ✦ **Múltiples usuarios concurrentes.** Las aplicaciones basadas en tecnologías Web pueden utilizarse por múltiples usuarios al mismo tiempo.
- ✦ **Almacenamiento de datos más seguro.** Si bien la ruptura de discos no va a desaparecer, es probable que los usuarios dejen de oír hablar de este tema. A medida que las compañías se hagan cargo del almacenamiento de los datos del usuario, granjas de almacenamiento de datos redundantes, altamente fiables, serán la norma más que la excepción, y los usuarios van a tener mucho menos riesgo de perder sus datos debido a una ruptura de disco impredecible o a un virus en el PC. Las compañías que proveen aplicaciones Web brindan amplios servicios de seguridad de datos.

Además, los navegadores ofrecen cada vez más y mejores funcionalidades, como por ejemplo para soportar aplicaciones Web enriquecidas o RIA, acrónimo de Rich Internet Applications (Aplicaciones de Internet Enriquecidas).

V.2.4. Aplicaciones Web Enriquecidas (RIA)

Las aplicaciones Web enriquecidas son aplicaciones Web que tienen la mayoría de las características de las aplicaciones tradicionales y del mismo modo utilizan un navegador Web estandarizado para ejecutarse. Sin embargo, por medio de un plugin o una máquina virtual independiente, se agregan las características adicionales. Este tipo de aplicaciones surge como una combinación de las ventajas que ofrecen las aplicaciones Web y las aplicaciones tradicionales, buscando mejorar la experiencia del usuario.

Veamos un poco más en que consiste este caso particular de las Aplicaciones Web. Las RIA son un nuevo tipo de aplicaciones con más ventajas que las tradicionales aplicaciones Web. Normalmente en las aplicaciones Web hay una recarga continua de páginas cada vez que el usuario pulsa sobre un enlace. De esta forma, se produce un tráfico muy alto entre el cliente y el servidor, llegando muchas veces a recargar la misma página tras un mínimo cambio.

En los entornos RIA, en cambio, no se producen recargas de página, ya que desde el principio se carga toda la aplicación, y sólo se produce comunicación con el servidor cuando se necesitan datos externos como datos de una Base de Datos o de otros ficheros externos.

Otra de las desventajas de las tradicionales aplicaciones Web es la poca capacidad multimedia que posee. Para ver un vídeo es necesario utilizar un programa externo para su reproducción. Las capacidades multimedia son totales gracias a que estos entornos tienen reproductores internos y no hace falta ningún reproductor del Sistema Operativo del usuario.

Además, las RIA pueden hacer que las interacciones de los clientes sean llamativas, dinámicas y útiles; es definitiva, más atractivas.

Hay muchas herramientas para la creación de entornos RIA. Entre estas se puede mencionar las plataformas *Adobe Flash*, *Adobe Flex* y *Adobe AIR* de **Adobe**, *AJAX*, *OpenLaszlo*, *Silverlight* de **Microsoft**, *JavaFX* de **Sun Microsystems**, *Bindows* y *Javascript*.

Normalmente, se requieren otros elementos adicionales instalados en el sistema que ya se ocupan de descargar, actualizar, verificar y ejecutar las aplicaciones. Éstos son algunos ejemplos: *ExtJs*, *Adobe Flash*, *Oracle ADF Faces*, *Adobe AIR*, *Adobe Flex*, *ICEfaces*, *Sun JavaFX*, *Sun Java SE*, *Microsoft Silverlight*, *Curl* y *ZK Framework*.

Centrándonos en el caso de Adobe Flex, podemos destacar que es un marco de trabajo de código abierto. Se trata de un software altamente productivo para la creación y el mantenimiento de aplicaciones Web, las cuales se trabajan empleando los principales navegadores, equipos de sobremesa y sistemas operativos. Aunque las aplicaciones Flex se pueden crear utilizando únicamente el kit gratuito de desarrollo de software de Flex, los desarrolladores pueden utilizar el software **Adobe Flex™ Builder™ 3** para acelerar el desarrollo de las mismas.

El lenguaje de Flex se basa en estándares modernos, con un modelo de programación que admite los patrones de diseño habituales. **MXML** es un lenguaje declarativo basado en XML, y empleado para describir el aspecto y comportamiento de la interfaz de usuario. Por su parte, *ActionScript 3* es un potente lenguaje de programación que permite manejar objetos. Flex incorpora una biblioteca de componentes muy completa con más de 100 componentes de interfaz de usuario extensibles y de eficacia demostrada para crear aplicaciones enriquecidas, así como un depurador interactivo de aplicaciones de Flex.

Las RIA creadas con Flex pueden ejecutarse en el navegador utilizando el software Adobe Flash Player 9 o simplemente a través del escritorio utilizando Adobe AIR®. Esta funcionalidad permite a Flex ejecutarse en los navegadores más populares como Firefox o Internet Explorer y en múltiples sistemas operativos.

Se puede acelerar el desarrollo de aplicaciones con Adobe Flex Builder 3 o la última versión, Adobe Flash Builder 4.6, que es un entorno de desarrollo basado en Eclipse™ muy productivo que ofrece nuevas y potentes funcionalidades para la creación de Aplicaciones Web Enriquecidas.

V.2.5. Seguridad en Aplicaciones Web

Aunque, en principio, todo parecen ventajas de cara a adoptar la tecnología Web en el desarrollo de aplicaciones que proporcionen acceso remoto, hay un aspecto muy a tener en cuenta, al que hay que prestarle mucha atención. Se trata de la seguridad, sobre todo si se desea que la aplicación pueda lanzarse desde cualquier ordenador conectado a Internet. En este sentido, es necesaria la adopción de medidas que garanticen su utilización únicamente

por usuarios autorizados, sobre todo si las operaciones que se pueden realizar son críticas para el sistema.

La seguridad en Internet, uno de sus puntos más débiles, constituye uno de los aspectos que más interesan y a cuya investigación se dedica más esfuerzo y recursos, sobre todo si además se une el hecho de que es necesario realizar un control de utilización de la aplicación. Por ejemplo, en el caso de que el sistema utilice scripts CGI hay que prestar especial interés, entre otros, al paso de parámetros entre el cliente y el servidor, así como al encadenamiento de scripts.

Por ejemplo, cuando se envía información confidencial sobre nuestra persona, a través de formularios CGI, nos interesa que el servidor conozca los datos, pero no el resto de la Red, especialmente si se está realizando una transacción comercial electrónica y se envía el número de tarjeta de crédito, o simplemente la dirección. Se pone de manifiesto la necesidad de asegurar mediante algún mecanismo la intimidad y la integridad en las sesiones con el servidor Web.

V.3. Tecnologías Web

Los avances en las comunicaciones y las nuevas tecnologías están acercando la información al usuario final, así como facilitando su procesamiento. Uno de los cambios más importantes tiene que ver con el soporte y canal de transmisión de la información. Internet y las tecnologías Web han conseguido que el usuario esté familiarizado con información hipermedia, incluyendo texto, imágenes, audio y vídeo. Cualquier ordenador conectado a la red constituye una fuente fácil de entrada de información y de servicios.

En la actualidad, la mayoría de aplicaciones que se utilizan en entornos empresariales están construidas en torno a una arquitectura cliente-servidor, en la cual uno o varios equipos (generalmente de una potencia considerable) son los servidores, que proporcionan servicios a un número mucho más grande de clientes conectados a través de la red. Los clientes suelen ser PCs de propósito general, menos potentes y más orientados al usuario final. A veces los servidores son intermediarios entre los clientes y otros servidores más especializados (por ejemplo, los grandes servidores de bases de datos corporativos basados en mainframes y/o sistemas Unix). En este caso se habla de aplicaciones de varias capas.

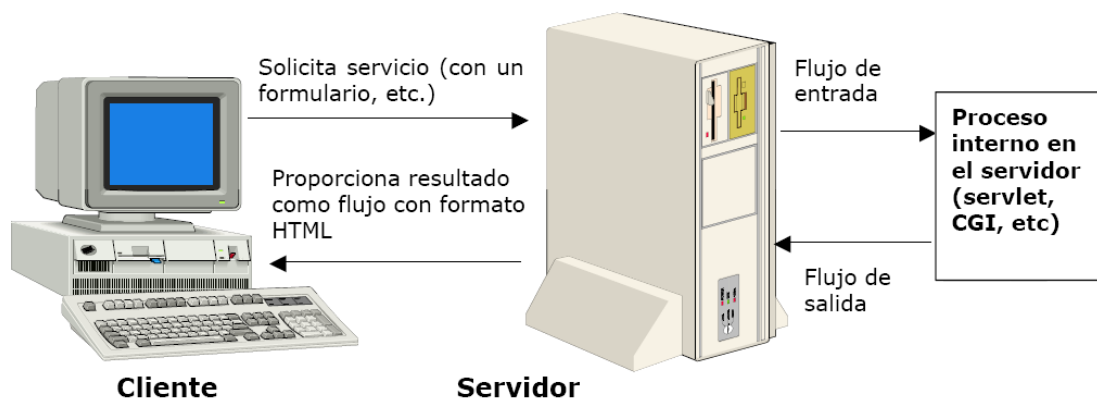


Figura V.1 Arquitectura Cliente - Servidor

Con el auge de Internet, la arquitectura cliente-servidor ha adquirido una mayor relevancia, ya que la misma es el principio básico de funcionamiento de la World Wide Web: un usuario que mediante un navegador (cliente) solicita un servicio (páginas HTML, etc.) a un computador que hace las veces de servidor. En su concepción más tradicional, los servidores HTTP se limitaban a enviar una página HTML cuando el usuario la requería directamente o accedía a ella a través de un enlace. La interactividad de este proceso era mínima, ya que el usuario podía pedir ficheros, pero no enviar sus datos personales de modo que fueran almacenados en el servidor u obtuviera una respuesta personalizada.

En las aplicaciones instaladas en el lado del servidor, el código de la aplicación se ejecuta en el servidor y en el cliente sólo recibe la respuesta normalmente en forma de páginas HTML, de este modo la compatibilidad está garantizada, ya que existe una independencia con respecto al navegador que interprete las páginas. Este modelo es el más adecuado cuando se utiliza una Base de Datos que ya reside en el servidor, de forma que se optimizan los accesos y se disminuye el ancho de banda utilizado. Otra de las situaciones en las que este mecanismo resulta adecuado cuando los procesos a ejecutar son operaciones críticas, que requieren unos chequeos de seguridad importantes, o cuando se necesita una potencia de cálculo que no se puede obtener en el cliente.

En cuanto a las tecnologías que permiten desarrollar aplicaciones de servidor nos encontramos con CGI y los Servlets de Java.

V.3.1. CGI

Las soluciones de primera generación incluyeron CGI (Common Gateway Interface) que no es más que una forma de comunicar información entre un servidor Web y un programa o script que está en la misma máquina que dicho servidor. No es un lenguaje de programación, sólo una forma de pasar información. La utilización de CGI permite al usuario introducir datos en una página (a través de formularios) y ejecutar procesos en un servidor, que como resultado, pueden devolver otra página Web, que es mostrada al usuario en el navegador. Esta página Web puede contener un nuevo formulario, que puede ser la entrada de otro script. De

esta forma, el sistema estará formado por una serie de programas que se ejecutarán independientemente y en distintos espacios de tiempo, que se comunicarán entre sí por medio de formularios y de ficheros en el servidor.

En principio, los programas CGI pueden estar implementados en cualquier lenguaje de programación, aunque en la práctica se han utilizado principalmente los lenguajes **Perl** y **C/C++**. Un claro ejemplo de un programa CGI sería el de un formulario en el que el usuario introdujera sus datos personales para registrarse en un sitio Web. El programa CGI recibiría los datos del usuario, introduciéndolos en la base de datos correspondiente y devolviendo al usuario una página HTML donde se le informaría de que sus datos habían sido registrados.

Una de las principales limitaciones de la utilización de este modelo consiste en la escalabilidad y la carga que supone para el servidor, ya que cada script que se ejecuta supone un proceso nuevo que se carga en memoria y, cuando varios usuarios acceden simultáneamente, provoca que se lancen múltiples copias del mismo código.

Es importante resaltar que estos procesos tienen lugar en el servidor. Esto a su vez puede resultar un problema, ya que, al tener múltiples clientes conectados al servidor, el programa CGI puede estar siendo llamado simultáneamente por varios clientes, con el riesgo de que el servidor se llegue a saturar. Como ya se ha mencionado, cada vez que se recibe una petición se arranca una nueva copia del programa CGI.

Un problema común en los CGI es que pueden permitir a un atacante ejecutar comandos en el Servidor. Un atacante habilidoso podría aprovecharse de programas CGI mal escritos para obtener los mismos privilegios que tendría un usuario a través de la línea de comandos (cuando se conecta vía Telnet o SSH al Servidor). Este problema de seguridad se relaciona con la forma en que se escriben los scripts y no con la seguridad del entorno del Servidor.

Además, existen otros riesgos adicionales que se veremos a continuación. El objetivo de este apartado es el estudio de la alternativa que Java ofrece a los programas CGI: los Servlets.

V.3.2. Servlets

Hasta no hace mucho, los programas CGI eran los únicos que permitían ejecutar código en servidores Web y capturar información enviada desde clientes Web. Los Servlets son la respuesta de la tecnología Java a la programación CGI. Ocurre que el modo de trabajo de los CGI es menos eficaz y ofrece menos posibilidades que los Servlets; no obstante, se siguen utilizando, aunque cada vez menos.

Hagamos un poco de historia. Java es un lenguaje desarrollado por Sun con la intención de competir con Microsoft en el mercado de la red. Sin embargo, su historia se remonta a la creación de una filial de Sun, FirstPerson, enfocada al desarrollo de aplicaciones para electrodomésticos, microondas, lavaplatos, televisiones... Esta filial desapareció tras un par de éxitos de laboratorio y ningún desarrollo comercial.

Sin embargo, para el desarrollo en el laboratorio, uno de los trabajadores de FirstPerson, James Gosling, desarrolló un lenguaje derivado de C++ que intentaba eliminar las deficiencias del mismo. Llamó a ese lenguaje Oak. Cuando Sun abandonó el proyecto de FirstPerson, se encontró con este lenguaje y, tras varias modificaciones (entre ellas la del nombre), decidió lanzarlo al mercado en verano de 1995.

El éxito de Java reside en varias de sus características. Java es un lenguaje sencillo o todo lo sencillo que puede ser un lenguaje orientado a objetos, eliminando la mayor parte de los problemas de C++, que aportó su granito (o tonelada) de arena a los problemas de C. Es un lenguaje independiente de plataforma, por lo que un programa hecho en Java se ejecutará igual en un PC con Windows que en una estación de trabajo basada en Unix. También hay que destacar su seguridad, ya que desarrollar programas que accedan ilegalmente a la memoria.

Cabe mencionar también su capacidad multihilo, su robustez o lo integrado que tiene el protocolo TCP/IP, lo que lo hace un lenguaje ideal para Internet. Pero es su sencillez, portabilidad y seguridad lo que le han hecho un lenguaje de tanta importancia.

La portabilidad se consigue haciendo de Java un lenguaje medio interpretado medio compilado. El código fuente se compila a un lenguaje intermedio cercano al lenguaje máquina pero independiente del ordenador y el sistema operativo en que se ejecuta (llamado en el mundo Java bytecodes) y, finalmente, se interpreta ese lenguaje intermedio por medio de un programa denominado máquina virtual de Java.

Este esquema lo han seguido otros lenguajes, como por ejemplo Visual Basic. Sin embargo, nunca se había empleado como punto de partida a un lenguaje multiplataforma ni se había hecho de manera tan eficiente. Cuando Java apareció en el mercado se hablaba de que era entre 10 y 30 veces más lento que C++. Ahora, con los compiladores JIT (Just in Time) se habla de tiempos entre 2 y 5 veces más lentos. Con la potencia de las máquinas actuales, esa lentitud es un coste que en la mayoría de los contextos se puede asumir sin problemas contemplando las ventajas de un lenguaje portable.

Dado que Java es un lenguaje orientado a objetos, es imprescindible entender qué es esto y en qué afecta a nuestros programas. Desde el principio, la carrera por crear lenguajes de programación ha sido una carrera para intentar realizar abstracciones sobre la máquina. Al principio no eran grandes abstracciones y el concepto de lenguajes imperativos es prueba de ello. Exigen pensar en términos del ordenador y no en términos del problema a solucionar. Esto provoca que los programas sean difíciles de crear y mantener al no tener una relación obvia con el problema que representa.

Muchos paradigmas de programación intentaron resolver este problema alterando la visión del mundo y adaptándola al lenguaje. Estas aproximaciones modelaban el mundo como un conjunto de objetos o de listas. Funcionaban bien para algunos problemas, pero no para otros. Los lenguajes orientados a objetos, más generales, permiten realizar soluciones que,

leídas, describen el problema. Permiten escribir soluciones pensando en el problema y no en el ordenador que debe solucionarlo en último extremo.

En el lado del servidor, Java es más popular que nunca, sobre todo desde la aparición de la especificación de Servlets y JSP (Java Server Pages). Hasta entonces, las aplicaciones Web dinámicas de servidor que existían se basaban fundamentalmente en componentes CGI y lenguajes interpretados. Como ya hemos visto ambos tenían diversos inconvenientes (fundamentalmente lentitud, elevada carga computacional o de memoria y propensión a errores por su interpretación dinámica).

Los Servlets y las JSPs supusieron un importante avance ya que:

- La API de programación es muy sencilla, flexible y extensible.
- Los Servlets no son procesos independientes (como los CGIs) y por tanto se ejecutan dentro del mismo proceso que la JVM mejorando notablemente el rendimiento y reduciendo la carga computacional y de memoria requeridas.
- Las JSPs son páginas que se compilan dinámicamente (o se pre-compilan previamente a su distribución) de modo que el código que se consigue una ventaja en rendimiento substancial frente a muchos lenguajes interpretados.

La especificación de Servlets y JSPs, que forma parte de la plataforma empresarial de Java (Java EE, Enterprise Edition), define una API de programación y los requisitos para un contenedor (servidor) dentro del cual se puedan desplegar estos componentes para formar aplicaciones Web dinámicas completas. En la actualidad existen multitud de contenedores (libres y comerciales) compatibles con estas especificaciones. Consecuentemente, los Servlets desarrollados para ejecutarse, por ejemplo en el servidor I-Planet Enterprise, se pueden ejecutar sin modificarse en Apache, Microsoft IIS, o WebStar.

Por tanto, la tecnología Servlet proporciona las mismas ventajas del lenguaje Java en cuanto a portabilidad (“*write once, run anywhere*”) y seguridad, ya que un Servlet es una clase de Java igual que cualquier otra y por tanto tiene en ese sentido todas las características del lenguaje. Esto es algo de lo que carecen los programas CGI, ya que hay que compilarlos para el sistema operativo del servidor y no disponen en muchos casos de técnicas de comprobación dinámica de errores en tiempo de ejecución.

A partir de su expansión entre la comunidad de desarrolladores, estas tecnologías han dado paso a modelos de desarrollo mucho más elaborados con frameworks (por ejemplo Struts, Spring) que se sobreponen sobre los Servlets y las JSPs para conseguir un entorno de trabajo mucho más poderoso y segmentado en el que la especialización de roles sea posible (desarrolladores, diseñadores gráficos,...) y se facilite la reutilización y robustez de código. A pesar de todo ello, las tecnologías que subyacen (Servlets y JSPs) son substancialmente las mismas.

Los Servlets, al estar programados en Java, son multiplataforma, es decir, se ejecutan del mismo modo independientemente del sistema operativo instalado en la máquina donde está corriendo el servidor. Como están escritos en Java, aportan todos los beneficios de este lenguaje. Java es lenguaje robusto y orientado a objetos, por lo que es fácilmente extensible a nuestras necesidades. Se ejecutan del mismo modo y sin retocar ninguna línea de su código cuando se cambian de servidor con soporte para Servlets. Son independientes del servidor utilizado y de su sistema operativo, lo que quiere decir que a pesar de estar escritos en Java, el servidor puede estar escrito en cualquier lenguaje de programación, obteniéndose exactamente el mismo resultado que si lo estuviera en Java.

Al igual que los programas CGI, los Servlets permiten la generación dinámica de código HTML dentro de una propia página HTML. Así, pueden emplearse Servlets para la creación de contadores, banners, etc.

Los Servlets nos permiten fácilmente hacer muchas cosas que son difíciles o imposibles con CGI normal. Por algo, los Servlets pueden hablar directamente con el servidor Web. Esto simplifica las operaciones que se necesitan para buscar imágenes y otros datos almacenados en el sistema de ficheros del servidor. Los Servlets también pueden compartir los datos entre ellos, haciendo las cosas útiles como almacenes de conexiones a bases de datos fáciles de implementar. También pueden mantener información de solicitud en solicitud, simplificando cosas como seguimiento de sesión y el caché de cálculos anteriores.

Además de la API de Servlets, se puede utilizar sin ninguna restricción la API estándar de Java. Por ejemplo, se puede utilizar JDBC para realizar consultas sobre una Base de Datos. En cambio, los programas CGI que, como ya hemos indicado, pueden programarse en varios lenguajes de programación como Perl, C++, etc. no siguen ningún tipo de especificación.

Los Servlets son más eficientes, fáciles de usar, más poderosos, más portables y más baratos que el CGI tradicional y otras muchas tecnologías del tipo CGI.

Frente a los CGI, muestran otras muchas ventajas entre las que destaca la eficiencia en el servidor. Los CGI pueden producir una sobrecarga en el servidor, ya que cada petición de un cliente supone la creación de un nuevo proceso. En cambio, mediante Servlets se optimiza la utilización del servidor, ya que por cada petición se crea un nuevo hilo de ejecución, pero sólo hay un proceso único ejecutándose en memoria, que atiende las diferentes peticiones. Por tanto, consumen menos recursos del servidor que las CGI, ya que no se cargan en memoria tantas veces como peticiones desde clientes se realicen sino que sólo se carga en memoria una instancia del Servlet la primera vez que se realiza una petición desde el cliente. El resto, se gestionan mediante hilos. En cambio, las CGI no admiten multithreading y cada petición implica carga en memoria de una CGI con la consiguiente lentitud y pesadez en su gestión. Puesto que sólo se cargan una vez, son mucho más rápidos

Además, los Servlets pueden comunicarse entre sí haciéndose posible una reasignación dinámica de la carga de procesos entre diversas máquinas. Con las CGI tradicionales se

arranca un nuevo proceso para cada solicitud HTTP. Si el programa CGI hace una operación relativamente rápida, la sobrecarga del proceso de arrancada puede dominar el tiempo de ejecución. Con los Servlets la máquina Virtual Java permanece arrancada y cada petición es manejada por un hilo de Java más ligero que un proceso del sistema operativo. De forma similar, en CGI tradicional, si hay N peticiones simultáneas para el mismo programa CGI, el código de este problema se cargará N veces en memoria. Sin embargo, con los Servlets hay N hilos, pero sólo una copia de la clase Servlet. Los Servlets también tienen más alternativas que los programas normales CGI para optimizaciones como los cachés de cálculos previos, mantener abiertas las conexiones de bases de datos, etc. Los Servlets pueden llamar a otros Servlets e incluso a métodos concretos de otros Servlets. De esta forma, se puede distribuir de forma más eficiente el trabajo a realizar. Por ejemplo, se podría tener un Servlet encargado de la interacción con los clientes y que llamara a otro Servlet para que a su vez se encargara de la comunicación con una base de datos. De igual forma, los Servlets permiten redireccionar peticiones de servicios a otros Servlets (en la misma máquina o en una máquina remota).

Otra de las principales ventajas de los Servlets con respecto a los programas CGI es la del rendimiento y, esto, a pesar de que Java no es un lenguaje particularmente rápido. Mientras que es necesario cargar los programas CGI tantas veces como peticiones de servicio existan por parte de los clientes, los Servlets, una vez que son llamados por primera vez, quedan activos en la memoria del servidor hasta que el programa que controla el servidor los desactiva. De esta manera, se minimiza en gran medida el tiempo de respuesta.

Además, los Servlets se benefician de la gran capacidad de Java para ejecutar métodos en ordenadores remotos, para conectar con bases de datos, para la seguridad en la información, etc. Se podría decir que las clases estándar de Java ofrecen resueltos muchos problemas que con otros lenguajes tiene que resolver el programador.

Los Servlets tienen una gran infraestructura para análisis automático y decodificación de datos de formularios HTML, leer y seleccionar cabeceras HTTP, manejar cookies, seguimiento de sesiones y muchas otras utilidades. Los Servlets pueden obtener fácilmente información acerca del cliente (la permitida por el protocolo HTTP), tal como su dirección IP, el puerto que se utiliza en la llamada, el método utilizado (GET, POST,...), etc. Permiten además la utilización de *cookies* y sesiones, de forma que se puede guardar información específica acerca de un usuario determinado, personalizando de esta forma la interacción cliente-servidor. Una clara aplicación es mantener la sesión con un cliente.

Los Servlets pueden actuar como enlace entre el cliente y una o varias bases de datos en arquitecturas cliente-servidor de 3 capas (si la base de datos está en un servidor distinto). Asimismo, pueden realizar tareas de proxy para un *applet*. Debido a las restricciones de seguridad, un applet no puede acceder directamente por ejemplo a un servidor de datos localizado en cualquier máquina remota, pero un Servlet sí podría hacerlo de su parte.

En cuanto a la seguridad en los Servlets, la única forma de invocar un Servlet es a través de un servidor de aplicaciones. Esto da un alto nivel de seguridad, especialmente si el servidor

de aplicaciones está protegido por un muro de contención (firewall). Esto significa que el cliente no puede borrar ni modificar nada del propio servidor. Para ampliar la seguridad, se pueden definir usuarios y grupos de usuarios. Por último, mencionar que también se pueden usar características nativas de seguridad, como el encriptamiento de mensajes.

Unido a la tecnología Servlet nos encontramos con las JSP. Las JavaServer Pages (JSP) están basadas en la tecnología de Servlets. Cuando se combina con el uso de componentes JavaBeans, JSP proporcionar una capacidad que es al menos tan poderosa como los Servlets, posiblemente más que un Servlet en crudo, y potencialmente mucho más fácil de usar.

La creación y compilación del Servlet es automática. Cada página JSP es compilada automáticamente en un Servlet por el motor JSP. (Sólo podemos usar JSP en servidores que sean compatibles con JSP).

La creación y compilación automática de un Servlet ocurre la primera vez que se accede a él. Dependiendo del comportamiento del servidor de aplicaciones, el Servlet será grabado durante algún periodo de tiempo para utilizarlo una y otra vez sin necesidad de recrearlo y recompilarlo. Por eso, la primera vez que se accede a la página, podría haber una pausa mientras que el servidor Web crea y compila el Servlet. Después de esto, los accesos a la página serán muchos más rápidos

Hay muchas formas diferentes de combinar JSP, JavaBeans y Servlets. Como se mencionó antes, el uso de JSP hace posible combinar las mejores capacidades del HTML con los componentes de software reutilizables para crear aplicaciones del lado del servidor.

Esto hace muy práctico separar la lógica del negocio de la representación de los datos. Con esto, los programadores especializados en escribir JavaBeans que implementen la lógica del negocio, y los diseñadores de páginas especializados en HTML pueden embeber llamadas a esos JavaBeans desde el HTML sin necesidad de convertirse en expertos programadores Java.

V.3.3. Seguridad en Servlets

Un requerimiento habitual a la hora de implementar aplicaciones Web es controlar el acceso a la aplicación. Las aplicaciones Web contienen recursos que pueden ser accedidos por muchos usuarios. Estos recursos atraviesan fronteras no protegidas (como Internet).

Cualquier empresa o institución que exponga sus servicios a Internet tendrá que proteger muy cuidadosamente su información y sus recursos. Un ejemplo sería el caso de los bancos que tendrán que prestar más atención en estos temas de la seguridad ya que la información que manejan es monetaria y no faltará más de una persona hábil en informática que quiera ganar dinero fácil. Pero en general, todas las aplicaciones Web deben estar resguardadas y aseguradas ante cualquier tipo de ataque. Generalmente, podemos hablar de 3 tipos de ataques:

- **Ataques anónimos.** Este tipo de ataques tratan de obtener información confidencial analizando (sniffing) las comunicaciones entre 2 PC's. Una manera de prevenir este tipo de ataques es encriptando los datos que son transmitidos. Por ejemplo, es común que las instituciones financieras utilicen el protocolo HTTPS para su aplicación Web online.
- **Ataques a la integridad.** Este tipo de ataques altera la información en tránsito con fines maliciosos. Se puede evitar esto usando técnicas de autenticación eficientes, como criptografía de llaves públicas.
- **Ataques de denegación de servicios.** Intenta inundar a un sistema de falsas peticiones y eso ocasiona que el sistema aparezca como no disponible para peticiones legítimas. Esto se previene usando firewalls que bloquean el tráfico de red en puertos no utilizados.

Otros tipos de ataques van referidos a:

- **Autenticación**, que se da con el ya conocido LOGIN; usuario y contraseña.
- **Autorización**, que se refiere a lo que el usuario puede hacer en el sistema luego de haber sido autenticado.
- **Integridad de los datos**, que se refiere a que si el usuario hace algún tipo de operación dentro del sistema, este no debe ser saboteado y sus datos no deber ser cambiados.
- **Confidencialidad**, que se refiere a que sólo el usuario indicado debe acceder a la información sensible. La diferencia con Autorización es que la confidencialidad asegura que incluso si la información cae en malos manos, esta sea inutilizable.

La especificación de los Servlets de Java proporciona métodos para implementar la seguridad en las aplicaciones Web. El servicio de Seguridad de la especificación de Servlets de Java garantiza que los recursos son accedidos por los usuarios autorizados, distinguiendo entre **autenticación** y **autorización**. Para ello, ofrece técnicas **declarativas** y **programáticas**.

Por otro lado, el protocolo SSL (Secure Socket Layer) es un sistema diseñado y propuesto por Netscape Communications Corporation, que proporciona sus servicios de seguridad cifrando los datos intercambiados entre el servidor y el cliente (un algoritmo de cifrado). Se utiliza una clave de sesión para cifrar los datos que se intercambian, generándose una clave distinta para cada transacción, lo cual permite que aunque sea reventada por un atacante en una transacción dada, no sirva para descifrar futuras transacciones.

Cuando se emplea la **seguridad programática**, el desarrollador es el encargado de implementar la seguridad escribiendo código en el programa para controlar el acceso a los recursos. Se trata de la implementación más común. Normalmente se utiliza cuando la

seguridad declarativa únicamente no es suficiente para expresar el modelo de seguridad de la aplicación.

La implementación habitual consiste en emplear formularios de acceso con los que solicitar a los usuarios los credenciales necesarias que les permitan acceder a la aplicación. Cuando el usuario rellena y envía el formulario, un Servlet pensado para controlar el acceso recibe la solicitud y autentifica al usuario. Normalmente, el Servlet se conecta a una base de datos y comprueba la correspondencia entre el login y la clave suministrados. Si tiene éxito, redirige al usuario al Servlet de entrada a la aplicación.

Es necesario evitar que los usuarios accedan directamente al Servlet de entrada: rastreo de sesiones. Se define un atributo en el objeto sesión cuando el usuario se autentifique.

Por otro lado, la **seguridad declarativa** se basa en el apoyo proporcionado por el contenedor. No existe un Servlet específico para controlar el acceso. Descansa sobre los mecanismos proporcionados por el contenedor de Servlets:

- **Autenticación:** las entidades que se comunican garantizan una a la otra que están actuando en nombre de usuarios específicos que tienen autorización de acceso a los recursos.
- **Control de acceso a los recursos:** la interacción con los recursos está limitada a usuarios o programas con el fin de garantizar la integridad, confidencialidad.
- **Integridad de datos:** para verificar que la información no ha sido modificada por terceros mientras estaba transmitiéndose.
- **Confidencialidad o privacidad de datos:** para asegurar que la información está disponible sólo para aquellos usuarios que tienen autorización de acceso a la misma.

Simplemente, la seguridad declarativa consiste en declarar, para cada recurso, los requerimientos de control de acceso y los privilegios (roles) que debería tener un usuario que acceda al mismo.

El contenedor implementa el mecanismo de autenticación, es decir, implementa el Servlet que realiza las funciones de login. Si detecta que no hay un usuario autenticado asociado con la sesión, redirige al usuario a una página de login, verifica el nombre y la clave y, si son válidos, permite el acceso del usuario

Para que el contenedor sepa que un recurso (un Servlet, una página Web o cualquier otro recurso) necesita autenticación de acceso, es necesario darle instrucciones al respecto. Por ello, el Desarrollador de la aplicación Web, debe añadir en el **descriptor de despliegue** la información necesaria para describir qué recursos necesitan autenticación de acceso.

La seguridad declarativa se refiere a los medios para expresar la estructura de seguridad de una aplicación. En la plataforma J2EE el mecanismo de autenticación se basa en roles: designan las responsabilidades y privilegios que tiene un usuario

El objetivo es que únicamente aquellos usuarios que tengan unos determinados privilegios, es decir, un rol, puedan invocar unos servicios determinados, unos Servlets específicos.

El descriptor de despliegue es el medio elemental para declarar esta seguridad en las aplicaciones Web. En él, el desarrollador de la aplicación indica qué requerimientos de seguridad tiene una aplicación

V.4. Java Platform, Enterprise Edition (Java EE)

La característica principal del lenguaje Java es que es un lenguaje de programación orientado a objetos. 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, que suelen inducir a muchos errores, como la manipulación directa de punteros o memoria.

La plataforma Java SE (Standard Edition) proporciona un entorno de ejecución y de desarrollo de aplicaciones Java: compilador, herramientas, módulos de ejecución y la API de Java que el que escribir, probar, implementar y ejecutar applets y aplicaciones.

Por otro lado, **la plataforma Java EE (Enterprise Edition) [16]** se apoya por completo en el lenguaje Java beneficiándose, por tanto, de sus características. Define el estándar para el desarrollo basado en componentes de aplicaciones empresariales distribuidas multinivel. Se basa en Java SE y ofrece servicios adicionales, herramientas y APIs para apoyar y simplificar el desarrollo de aplicaciones empresariales (JSP, Beans, Servlets, JSF, EJB,...). Añade librerías que proporcionan las funcionalidades de tolerancia a fallos y de software multinivel y distribuido, basado ampliamente en componentes modulares corriendo en un servidor de aplicaciones. Esta plataforma fue conocida como J2EE o Java 2 Platform, Enterprise Edition, hasta que su nombre cambió a Java EE en la versión 5. La última versión es Java EE 7.

A grandes rasgos, la plataforma Java EE proporciona:

- un modelo de desarrollo de componentes Web (Servlet, JSP) y de componentes activos (EJB) bajo la forma de APIs de Java,
- un conjunto de servicios (JDBC, JTA, JNDI, JMS, RMI/IIOP, JavaMail, XML), herramientas para los componentes, bajo la forma de APIs de Java.
- Un modelo de creación de módulos Web (.war), de módulos EJB (.jar) y de módulos corporativos (.ear), asociados a descriptores de despliegue en formato XML, herramientas para el desarrollo de aplicaciones de empresa.
- Contenedores (Web y EJB), para la ejecución de componentes.

La plataforma Java EE está definida por una especificación. Similar a otras especificaciones del Java Community Process (JCP), Java EE es también considerada

informalmente como un estándar debido a que los proveedores deben cumplir ciertos requisitos de conformidad para declarar que sus productos son conformes a Java EE.

V.4.1. Componentes de la Arquitectura Java EE

Como se puede observar en la **Figura V.2** la plataforma Java EE ofrece un modelo de aplicaciones distribuidas multinivel, componentes reutilizables, un modelo de seguridad unificado, control flexible de transacciones y soporte para Web Services a través del intercambio integrado de datos con estándares y protocolos basados en XML (Extensible Markup Language).

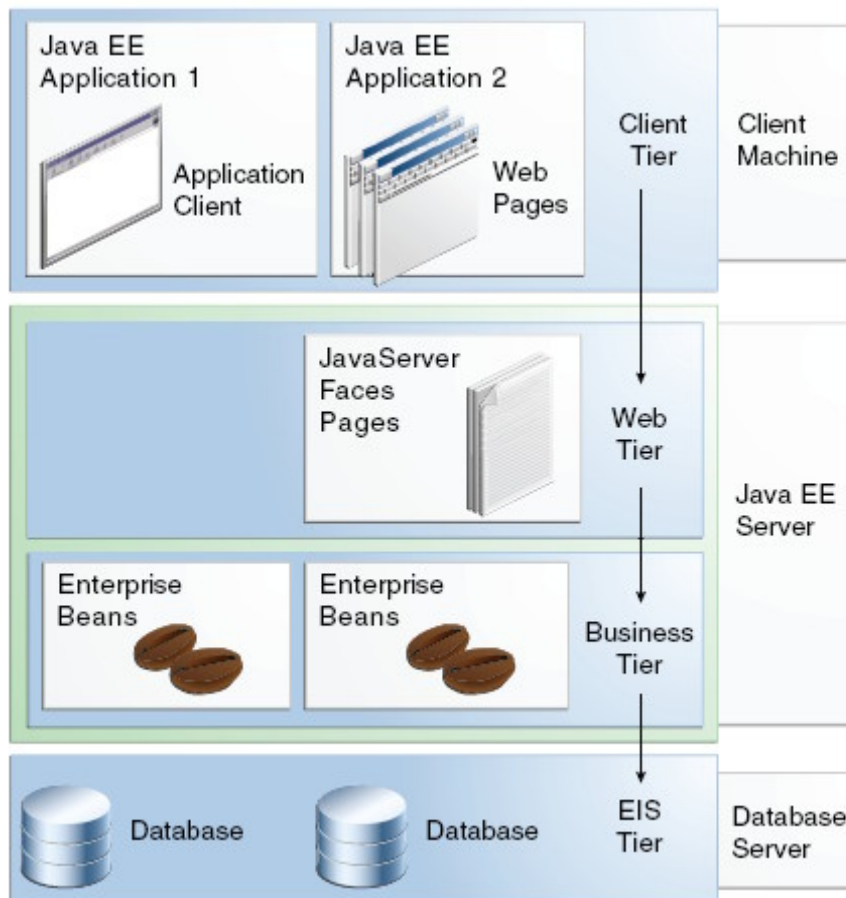


Figura V.2 Arquitectura Multinivel proporcionada por Java EE

Las aplicaciones Java EE están basadas en componentes. Un componente es una unidad de software auto-contenida que es utilizada en una aplicación Java EE con sus clases relacionadas y archivos adicionales que además se comunican con otros componentes. La especificación Java EE define los siguientes componentes Java EE:

- Clientes de aplicación y Applets clientes Web (navegadores), clientes de escritorio Java (Swing, AWT) y otros clientes de escritorio que no tienen por qué desarrollarse en Java, que encontramos en el **lado del cliente**.

- **Componentes Web**, como Java Servlets, Java Server Pages (JSP) y Java Server Faces (JSF), que encontramos en el **lado del servidor** y son los responsables de la presentación de datos apoyándose en el uso de componentes JavaBeans
- **Componentes de Negocio** como EJB (Enterprise JavaBeans) y entidades JPA (Java Persistence API) que también encontramos en **lado del servidor** y que implementan la lógica de la aplicación. Los EJBs gestionan interacciones con los clientes e implementan reglas de negocio mientras que las entidades JPA representan los datos del Nivel EIS (Enterprise Information Systems).
- **EIS (Enterprise Information System)**, que gestiona la información permanente del sistema. Nos podemos encontrar con Bases de Datos o aplicaciones empresariales "heredadas" (*legacy systems*) que actúan como almacenes de datos.

V.4.2. Contenedores Java EE

Los contenedores son la interfase entre los componentes y la funcionalidad específica de la plataforma a bajo nivel que soporta el componente. Antes que un componente Web, bean empresarial o cliente de aplicación pueda ejecutarse, debe ensamblarse en un módulo Java EE y desplegado dentro de su contenedor.

El proceso de ensamblado involucra configuración específica de contenedor para cada componente en la aplicación Java EE y para la aplicación Java EE misma. La configuración de los componentes define el soporte de capas bajas proporcionadas por el servidor, incluyendo servicios como seguridad, gestión de transacciones, consulta de directorios de nombres (JNDI) y conectividad remota. Algunos de los principales son:

- **El modelo de seguridad de Java EE** que permite configurar un componente Web o bean empresarial para que los recursos del sistema sean accedidos solo por usuarios autorizados.
- **El modelo de transacciones de Java EE** que nos permite especificar las relaciones entre métodos que conformen una sola transacción de tal forma que todos los métodos en una transacción sean tratados como una sola unidad.
- **Los servicios de búsqueda JNDI** que proporcionan una interfase unificada para muchos servicios de nombres y directorio en la empresa de tal manera que los componentes de la aplicación puedan acceder a estos servicios.
- **El modelo de conectividad remota de Java EE** que maneja las comunicaciones a bajo nivel entre clientes y beans empresariales. Tras la creación de un bean empresarial, un cliente podría invocar los métodos de éste como si estuviera en la misma máquina virtual.

Dado que la arquitectura Java EE proporciona servicios configurables, los componentes de aplicación dentro de la misma aplicación Java EE pueden comportarse de forma diferente dependiendo de donde se hayan desplegado. Por ejemplo, un bean empresarial puede tener configuraciones de seguridad que le permitan un cierto nivel de acceso a los datos de una base de datos en un entorno de producción y otro nivel de acceso a base de datos en otro entorno de producción.

El contenedor también maneja servicios que no son configurables como EJBs y los ciclos de vida de los Servlets, agrupaciones de recursos de conexiones a bases de datos, persistencia y el acceso a las APIs de la plataforma Java EE.

Los contenedores Java EE ofrecen servicios a los componentes actuando como interfaz entre un componente y los aspectos de bajo nivel de cada plataforma. Simplifican el desarrollo de los componentes ocultando los detalles complejos. Sus tareas concretas dependen de la capa a la que pertenezcan. Estos servicios ofrecidos son los que ya se han comentado: seguridad, gestión de transacciones, consulta de directorios de nombres (JNDI),...

El proceso de despliegue instala componentes de aplicación Java EE en los contenedores tal y como se ilustra en la **Figura V.3**.

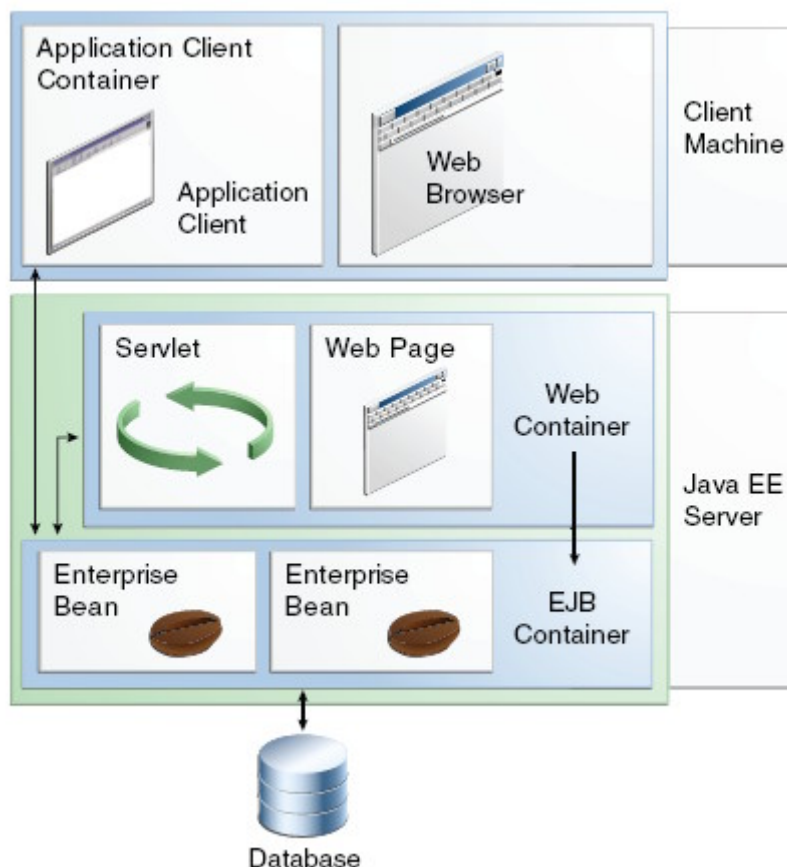


Figura V.3 Modelo de Contenedores proporcionado por Java EE

Los tipos de contenedores definidos en la plataforma Java EE son los siguientes:

- **Servidor de Aplicaciones Java EE**, que provee el entorno de ejecución de aplicaciones Java EE. Proporciona el contenedor Web y/o el contenedor de EJBs.
- **Contenedor EJB**, que controla la ejecución de beans empresariales para aplicaciones Java EE. Los bean empresariales y su contenedor se ejecutan en el servidor Java EE. Su función es la de gestionar la ejecución de los EJBs (su ciclo de vida), dar soporte a los componentes que implementan la capa de lógica de negocio y ofrecer servicios de control de acceso y seguridad, control de transacciones y ejecución concurrente,... Hay otras alternativas que ofrecen funcionalidades similares para implementar la lógica de negocio sin tener que usar EJBs, como por ejemplo Spring. Algunos ejemplos libres son JBoss, OpenEJB, ...
- **Contenedor Web**, que maneja la ejecución de las JSPs y los componentes Servlets para aplicaciones Java EE. Los componentes Web y sus contenedores se ejecutan en el servidor Java EE. Su función es la de gestionar la ejecución de los Servlets y JSP y dar soporte a los componentes que implementan la capa Web. Implementa la API de Servlets, así como las APIs de JSP, JSTL, JSF, JavaBeans. Algunos ejemplos libres son Tomcat, Jetty, ...
- **Contenedor de aplicación cliente**, que maneja la ejecución de los componentes de una aplicación cliente. La aplicación cliente y sus contenedores se ejecutan en el cliente.
- **Contenedor de applet**, que maneja la ejecución de Applets. Consiste en un navegador Web y un plugin que se ejecutan juntos en el cliente.

V.4.3. Tecnologías Java EE

Java y la plataforma Java EE aportan en la actualidad una solución única, fiable y eficaz para el desarrollo e implementación de aplicaciones Web y aplicaciones distribuidas mediante tecnologías modernas, permitiendo además acceder a los sistemas existentes.

Java EE incluye varias especificaciones (API), tales como JDBC, RMI, JavaMail, JMS, Servicios Web, XML,... y define cómo coordinarlos. Java EE también configura algunas especificaciones únicas para Java EE para componentes. Estas incluyen Enterprise JavaBeans, Servlets, JavaServer Pages y varias tecnologías de servicios Web. Esto permite al desarrollador crear una Aplicación Web portable entre plataformas y escalable, a la vez que integrable con las tecnologías anteriores. Otros beneficios añadidos son, por ejemplo, que el servidor de aplicaciones puede manejar transacciones, la seguridad, escalabilidad, concurrencia y gestión de los componentes desplegados, significando que los desarrolladores pueden concentrarse más en la lógica de negocio de los componentes en lugar de en tareas de mantenimiento de bajo nivel.

Java EE se basa en Java SE, por lo que todas las APIs de Java SE están disponibles. Sin embargo, en Java EE se definen nuevas API específicas para el desarrollo de sistemas distribuidos multinivel. Con el tiempo, algunas han acabado formando parte de Java SE (por ejemplo JDBC y JPA). A continuación se procederá a realizar una breve introducción a estas APIs con el objetivo de mostrar que tipo de funcionalidades ofrece esta plataforma.

- **Tecnología Enterprise JavaBeans**

Un componente JavaBean Empresarial (EJB) o bean empresarial es un elemento de código que tiene campos y métodos para implementar módulos de lógica de negocio. Se puede pensar en un bean empresarial como un ladrillo que puede utilizarse solo con otros beans empresariales para ejecutar lógica de negocio en el servidor Java EE.

Hay dos tipos de beans empresariales: los beans de sesión (Session EJBs) y los beans dirigidos por mensajes (Message-driven EJBs). Un bean de sesión representa una conversación transitoria con un cliente. Cuando el cliente termina de ejecutarse, el bean de sesión y sus datos desaparecen. Un bean dirigido por mensajes combina las características de un bean de sesión y un listener de mensajes permitiendo a un componente de negocio recibir mensajes de forma asíncrona. Comúnmente, estos son mensajes JMS (Java Message Service).

La API de EJB proporciona un modelo de diseño de componentes utilizadas en las aplicaciones basadas en una arquitectura distribuida multinivel. La API proporciona también un conjunto de servicios (persistencia, transacción,...) asociados a los componentes cuya gestión se delega al contenedor de EJB.

- **Tecnología Java Servlet**

La tecnología de Servlet de Java permite definir una clase Servlet específica para HTTP. Una clase Servlet extiende las habilidades de los servidores que hospedan aplicaciones que son accedidas mediante el modelo de programación petición-respuesta. A pesar de que los Servlets pueden responder a cualquier tipo de solicitud, comúnmente son utilizados para extender las aplicaciones ofrecidas por servidores Web.

La API de Servlet proporciona los elementos básicos que permiten la creación de componentes Web dinámicos en Java.

- **Tecnología JavaServer Faces**

La tecnología de JavaServer Faces (JSF) es un marco de trabajo de interfaces de usuario para crear aplicaciones Web. Los componentes principales de la tecnología de JavaServer Faces son los siguientes:

- Un marco de trabajo para componentes GUI.
- Un modelo flexible para dibujar componentes en diferentes tipos de HTML, lenguajes de marcas y tecnologías.
- Un *RenderKit* estándar para generar marcas HTML/4.01.

Los componentes GUI, soportan las siguientes características:

- Validación de entrada.
- Manejo de eventos.
- Conversión de datos entre objetos de modelo y componentes.
- Creación administrada de objetos modelo.
- Configuración de navegación en la página.

Toda esta funcionalidad está disponible utilizando la API estándar y los ficheros de configuración basados en XML.

▪ **Tecnología JavaServer Pages**

La tecnología de JavaServer Pages (JSP) permite colocar partes del código de un Servlet directamente en un documento de texto. Una JSP es un documento de texto que contiene dos tipos de texto: datos estáticos (que pueden expresarse en formatos basados en texto como HTML, WML y XML) y elementos JSP, que determinan como la página construye el contenido dinámico.

La API de JSP permite llevar a cabo la creación de páginas Web dinámicas, basadas en un esqueleto HTML con datos incluidos dinámicamente en código Java.

▪ **Librería estándar de etiquetas de JavaServer Pages**

La librería estándar de etiquetas de JSPs (JSTL) encapsula la funcionalidad principal común a muchas JSP. En lugar de etiquetas mezcladas de varios proveedores en sus JSP, se emplea un solo grupo estándar de etiquetas. Esta estandarización permite, desplegar sus aplicaciones en cualquier contenedor que soporte JSTL y lo hace más deseable que la implementación de etiquetas optimizada.

JSTL facilita etiquetas para crear iteraciones y condiciones para manejar flujo de la respuesta, etiquetas para manejar documentos XML, etiquetas de internacionalización, etiquetas para acceder a bases de datos utilizando SQL y las funciones mas comúnmente utilizadas.

▪ **API Java de Persistencia**

La API Java de persistencia (JPA) es una solución Java basada en estándares para persistencia. La persistencia utiliza una estrategia de "mapeo" objeto relacional para unir la

brecha entre el modelo orientado a objetos y la base de datos relacional. La persistencia de Java consiste en tres áreas:

- La API Java de persistencia
- El lenguaje de consultas
- Los datos de alto nivel con el mapeo objeto relacional

▪ **API Java de Transacciones**

La API Java de transacciones (JTA) proporciona una interfaz estándar para distinguir transacciones. La arquitectura Java EE proporciona un *commit* automático para manejar *commit* y *rollbacks*. Un *auto commit* significa que otras aplicaciones que están viendo estos datos verán los datos actualizados después de que cada base de datos realice la operación de lectura o escritura. Sin embargo, si la aplicación realiza dos operaciones de acceso a base de datos que dependen una de otra es conveniente utilizar JTA para distinguir cuando la transacción se ha completado incluyendo ambas operaciones.

JTA permite utilizar la gestión de las transacciones en las aplicaciones distribuidas, dividiendo una misma transacción entre distintos componentes EJB y/o compartiendo una misma transacción entre distintas conexiones de base de datos.

▪ **API Java para Servicios Web REST**

JAX-RS proporciona una API para la construcción de servicios Web al estilo de la arquitectura REST (Representational State Transfer). La API consiste en una serie de anotaciones, interfaces y clases asociadas. Aplicando las anotaciones a POJOs (Plain Old Java Objects) se pueden exponer servicios REST.

▪ **Managed Beans**

Los Managed Beans son objetos ligeros gestionados por el contenedor, también conocidos como POJOs (Plain Old Java Objects) con unos requisitos mínimos, soportan un pequeño conjunto de servicios básicos, inyección de recursos, ciclos de vida e interceptores.

▪ **Contextos e Inyección de Dependencias en Java EE**

Los contextos y la inyección de la dependencias (CDI) en la plataforma de Java EE define un sistema de servicios contextuales, proporcionados por los contenedores Java EE, que facilitan a los desarrolladores el empleo de EJBs junto con la tecnología JavaServer Faces en aplicaciones Web. Diseñado para el uso con los objetos con estado, el CDI también tiene muchas aplicaciones más amplias, permitiendo a los desarrolladores mucha de flexibilidad para integrar diversas clases de componentes de una manera débilmente acoplada pero segura.

- **Inyección de Dependencias en Java**

En Java, la inyección de dependencias define un conjunto estándar de anotaciones (y un interfaz) para su uso en clases inyectables. En la plataforma de Java EE, el CDI proporciona el soporte necesario para el empleo de Inyección de Dependencias. Específicamente, sólo se puede utilizar los puntos de inyección en aplicaciones en las que esté habilitada la inyección de dependencias.

- **Bean Validation**

Bean Validation es una especificación de Java (JSR-303) que define un modelo de los metadatos y una para validar datos en los componentes de los JavaBeans. En lugar de utilizar una validación de los datos distribuida sobre varias capas, tales como el navegador y el lado servidor, se pueden definir restricciones de validación en un lugar y que éstos se compartan a través de las diferentes capas.

Se trata de una API para validación en JavaBeans basada en anotaciones, que simplifica las validaciones y reduce duplicidad y errores en beans.

- **API Java de servicio de mensajes**

La API del servicio de mensajes de Java (JMS) es un estándar de mensajería que permite a componentes de aplicación Java EE crear, enviar, recibir y leer mensajes. Para habilitar comunicación distribuida con bajo acoplamiento, confiable y asíncrona.

- **Arquitectura de Conectores J2EE**

La arquitectura de conectores J2EE es utilizada por los proveedores de herramientas e integradores de sistemas para crear adaptadores que soporten el acceso a sistemas de información empresarial que puedan conectarse en cualquier producto Java EE. Un adaptador de recurso es un componente de software que permite a los componentes de aplicación Java EE acceder e interactuar con el manejador de recursos de capas bajas del EIS. Dado que un adaptador de recurso es específico de su manejador de recursos, típicamente hay un adaptador de recursos diferente para cada tipo de base de datos o sistema de información empresarial.

La arquitectura del conector J2EE también proporciona una integración transaccional orientada a rendimiento, seguro, escalable y basada en mensajes de servicios Web basados en Java EE con EISs existentes que pueden ser sincrónicos o asíncrónicos. Aplicaciones existentes y EISs integrados a través de la arquitectura del conector J2EE dentro de la plataforma Java EE pueden exponer servicios Web basados en XML utilizando JAX-WS y modelos de componentes Java EE. De esta manera JAX-WS y la arquitectura del conector J2EE son tecnologías complementarias para integraciones de aplicaciones empresariales (EAI) e integraciones de negocios de un extremo al otro.

- **API JavaMail**

La API JavaMail ofrece a las aplicaciones Java funcionalidades de gestión de correo electrónico. Esta API tiene dos partes: una interfaz a nivel de aplicación utilizada por los componentes de la aplicación para enviar correo y una interfaz de proveedor de servicio. La plataforma Java EE incluye JavaMail con un proveedor de servicio que permite a los componentes de aplicación enviar correo por la Internet.

- **Java Authorization Contract for Containers**

La especificación Java Authorization Contract for Containers (JACC) define los contratos de seguridad entre un servidor de aplicaciones Java EE y un proveedor de políticas de autorizaciones. Todos los contenedores Java EE soportan este contrato. La especificación JACC define las clases *java.security.Permission* que satisfacen el modelo de autorización Java EE.

- **Java Authentication Service Provider Interface for Containers**

La especificación Java Authentication Service Provider Interface for Containers (JASPIC) define una interfaz proveedora de servicio (SPI, Service Provider Interface) por el cual los proveedores de autenticación que implementan mecanismos de mensajería de autenticación puedan integrarse en clientes, contenedores o en cualquier entorno de ejecución.

Los proveedores de autenticación integrados a través de este interfaz funcionan a través de mensajes de red proporcionados por el contenedor llamante. Los proveedores de autenticación transforman mensajes de salida para poder autenticar la fuente del mensaje por el contenedor de recepción, y el recipiente del mensaje pueda autenticarse por el remitente del mensaje. Los proveedores de autenticación, autentican mensajes entrantes y vuelven a su contenedor llamante la identidad establecida como resultado del mensaje de autenticación.

- **API JAVA de conectividad a bases de datos**

La API Java de conectividad a base de datos (JDBC) permite invocar comandos SQL desde métodos del lenguaje de programación Java. Se utiliza JDBC en un bean empresarial cuando se tiene un bean de sesión accediendo a la base de datos. Se puede utilizar también la API de JDBC desde un Servlet o una JSP para acceder a la base de datos directamente sin pasar a través de un bean empresarial.

JDBC tiene dos partes: una interfaz a nivel de aplicación usado por los componentes de aplicación para acceder a la base de datos y una interfaz de proveedor de servicio para anexar un controlador JDBC a la plataforma Java EE.

La API de JDBC (Java Database Connectivity) permite la ejecución de operaciones sobre bases de datos desde el lenguaje de programación Java. Además la extensión de la API JDBC 2.0 facilita la obtención de conexiones JDBC para la utilización de fuentes de datos que

ocupan de la gestión de un conjunto de conexiones. La API proporciona además un mecanismo que permite gestionar transacciones distribuidas entre distintas bases de datos basándose en JTA.

- **Interfase de Nombres y directorios Java**

La Java Naming and Directory Interface (JNDI) proporciona la funcionalidad de nombres y directorios, habilitando las aplicaciones a acceder a múltiples servicios de nombres y directorios, incluyendo los servicios de nombres y directorios existentes como LDAP, NDS, DNS, y NIS, proporcionando una implementación de SPI (Service Provider Interface). Ésta le proporciona a las aplicaciones métodos para realizar operaciones de directorio estándares, como lo es asociar atributos con objetos y la búsqueda de objetos utilizando sus atributos. Usando JNDI una aplicación Java EE puede almacenar y recuperar cualquier tipo de objeto nombrado, permitiendo a las aplicaciones Java EE coexistir con muchas aplicaciones y sistemas legados.

Un componente Java EE puede acceder a objetos nombrados provistos por el sistema y definidos por el usuario. Los nombres de los objetos provistos por el sistema, como objetos UserTransaction de JTA, son almacenados en el contexto de nombres, `java:comp/env`. La plataforma Java EE permite a un componente nombrar objetos definidos por el usuario como bean empresariales, objetos DataSource de JDBC y conexiones de mensajes. Un objeto puede nombrarse dentro de un subcontexto del contexto de nombres de acuerdo al tipo de objeto. Por ejemplo, los beans empresariales son nombrados dentro del subcontexto `java:comp/env/ejb`, y las referencias a objetos DataSource de JDBC en el subcontexto `java:comp/env/jdbc`.

- **Marco de trabajo de activación de JavaBeans**

El marco de trabajo de activación JavaBeans (JAF) está incluido porque JavaMail lo utiliza. JAF proporciona servicios estándares para determinar el tipo de un trozo arbitrario de datos, acceso encapsulado al, descubrimiento de operaciones disponibles y creación de los componentes JavaBeans apropiados para realizar estas operaciones.

- **API Java para procesamiento XML**

La API de Java para procesamiento XML (JAXP), parte de la plataforma Java SE, soporta el procesamiento de documentos XML utilizando el Modelo de objetos del documento (DOM), API simple para XML (SAX) y Transformaciones del Lenguaje Extensible de Hojas de estilo (XSLT). JAXP habilita a las aplicaciones a analizar la sintaxis y transformar documentos XML independientemente de una implantación de procesado XML particular.

JAXP también proporciona soporte para espacios de nombres, lo que permite trabajar con esquemas que puedan, de otra forma tener nombres con conflicto.

- **Arquitectura Java para enlazar XML (JAXB)**

La arquitectura Java para enlazar XML (JAXB) proporciona una forma conveniente de vincular un esquema XML para representar en programas en lenguaje Java. JAXB puede utilizarse independientemente o en combinación con JAX-WS, donde proporciona una vinculación estándar de datos para mensajes de servicios Web. Todos los contenedores de aplicaciones cliente de Java EE, contenedores Web y EJB soportan la API de JAXB.

- **API de SOAP con adjuntos para Java**

La API de SOAP con adjuntos para Java (SAAJ) es una API de bajo nivel de la cual dependen JAX-WS y JAXR. SAAJ habilita la producción y el consumo de mensajes que cumplen con la especificación de SOAP1.1 y mensajes SOAP con adjuntos. Muchos desarrolladores no utilizan SAAJ y en lugar de esto utilizan la API de alto nivel JAX-WS.

- **API Java para Servicios Web XML (JAX-WS)**

La especificación JAX-WS proporciona soporte para servicios Web que utiliza la API de JAXB para vincular datos XML en objetos Java. La especificación JAX-WS define APIs clientes para acceder a servicios Web así como técnicas para implementar puntos finales de servicios Web. Los servicios Web para la especificación J2EE describen el despliegue de servicios basados en clientes JAX-WS. La especificación EJB y Servlet también describe aspectos de ese desarrollo. Esto debe posibilitar el despliegue de aplicaciones basadas en JAX-WS utilizando cualquiera de estos modelos de despliegue.

La especificación JAX-WS describe el soporte para los manejadores de mensajes que pueden procesar mensajes de petición y respuesta. En general, estos manejadores de mensajes se ejecutan en el mismo contenedor y con los mismo privilegios y contextos de ejecución que el JAX-WS cliente o en el componente punto final con el que esta asociado. Estos manejadores de mensajes tienen acceso a el mismo espacio de nombres JNDI *java:comp/env* como su componente asociado.

- **Servicio de autenticación y autorización Java**

El servicio de autenticación y autorización Java (JAAS) proporciona los mecanismos necesarios para autenticar y autorizar a un usuario o grupo de usuarios a ejecutar una aplicación Java EE.

JAAS es la implementación en Java del estándar PAM (Pluggable Authentication Module), extendiendo la arquitectura de seguridad de la plataforma Java para soportar autorización basada en usuario.

V.4.4. Servidores de Aplicaciones Java EE

Como consecuencia del éxito del lenguaje de programación Java, el término servidor de aplicaciones usualmente hace referencia a un servidor de aplicaciones Java EE. WebSphere (IBM) y WebLogic (Oracle, antes BEA Systems) están entre los servidores de aplicación Java EE comerciales más conocidos. EAServer (Sybase Inc.) es también conocido por ofrecer soporte a otros lenguajes diferentes a Java, como PowerBuilder. El servidor de aplicaciones JOnAS, desarrollado por el consorcio ObjectWeb, fue el primer servidor de aplicaciones libre en lograr certificación oficial de compatibilidad con J2EE. JBoss es otro servidor de aplicaciones libre y muy popular en la actualidad, así como GlassFish de SUN.

Existen además otras opciones como Tomcat, que en ocasiones se le denomina como servidor de aplicaciones; aunque solamente es un contenedor de Servlets.

Java EE provee estándares que permiten a un servidor de aplicaciones servir como contenedor de los componentes que conforman dichas aplicaciones. Estos componentes, escritos en lenguaje Java, usualmente se conocen como Servlets, Java Server Pages (JSPs) y Enterprise JavaBeans (EJBs) y permiten implementar diferentes capas de la aplicación, como la interfaz de usuario, la lógica de negocio, la gestión de sesiones de usuario o el acceso a bases de datos remotas.

El objetivo de un servidor de aplicaciones es hacer funcionar aplicaciones distribuidas implementadas a base de componentes Java (Servlets, JSP, EJB), haciéndolas accesibles a los clientes Web (navegadores) y a las aplicaciones corporativas escritas en Java.

El servidor debe encargarse de la creación y de la carga de memoria de las instancias de los componentes, así como de la gestión de una cola de espera para satisfacer las peticiones de los clientes.

Además, para satisfacer las exigencias de las aplicaciones corporativas, el servidor de aplicaciones deber ser potente y fiable. Es por tanto capaz de gestionar la disponibilidad (escalabilidad) de las aplicaciones (balanceo de carga, tolerancia a fallos) utilizando los denominadas cluster de servidores.

También se ocupa de la implementación de distintos servicios útiles para los componentes y para el funcionamiento de las aplicaciones:

- **Servicio de Nombres:**

Se trata del servicio más importante para un servidor de aplicación. En efecto, el servicio de nombres permite almacenar de forma jerárquica referencias a recursos de objetos (EJBs, DataSources JDBC, JMS, JTA, información de la configuración), de forma que los clientes pueden realizar en el índice búsqueda sobre los recursos, para trabajar con los correspondientes objetos remotos. El servicio de nombre implementa la interfaz SPI (Service Provider Interface) y se basa normalmente en el protocolo LDAP (Lightweight Directory Access

Protocol), que proporciona fundamentalmente mecanismos de búsqueda, de localización y de enlace.

- **Servicio de Gestión de Transacciones:**

El gestor de transacciones es una implementación de la API JTS. Permite gestionar transacciones locales, pero sobre todo, se encarga de la gestión de las transacciones distribuidas, indispensables para las aplicaciones JEE. La misma transacción puede compartirse por varios componentes y por varias fuentes de datos. El servicio de control de transacciones vela para que la transacción respete las propiedades ACID (Atomic, Consistent, Isolated, Durable) y procede a validar o a denegar la transacción basándose en el protocolo de validación en dos fases 2PC (Two Phase Commit).

- **Servicio de gestión de la disponibilidad de las aplicaciones (Balanceo de carga y tolerancia a fallos)**

Por balanceo de carga (load balancing) se entiende la capacidad de repartir el procesamiento entre distintos servidores. Las peticiones de los clientes se redirigen a la máquina que más desocupada se encuentre en ese momento. Mejora de rendimiento de la aplicación. No es tan sencillo como añadir una nueva máquina y ya está. Además de la escalabilidad, se consigue una mayor tolerancia a fallos.

El objetivo es que el servidor de aplicaciones sea capaz de gestionar un elevado número de accesos y respuestas simultáneas que, además, pueden invocar al mismo recurso. Por tanto es necesario controlar que el recurso sea capaz de satisfacer las peticiones de los clientes en un tiempo mínimo.

El Servidor de Aplicaciones es la aplicación que implementa las APIs y especificaciones Java EE. Ofrecen soporte y servicios a los componentes de la aplicación Java EE. Provee la infraestructura necesaria para las aplicaciones Web empresariales. De esta forma los programadores van a poder dedicarse casi en exclusiva a implementar la lógica del dominio, ya que servicios de uso común, como transacciones, seguridad, persistencia, etc. ya son proporcionados por el servidor Web.

Se ha convertido en una pieza de software clave para cualquier empresa dedicada al comercio electrónico. Es una capa intermedia (middleware) que se sitúa entre el servidor Web y las aplicaciones y bases de datos subyacentes.

Además se ocupa de la gestión de la sesión. Como sabemos, HTTP es un protocolo sin sesión. No permite mantener una conexión abierta entre el cliente y el servidor más allá de lo que dura la transferencia del documento en cuestión. En cualquier aplicación de comercio electrónico, es necesario poder identificar al usuario a través de su navegación por el sitio Web. Lo lógico es que sea el servidor de aplicaciones quien se encargue de gestionar la sesión. Además, debería ser más eficiente que si lo programamos nosotros mismos.

La portabilidad de Java también ha permitido que los servidores de aplicación Java EE se encuentren disponibles sobre una gran variedad de plataformas, como Unix, Microsoft Windows y GNU/Linux.

Los servidores de aplicaciones Java se soportan en las APIs de la plataforma J2EE. Por ello, sea cual sea la herramienta utilizada para desarrollar los componentes (Servlet, JSP, EJB,...) estos podrán utilizarse casi al 100% con independencia del servidor de aplicaciones en que se desarrolle la aplicación.

Capítulo VI

Implementación y Pruebas del Sistema

VI.1. Introducción

Los transformadores de potencia son una de las partes más importantes dentro del proceso de generación y distribución de la energía. Dada la naturaleza de este tipo de instalaciones, la monitorización de los transformadores de potencia se ha convertido en una tarea crítica dentro del proceso de mantenimiento de este tipo de equipos. Durante los primeros capítulos de esta documentación se ha llevado a cabo un estudio a cerca del comportamiento de los transformadores de alta potencia, y cómo éstos ofrecen cierta información durante su vida útil que nos permite estudiar el comportamiento de los mismos y evitar fallos cuyas consecuencias puedan ser catastróficas.

Por otro lado, se han estudiado algunas tecnologías que nos permiten desarrollar Sistemas que nos faciliten la predicción de posibles fallos en los transformadores de potencia de una manera sencilla.

VI.2. Funcionalidades

El Sistema de Monitorización Continua de Descargas Parciales para Transformadores de Alta Potencia debe proporcionar una serie de funcionalidades que permitan el estudio del comportamiento de los Transformadores de Potencia en cualquiera de los contextos que se pueden presentar:

1. Capacidad para capturar las emisiones acústicas derivadas de las descargas parciales producidas en los Transformadores de Potencia de Alta Potencia.

2. Tiempos de rearme, o el tiempo en el que el digitalizador no está listo para un disparo, extremadamente bajos, permitiendo la captura de datos si los disparos ocurren cada microsegundo como si ocurren con varios días de separación.
3. Ofrecer un interfaz amigable de configuración de cada uno de los parámetros de la adquisición.
4. Permitir dos variedades de adquisición: **Simple** y **Continua**.
5. En la **adquisición simple**, la adquisición obtenida debe representarse gráficamente permitiendo al usuario la posibilidad de almacenar la captura.
6. En la **adquisición continua**, tras una adquisición, ésta debe almacenarse en una base de datos y se procederá a realizar una nueva adquisición y así sucesivamente.
7. Debe proporcionar la capacidad para representar gráficamente cualquier adquisición almacenada en la base de datos.
8. Debe proporcionar la capacidad de exportar una adquisición a ficheros compatibles con **MatLab** o **LabView**.
9. Permitir la **interacción remota**, por ejemplo desde un centro de control o un laboratorio de investigación.

VI.3. Casos de Uso

Partiendo de las funcionalidades especificadas en el apartado anterior, se han establecido una serie de casos de uso que se describen a continuación.

La especificación de los casos de uso del sistema nos proporcionará los escenarios en los cuales los usuarios finales interactuarán con el sistema. Se utilizará el diagrama de casos de uso junto con la descripción de cada uno de ellos.

VI.3.1. Diagramas

En los siguientes diagramas se representan las funcionalidades y comportamientos del sistema en su iteración tanto con los usuarios con perfil de operador y visitante, como con los usuarios con perfil de administrador. Por lo tanto el sistema contará con tres actores como entidades externas, que serán:

- Usuario Administrador (**Figura VI.1**)
- Usuario Operador (**Figura VI.2**)
- Usuario Visitante (**Figura VI.3**)

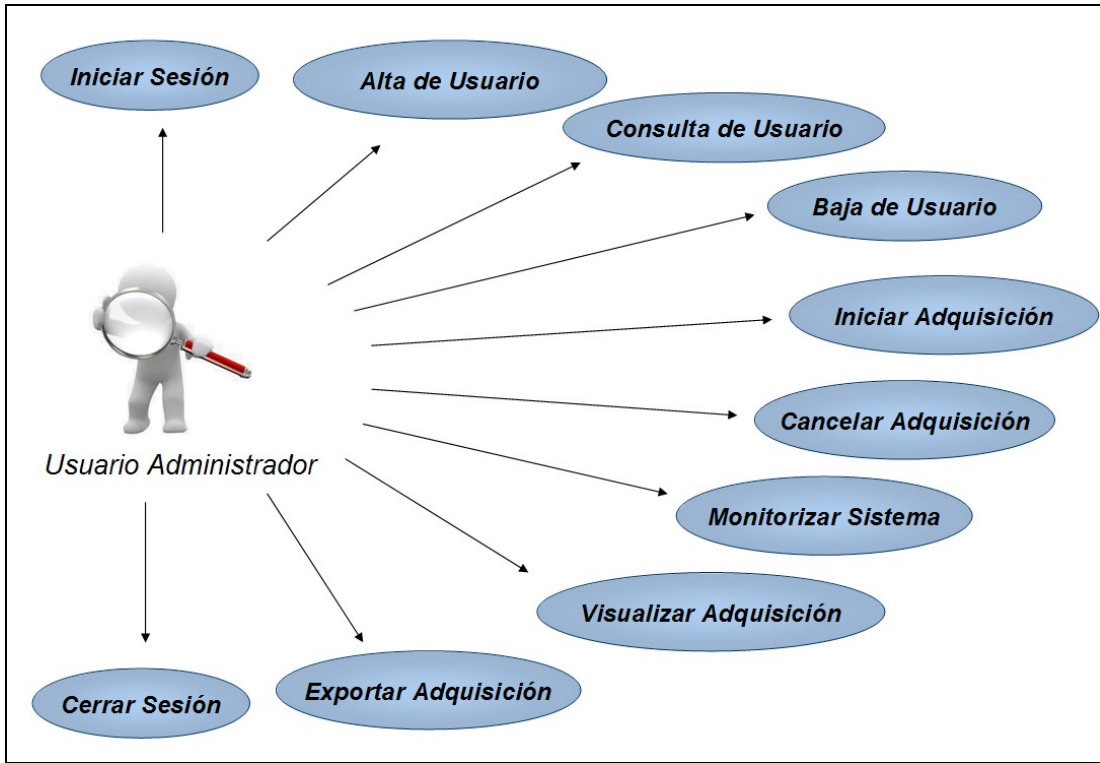


Figura VI.1 Diagrama de Casos de Uso del Usuario Administrador

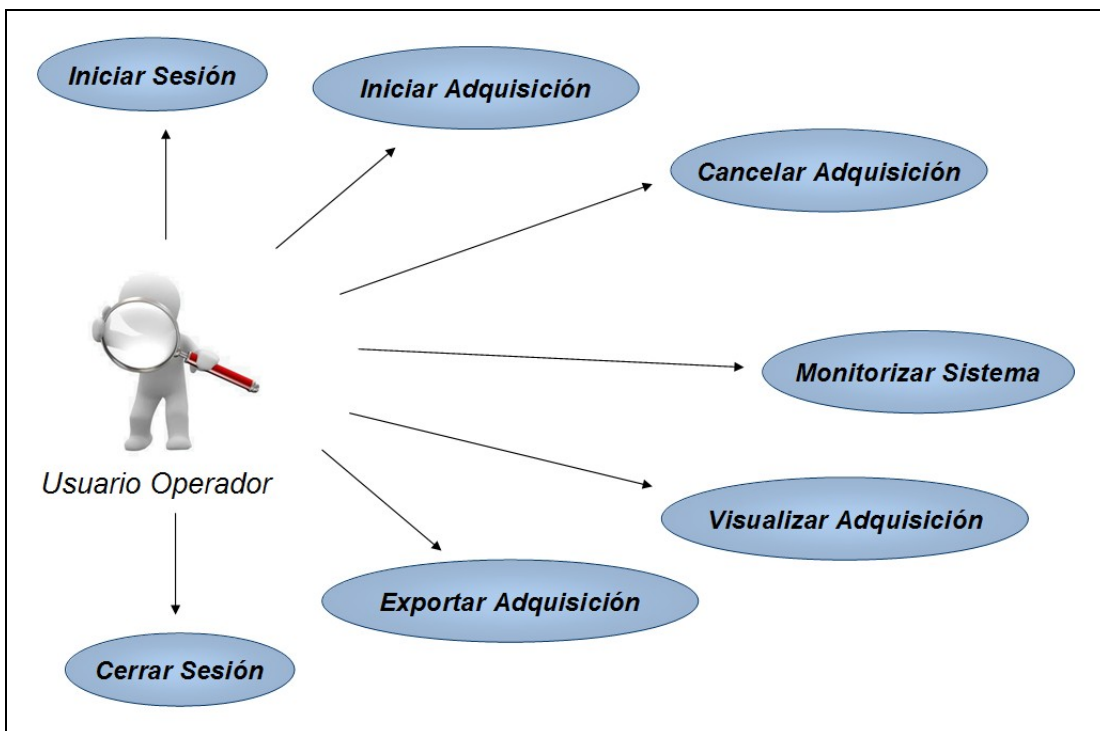


Figura VI.2 Diagrama de Casos de Uso del Usuario Operador

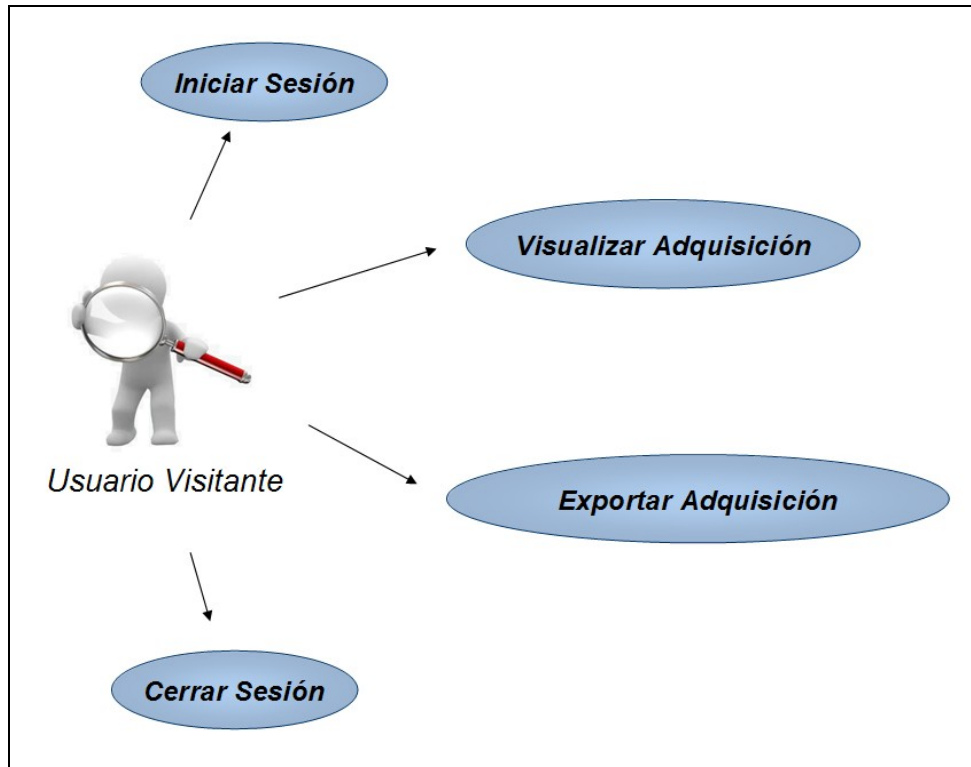


Figura VI.3 Diagrama de Casos de Uso del Usuario Visitante

VI.3.2. Descripciones

Para describir adecuadamente los casos de uso del sistema, se utilizarán las siguientes tablas:

Caso de Uso	Iniciar Sesión
Actores	Administrador, Operador y Visitante
Descripción	Auntenticación y Autorización del Usuario en el Sistema
Especificación	<ol style="list-style-type: none"> 1. El usuario accede al formulario de acceso al Sistema 2. El usuario introduce en el sistema su identificador de usuario y su clave 3. El sistema comprueba las credenciales del usuario 4. Se muestra el escritorio de la aplicación en función del perfil del usuario

Tabla VI.1 Descripción Caso de Uso: Iniciar Sesión

Caso de Uso	Alta de Usuario
Actores	Administrador
Descripción	Alta de Usuario en el Sistema
Especificación	<ol style="list-style-type: none"> 1. El usuario acceder al formulario de Alta de Usuario. 2. El usuario introduce en formulario de Alta de Usuario el identificador del nuevo usuario así como su clave y su perfil. 3. El sistema comprueba que el usuario que está realizando la operación tiene permisos para hacerlo. 4. El sistema comprueba que los datos del nuevo usuario son correctos. 5. Se crea el usuario en el sistema

Tabla VI.2 Descripción Caso de Uso: Alta de Usuario

Caso de Uso	Consulta de Usuario
Actores	Administrador
Descripción	Consulta y Modificación de los Datos del usuario
Especificación	<ol style="list-style-type: none"> 1. El usuario accede a la lista de usuarios del sistema. 2. El usuario selecciona el usuario cuyos datos desea visualizar 3. El sistema comprueba que el usuario que está realizando la operación tiene permisos para hacerlo. 4. Se visualizan los datos del usuario en el sistema

Tabla VI.3 Descripción Caso de Uso: Consulta de Usuario

Caso de Uso	Baja de Usuario
Actores	Administrador
Descripción	Baja del Usuario en el Sistema
Especificación	<ol style="list-style-type: none"> 1. El usuario accede a la lista de usuarios del sistema. 2. El usuario selecciona el usuario que desea eliminar del sistema. 3. El sistema comprueba que el usuario que está realizando la operación tiene permisos para hacerlo. 4. Se usuario es eliminado del sistema.

Tabla VI.4 Descripción Caso de Uso: Baja de Usuario

Caso de Uso	Iniciar Adquisición
Actores	Administrador y Operador
Descripción	Configurar tarjeta de adquisición e iniciar la adquisición
Especificación	<ol style="list-style-type: none"> 1. El usuario accede al formulario de configuración de la adquisición. 2. El usuario modifica cada uno de los parámetros de configuración. 3. El usuario inicia la adquisición. 4. El sistema comprueba que el usuario que está realizando la operación tiene permisos para hacerlo. 5. El sistema comprueba que no existe ninguna adquisición en curso. 6. La adquisición se inicia.

Tabla VI.5 Descripción Caso de Uso: Iniciar Adquisición

Caso de Uso	Cancelar Adquisición
Actores	Administrador y Operador
Descripción	Cancelar Adquisición en curso
Especificación	<ol style="list-style-type: none"> 1. El usuario accede al monitor del sistema. 2. El usuario selecciona la adquisición que desea cancelar. 3. El sistema comprueba que el usuario que está realizando la operación tiene permisos para hacerlo. 4. La adquisición se cancela.

Tabla VI.6 Descripción Caso de Uso: Cancelar Adquisición

Caso de Uso	Monitorizar el Sistema
Actores	Administrador y Operador
Descripción	Monitorizar el Estado de las Adquisiciones en el Sistema
Especificación	<ol style="list-style-type: none"> 1. El usuario accede al monitor del sistema. 2. El sistema comprueba que el usuario que está realizando la operación tiene permisos para hacerlo. 3. Se muestra el estado de las adquisiciones en el sistema.

Tabla VI.7 Descripción Caso de Uso: Monitorizar el Sistema

Caso de Uso	Visualizar Adquisición
Actores	Administrador, Operador y Visitante
Descripción	Visualizar las Adquisiciones realizadas por la tarjeta de adquisición
Especificación	<ol style="list-style-type: none"> 1. El usuario accede a la base de datos de adquisiciones. 2. El usuario selecciona la adquisición que desea visualizar. 3. El sistema comprueba que el usuario que está realizando la operación tiene permisos para hacerlo. 4. Se muestran la adquisición seleccionada.

Tabla VI.8 Descripción Caso de Uso: Visualizar Adquisición

Caso de Uso	Exportar Adquisición
Actores	Administrador, Operador y Visitante
Descripción	Exportar la adquisición para trabajar con ella con otras aplicaciones
Especificación	<ol style="list-style-type: none"> 1. El usuario accede a la base de datos de adquisiciones. 2. El usuario selecciona la adquisición que desea exportar. 3. El usuario seleccionar el formato en el que desea exportar la adquisición (MatLab / LabView). 4. El sistema comprueba que el usuario que está realizando la operación tiene permisos para hacerlo. 5. La adquisición se exporta en el formato seleccionado.

Tabla VI.9 Descripción Caso de Uso: Exportar Adquisición

Caso de Uso	Cerrar Sesión
Actores	Administrador, Operador y Visitante
Descripción	Cerrar la Sesión del Usuario en el Sistema
Especificación	<ol style="list-style-type: none"> 1. El usuario cierra la sesión en el sistema. 2. La sesión se cierra y se muestra el formulario de acceso al sistema

Tabla VI.10 Descripción Caso de Uso: Cerrar Sesión

VI.4. Arquitectura

El Sistema de Monitorización Continua de Descargas Parciales para Transformadores de Alta Potencia que estamos planteando en este proyecto tiene una característica que influye por

completo en la arquitectura del sistema. Así, al permitir la interacción remota debemos centrarnos en arquitecturas cliente-servidor o arquitecturas multinivel.

Además de necesitar una Base de Datos donde almacenar los datos recuperados por el sistema, este tipo de arquitecturas nos obliga a solucionar los problemas derivados de la seguridad en aplicaciones distribuidas y a tener en cuenta otras limitaciones derivadas de las propias redes como son el ancho de banda, la latencia, ...

Dado que el protocolo HTTP, utilizado para el intercambio de información entre Clientes y Servidores Web, es un protocolo sin estado, de algún modo es necesario identificar que una serie de solicitudes de un mismo cliente están relacionadas. Una **sesión** puede definirse como un conjunto de interacciones entre un Cliente y un Servidor Web que tienen lugar durante un período de tiempo. Los contenedores son responsables de proporcionar los mecanismos básicos para crear y mantener sesiones. Así, la especificación de **Servlets** ofrece al desarrollador la posibilidad de trabajar con objetos sesión, permitiendo tener un control de las sesiones de los usuarios en el sistema. Además, la utilización de **filters** nos permite interceptar cualquier petición antes de que ésta llegue al Servlet o a la JSP correspondiente, permitiéndonos centralizar la seguridad de la aplicación en estos **filters**, y dejando a los **Servlets** la lógica propia de la aplicación.

Teniendo en cuenta estos detalles, la arquitectura del sistema queda reflejada en la **Figura VI.4**.

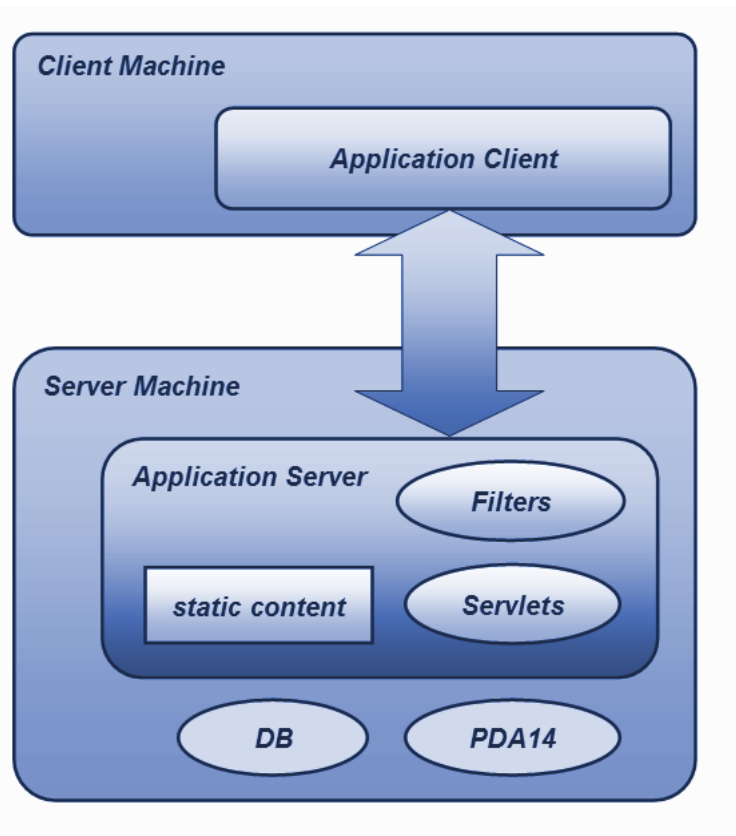


Figura VI.4 Arquitectura del Sistema

Uno de los servidores HTTP más sencillos y que además proporciona un contenedor de Servlets es **Jetty [17]**. Además Jetty es un proyecto de software libre bajo la licencia Apache 2.0 y que de forma muy simple permitirá validar la aplicación.

En este caso, al no disponer de unos requisitos de seguridad bien definidos, se ha planteado un sistema de seguridad sencillo basado en perfiles de usuario, donde los **filters** serán los elementos del sistema encargados de la autenticación y de la autorización.

Del mismo modo, al no tener una previsión del volumen de datos que manejará la aplicación se ha optado por utilizar como base de datos el propio sistema de ficheros del sistema operativo aplicando asimismo uno de los patrones de diseño más utilizados para el acceso a datos: el Patrón DAO [18]. De esta forma, si fuera necesario utilizar algún sistema de almacenamiento más potente, el impacto sobre la aplicación sería mínimo.

Todas estas funcionalidades son proporcionadas por la tecnología Java. Sin embargo, el fabricante de la PDA14 proporciona librerías C para interactuar con la tarjeta y no una librería Java. Por este motivo, es necesario implementar una librería Java que permite interactuar con la tarjeta de adquisición.

VI.5. Software de Adquisición

La Tarjeta de Adquisición PDA14 de Signatec es una opción muy interesante para la monitorización de señales de la naturaleza de las descargas parciales producidas por un transformador de alta potencia. Junto con ella, el fabricante proporciona una librería de funciones implementadas en C con la que interactuar con la Tarjeta de Adquisición PDA14 y así configurar los parámetros de adquisición de señales.

Ya hemos visto que las ventajas de emplear Java para desarrollar una aplicación que permita interactuar con la Tarjeta de Adquisición PDA14 son muchas, sobre todo si queremos ofrecer acceso remoto. Por eso, nuestro primer objetivo es ofrecer una librería de funciones con la que interactuar con la Tarjeta de Adquisición PDA14 desde cualquier aplicación desarrollada en Java.

VI.5.1. JNI

JNI [19] es un mecanismo que permite ejecutar código nativo desde Java y viceversa. El código nativo son funciones escritas en un lenguaje de programación como C o C++ para un sistema operativo donde se está ejecutando la máquina virtual de Java.

Estas librerías escritas en código nativo pueden ser de dos tipos, **librerías de enlace estático** y **librería de enlace dinámico**.

Las **librerías de enlace estático** son ficheros destinados a almacenar funciones, clases y variables globales y tradicionalmente se crean a partir de varios ficheros de código objeto .o

(UNIX) o *.obj* (Windows). En UNIX las librerías de enlace estático suelen tener la extensión *.a* y en Windows la extensión *.lib*. Mac OS X usa la extensión *.a* de los sistemas UNIX tradicionales. Las funciones de la librería se incluyen dentro del ejecutable durante la fase de enlazado, con lo que una vez generado el ejecutable ya no es necesario disponer de las librerías de enlace estático.

Las **librerías de enlace dinámico** son ficheros cuyas funciones no se incrustan en el ejecutable durante la fase de enlazado, sino que en tiempo de ejecución el programa busca el fichero, carga su contenido en memoria y enlaza su contenido según va siendo necesario, es decir según vamos llamando a las funciones. Esto tiene la ventaja de que varios programas pueden compartir las mismas librerías, lo cual reduce el consumo de disco duro, especialmente con las llamadas al Sistema Operativo (APIs) que suelen usarse por muchas aplicaciones a la vez. Su extensión también varía dependiendo del Sistema Operativo en el que estemos trabajando.

JNI se usa para escribir métodos nativos que permitan solventar situaciones en las que una aplicación no puede ser enteramente escrita en Java, como por ejemplo en el caso de que la biblioteca estándar de clases no proporcione soporte para funcionalidades dependientes de la plataforma

También se usa para modificar programas existentes escritos en algún otro lenguaje, permitiéndoles ser accesibles desde aplicaciones Java. Muchas de las clases de la API estándar de Java dependen del JNI para proporcionar funcionalidad al desarrollador y al usuario, por ejemplo las funcionalidades de sonido o lectura/escritura de ficheros. El desarrollador debe asegurarse que la API estándar de Java no proporciona una determinada funcionalidad antes de recurrir al JNI, ya que la primera ofrece una implementación segura e independiente de la plataforma.

JNI permite a un método nativo utilizar los objetos Java de la misma forma en que el propio código de Java lo hace. Un método nativo puede crear objetos Java; y examinarlos y utilizarlos para que lleven a cabo su función. Un método nativo puede asimismo examinar y utilizar objetos que han sido creados por código de aplicación escrito en Java.

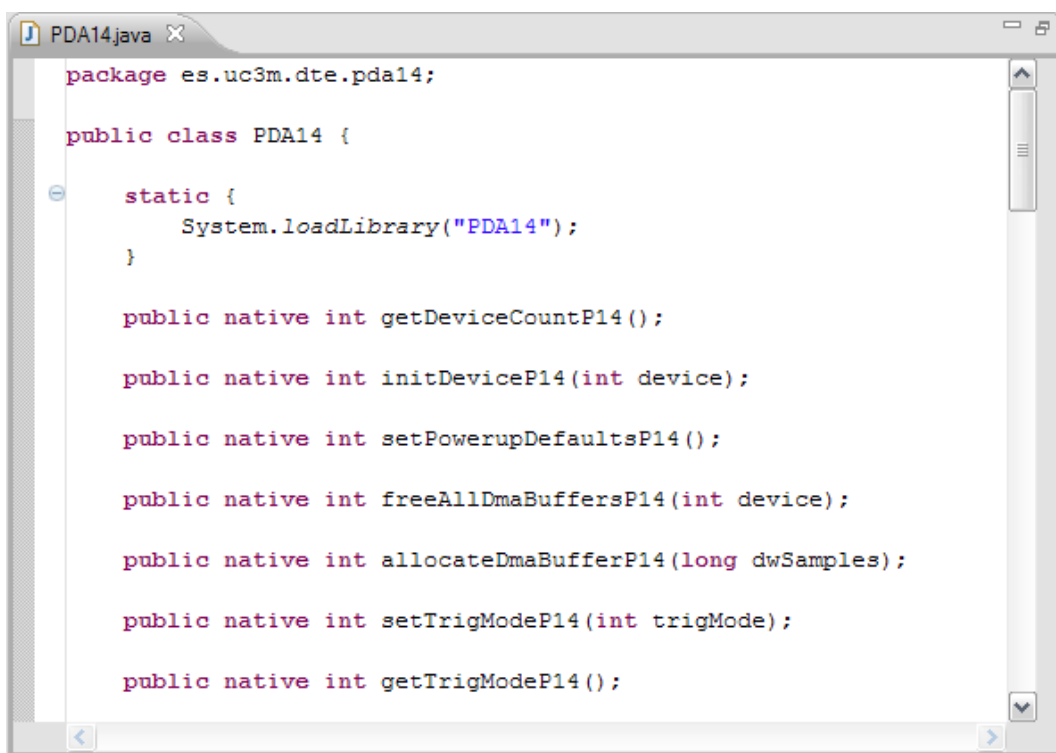
A menudo se denomina a JNI no es otra cosa que una válvula de escape para desarrolladores dado que les permite añadir funcionalidades a sus aplicaciones que el API de Java no puede proporcionar.

Dado que puede usarse para interactuar con código escrito en otros lenguajes como C++, también se usa para operaciones y cálculos de alta complejidad temporal, porque el código nativo es por lo general más rápido que el que se ejecuta en una máquina virtual.

VI.5.2. Implementación con JNI

Junto con la Tarjeta de Adquisición PDA14, el fabricante proporciona una librería de enlace estático *PDA14Win.lib* y su correspondiente fichero de encabezado *PDA14Win.h* con los que podemos desarrollar cualquier aplicación en C o C++.

Por lo tanto, es necesario crear una librería de enlace dinámico con la que acceder desde la máquina virtual de Java. Para ello se creará una Clase Java denominada PDA14 que nos permita interactuar con la Tarjeta de Adquisición PDA14 a través de la librería de enlace estático proporcionada por el fabricante. Cada uno de sus métodos serán cada una de las funciones que ofrece la librería de enlace estático proporcionada por el fabricante y que podemos encontrar tanto en el fichero de encabezado como en el manual de usuario. Como se muestra en la **Figura VI.5**, dichos métodos serán declarados como nativos, mediante el modificador *native*, indicando que están sin implementar ya que su implementación se encuentra en la librería nativa.



```

package es.uc3m.dte.pda14;

public class PDA14 {

    static {
        System.loadLibrary("PDA14");
    }

    public native int getDeviceCountP14();

    public native int initDeviceP14(int device);

    public native int setPowerupDefaultsP14();

    public native int freeAllDmaBuffersP14(int device);

    public native int allocateDmaBufferP14(long dwSamples);

    public native int setTrigModeP14(int trigMode);

    public native int getTrigModeP14();
}

```

Figura VI.5 Extracto de la Clase PDA14

En la implementación de nuestra clase PDA14, conviene destacar la llamada al método

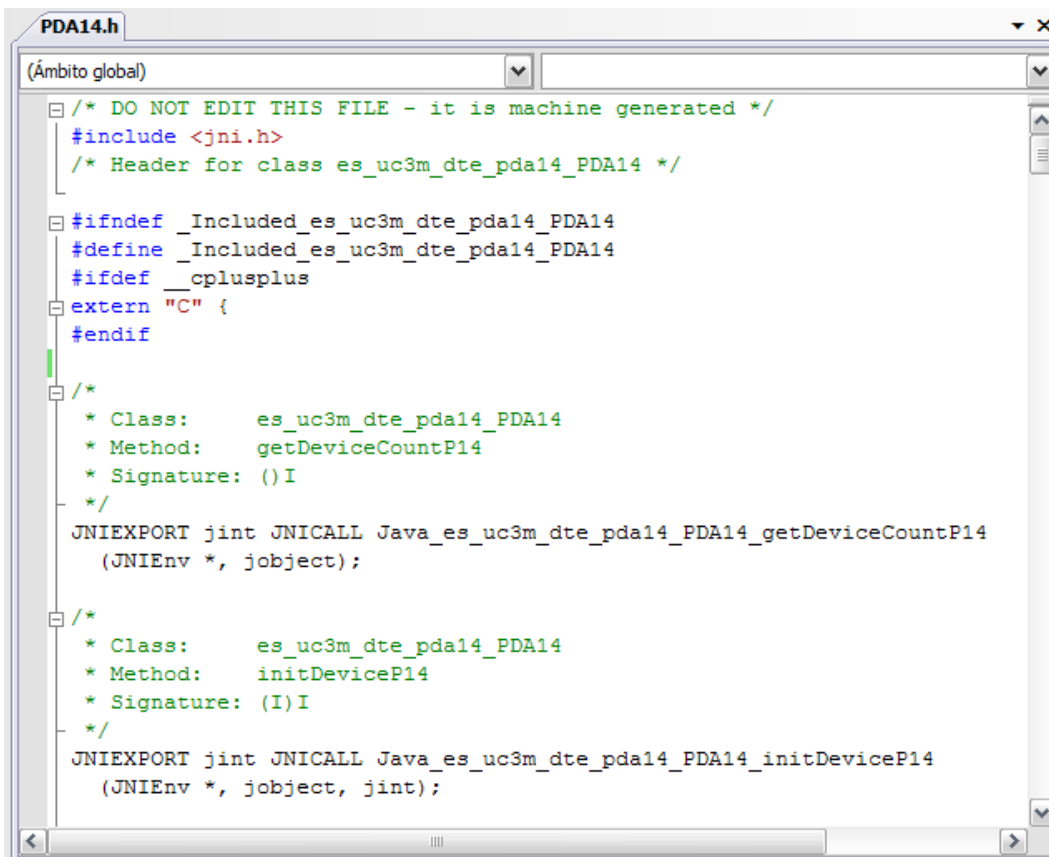
```
void <System> loadLibrary(String libraryName)
```

que recibe el nombre de una librería de enlace dinámico y la carga. Esta librería debe cargarse antes de llamar a cualquier método nativo.

Una vez creada la clase, el entorno de desarrollo de Java proporciona una herramienta (**javah**) con la que se crea el fichero de encabezado con el prototipo de los métodos nativos a implementar a partir del fichero compilado.

```
javah es.uc3m.dte.pda14.PDA14
```

El fichero de encabezado generado se muestra en la **Figura VI.6**:



```
PDA14.h
(Ámbito global)
/* DO NOT EDIT THIS FILE - it is machine generated */
#include <jni.h>
/* Header for class es_uc3m_dte_pda14_PDA14 */

#ifndef _Included_es_uc3m_dte_pda14_PDA14
#define _Included_es_uc3m_dte_pda14_PDA14
#ifdef __cplusplus
extern "C" {
#endif

/*
 * Class:      es_uc3m_dte_pda14_PDA14
 * Method:    getDeviceCountP14
 * Signature: ()I
 */
JNIEXPORT jint JNICALL Java_es_uc3m_dte_pda14_PDA14_getDeviceCountP14
    (JNIEnv *, jobject);

/*
 * Class:      es_uc3m_dte_pda14_PDA14
 * Method:    initDeviceP14
 * Signature: (I)I
 */
JNIEXPORT jint JNICALL Java_es_uc3m_dte_pda14_PDA14_initDeviceP14
    (JNIEnv *, jobject, jint);
```

Figura VI.6 Extracto del fichero de encabezado para la Clase PDA14

Crearemos un fichero **PDA14.c** como el que se muestra en la **Figura VI.7** con la implementación de cada uno de los métodos nativos definidos en el fichero de encabezado.

```

PDA14.c
(Ámbito global)
#include "PDA14win.h"
#include "PDA14.h"
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <jni.h>

P14ID id;
p14_dma_buf_t pDmaBuf;
long totalSamples;

JNIEXPORT jint JNICALL Java_es_uc3m_dte_pda14_PDA14_getDeviceCountP14 (JNIEnv *env, jobject obj)
{
    return GetDeviceCountP14();
}

JNIEXPORT jint JNICALL Java_es_uc3m_dte_pda14_PDA14_initDeviceP14 (JNIEnv *env, jobject obj, jint device)
{
    return InitDeviceP14(&id, device);
}

JNIEXPORT jint JNICALL Java_es_uc3m_dte_pda14_PDA14_setPowerupDefaultsP14 (JNIEnv *env, jobject obj)
{
    return SetPowerupDefaultsP14(&id);
}
    
```

Figura VI.7 Extracto del fichero PDA14.c

El último paso consiste en compilar este fichero para crear la librería de enlace dinámico, en nuestro caso un fichero llamado **PDA14.dll** incluyendo en la compilación los directorios:

- **JAVA_HOME/include**
- **JAVA_HOME/include/win32**

Al ejecutar cualquier aplicación Java que utilice la nueva librería, se producirá una excepción **UnsatisfiedLinkError** si la función **System.loadLibrary** no encuentra la librería de enlace dinámico que hemos creado. Una de las opciones es fijar las variables de entorno del sistema operativo dónde están las librerías de enlace dinámico. La otra opción es indicar la localización de la librería de enlace dinámico en el momento de la ejecución con la propiedad `java.library.path`.

VI.5.3. API Java

A continuación se puede encontrar la API Java de acceso a la Tarjeta de Adquisición PDA14 desarrollado para el Sistema de Monitorización Continua de Descargas Parciales para Transformadores de Alta Potencia y que puede extrapolarse a cualquier aplicación que quiera utilizar la PDA14:

```

es.uc3m.dte.pda14
Class PDA14

java.lang.Object
    → es.uc3m.dte.pda14.PDA14

public class PDA14 extends Object
    
```

Constructor Summary	
PDA14()	Initializes a newly created PDA14 object so that it represents an PDA14 dual channel waveform capture board.
Method Summary	
int	getDeviceCountP14();
int	initDeviceP14(int device);
int	setPowerupDefaultsP14();
int	freeAllDmaBuffersP14(int device);
int	allocateDmaBufferP14(long dwSamples);
int	setTrigModeP14(int trigMode);
int	getTrigModeP14();
int	setTrigSourceP14(int sourceMode);
int	getTrigSourceP14();
int	setTrigSlopeP14(int trigSlope);
int	getTrigSlopeP14();
int	setVoltRangeCh1P14(int voltRangeCh1);
int	getVoltRangeCh1P14();
int	setVoltRangeCh2P14(int voltRangeCh2);
int	getVoltRangeCh2P14();
int	setChannelsP14(int channels);
int	getChannelsP14();
int	setClockP14(int clock);
int	getClockP14();
int	setClockDividerP14(int clockDivider);
int	getClockDividerP14();
int	setTrigLevelP14(int trigLevel);
int	getTrigLevelP14();
int	setOffsetCh1P14(int offsetCh1);
int	getOffsetCh1P14();

<code>int</code>	<code>setOffsetCh2P14(int offsetCh2);</code>
<code>int</code>	<code>getOffsetCh2P14();</code>
<code>int</code>	<code>setTotalSamplesP14(int dwSamples);</code>
<code>int</code>	<code>getTotalSamplesP14();</code>
<code>int</code>	<code>setStartSampleP14(int startSample);</code>
<code>int</code>	<code>getStartSampleP14();</code>
<code>int</code>	<code>setModeP14(int mode);</code>
<code>int</code>	<code>getModeP14();</code>
<code>int</code>	<code>setSegmentSizeP14(int segmentSize);</code>
<code>int</code>	<code>getSegmentDetectP14();</code>
<code>int</code>	<code>getTimeStampDataP14();</code>
<code>int</code>	<code>resetTimeStampDataP14();</code>
<code>int</code>	<code>acquireToPciP14(long dwSamples);</code>
<code>int</code>	<code>acquireToBoardP14();</code>
<code>int</code>	<code>boardToPcP14(long dwSamples);</code>
<code>int</code>	<code>getCancelableFlagP14();</code>
<code>int</code>	<code>setCancelableFlagP14(boolean bool);</code>
<code>int</code>	<code>cancelCurrentPendingOpP14();</code>
<code>int[]</code>	<code>getAcquisitionDataP14();</code>
<code>int</code>	<code>freeDmaBufferP14();</code>
<code>int</code>	<code>closeDeviceP14();</code>
<code>int</code>	<code>cancelCurrentPendingOpP14();</code>
<code>int[]</code>	<code>acquireToBoardWrapperP14(long dwSamples);</code>
<code>int[]</code>	<code>acquireAndTransferWrapperP14(int startSample, long dwSamples);</code>

Tabla VI.11 API de la Librería JAVA para la PDA14

De todas las funciones proporcionadas en la librería, conviene destacar las dos últimas donde se encapsulan operaciones habituales que engloban una serie de funciones como son la reserva de memoria, la adquisición, la recuperación de los datos de la memoria reservada y la liberación de dicha memoria:

acquireToBoardWrapperP14

```
JNIEXPORT jintArray JNICALL
Java_es_uc3m_dte_pda14_PDA14_acquireToBoardWrapperP14
(JNIEnv *env, jobject obj, jlong dwSamples)
{
    p14_dma_buf_t pDmaBuf;
    int index;
    jintArray data;
    jint* dataPointer;
    jsize size;
    AllocateDmaBufferP14(&id, (ULONG) dwSamples, &pDmaBuf);
    AcquireToBoardP14(&id);
    BoardToPcP14 (&id, 0, (ULONG) dwSamples, pDmaBuf,
        INVALID_HANDLE_VALUE);
    data = (*env)->NewIntArray(env, (ULONG) dwSamples);
    dataPointer = (jint*)(*env)->
        GetPrimitiveArrayCritical(env, data, NULL);
    size = (*env)->GetArrayLength(env, data);
    for(index = 0; index < size; index++)
        dataPointer[index] = (unsigned)pDmaBuf[index];
    (*env)->ReleasePrimitiveArrayCritical(env, data, dataPointer,
        0);
    FreeDmaBufferP14(&id, pDmaBuf);
    return data;
}
```

acquireAndTransferWrapperP14

```

JNIEXPORT jintArray JNICALL
Java_es_uc3m_dte_pda14_PDA14_acquireAndTransferWrapperP14
(JNIEnv *env, jobject obj, jint dwStart, jlong dwSamples)
{
    p14_dma_buf_t pDmaBuf;
    int index;
    jintArray data;
    jint* dataPointer;
    jsize size;
    if (AllocateDmaBufferP14(&id, (ULONG) dwSamples, &pDmaBuf) !=
        SIG_SUCCESS) {
        return (void*)0;
    }
    if (AcquireAndTransferP14(&id, (ULONG) dwStart, (ULONG)
        dwSamples, pDmaBuf) != SIG_SUCCESS) {
        FreeDmaBufferP14(&id, pDmaBuf);
        return (void*)0;
    }
    data = (*env)->NewIntArray(env, (ULONG) dwSamples);
    dataPointer = (jint*)(*env)->
        GetPrimitiveArrayCritical(env, data, NULL);
    size = (*env)->GetArrayLength(env, data);
    for (index = 0; index < size; index++) {
        dataPointer[index] = (unsigned)pDmaBuf[index];
    }
    (*env)->ReleasePrimitiveArrayCritical(env, data, dataPointer,
        0);
    FreeDmaBufferP14(&id, pDmaBuf);
    return data;
}

```

La diferencia entre ambas funciones está en el método empleado para la adquisición y posterior transferencia al PC. En el caso de la segunda función, se está haciendo uso de la interrupción que determina que la adquisición se ha completado de forma que mientras se está esperando a que la adquisición se complete, no se está produciendo ningún consumo de CPU.

VI.6. Herramienta de Administración Local

Una vez desarrollada la librería que nos permita interactuar con la Tarjeta de Adquisición PDA14 desde una aplicación Java, utilizando diversas tecnologías y frameworks como **Spring [20]** y **JFreeChart [21]** se ha llevado a cabo la implementación de una Herramienta de Administración Local para la PDA14. Con esta herramienta, una manera sencilla, se puede configurar la PDA14 para llevar a cabo una adquisición. Además, permite trabajar con adquisiciones ya realizadas.

La **Figura VI.8** muestra el interfaz de la herramienta. Se trata una herramienta muy sencilla pero que además ser la primera herramienta desarrollada en Java para Interactuar con la PDA14, destaca por ser la principal herramienta de test utilizada durante el desarrollo del software de adquisición.

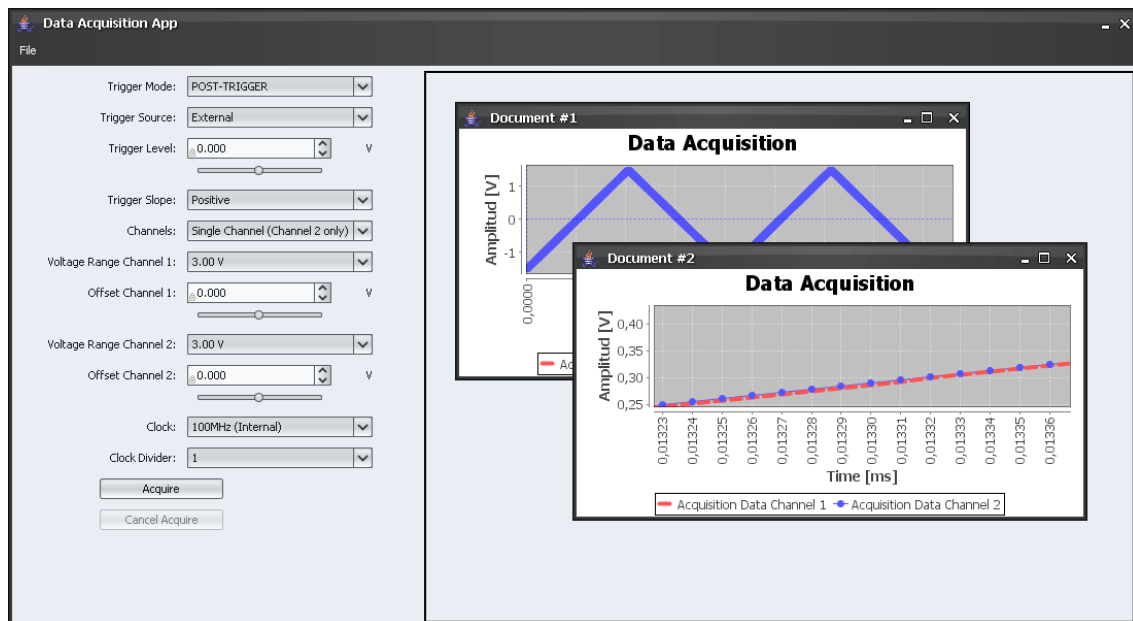


Figura VI.8 Intefaz de la Herramienta de Administración Local

La funcionalidad de esta herramienta es muy similar a las herramientas que proporciona Signatec junto con la Tarjeta de Adquisición. Sin embargo, esta herramienta ofrece una funcionalidad que la diferencia de las proporcionadas por el fabricante. Se trata de la posibilidad de exportación a ficheros de MatLab [22] o LabView [23] de las adquisiciones realizadas por la tarjeta.

En la **Figura VI.9** podemos observar las opciones ofrecidas por la Herramienta de Administración Local desarrollada para la PDA14.

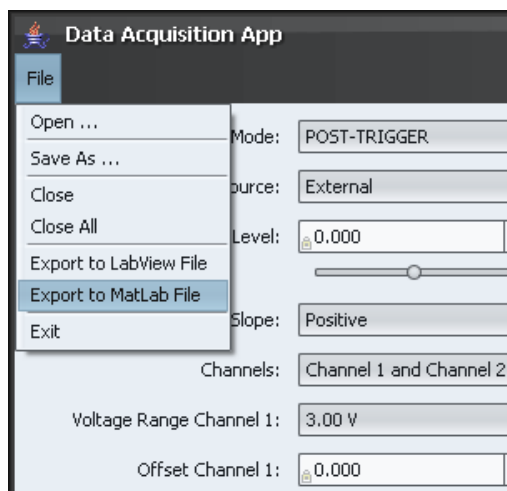


Figura VI.9 Exportación de Adquisiciones desde la Herramienta de Administración Local

Las pruebas realizadas con la ayuda de esta herramienta han sacado a relucir algunas limitaciones que poseen las versiones del firmware y del software de la PDA14 instalado en el laboratorio:

- El modo de disparo pre-trigger no está todavía implementado.
- Las funciones para el tratamiento de las marcas de tiempo en el modo segmentado no están todavía implementadas

Por lo tanto, a la espera de que estas limitaciones se superen según el fabricante vaya actualizando tanto firmware como el software de la tarjeta, se ha tenido que reducir algunas de las capacidades que esperaba ofrecer el Sistema de Monitorización Continua de Descargas Parciales.

VI.7. Interfaz de Acceso Remoto

La plataforma J2EE ofrece gran variedad de tecnologías que juntas permiten soluciones para el desarrollo, despliegue y gestión de aplicaciones distribuidas centradas en servidor. La solución propuesta establece el acceso al sistema de adquisición mediante un servidor de aplicaciones. Una aplicación Web, instalada en el servidor de aplicaciones será el interfaz de acceso con el Sistema de Adquisición. De éste modo, el acceso solo se llevará acabo a través del protocolo HTTP, a través del puerto 80.

En la **Figura VI.10**, se muestra un pantallazo del Interfaz de Acceso Remota del Sistema de Monitorización Continua de Descargas Parciales.

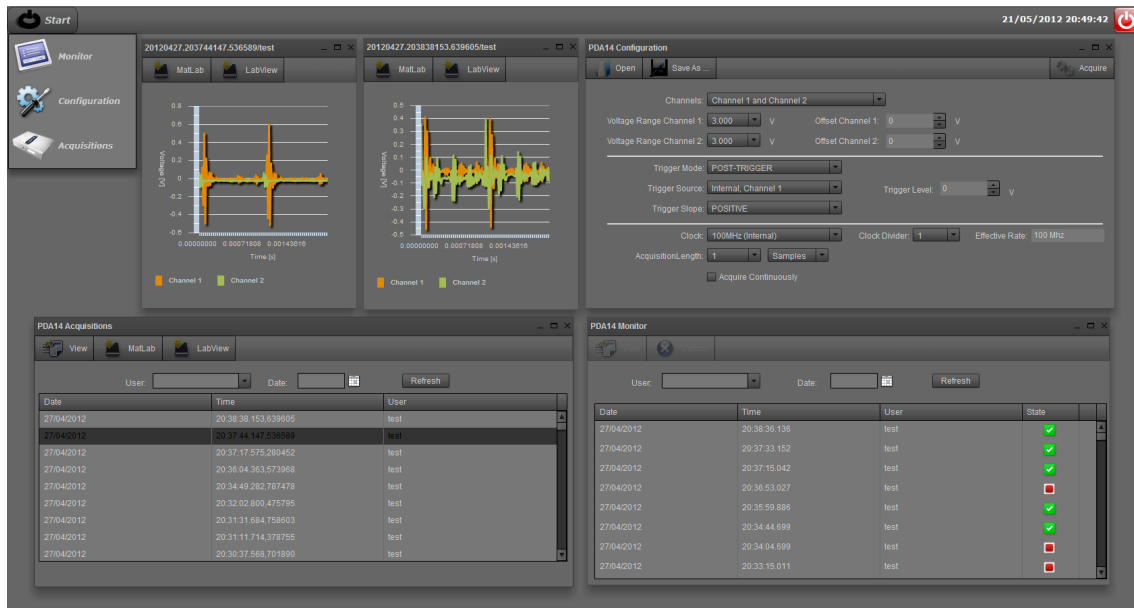


Figura VI.10 Intefaz de Acceso Remoto

Esta interfaz, corresponde al nivel de Presentación dentro de la arquitectura multinivel y ha sido desarrollada mediante la herramienta de desarrollo Adobe Flash Builder 4.5, que facilita la implementación de Aplicaciones Web Enriquecidas.

VI.8. Pruebas y Resultados

Para realizar las pruebas de la aplicación, en uno de los laboratorios del Departamento de Tecnología Electrónica de la Universidad Carlos III de Madrid, se ha instalado en una cuba un sistema capaz de generar emisiones acústicas similares a las generadas por las descargas parciales en transformadores de alta potencia. Como se muestra en la **Figura VI.11**, en las paredes de la cuba se han instalado dos sensores de ultrasonidos R15I-AST. Además, en el interior de la cuba se dispone de otro sensor B&K 8103.¹

¹ Las hojas de características de ambos sensores se pueden encontrar al final de esta memoria

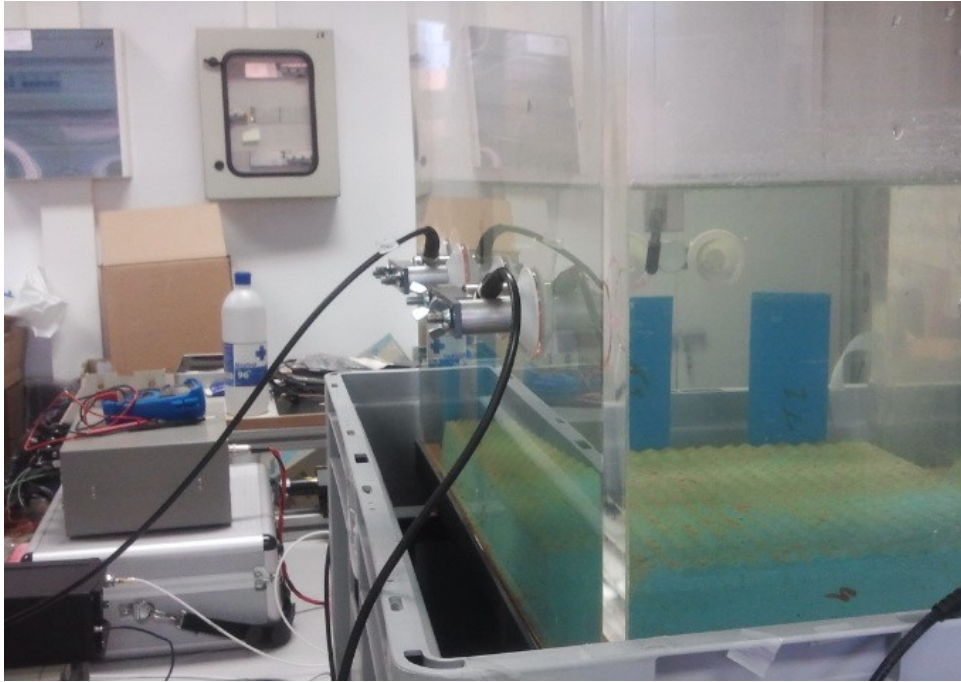


Figura VI.11 Simulación de Emisiones Acústicas

Una vez configurado el sistema para capturar las emisiones acústicas, en la **Figura VI.12** observamos como desde un PC podemos empezar a observar señales.



Figura VI.12 Vista del Sistema desde el puesto de un operador

A continuación analizaremos algunas de las adquisiciones de la PDA14 para mostrar algunas de las capacidades de la aplicación:

❖ Adquisición 1

En esta primera adquisición, se ha hecho uso del modo segmentado. Además, conviene destacar que la señal de disparo se sitúa sobre Canal 1 que corresponde al sensor externo, lo que lógicamente influye en los datos adquiridos.

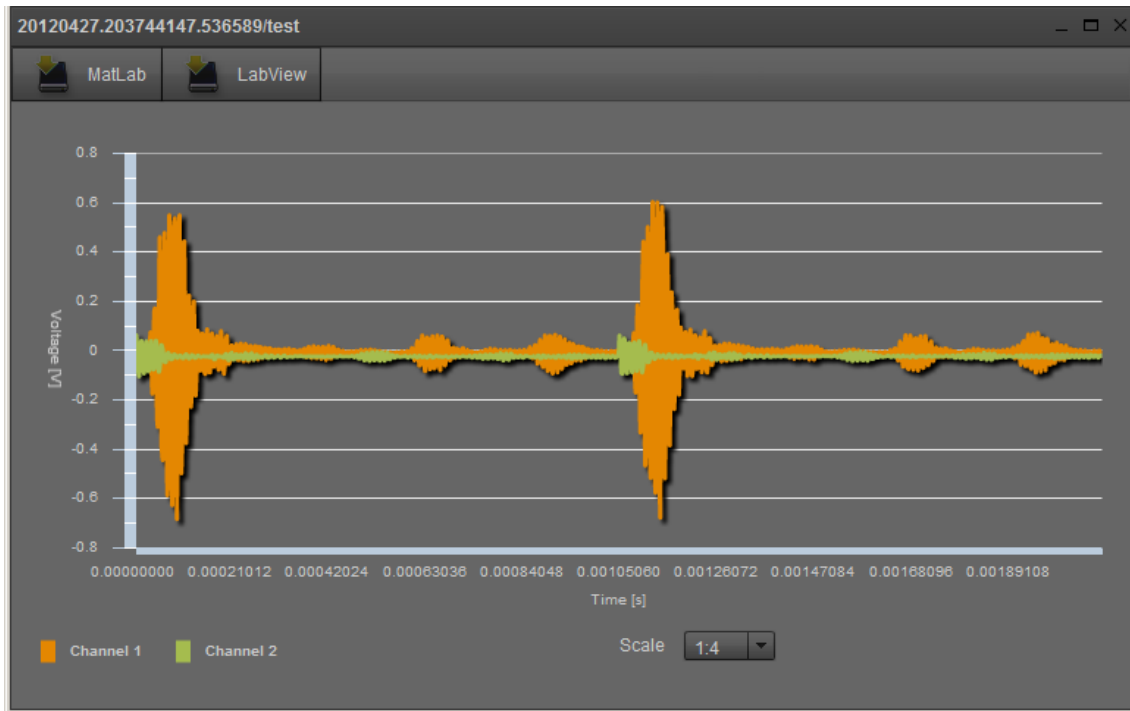


Figura VI.13 Adquisición de Prueba con la Señal de Disparo sobre el Canal 1

En la **Figura VI.13**, se pueden apreciar los dos segmentos capturados por la tarjeta. Asimismo, se observan la llegada de réplicas de menor intensidad. Además, conviene destacar como la intensidad de la señal capturada por el Canal 2 es muy reducida respecto a la capturada por el Canal 1. Esto se debe a que la señal de disparo se sitúa en el Canal 1 y parece que la emisión acústica está llegando antes al sensor conectado al canal 2.

❖ Adquisición 2

En esta nueva adquisición, mostrada en la **Figura VI.14**, la señal de disparo se sitúa sobre Canal 2 que corresponde al sensor interno.

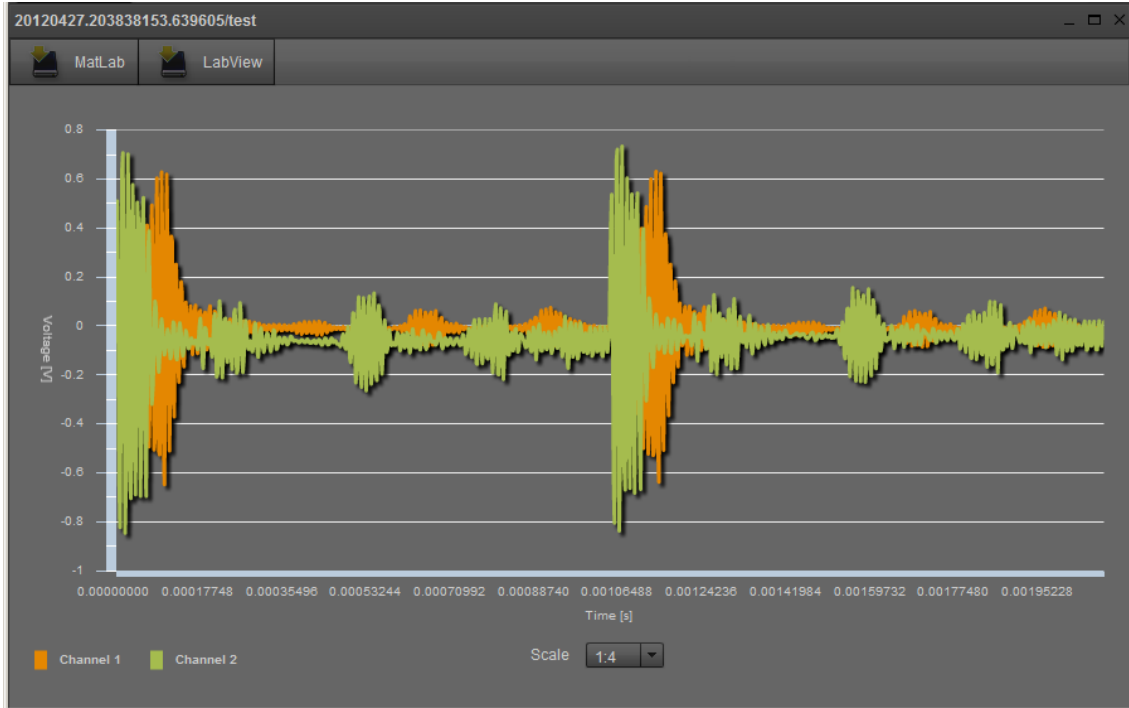


Figura VI.14 Adquisición de Prueba con la Señal de Disparo sobre el Canal 2

Cuando la señal de disparo se sitúa sobre el sensor externo, parte de la señal proporcionada por el sensor interno se pierde, al llegar la emisión acústica, antes al sensor interno que al externo

❖ Adquisición 3

En esta última adquisición, la mostrada en **Figura VI.15**, se ha variado el tamaño de los segmentos a la mitad, lo que provoca que se capture menos cantidad de señal pero mayor número de señales, reduciendo la posibilidad de pérdida de una señal acústica debido al tiempo de rearme de la tarjeta de adquisición.

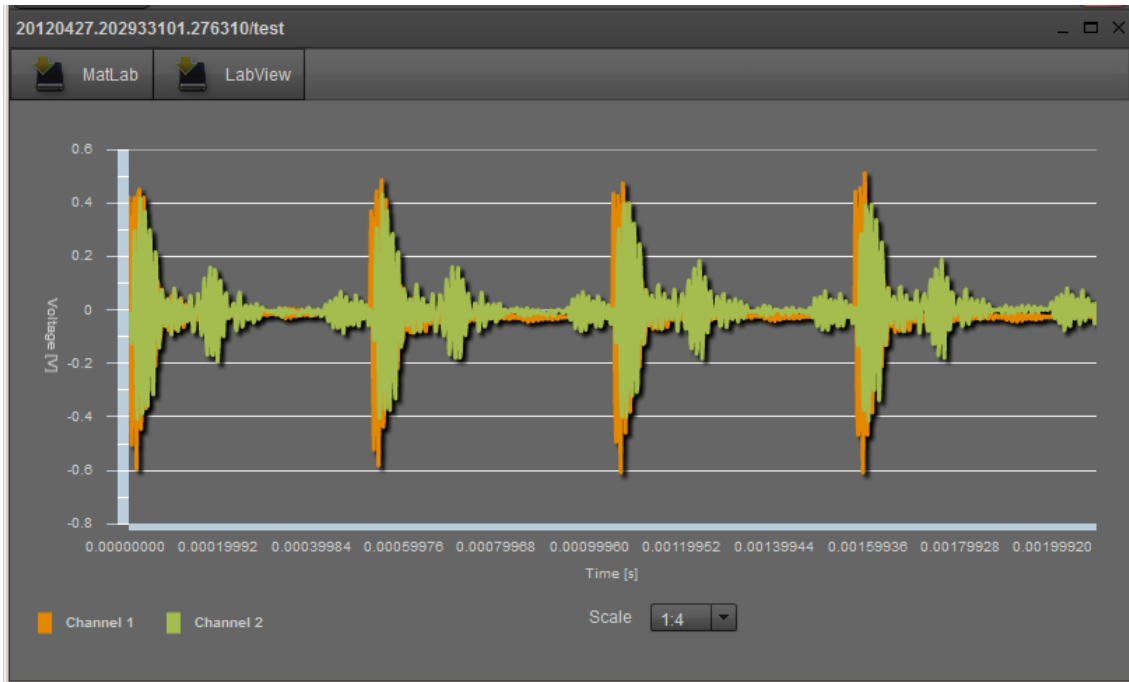


Figura VI.15 Adquisición de Prueba disminuyendo el tamaño de los Segmentos

Capítulo VII

Conclusiones

La medida de descargas parciales ha demostrado ser un buen indicador del estado de degradación de los aislantes eléctricos. Sin embargo, los fenómenos de las descargas parciales no pueden medirse directamente. La energía producida por estos fenómenos produce una serie de transitorios eléctricos, cambios químicos, emisiones electromagnéticas, vibraciones, emisiones acústicas de ultrasonidos, luz y calor que si que pueden medirse. La identificación de descargas parciales basada en la detección de cada una de estas alteraciones tiene sus ventajas e inconvenientes. Destacan las ondas acústicas generadas por las descargas parciales que son inmunes a ruido electromagnético, que es la principal fuente de problemas en cuanto a la detección eléctrica se refiere.

El estudio de las características de las emisiones acústicas derivadas de las descargas parciales producidas en el aislante de los transformadores de potencia ha permitido comprobar que la Tarjeta de Adquisición PDA14 de Signatec posee una serie de características que la convierten en un Sistema de Adquisición idóneo para capturar este tipo de señales. Además, a pesar de tratarse de un sistema de dos canales, por su ancho de banda y profundidad de memoria es posible multiplexar un número elevado de canales de medida acústica fáciles de demultiplexar por software. Estos sistemas de adquisición más complejos pueden proporcionar mayor información de medidas espacialmente distribuidas, así como la localización de la emisión acústica teniendo en cuenta los tiempos de vuelo.

No sólo las características técnicas de la Tarjeta de Adquisición PDA14 la convierten en una candidata adecuada para la detección de las emisiones acústicas procedentes de descargas parciales. Además, el software proporcionado por el fabricante permite adaptar la tarjeta de adquisición a las nuevas necesidades ofreciendo la posibilidad de construir un sistema de monitorización continua. Al ampliar el software proporcionado por el fabricante, permitiéndole ser accesible desde aplicaciones Java, se ha posibilitado la implementación de

un sistema distribuido, accesible remotamente, satisfaciendo la necesidad de sistemas que permitan monitorizar de forma continua.

La solución propuesta, el Sistema de Monitorización Continua de Descargas Parciales para Transformadores de Alta Potencia, es una herramienta de trabajo pensada para facilitar el estudio y la detección de las descargas parciales generadas por los transformadores de alta potencia. Permite tanto monitorizar remotamente experimentos de descargas en un laboratorio de alta tensión, como, trasladando esta instrumentación a una subestación eléctrica, monitorizar continuamente los transformadores de la misma, ofreciendo la posibilidad de mejorar la eficiencia con la cual se realiza normalmente la monitorización de estos equipos. El uso de esta herramienta permite reducir las tasas de fallos de los transformadores de alta potencia y alargar la vida útil de los mismos ya que facilita el paso del tradicional mantenimiento correctivo a una política de mantenimiento predictivo basado en el diagnóstico.

El Sistema de Monitorización Continua desarrollado permite registrar las señales acústicas de ultrasonidos en varios canales, disminuyendo el riesgo de pérdida de eventos debido al tiempo de rearme. Por otro lado, permite monitorizar en largos periodos de tiempo sin pérdidas de datos, gracias a la profundidad de memoria de la tarjeta de adquisición y gracias a la gestión segmentada de la misma. Con las marcas de tiempo y las señales almacenadas se posee toda la información necesaria para caracterizar cada canal de escucha y la relación entre los mismos para la detección y localización de la emisión acústica.

Por último, conviene destacar las posibilidades que ofrecen las arquitecturas como la empleada en esta solución, facilitando la escalabilidad, la incorporación de otros software de adquisición y la ampliación de la solución con nuevas funcionalidades como el acceso a través de *smartphones* o *tablets*.

Además, sería interesante continuar trabajando en este sistema. Una posibilidad sería realizar pruebas experimentales en el Laboratorio de Alta Tensión de la Universidad Carlos III con el sistema instalado y adquiriendo descargas parciales reales. Posteriormente sería muy interesante la instalación del sistema en una subestación con la que testear la utilidad de la aplicación y analizar posibles propuestas de mejora.

Otra línea de trabajo podría consistir en integrar el sistema desarrollado con un sistema de multiplexación para detección en múltiples canales acústicos. El software de procesamiento de señales acústicas adecuado podría proporcionar la información necesaria para el diagnóstico basándose en la identificación de descargas parciales y en la localización de la fuente de emisión acústica. El sistema de multiplexación se integraría con el hardware en el sistema remoto instalado en campo y el software de procesamiento podría integrarse en un sistema central de diagnóstico separado del transformador, incluso alejado de la subestación.

Capítulo VIII

Presupuesto

VIII.1. Material

A continuación, se detallan los costes asociados al material empleado en la implementación del sistema.

Descripción	Precio (€)	Unidades	Importe (€)
PDA14	5500	1	5500
PC (incluida Licencia Windows XP) <i>Ordenador AMD doble núcleo de 2,1GHz de velocidad, tarjeta gráfica NVIDIA GeForce 6200 y 4GB de memoria RAM. Placa base dotada con 1 puerto PCI.</i>	450	1	450
Licencia Microsoft Visual C++ 2008 Express Edition	500	1	500
Java EE 6	-	1	-
Licencias Frameworks Java (Spring, JFreeChart y Jetty)	-	1	-

Licencia Eclipse IDE	-	1	-
Licencia Flash Builder 4.6 Standard <i>Versión para Estudiantes</i>	-	1	-
Licencia SVN	-	1	-
TOTAL COSTES MATERIAL			6450

Tabla VIII.1 Costes de Material

VIII.2. Personal

A continuación, se detallan los costes asociados al personal que ha intervenido en el desarrollo de la solución, incluyendo las tareas de análisis de viabilidad, análisis de requisitos, diseño funcional, implementación, pruebas y documentación.

Descripción	Precio (€/h)	Horas	Importe (€)
Ingeniero Senior	32	360	11520
- <i>Análisis de Viabilidad</i>		20	
- <i>Análisis de Requisitos</i>		40	
- <i>Diseño Funcional</i>		40	
- <i>Implementación</i>		200	
- <i>Pruebas</i>		40	
- <i>Documentación</i>		20	
Ingeniero	20	360	7200
- <i>Análisis de Viabilidad</i>		-	
- <i>Análisis de Requisitos</i>		-	
- <i>Diseño Funcional</i>		-	
- <i>Implementación</i>		220	
- <i>Pruebas</i>		80	
- <i>Documentación</i>		60	
TOTAL COSTES PERSONAL			18720

Tabla VIII.2 Costes de Personal

VIII.3. Total

<i>Descripción</i>	<i>Importe (€)</i>
TOTAL COSTES MATERIALES	6450
TOTAL COSTES PERSONAL	18720
TOTAL	25170

Tabla VIII.3 Total de Costes

El presupuesto total de este proyecto asciende a la cantidad de 25170 EUROS.

Leganés a 27 de Junio de 2012

El ingeniero proyectista

Fdo. Daniel Lastra Rodríguez

Anexo A

Manual del Sistema

A.1. Introducción

En este capítulo se proporciona el manual de instalación y operación del Sistema de Monitorización Continua de Descargas Parciales. En primer lugar se indicarán los requisitos mínimos necesarios para instalar y arrancar la aplicación. A continuación, una guía de instalación y, para terminar, una guía de usuario que permitirá obtener un mayor rendimiento de la aplicación.

A.2. Requisitos Mínimos

Antes de instalar la aplicación, es necesario disponer de un equipo plataformado adecuadamente. Puesto que el fabricante suministra junto con la Tarjeta de Adquisición PDA14 un instalador de Windows que, entre otras cosas, instala los *drivers* y otras librerías con las que interactuar con la Tarjeta de Adquisición PDA14, será necesario que nuestro equipo disponga del Sistema Operativo **Windows 2000** o **Windows XP** [12].

A continuación se enumera cada uno de los componentes que necesitamos tener instalados para poder arrancar nuestra aplicación:

1. **Software de la Tarjeta de Adquisición PDA14**, proporcionado por Signatec. Instalando este software, se instalan los *drivers* de la tarjeta y además la librería de enlace dinámico **PDA14Win.dll** que es la que sirve de interfaz con los *drivers* de la tarjeta y que, tras la instalación, se puede encontrar en el directorio System32 de

Windows. También se instala la aplicación **UpdateP14** que permite actualizar el firmware de la tarjeta.

2. **J2SE 6.0**, que podemos obtener directamente de la Web de Oracle (<http://www.oracle.com>). Tras la instalación, será necesario comprobar que la instalación se ha realizado correctamente.

Si simplemente se desea acceder a la aplicación, basta con disponer de un navegador compatible con Flash y tener **Adobe Flash Player** instalado en el equipo. Al igual que el entorno de Ejecución de Java, Adobe Flash Player se puede obtener de forma gratuita directamente desde la página de Adobe (<http://get.adobe.com/es/flashplayer/>).

A.3. Instalación y Ejecución

El Sistema de Monitorización Continua de Descargas Parciales se distribuye en un fichero comprimido. Para instalar la aplicación bastará con descomprimir dicho archivo en el directorio que seleccionemos.

Tras la instalación, el sistema de archivos en el directorio de instalación deberá ser como el mostrado en la **Figura A.1**.

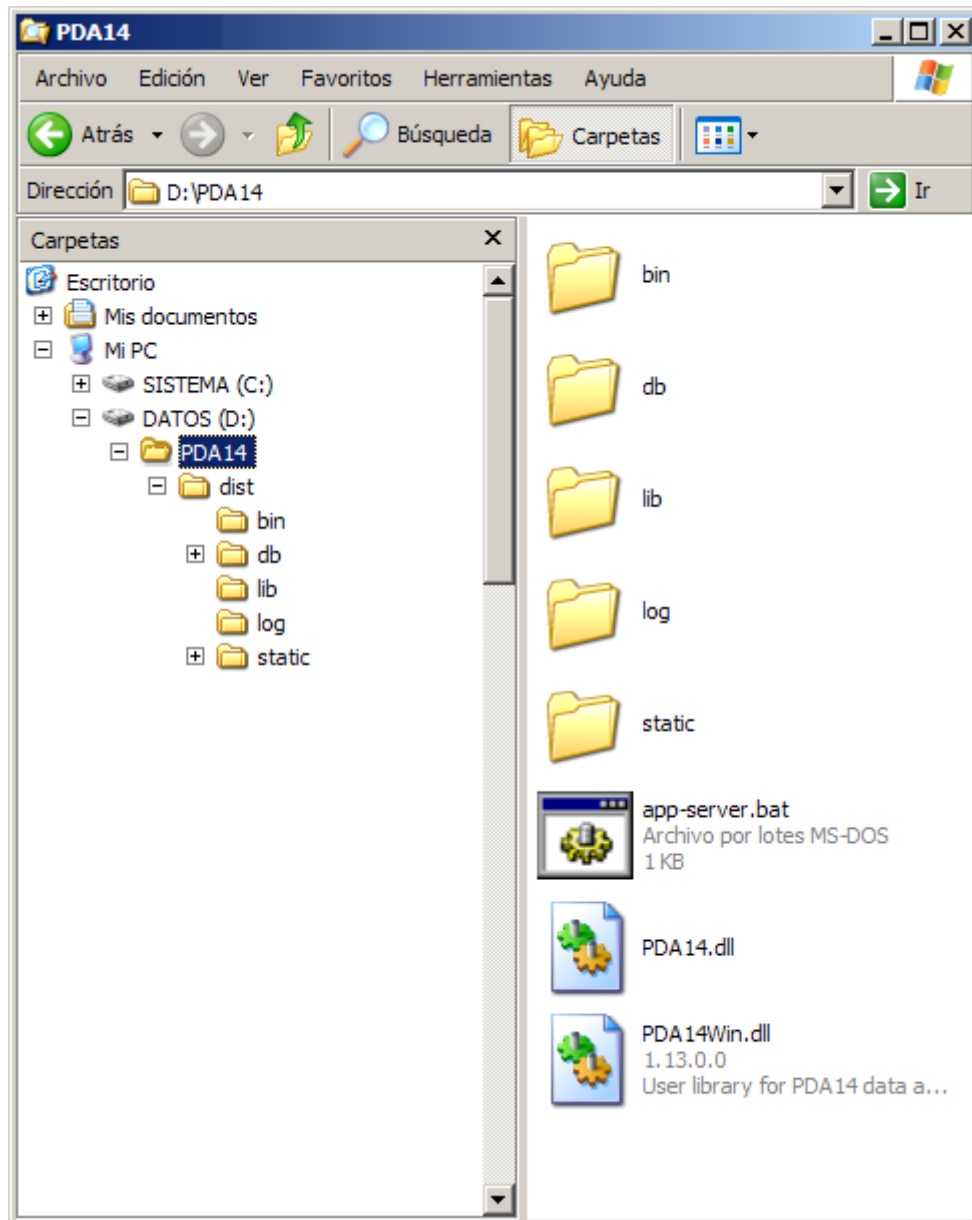


Figura A.1 Directorio de Instalación de la Aplicación

Una vez instalado, podemos arrancar la aplicación ejecutando el archivo **app-server.bat**.

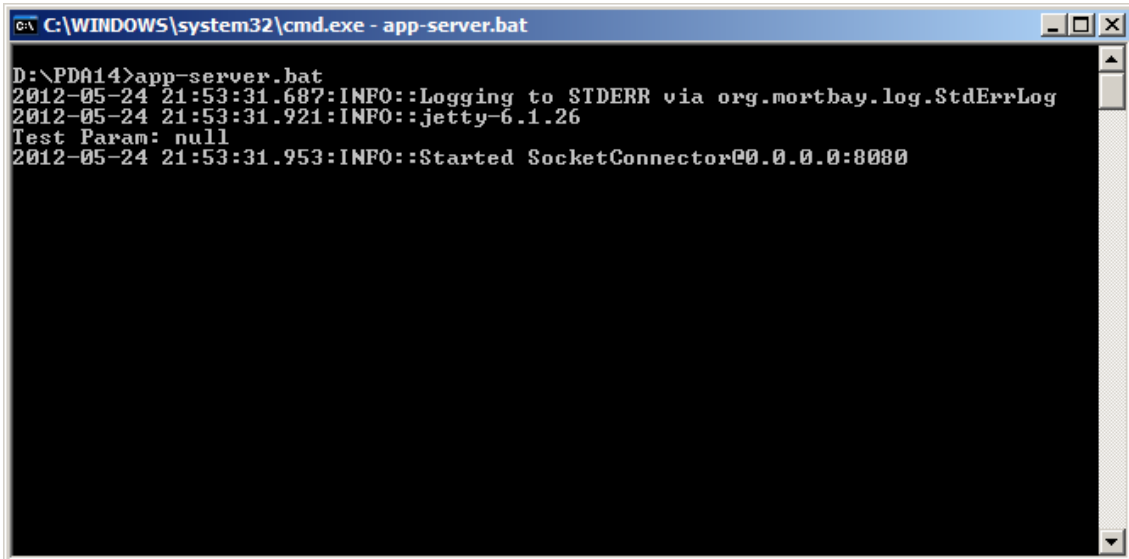


Figura A.2 Ejecución de la Aplicación

Por último, se puede comprobar que el servidor de aplicaciones se ha levantado correctamente accediendo a la aplicación a través de un navegador Web y comprobando como se carga la pantalla de acceso al sistema.

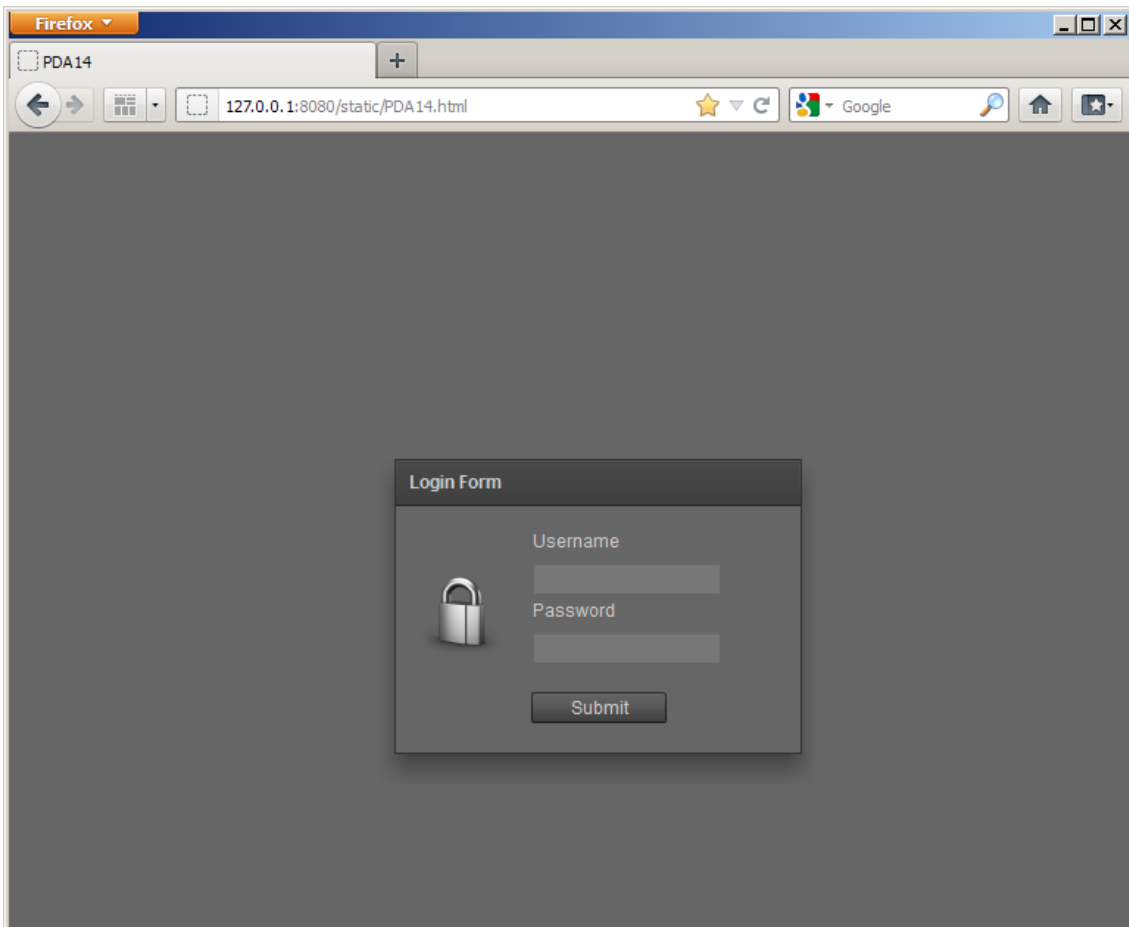


Figura A.3 Vista de la Aplicación desde un Navegador Web

A.4. Guía de Usuario

A continuación, se proporciona una guía de usuario básica con la que se pretende facilitar a cualquier usuario la utilización de la aplicación. En esta guía se describen cada una de las opciones que ofrece la aplicación, haciendo hincapié en aquellas que pueden resultar más atractivas al usuario.

A.4.1. Inicio de Sesión

Antes de comenzar a utilizar la aplicación, será necesario iniciar sesión. Con una pantalla de Inicio de Sesión como la que se muestra en la **Figura VII.4**, el usuario podrá registrarse en el sistema de forma que éste validará no sólo si el usuario tiene permiso para acceder a la aplicación, sino que también, validará que permisos tiene dicho usuario dentro de la aplicación.

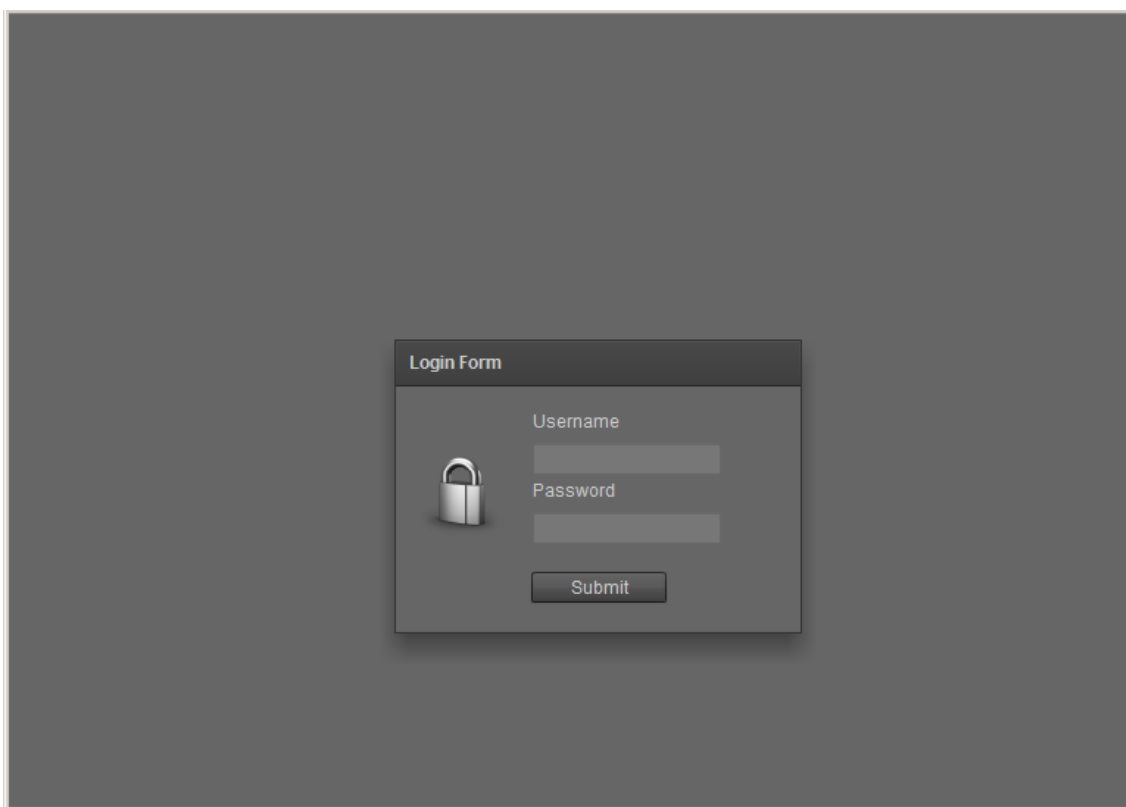


Figura A.4 Inicio de Sesión en la Aplicación

A.4.2. Escritorio

Una vez que se ha iniciado la sesión en el sistema, se carga el escritorio de la aplicación. En la **Figura VII.5** se muestra un pantallazo de dicho escritorio. En la parte superior, se encuentra la barra de menús, donde destaca el Menú de Inicio, un indicador de la hora del sistema y un botón con el que el usuario podrá cerrar la sesión una vez finalizadas sus tareas.

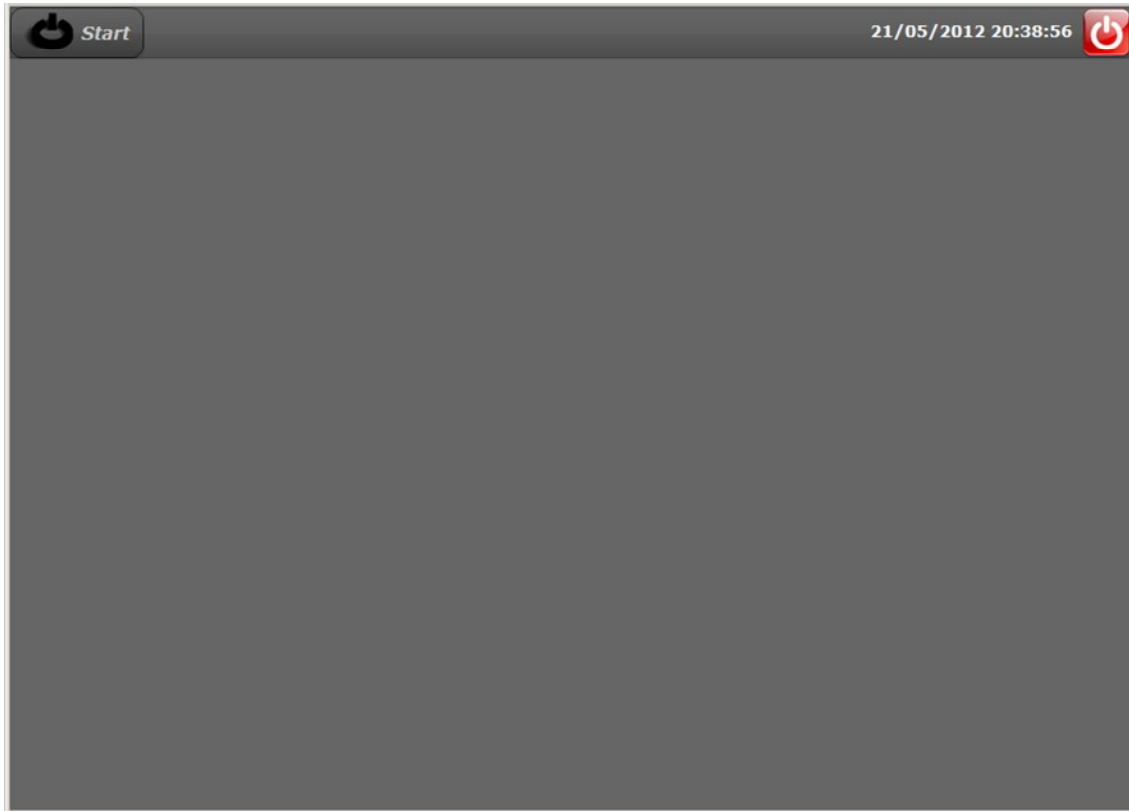


Figura A.5 Escritorio de la Aplicación

A.4.3. Menú de Inicio

En el Menú de Inicio se muestra cada una de las funcionalidades que ofrece el sistema en función del Perfil del Usuario con el que se ha iniciado sesión. Así, un usuario con el perfil de **Administrador**, tendrá acceso a todas las funcionalidades que ofrece el sistema:



Figura A.6 Menú de Inicio de los usuarios con perfil de Administrador

Un usuario con el perfil de **Operador**, tendrá acceso a todas las funcionalidades que ofrece el sistema relacionadas con la operación del sistema, pero no a las de Administración de la Aplicación.

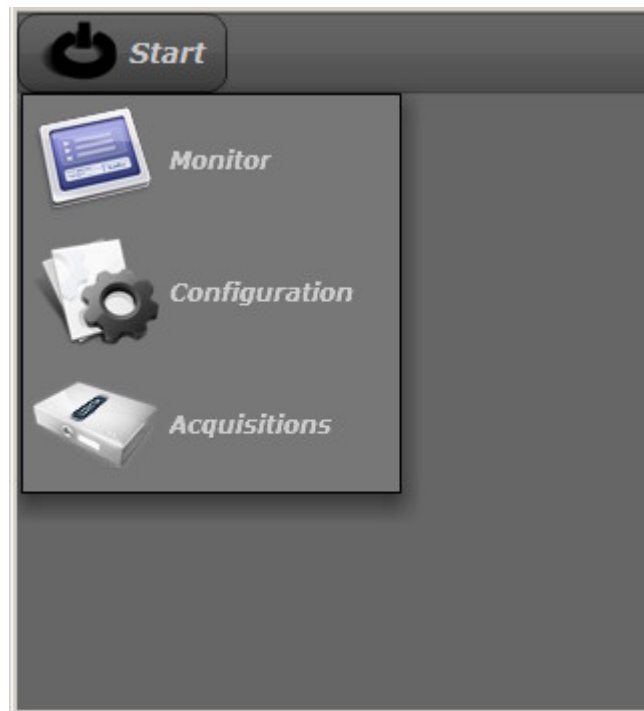


Figura A.7 Menú de Inicio de los usuarios con perfil de Operador

Por último, un usuario con el perfil de **Visitante**, sólo tendrá acceso a las operaciones de consulta que consisten básicamente en el acceso a las adquisiciones realizadas:



Figura A.8 Menú de Inicio de los usuarios con perfil de Visitante

A.4.4. Administración de Usuarios

La ventana de administración de usuarios sólo está accesible para usuarios con el perfil de administrador. Desde esta ventana, un administrador podrá dar de alta nuevo usuario, eliminar usuarios y cambiar los permisos de cada usuario modificando el perfil del mismo.

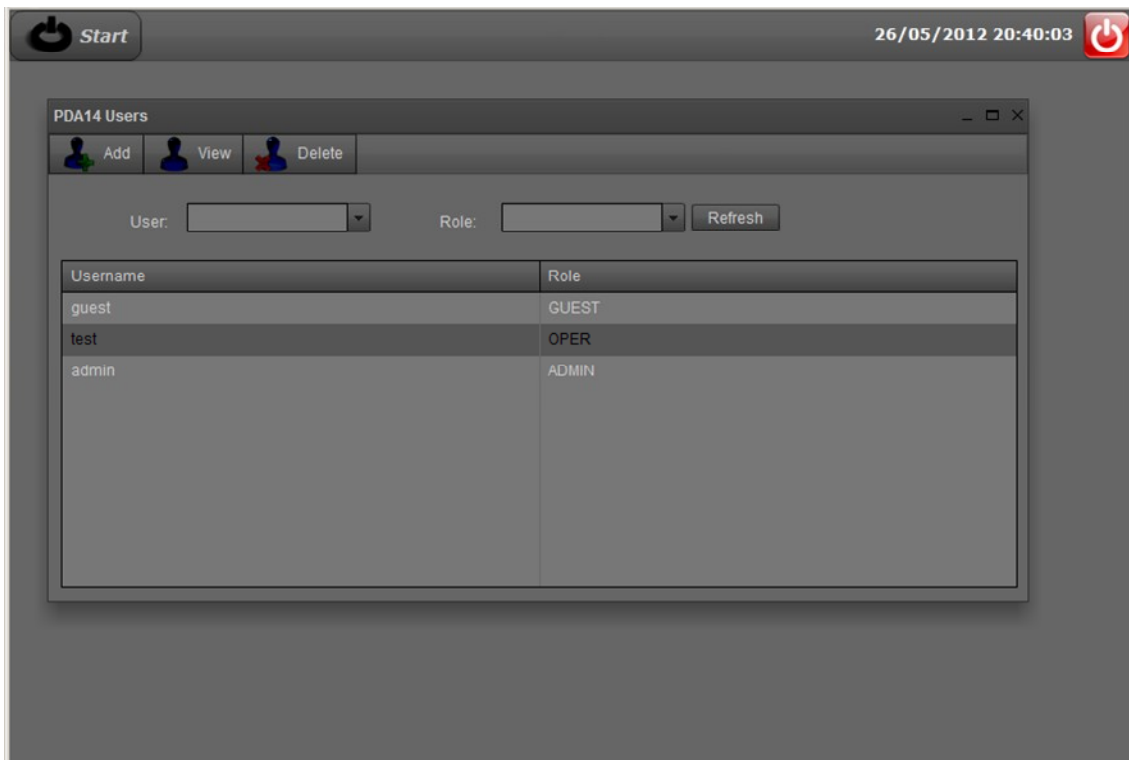


Figura A.9 Ventana de Administración de Usuarios

A.4.5. Configuración

La ventana de configuración permite al usuario configurar la tarjeta de adquisición, estableciendo los parámetros que considere oportunos para llevar a cabo la monitorización del equipo en examen. Como se observa en la **Figura VI.9**, se puede configurar el número de canales sobre los que hacer la adquisición, el nivel de disparo, la frecuencia de muestreo,...

Además, conviene destacar la existencia del parámetro **Adquirir Continuamente**. Este parámetro permite seleccionar si lo que se quiere llevar a cabo es una adquisición única, o una monitorización continua.

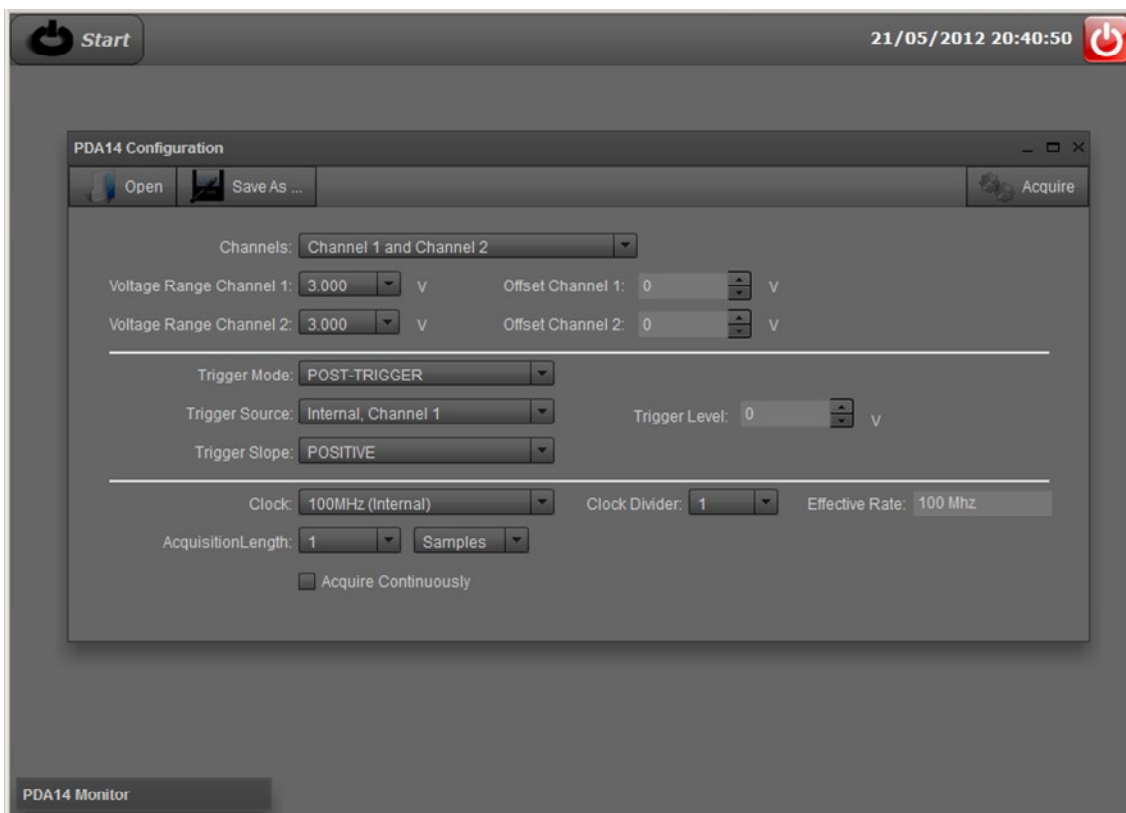


Figura A.10 Ventana de Configuración

A.4.6. Monitorización

Cada una de las configuraciones que recibe el sistema, se van almacenando en forma de adquisiciones pendientes que se van procesando en orden de llegada. Mediante la Ventana de Monitorización, el usuario puede conocer el estado del Sistema, es decir, que adquisiciones ya han finalizado, si alguna ha sido cancelada, si existe alguna adquisición en curso y si queda alguna pendiente.

Además, el usuario tiene la capacidad de visualizar cualquier adquisición, sea cual sea el estado en el que se encuentre y además, podrá cancelar una adquisición en curso si lo desea.

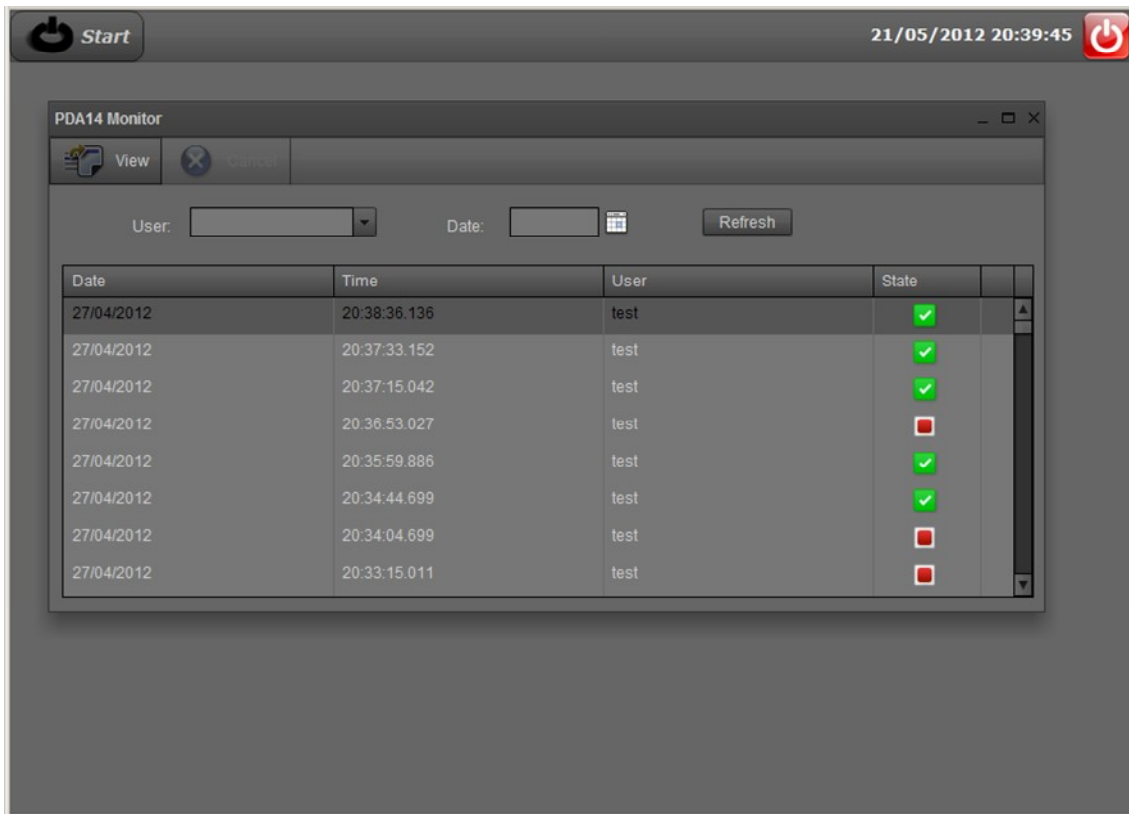


Figura A.11 Ventana de Monitorización

A.4.7. Adquisiciones

Cada una de las adquisiciones que se van realizando, quedan registradas y almacenadas en el sistema. Mediante la ventana de adquisiciones, el usuario podrá acceder a dichas adquisiciones.

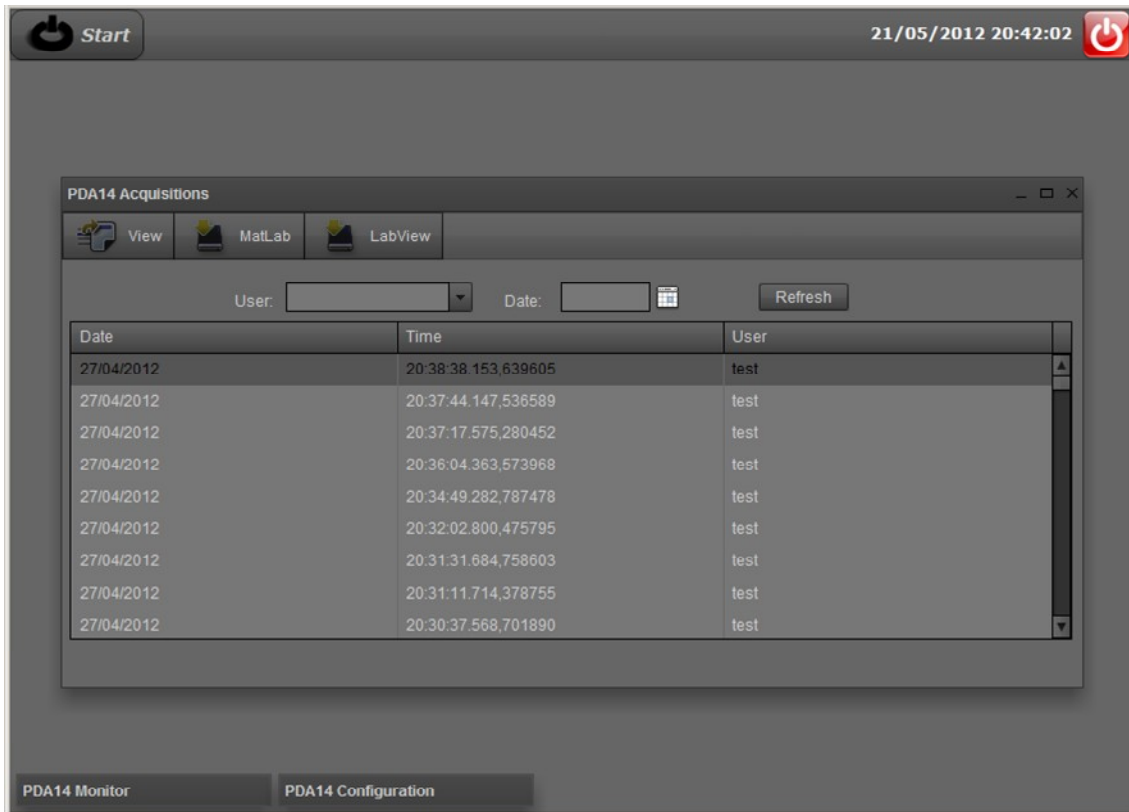


Figura A.12 Ventana de Adquisiciones

La aplicación permite al usuario visualizar la adquisición que desee. Podemos encontrarnos situaciones en las que el sistema se encuentre instalado en zonas donde la conexión a Internet sea muy lenta, por ejemplo en lugares de poca cobertura. Por este motivo, por defecto, la aplicación no muestra la señal completa, sino que se trae sólo algunas muestras con las que realizar una visualización previa de la señal. Si lo desea, modificando la escala, el usuario podrá obtener una visualización más exacta de la señal adquirida.

Asimismo, el usuario no sólo podrá visualizar cada una de las adquisiciones desde la propia aplicación, también tendrá la opción de descargarlas en ficheros de datos compatibles con **MatLab [20]** o **LabView [21]** permitiéndole trabajar con los datos adquiridos desde cualquiera de estas herramientas.



Figura A.13 Visualización de una señal adquirida

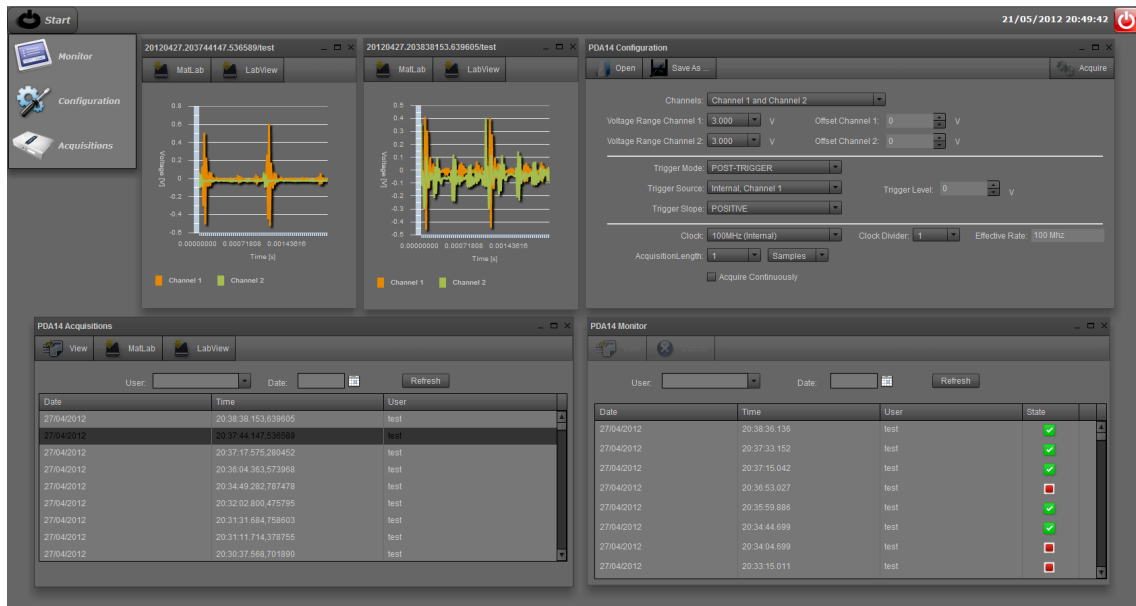


Figura A.14 Vista Completa de la Aplicación

Anexo B

Manual de Desarrollo

Este proyecto se centra en la implementación de un Sistema de Monitorización Continua de Descargas Parciales para Transformadores de Alta Potencia. Para ello, ha sido necesario utilizar diferentes tecnologías que han permitido entre otras cosas acceder desde Java a la librería de funciones en C proporcionada por Signatec con la Tarjeta de Adquisición PDA14.

En este anexo se encuentra una serie de guías cuyo objetivo es el de facilitar el trabajo a la hora de utilizar tanto el software proporcionado por Signatec como el software desarrollado en este proyecto para interactuar con la PDA14.

B.1. Desarrollo de Aplicaciones para la PDA14 con Microsoft Visual C++ 2008 Express Edition

La PDA14 de Signatec trae consigo una librería de funciones implementadas en C. Se trata de una librería de enlace estático para Windows, que permite al usuario desarrollar sus propias aplicaciones para interactuar con la propia tarjeta. Aunque existen diferentes entornos de desarrollo para programar en C en Windows, en este caso se ha optado por utilizar el Entorno de Desarrollo Microsoft Visual C++ 2008 Express Edition.

Antes de continuar, destacaremos que para desarrollar nuestras propias aplicaciones, es necesario disponer de **Microsoft Visual C++ 2008 Express Edition** y tener instalado el **Software de Adquisición de la PDA14** que se suministra junto con la tarjeta.

B.1.1. Configuración de un Nuevo Proyecto

En este apartado se muestra cómo construir una aplicación muy sencilla que permita configurar la Tarjeta de Adquisición PDA14 y llevar a cabo una adquisición utilizando el Modo Segmentado.

Para ello, se ha utilizado el Entorno de Desarrollo Microsoft Visual C++ 2008 Express Edition y que podemos ver en la **Figura B.1**.

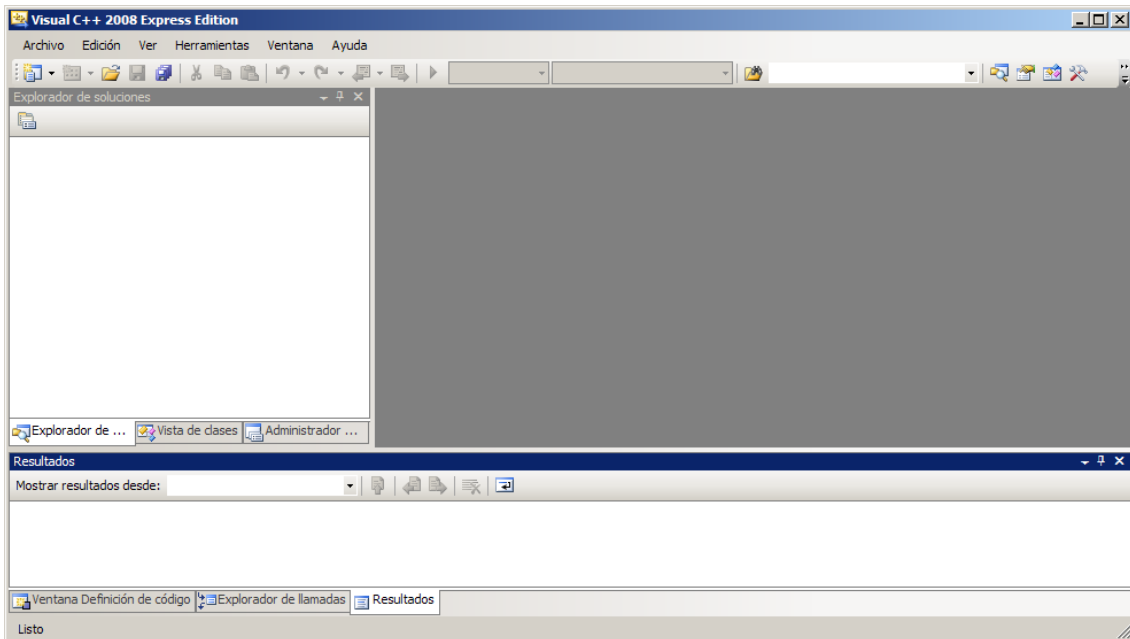


Figura B.1 Entorno de Desarrollo Microsoft Visual C++ 2008 Express Edition

En primer lugar es necesario crear y configurar adecuadamente un nuevo proyecto dentro del Entorno de Desarrollo. Como se muestra en la **Figura B.2**, desde el Menú Archivo podemos crear un nuevo proyecto:

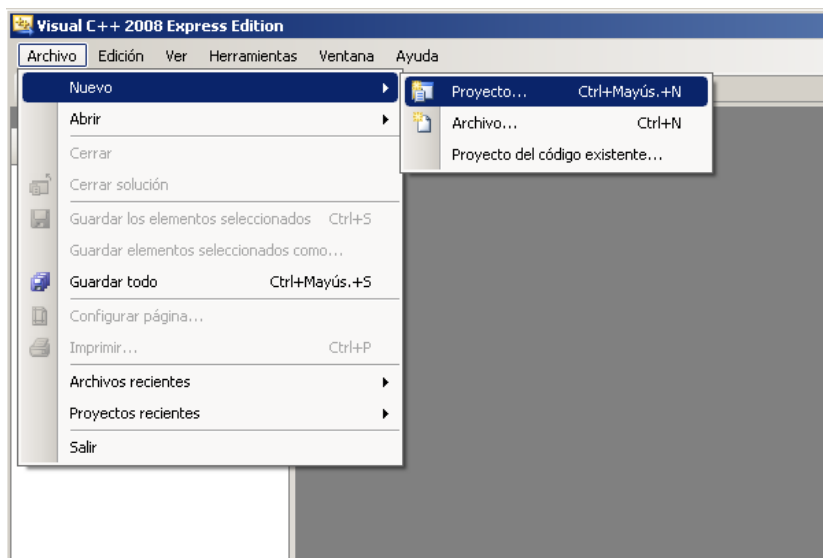


Figura B.2 Creación de un Nuevo Proyecto

Aparecerá una venta como la de **Figura B.3** que permite seleccionar que Tipo de Proyecto se desea crear. A modo de ejemplo, se creará una aplicación de consola por lo que en el Tipo de Proyecto, se debe seleccionar **Win32** y en la plantilla **Aplicación de consola Win32**. Además, es necesario indicar el nombre del Proyecto, **SampleSegmentedTriggerMode** y el directorio donde se ubicará.

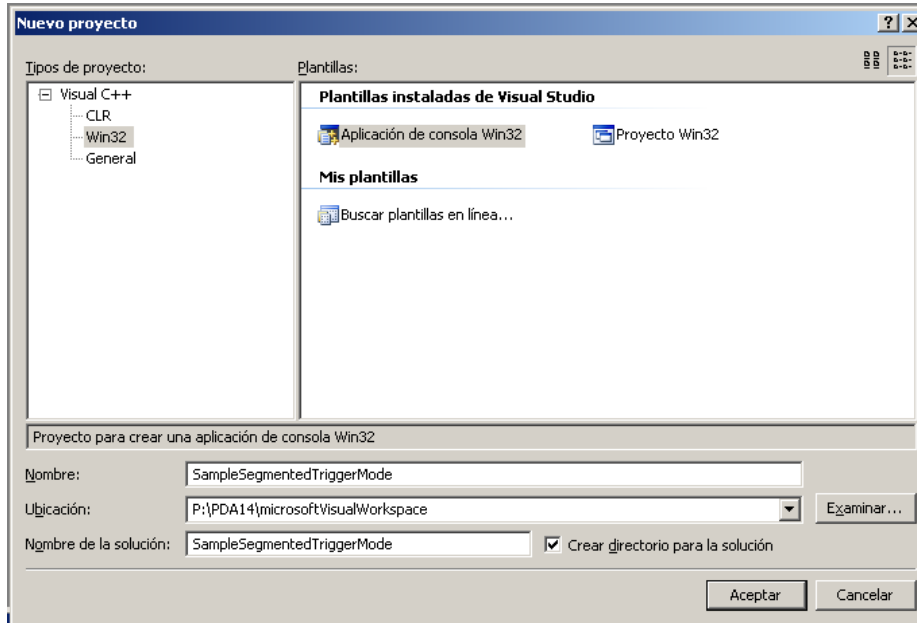


Figura B.3 Selección de Propiedades del Proyecto

Al aceptar, aparecerá el Asistente para Aplicaciones Win32 que se muestra en la **Figura B.4** y que permite configurar el tipo de aplicación que se va a desarrollar.

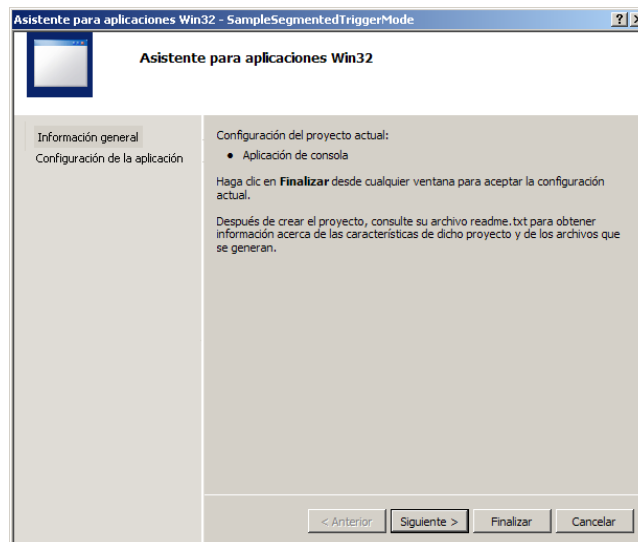


Figura B.4 Asistente para Aplicaciones Win32

En lugar de hacer clic en Finalizar, hacer clic en Siguiente permite confirmar que la configuración es la deseada, es decir que el Tipo de Aplicación es Aplicación de Consola.

Además, hay que asegurarse que se genera un proyecto vacío. Por tanto, la configuración de la aplicación debe quedar como se observa en **Figura B.5**:

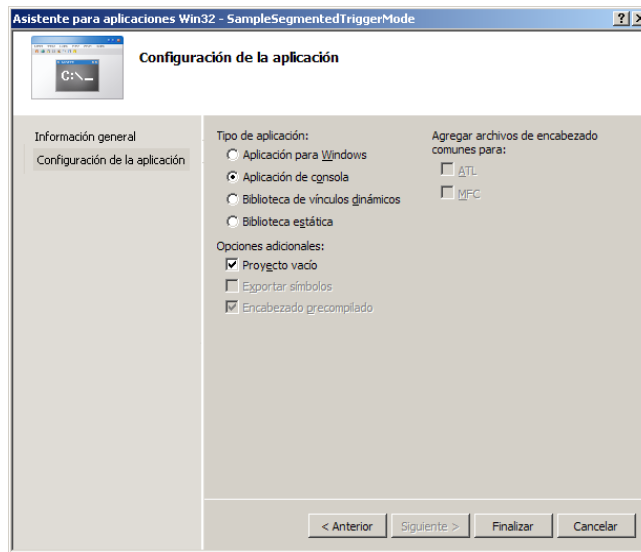


Figura B.5 Configuración de la Aplicación

Al hacer clic en Finalizar, en el Explorador de Soluciones del Microsoft Visual C++ 2008 Express Edition se creará un nuevo proyecto denominado **SampleSegmentedTriggerMode** como se observa en la **Figura B.6**:

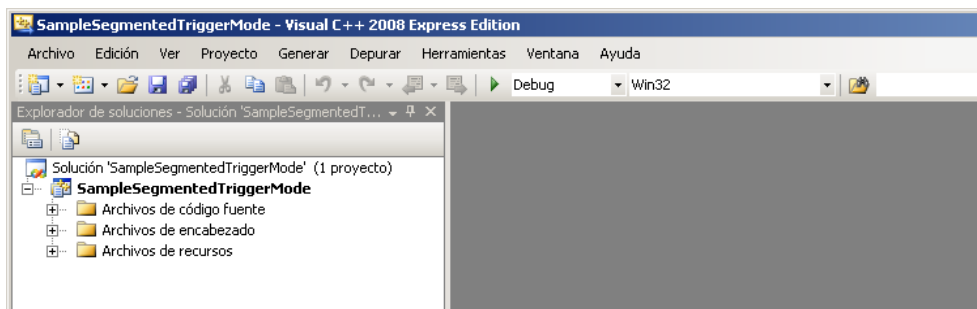


Figura B.6 Nuevo Proyecto en el Explorador de Soluciones

El siguiente paso, será añadir al proyecto los ficheros **PDA14Win.h** y **PDA14Win.lib** proporcionados por el fabricante de la PDA14 y que componen la librería de enlace estático que permite interactuar con el Hardware de Adquisición.

Los ficheros **PDA14Win.h** y **PDA14Win.lib** son un Archivo de Encabezado y un Archivo de Recursos respectivamente, que se pueden agregar al proyecto desde el Explorador de Soluciones como se muestra en la **Figura B.7** y en la **Figura B.8**.

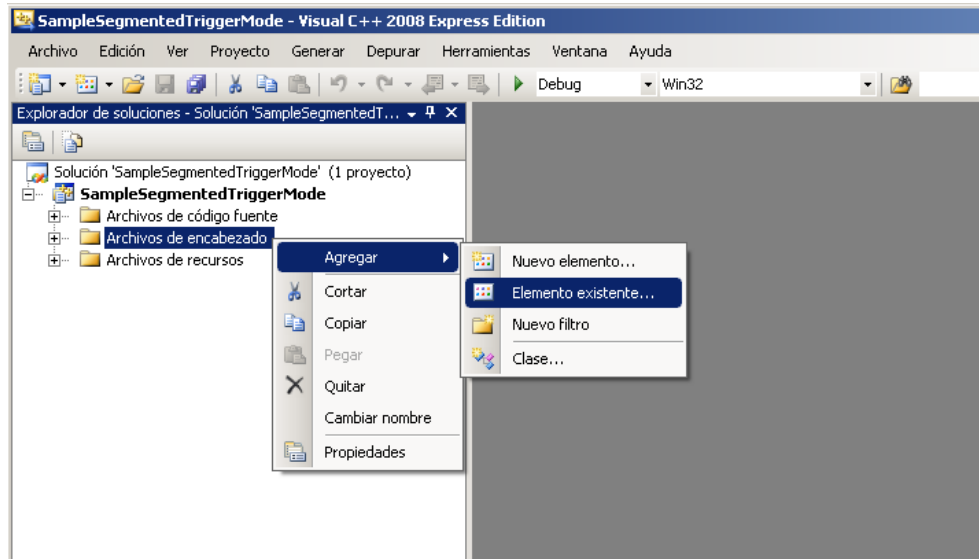


Figura B.7 Agregación de Archivo de encabezado

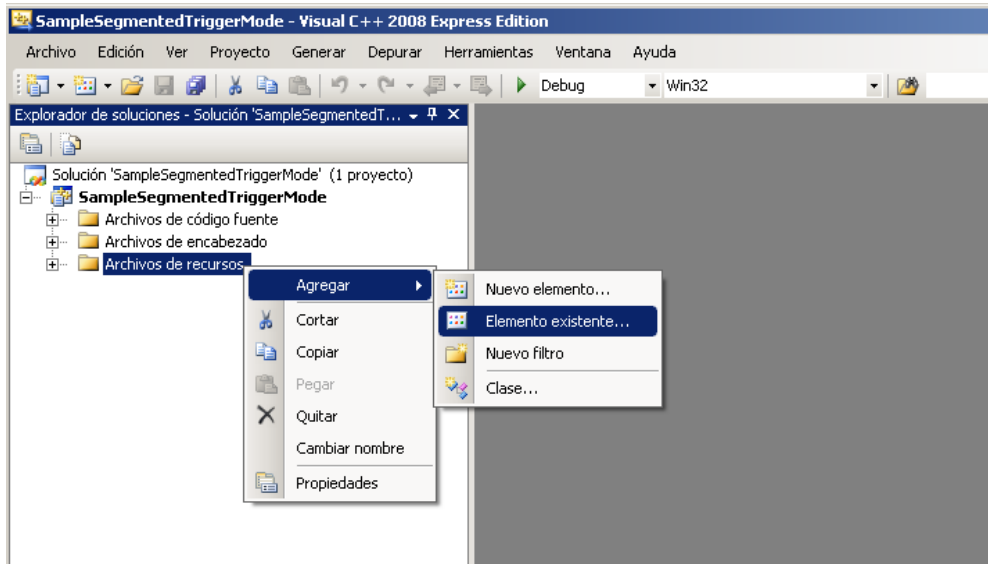


Figura B.8 Agregación de Archivos de recursos

B.1.2. Implementación de la Aplicación

Una vez configurado el nuevo proyecto, se va proceder a desarrollar una aplicación de ejemplo de utilización del Modo Segmentado de la PDA14.

En primer lugar, se añadirán los **Archivos de Código Fuente** que sean precisos, como se muestra en la **Figura B.9**. En este caso, uno sólo llamado **main.cpp**:

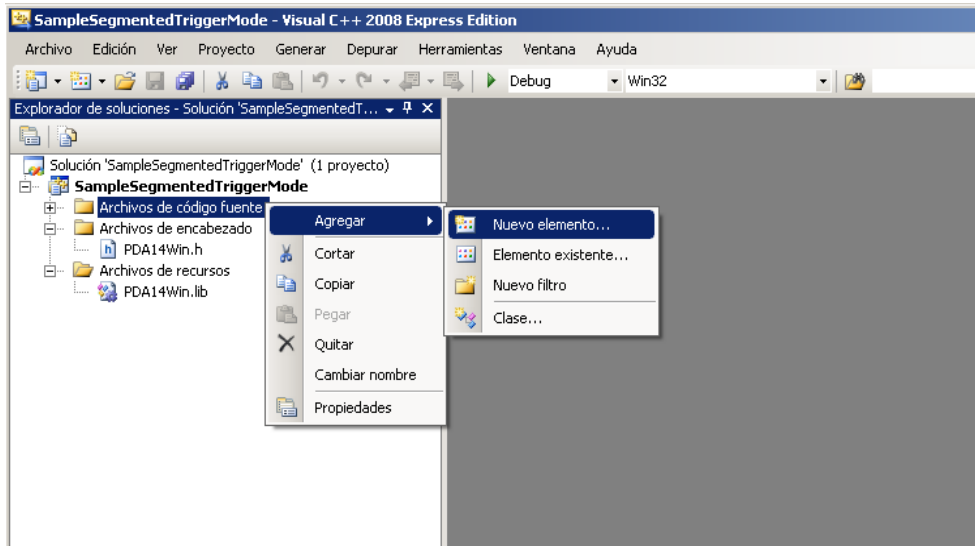


Figura B.9 Agregación de Archivos de código Fuente

Tal y como se muestra en la **Figura B.10**, en el asiste para Agregar nuevo elemento al proyecto se debe seleccionar la plantilla **Archivo C++ (.cpp)**:

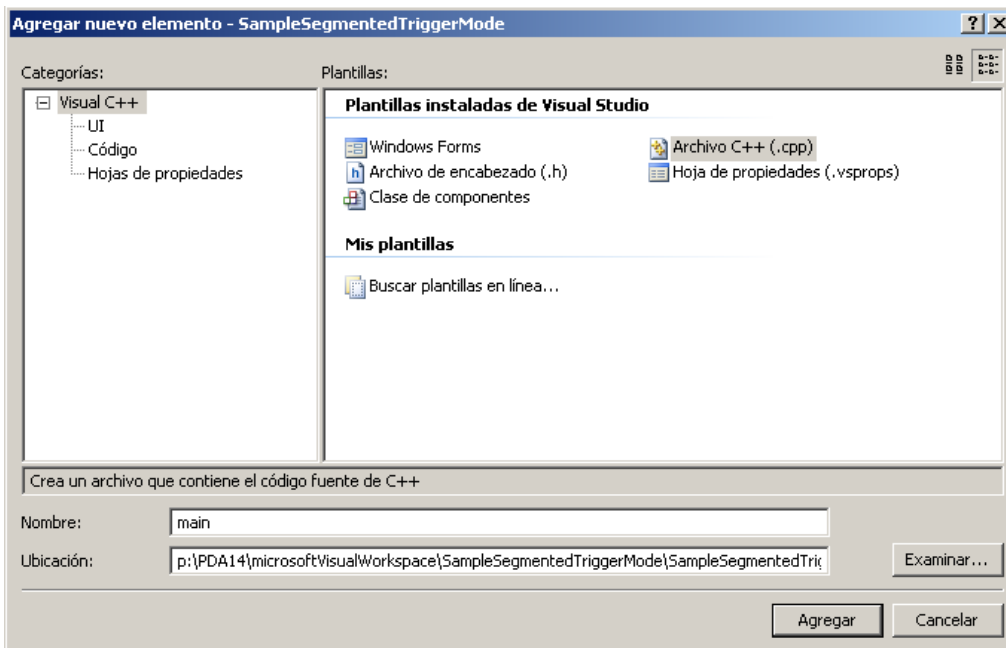


Figura B.10 Asistente para Agregar nuevo Elemento

Como se observa en la **Figura B.11** el nuevo Archivo de Código Fuente aparecerá en el proyecto y sólo hay que seleccionarlo para editarlo.

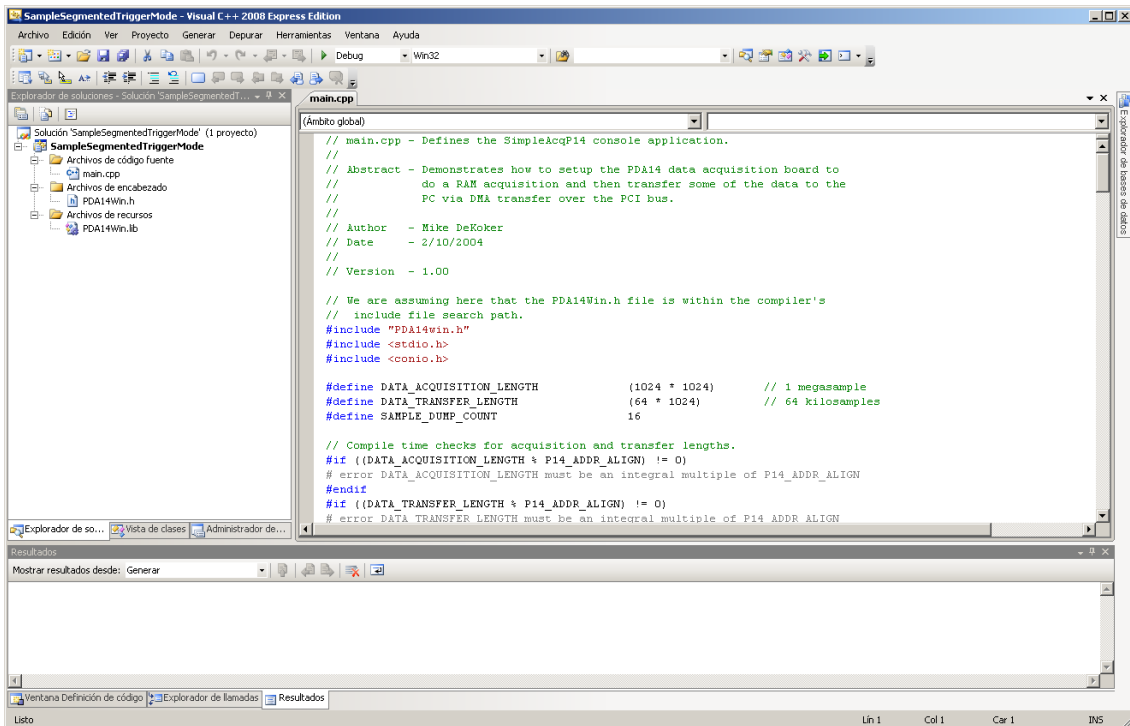


Figura B.11 Codificación del Nuevo Elemento

En el Archivo de Código Fuente, se implementa el ejemplo. En este caso, el código de ejemplo se ha obtenido directamente del Manual de la PDA14 proporcionado por el fabricante [12].

B.1.3. Generación y Ejecución de la Aplicación

El propio entorno de desarrollo facilita las tareas de generar y ejecutar la solución. En la **Figura B.14** se observa como desde el Menú Generar, o simplemente pulsando F7, se puede generar la solución.

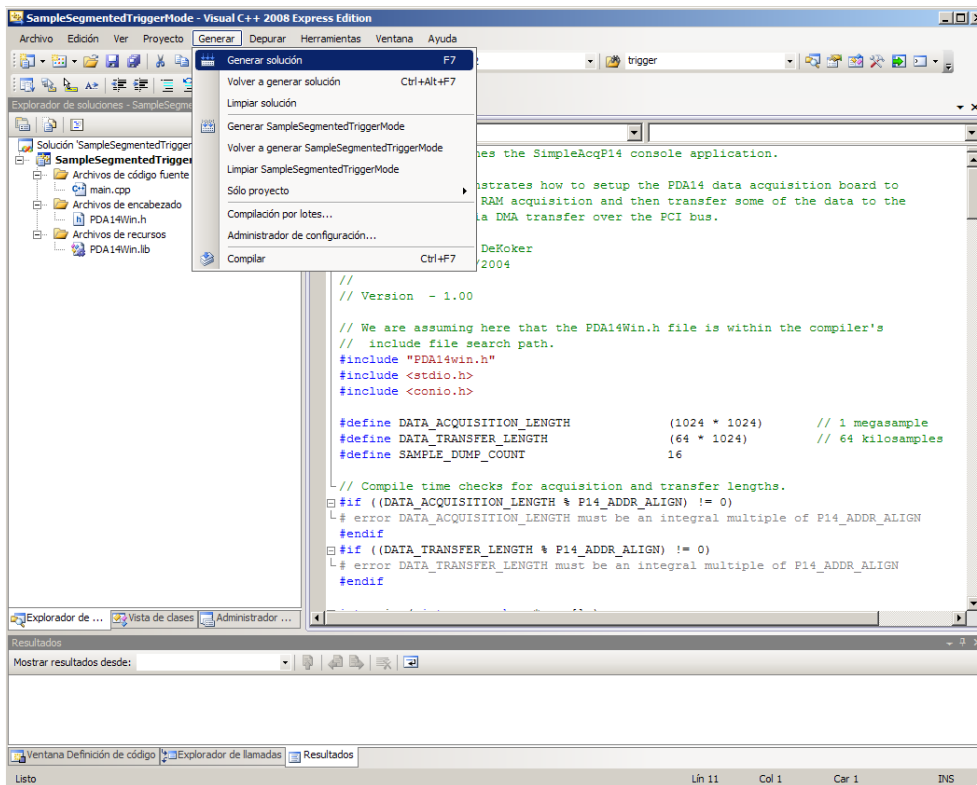


Figura B.12 Generación de la Solución

Como se observa en la **Figura B.13**, en la carpeta Debug, dentro del directorio de del proyecto, se creará el archivo ejecutable con nuestra solución:

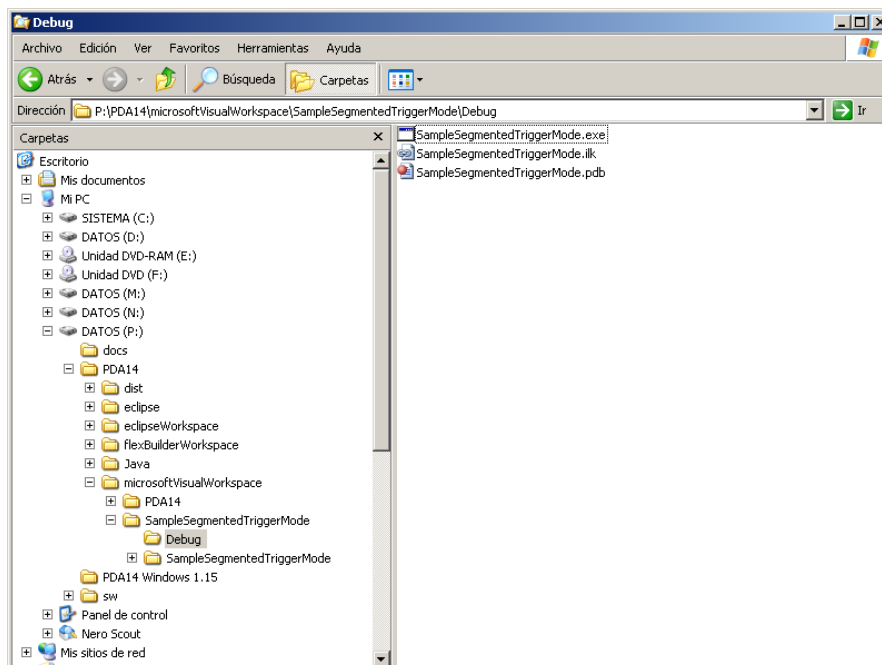
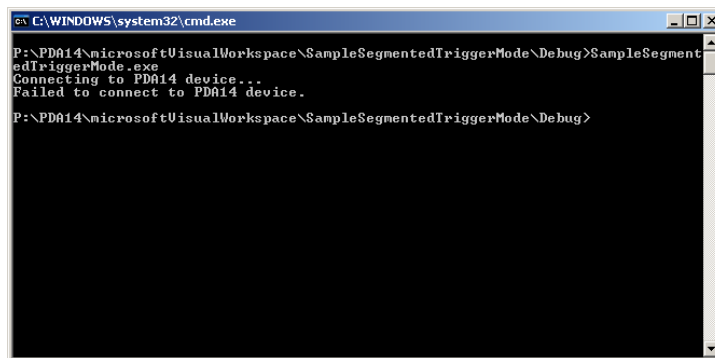


Figura B.13 Solución generada en el Sistema de Archivos

En la **Figura B14**, se encuentra un ejemplo de ejecución de la aplicación desde un intérprete de comandos de Windows:

134



```

C:\WINDOWS\system32\cmd.exe
P:\PDA14\microsoftVisualWorkspace\SampleSegmentedTriggerMode\Debug>SampleSegmentedTriggerMode.exe
Connecting to PDA14 device...
Failed to connect to PDA14 device.
P:\PDA14\microsoftVisualWorkspace\SampleSegmentedTriggerMode\Debug>

```

Figura B.14 Ejemplo de Ejecución

Para que la aplicación se ejecute correctamente, es necesario asegurarse que se tiene acceso a la librería de enlace dinámico de Windows **PDA14Win.dll** proporcionada por Signatec. Para ello, basta con tener instalados los drivers de la PDA14 que proporciona el fabricante.

B.2. Desarrollo de Aplicaciones Java para la PDA14 con Eclipse IDE

Ya se ha visto como desarrollar una aplicación en C o en C++ con el Entorno de Desarrollo Microsoft Visual C++ 2008 Express Edition. Sin embargo, uno de los objetivos de este proyecto era el de proporcionar un entorno de desarrollo que permitiera construir aplicaciones en Java para interactuar con la PDA14 de Signatec.

En este proyecto se ha desarrollado una librería Java y una librería de enlace dinámico en C con la que podemos interactuar con la PDA14 desde una aplicación Java. A continuación se muestra como configurar el Entorno de Desarrollo en Eclipse para crear aplicaciones Java utilizando éstas nuevas librerías.

Antes, es necesario tener instalado alguna de las versiones disponibles de la plataforma **Java** y que dependerá del tipo de aplicación que vayamos a desarrollar. Además, habrá que tener instalado el **Software de Adquisición de la PDA14**. Por otro lado, será necesario disponer de **Eclipse IDE** (o de cualquier entorno de desarrollo basado en Eclipse IDE), que se puede obtener directamente la página Web de la Fundación Eclipse (<http://www.eclipse.org>)

Por último, es necesario disponer de las librerías **pda14.jar** y **PDA14.dll** desarrolladas en este proyecto.

135

B.2.1. Configuración de un Nuevo Proyecto

En este apartado se va a construir una aplicación muy sencilla que permita configurar la Tarjeta de Adquisición PDA14 y llevar a cabo una adquisición.

Para ello, se utilizará el Entorno de Desarrollo Eclipse y cuya apariencia se puede observar en la **Figura B.15**.

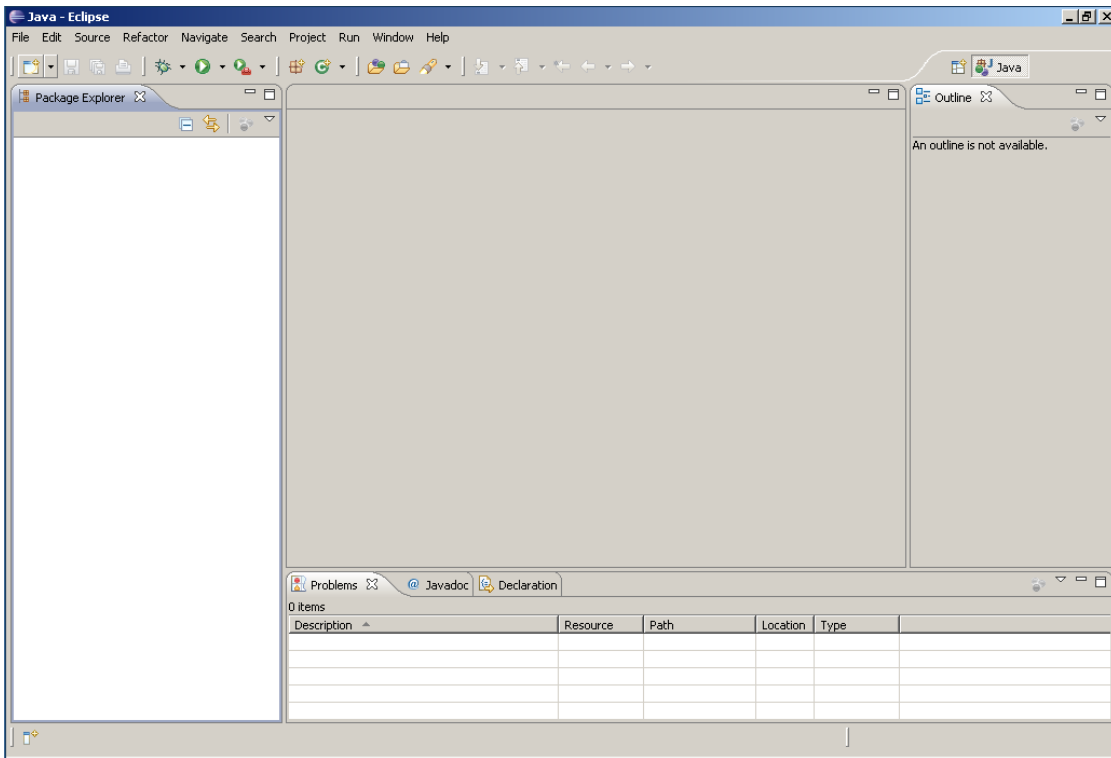


Figura B.15 Escritorio de Eclipse IDE

En primer lugar se va a crear y configurar adecuadamente un nuevo proyecto dentro del Entorno de Desarrollo. Como se puede observar en la **Figura B.16**, desde el Menú File, se puede crear un nuevo proyecto Java.

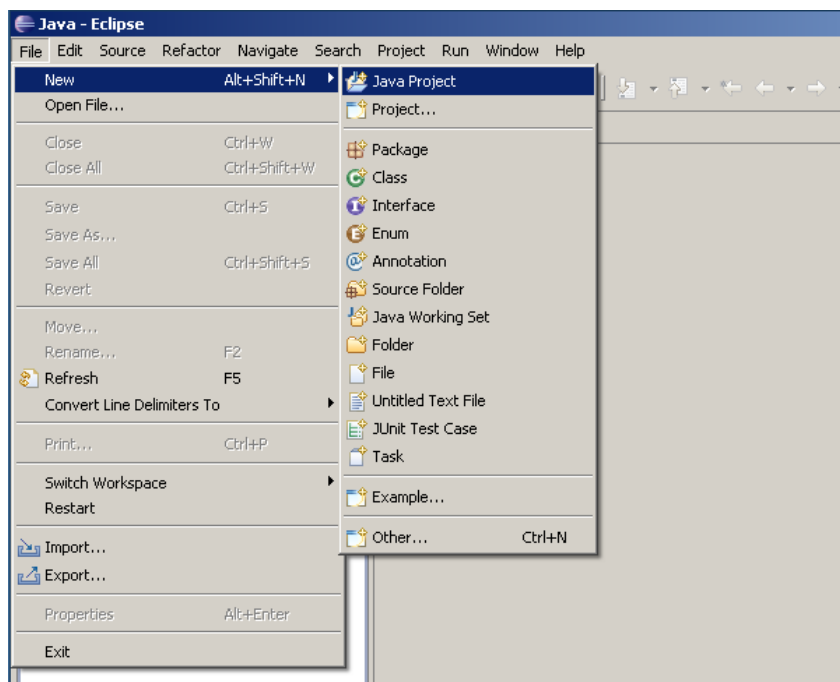


Figura B.16 Creación de Nuevo Proyecto desde Eclipse IDE

Aparecerá un Asistente para Proyectos Java como el que se muestra en la **Figura B.17** y que permitirá configurar algunos parámetros del proyecto como el nombre, **PDA14SampleSignalAcquisition**.

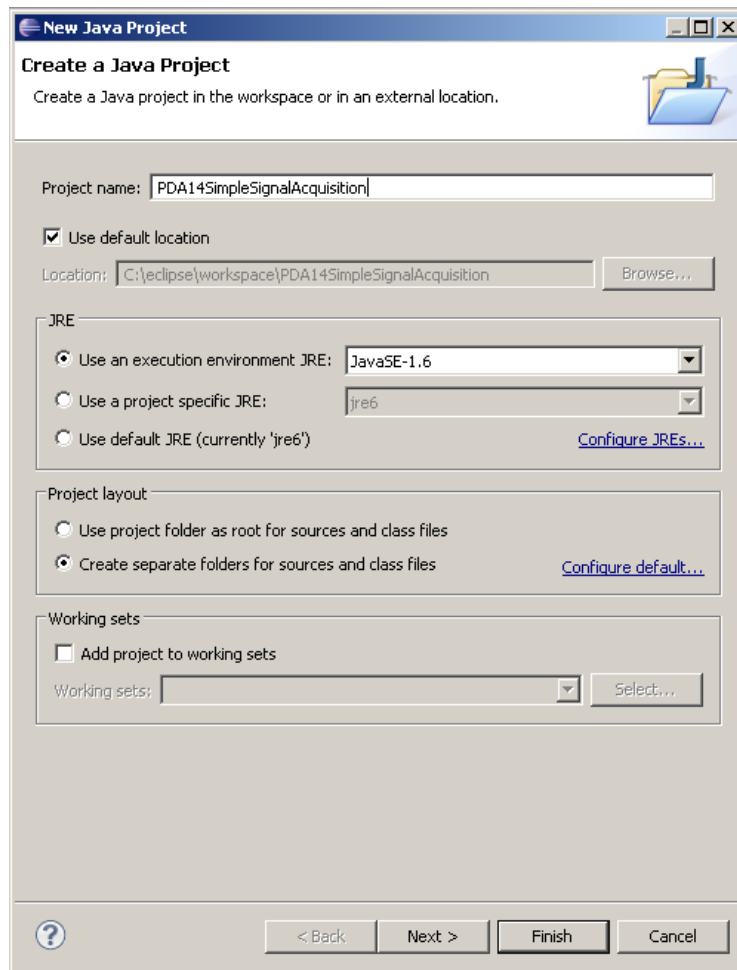


Figura B.17 Asistente para creación de proyectos Java

Al hacer clic en Finish, en el Explorador del Eclipse aparecerá el nuevo proyecto tal y como se muestra en la **Figura B.18**

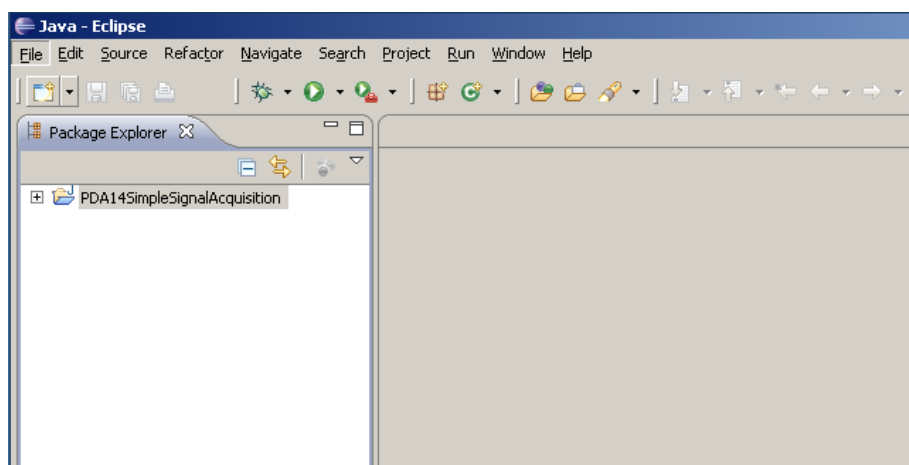


Figura B.18 Explorador de paquetes del Eclipse IDE

Para agrupar todas las dependencias del proyecto en un mismo directorio, como se indica en la **Figura B.19**, dentro del proyecto se añade una nueva carpeta denominada **lib**:

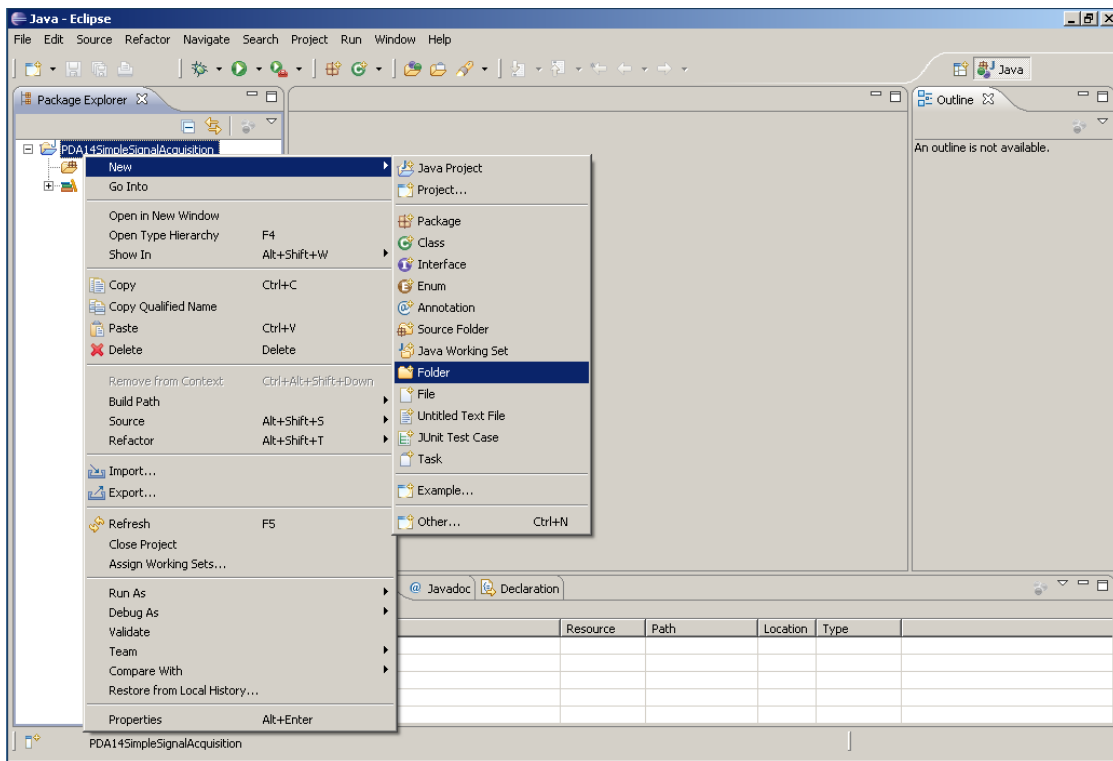


Figura B.19 Nueva carpeta en el Proyecto

En el Asistente para Añadir un nuevo directorio al proyecto donde se indica el nombre del directorio tal y como se muestra en la **Figura B.20**:

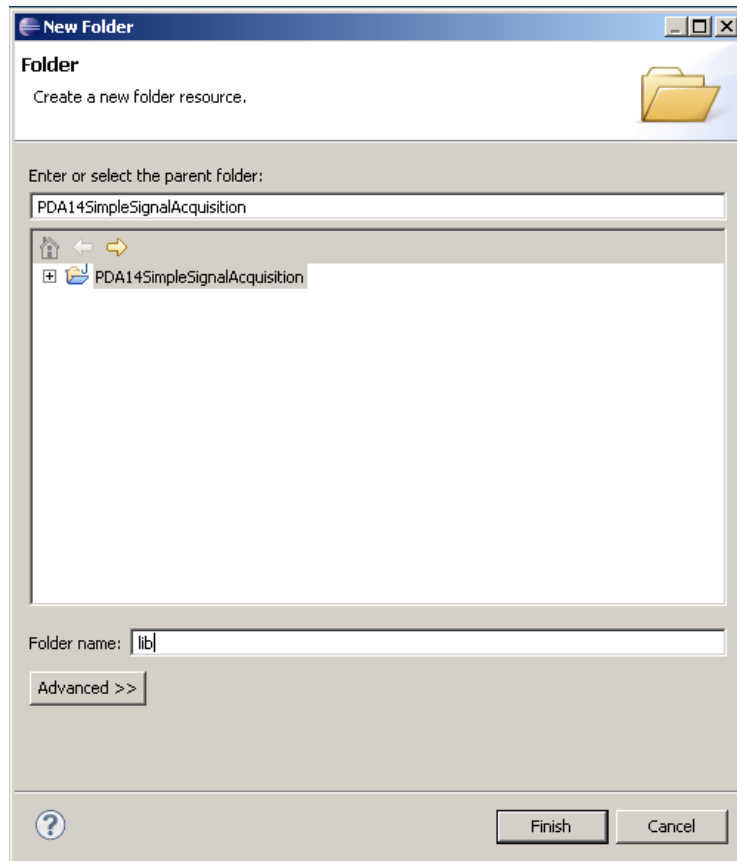


Figura B.20 Asistente de Creación de Carpetas

Al hacer clic en Finish, dentro del proyecto, aparecerá este nuevo directorio donde se deberá añadir las librerías implementadas en este proyecto para interactuar con la PDA14. Se trata de los ficheros **pda14.jar** y **PDA14.dll**:

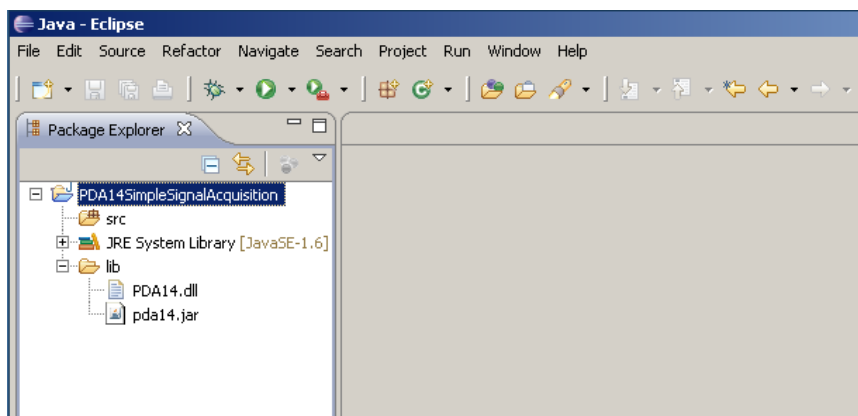


Figura B.21 Explorador de paquetes con la carpeta de dependencias

Para que el proyecto utilice la librería pda14.jar tanto en tiempo de compilación como en tiempo de ejecución, es necesario añadirla al path del proyecto como se indica en la **Figura B.22**.

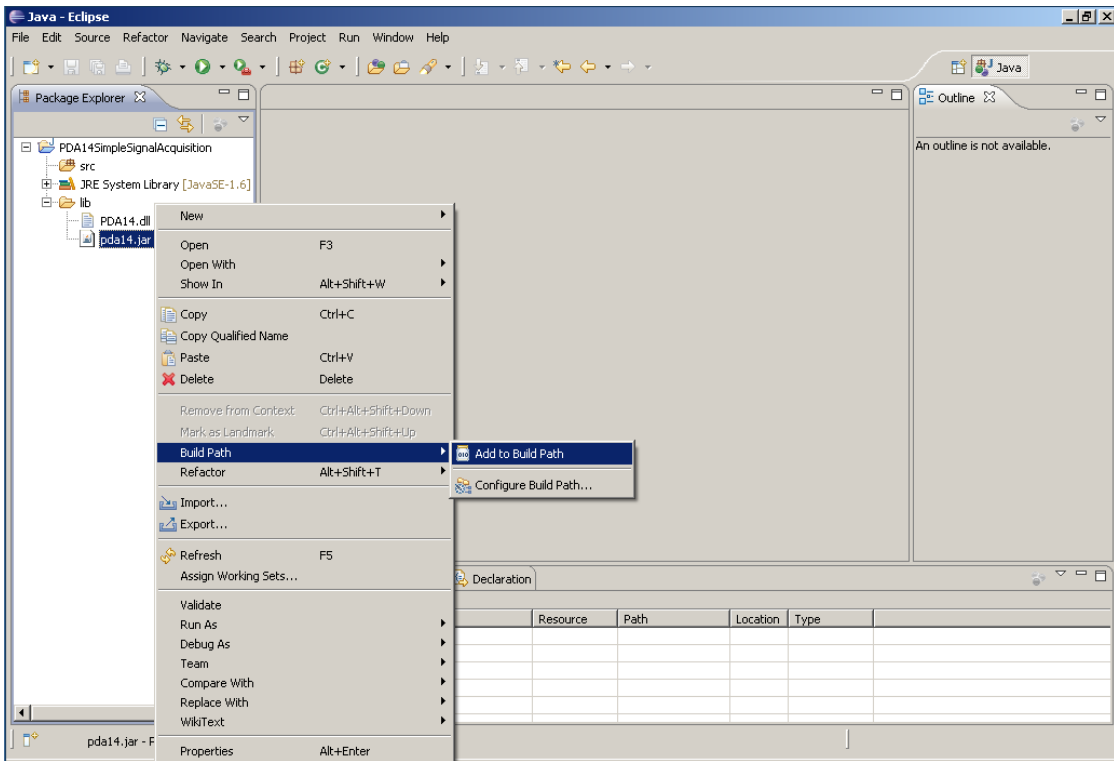


Figura B.22 Añadir dependencias al proyecto

Como se puede observar en la **Figura B.23**, colgando del icono **Referenced Libraries** aparecerá la librería pda14. Si nuestro proyecto tuviera otras dependencias, habría que repetir este proceso para cada una de ellas.

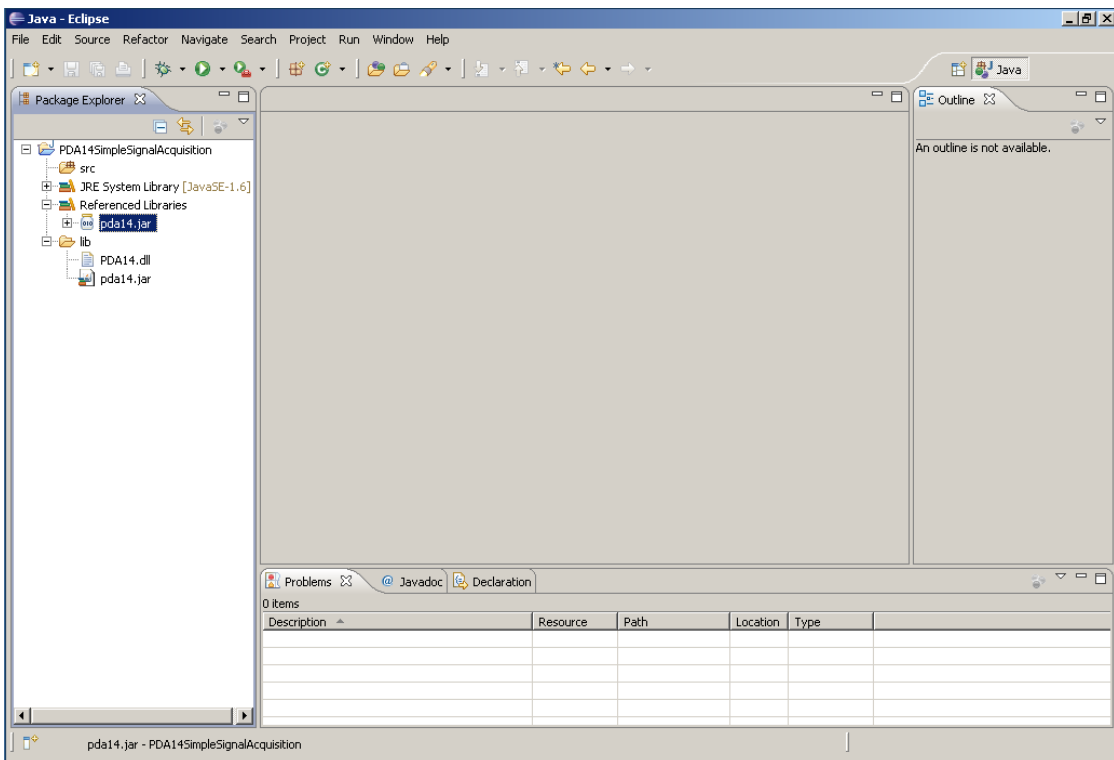


Figura B.23 Explorador de paquetes con las dependencias añadidas

B.2.2. Implementación de la Aplicación

Para construir una aplicación Java, lo primero es crear una clase que contenga un método *main*. Para ello, como se muestra en la **Figura B.24** crearemos una nueva clase en directorio *src* y que se creó automáticamente al crear el proyecto:

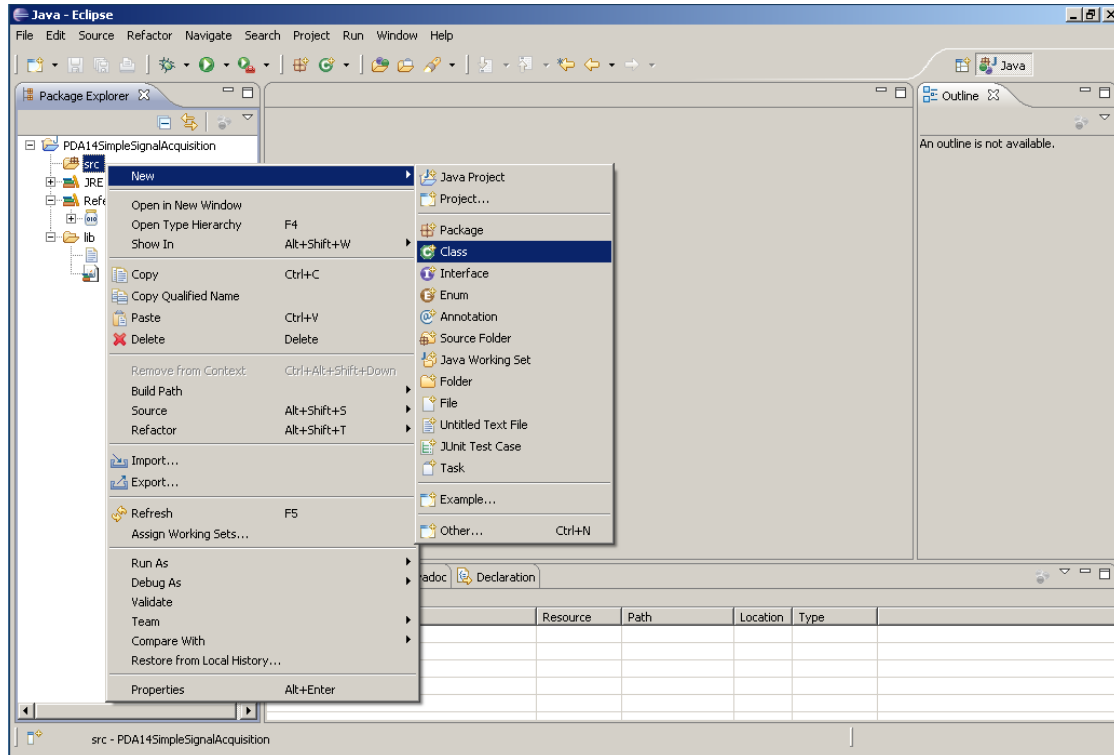


Figura B.24 Añadir nueva clase al proyecto Java

Aparecerá un Asistente para Añadir la nueva clase como el que se muestra en la **Figura B.25**. Se debe indicar el nombre de la clase y el nombre del paquete al que queremos que pertenezca esta clase. Además, podemos indicar si se desea que esta clase contenga un método *main*:

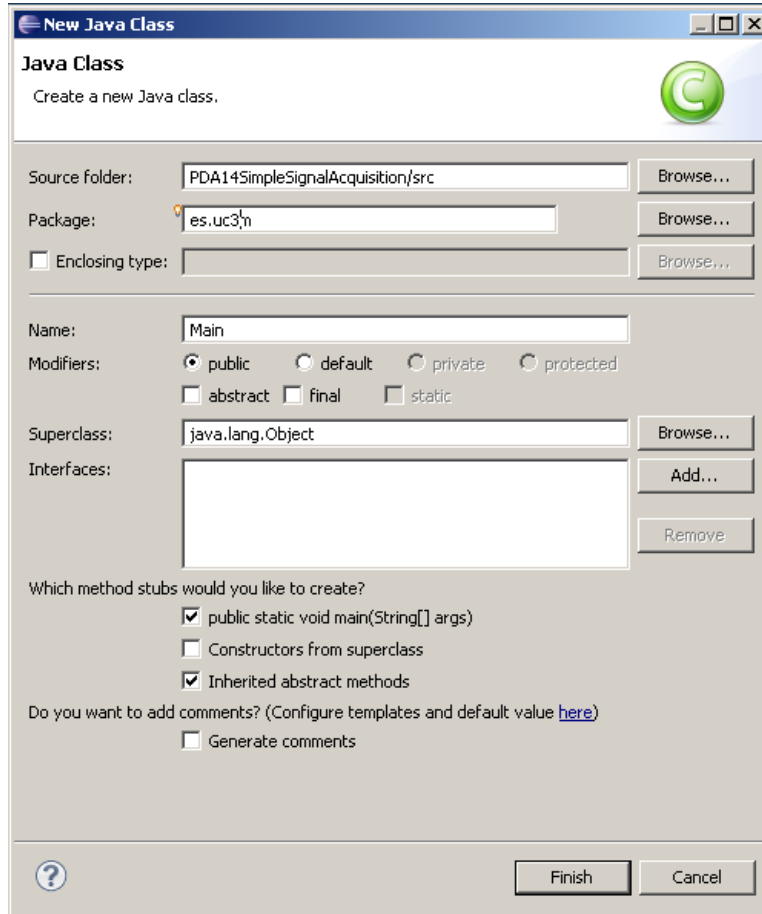


Figura B.25 Asistente de Creación de Nueva Clase

Tras finalizar con el asistente, como se observa en la **Figura B.26**, automáticamente se mostrará en el editor la nueva clase. Ésta contendrá la estructura típica de una clase Java con la definición del método *main* ya que fue indicado anteriormente:

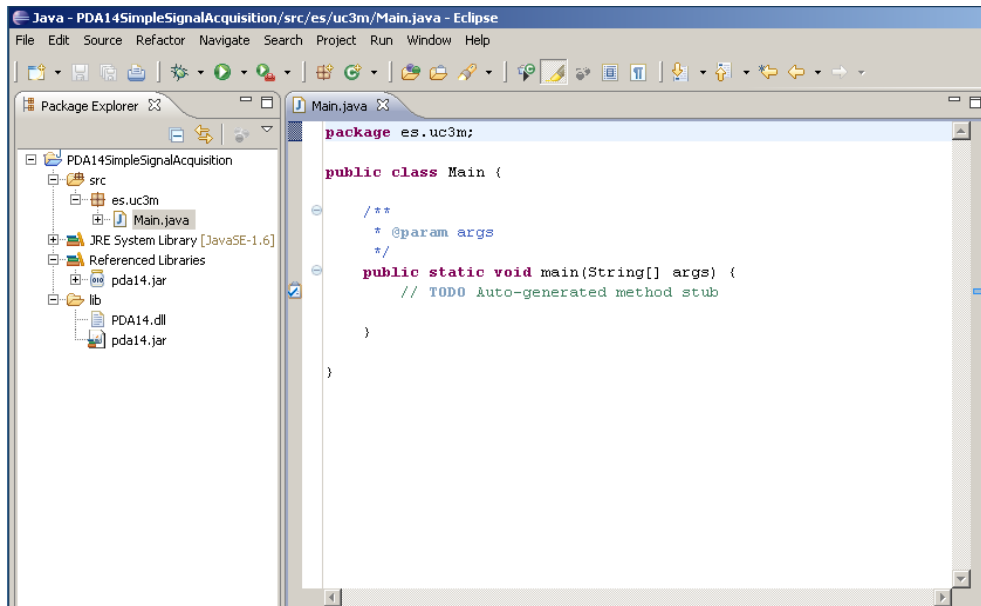
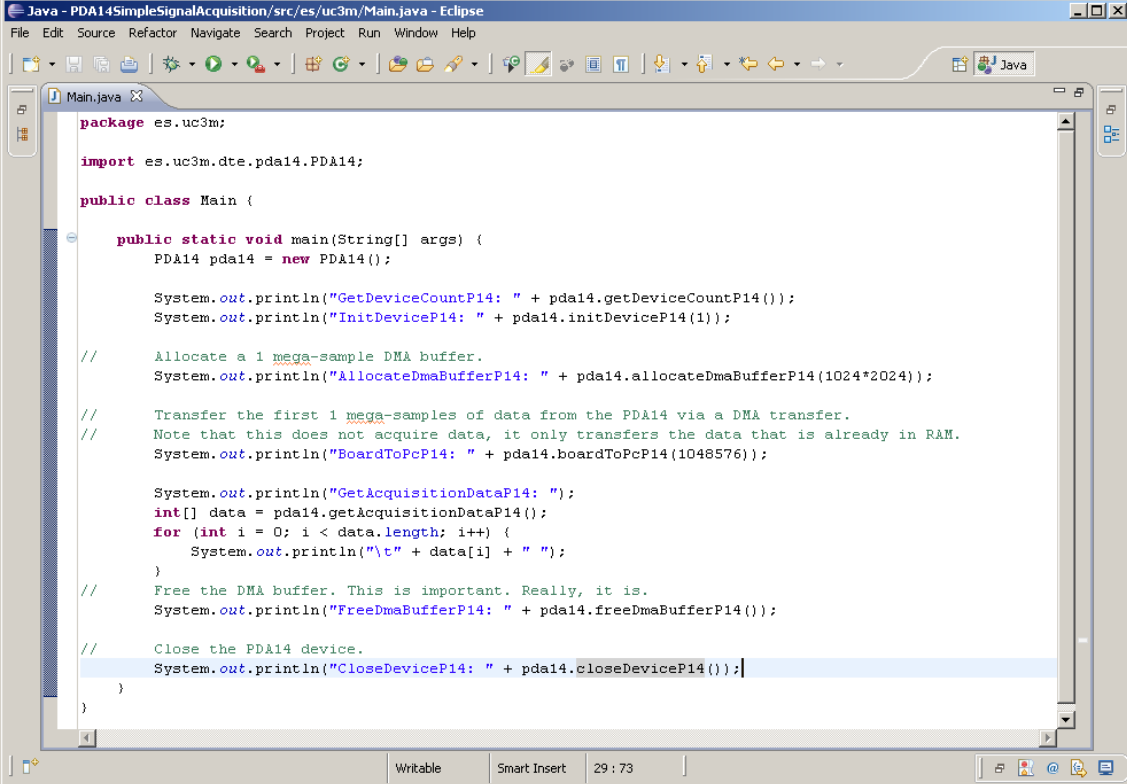


Figura B.26 Editor del Elipse IDE

El siguiente paso consiste en completar el método *main* con la implementación de una configuración para la PDA14 que permita interactuar con ella. Un ejemplo sencillo es el que se muestra en la **Figura B.27**.



```
Java - PDA14SimpleSignalAcquisition/src/es/uc3m/Main.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help

Main.java
package es.uc3m;

import es.uc3m.dte.pda14.PDA14;

public class Main {

    public static void main(String[] args) {
        PDA14 pda14 = new PDA14();

        System.out.println("GetDeviceCountP14: " + pda14.getDeviceCountP14());
        System.out.println("InitDeviceP14: " + pda14.initDeviceP14(1));

        // Allocate a 1 mega-sample DMA buffer.
        System.out.println("AllocateDmaBufferP14: " + pda14.allocateDmaBufferP14(1024*2024));

        // Transfer the first 1 mega-samples of data from the PDA14 via a DMA transfer.
        // Note that this does not acquire data, it only transfers the data that is already in RAM.
        System.out.println("BoardToPcP14: " + pda14.boardToPcP14(1048576));

        System.out.println("GetAcquisitionDataP14: ");
        int[] data = pda14.getAcquisitionDataP14();
        for (int i = 0; i < data.length; i++) {
            System.out.println("\t" + data[i] + " ");
        }
        // Free the DMA buffer. This is important. Really, it is.
        System.out.println("FreeDmaBufferP14: " + pda14.freeDmaBufferP14());

        // Close the PDA14 device.
        System.out.println("CloseDeviceP14: " + pda14.closeDeviceP14());
    }
}

Writable Smart Insert 29 / 73
```

Figura B.27 Aplicación Java de Adquisición con la PDA14

El código fuente de esta aplicación es el siguiente:

```
package es.uc3m;

import es.uc3m.dte.pda14.PDA14;

public class Main {

    public static void main(String[] args) {
        PDA14 pda14 = new PDA14();

        System.out.println("GetDeviceCountP14: " +
            pda14.getDeviceCountP14());
        System.out.println("InitDeviceP14: " + pda14.initDeviceP14(1));

        // Allocate a 1 mega-sample DMA buffer.
        System.out.println("AllocateDmaBufferP14: " +
            pda14.allocateDmaBufferP14(1024*2024));

        // Transfer the first 1 mega-samples
        // of data from the PDA14 via a DMA transfer.
        // Note that this does not acquire data,
        // it only transfers the data that is already in RAM.
        System.out.println("BoardToPcP14: " +
            pda14.boardToPcP14(1048576));

        System.out.println("GetAcquisitionDataP14: ");
        int[] data = pda14.getAcquisitionDataP14();
        for (int i = 0; i < data.length; i++) {
            System.out.println("\t" + data[i] + " ");
        }
        // Free the DMA buffer. This is important. Really, it is.
        System.out.println("FreeDmaBufferP14: " +
            pda14.freeDmaBufferP14());

        // Close the PDA14 device.
        System.out.println("CloseDeviceP14: " + pda14.closeDeviceP14());
    }
}
```

B.2.3. Ejecución de la Aplicación

Una vez implementado hay que configurar la ejecución de la aplicación. Para ello, tal y como se muestra en la **Figura B.28**, desde el Menú Run, se puede utilizar el asistente para configurar ejecuciones.

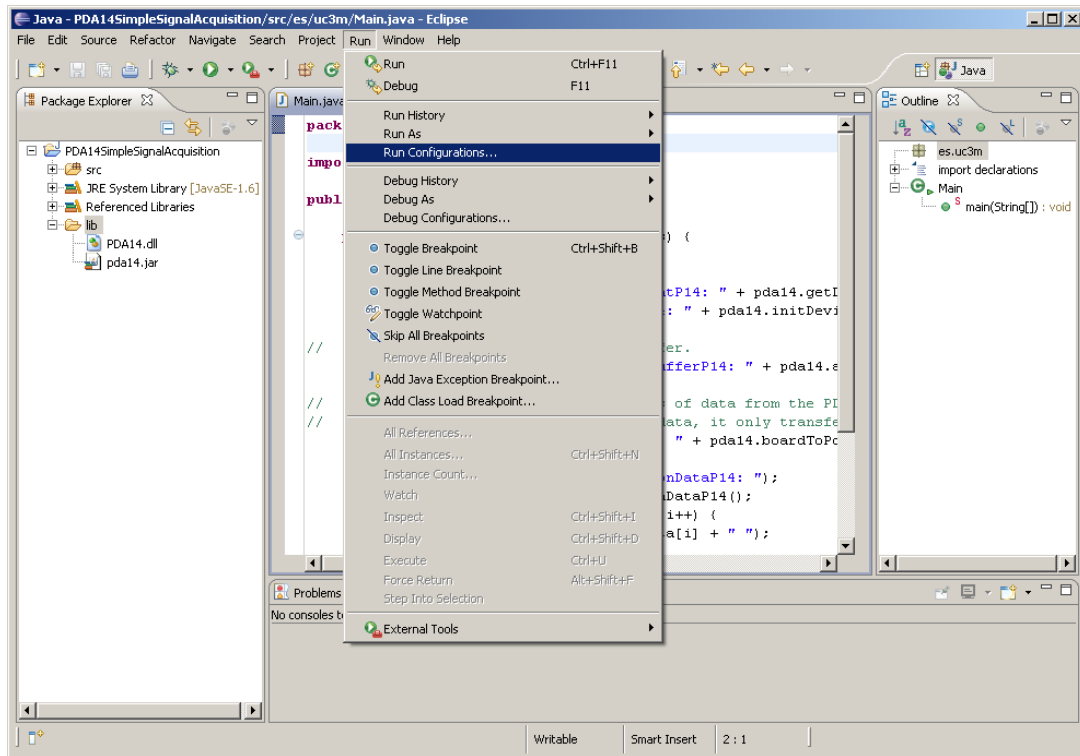


Figura B.28 Arrancar asistentes de configuración de ejecuciones

Como se muestra en la Figura B.29, se añade una nueva aplicación Java.

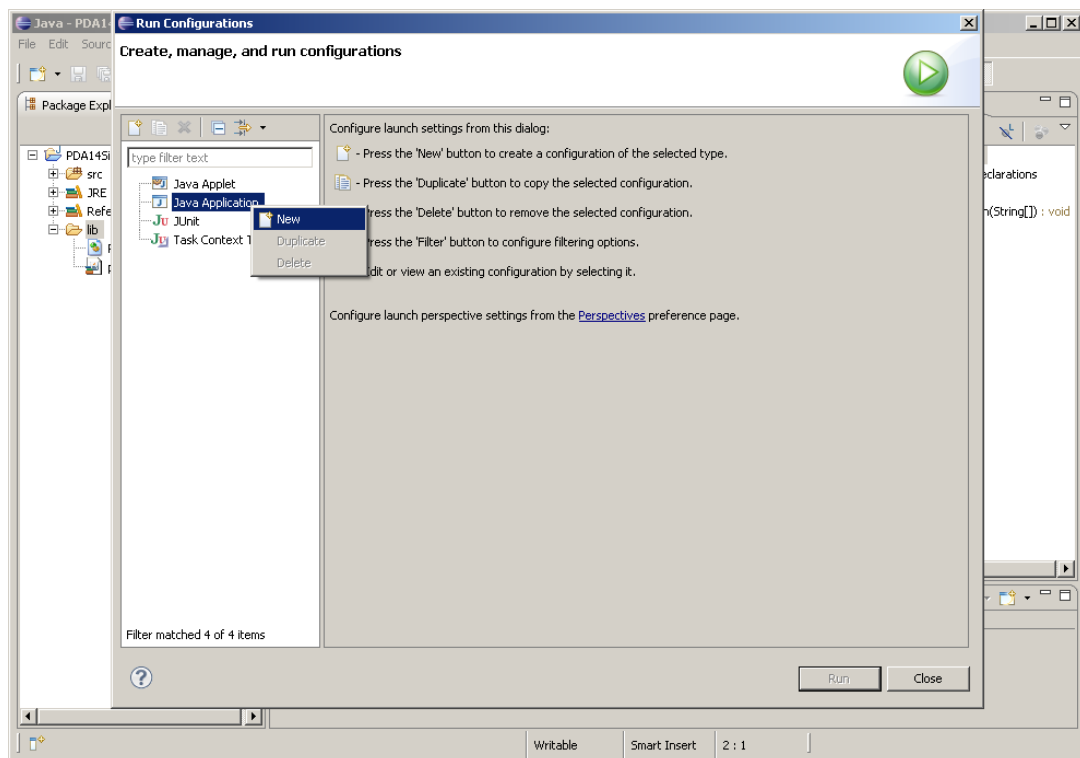


Figura B.29 Añadir nueva aplicación Java

Aparecerá una asistente como el que se muestra en la Figura B.30 y que permite seleccionar el nombre del proyecto y la clase que contiene el método *main* y que es la que se pretende ejecutar.

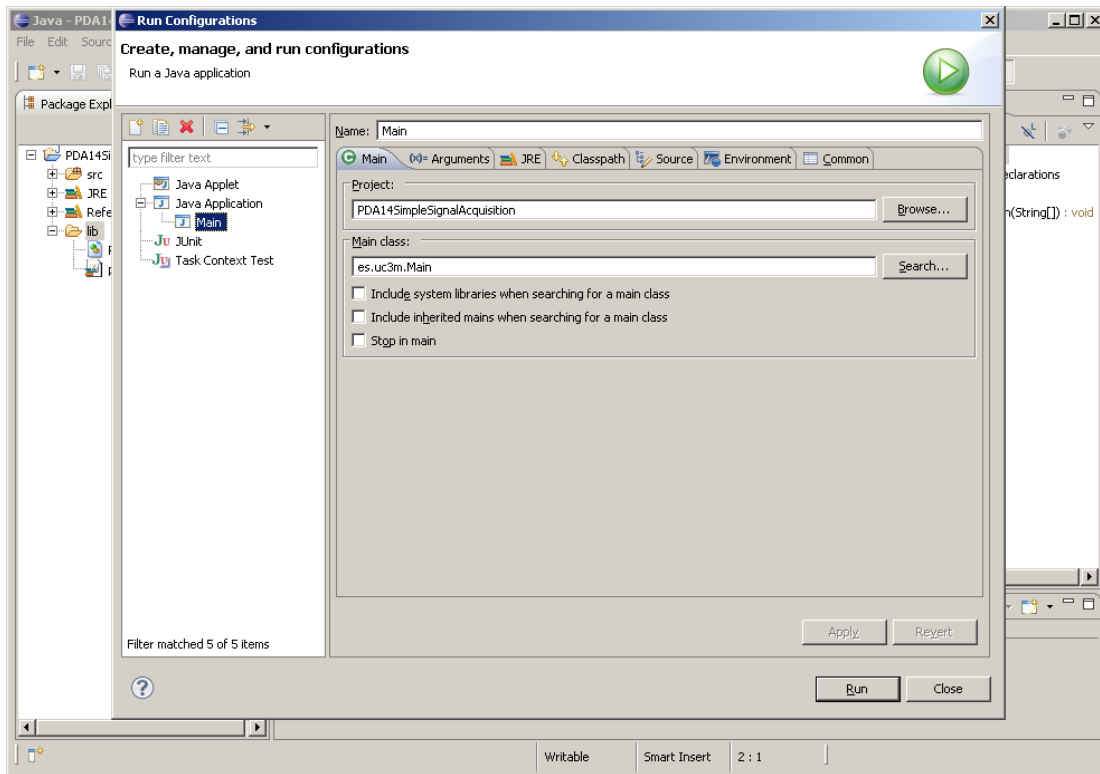


Figura B.30 Configuración de la Ejecución de la Aplicación Java

La aplicación Java que ha sido desarrollada para este ejemplo tiene que tener acceso al fichero PDA14.dll en tiempo de ejecución. Para ello basta con instalar el Software de Adquisición de la PDA14 o añadir la ruta donde se encuentra dicho fichero tal y como se indica en la **Figura B.30**.

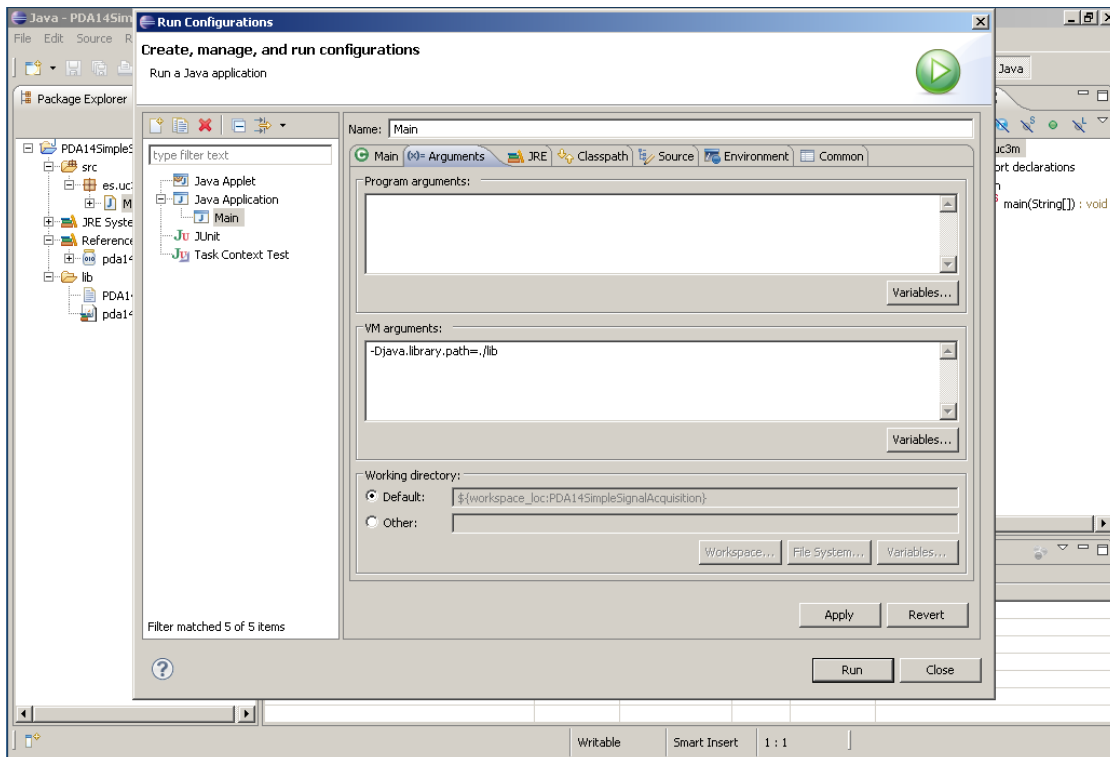


Figura B.31 Argumentos de la ejecución

Así, tras pulsar en Run, el resultado que se obtiene será el que se muestra en la **Figura B.32**.

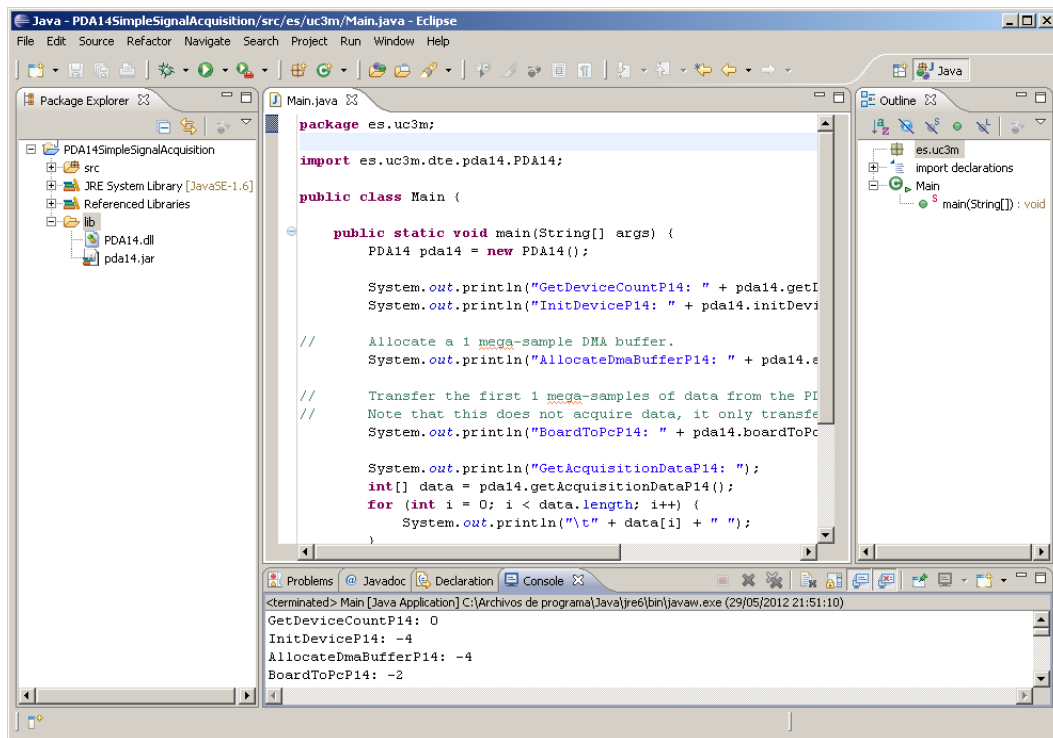


Figura B.32 Ejecución de la Aplicación

Si la aplicación no tuviera acceso a la librería de enlace dinámico, el resultado de la ejecución sería que encontramos en la **Figura B.33**.

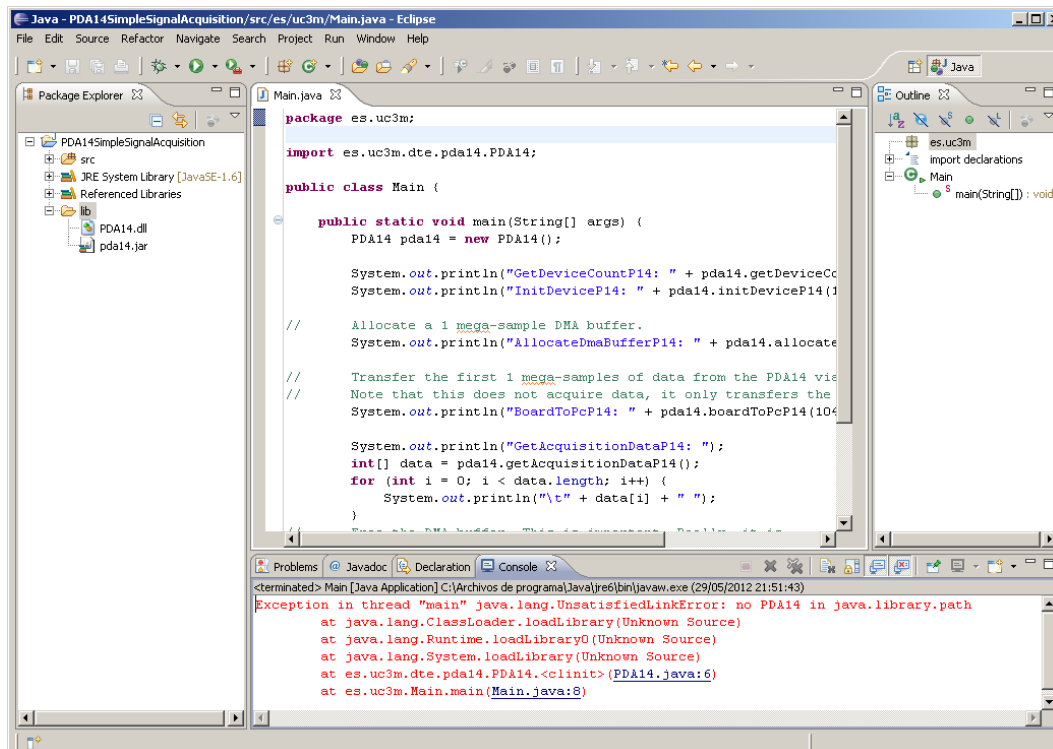


Figura B.33 Ejecución Incorrecta de la Aplicación

Bibliografía

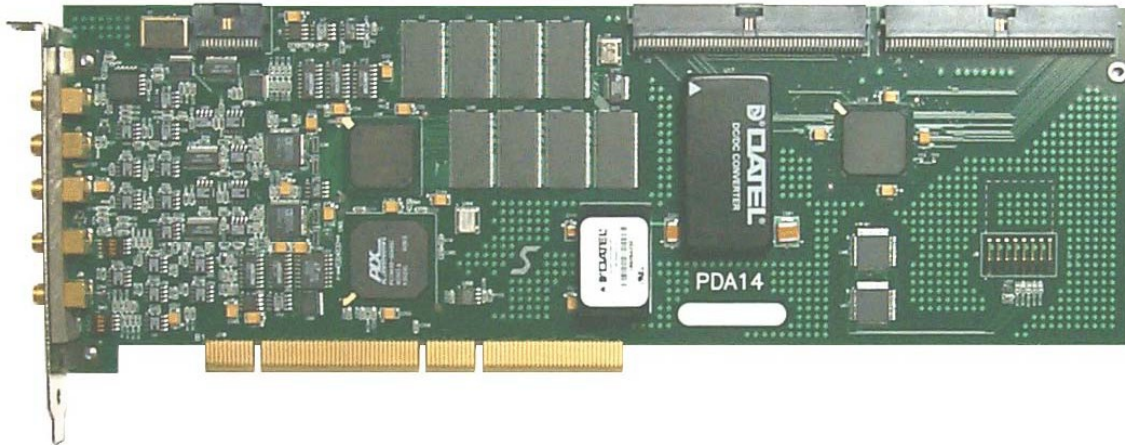
- [1] Roger S. Pressman, "Ingeniería del Software: Un enfoque práctico".
Sexta Edición McGraw-Hill, 2005. 900 p.
ISBN: 9701054733
- [2] Ben Collins-Sussman, Brian W. Fitzpatrick, y C. Michael Pilato,
"Version Control with Subversion For Subversion 1.7".
Recuperado el 15 de Julio de 2010 de
<http://svnbook.red-bean.com/nightly/en/svn-book.pdf>
- [3] Moshe Bar y Karl Fogel "Open Source Development with CVS"
Recuperado el 15 de Julio de 2010 de
http://cvsbook.red-bean.com/OSDevWithCVS_3E.pdf
- [4] William H. Bartley. "Analysis of Transformer Failures".
The Hartford Steam Boiler Inspection & Insurance Co. Hartford, CT USA. 2003
- [5] IEEE Guide for Protective Relay Applications to Power Transformers.
Marzo 2000.
IEEE Std. C37.91-2000.
- [6] Mario Mañana Canteli, "Protección del transformador".
Departamento de Ingeniería Eléctrica y Energética.
Universidad de Cantabria

- [7] Thomas Leibfried, "Online Monitors Keep Transformers in Service".
IEEE Computer Applications in Power. Julio 1998. p. 36 – 42.
- [8] IEEE Guide for the Interpretation of Gases Generated in Oil-Immersed Transformers.
Junio 1991.
IEEE Std C57.104-1991, Institute of Electrical and Electronics Engineers
- [9] IEEE Guide for the Detection and Location of Acoustic Emissions from
Partial Discharges in Oil-Immersed Power Transformers and Reactors.
Agosto 2007.
IEEE Std C57.127-2007 , Institute of Electrical and Electronics Engineers
- [10] Sacha M. Markalous, Stefan Tenbohlen y Kurt Feser,
"Detection and Location of Partial Discharges in Power Transformers using acoustic and
Electromagnetic Signals".
Universidad de Stuttgart (Alemania).
- [11] Carlos Macià Sanahuja
"Sensores interferométricos de fibra óptica para la detección de descargas parciales en
transformadores de potencia".
Proyecto Fin de Carrera. 2002. Departamento de Tecnología Electrónica, Universidad
Carlos III de Madrid
- [12] Signatec, Incorporated
"PDA14 Operator's Manual"
Revision 1.08 -- April 14, 2006
- [13] David Navidad Mencía,
"Sistema Multiplexado de Adquisición de Señales Ultrasónicas con Aplicación en
Emisión Acústica de Descargas Parciales"
Proyecto Fin de Carrera. 2009. Departamento de Tecnología Electrónica, Universidad
Carlos III de Madrid
- [14] Sitio Web Oficial del TJ-II
http://www-fusion.ciemat.es/New_fusion/es/TJII/
Accedido el 20 de Mayo de 2012
- [15] Sitio Web Oficial de ITER
<http://www.iter.org/>
Accedido el 20 de Mayo de 2012

- [16] Oracle Corporation “The Java EE 6Tutorial”
<http://download.oracle.com/javaee/6/tutorial/doc/javaeetutorial6.pdf>
Accedido el 15 de Julio de 2010
- [17] Sitio Web Oficial de Jetty
<http://jetty.codehaus.org/jetty/>
Accedido el 20 de Mayo de 2012
- [18] Core J2EE Patterns - Data Access Object
<http://java.sun.com/blueprints/corej2eepatterns/Patterns/DataAccessObject.html>
Accedido el 20 de Mayo de 2012
- [19] Sheng Liang, “Java Native Interface: Programmer's Guide and Specification: Programmer’s Guide and Specification”.
Primera Edición Addison-Wesley, 1999. 320p.
ISBN: 0201325772
- [20] Sitio Web Oficial de Spring
<http://www.springsource.org/>
Accedido el 20 de Mayo de 2012
- [21] Sitio Web Oficial de JFreeChart
<http://www.jfree.org/jfreechart/>
Accedido el 20 de Mayo de 2012
- [22] The MathWorks, Inc. “MAT-File Format”
http://www.mathworks.com/help/pdf_doc/matlab/matfile_format.pdf
Accedido el 20 de Mayo de 2012
- [23] National Instruments Corporation, “Specification for the LabVIEW Measurement File (.lvm)”
<http://www.ni.com/white-paper/4139/en>
Accedido el 20 de Mayo de 2012

Hojas de Características

- ❖ ***Signatec PDA14***
- ❖ ***Mistras Acoustic Emission Sensor - R15I-AST Sensor***
- ❖ ***Hydrophones Brüel & Kjaer - Types 8103, 8104, 8105 and 8106***



FEATURES

- 2 Channels at up to 100 MHz Sample Rate
- 14 Bits of Resolution
- Bandwidth from DC-50 MHz
- 512 Megabytes of On-Board Memory
- 500 MB/s Transfer via Signatec Auxiliary Bus (SAB)
- 266 MB/s Transfer (Peak) Over PCI Bus
- 64/32 Bit PCI Plug and Play Compatible Board

APPLICATIONS

- Radar
- Mass Spectroscopy
- Mass Spectrometry – Time of Flight
- Communications
- Ultrasound
- Medical Diagnostics / Non Destructive Testing
- Laser Doppler Velocimetry
- High Speed Waveform Capture

OVERVIEW

The PDA14 is a dual channel waveform capture board which provides a tremendous combination of high speed and high resolution along with an extremely large memory capacity. The entire 512 MB memory may be used as a giant FIFO for acquiring data directly to either the SAB or PCI bus. Tests have shown that a very large FIFO memory is required to prevent data loss when performing continuous data streaming over the PCI bus.

The PDA14 is a 64-bit PCI compatible board equipped with standard 'Plug and Play' features common in PCI systems. It is capable of Bus Master DMA data transfers at up to a peak of 266 megabytes/second and a sustained rate of 250 megabytes/second.

It can also operate in 32-bit PCI slots in which case the transfer rate will be limited to about 125 megabytes per second.

The PDA14 incorporates the advanced Signatec Auxiliary Bus (SAB) that allows for data transfers of up to 500 megabytes/second. This allows for the high-speed transfer of data to fast processor boards, such as Signatec's PMP8A, or other peripherals. The SAB also incorporates device control features for operating the PDA 14 independent of the host bus.

The PDA14 is equipped with an interconnect port to allow multiple boards to be interconnected in a Master/Slave configuration. Up to three Slave boards may be operated with one Master. Master/Slave connections are via a ribbon cable that connects at the top of the board. In this configuration the clock and trigger signals from the Master drive the Slave boards so that data sampling on all boards occurs simultaneously.

The PDA14 has six software selectable signal amplitude ranges from a maximum of 3.0 volts down to 200 millivolts full scale.

External clock and trigger signals are provided via SMA connectors on the back bracket. Also provided is a user selectable digital output signal for synchronization purposes. Effectively, twenty-two internal clock frequencies may be selected, from 100 MHz down to 97.7 kHz in factors of 2 or from 62.5 MHz down 61.0 kHz in factors of 2. The PDA14 supports single shot, segmented, and pretrigger triggering modes.

HARDWARE DESCRIPTION

The figure on the next page shows a simplified mechanization for the PDA14. The input signals may be set for either DC or AC coupling. The signal conditioning provides six voltage ranges from 200 millivolts to 3.0 volts peak-to-peak full scale. A low pass filter sets the channel bandwidth to 50 MHz. For test purposes, the ADC input may be fed a sine wave test signal (not shown) for measuring the AC performance at 25 MHz.

The data input to RAM FIFO 1 can be considered to be two separate 16-bit data streams. The latches and multiplexers at the output of the ADCs allows for capturing data from both channels or from channel 1 only or channel 2 only.

The Pretrigger Samples Shift Registers are programmable in length up to 4k samples. They can be used to capture pretrigger samples in either the Single Shot or Segmented trigger modes. Before a trigger is received, data is written into the shift registers but goes no further. After receiving a trigger, data samples start to be written into RAM FIFO 1. See the section "Trigger Modes and Options" for trigger mode details.

The ADCs always operate at either 100 MHz or 62.5 MHz when the internal clock is used or else at the external clock frequency. Operation at reduced rates is accomplished by dropping out the appropriate data bytes from the data stream. Thus the effective sample rate is divided from 2 to 1024 in factors of 2.

Data is written into the SDRAM via FIFO1 and read from RAM via FIFO2. The RAM operates at a clock rate of 133 MHz so it has a bandwidth capability of slightly greater than 500 MB/s. When data is being acquired at a rate of 250 MB/s or less (62.5 MHz on 2 channels or 100 MHz on single channel), it is possible to operate the board in Buffered Acquisition mode. In this mode the RAM is operated as a very large FIFO for acquiring data directly to the PCI bus or SAB.

The PDA14 has 12 operating modes as follows:

- Standby
- Acquisition to RAM
- Acquisition to PCI Bus
- Buffered Acquisition to PCI bus (RAM as FIFO)
- Acquisition to SAB
- Buffered Acquisition to SAB (RAM as FIFO)
- Data Transfer, RAM to PCI bus
- Data Transfer, RAM to SAB
- Write RAM (from PCI bus)
- Write RAM (from SAB)
- Read Time Stamps (from PCI bus)
- Read Time Stamps (from SAB)

The RAM write mode is typically used to test the on-board memory by writing data via the PCI bus and reading back via a RAM-to-PCI transfer.

External Inputs/Outputs

Besides the signal input, the PDA14 also provides SMA connections for a clock input, a trigger input, and a digital output signal. The clock input can be selected as the ADC sample clock. The external trigger can be used to synchronize the start of data acquisition with an external event. Trigger parameters such as trigger level, slope, etc. are user programmable. The digital output is a user selectable signal. The list of selections is TBD.

Trigger Modes and Options

In data acquisition mode three triggering modes are available: single shot, segmented, or pretrigger. In the single shot mode, following the detection of a trigger signal, all of the active memory is filled. In the segmented mode a separate trigger signal is required to successively fill each memory segment until all of the active memory is filled. In the pretrigger mode the board is armed and continuously fills the entire active memory until a stop trigger is detected, after which a programmed number of post trigger samples are taken before acquisition is terminated. The pretrigger mode may be used to see signal information both before and after the trigger signal.

Samples Settings

There are several board settings that affect the quantity and method of acquiring samples.

Active Memory Size – In the "post-trigger modes" this is the number of samples that will be taken after which the memory will be considered "full" and the acquisition is terminated. When a full condition is detected, a flag is set which may be read by the PC or software selected to cause a PC interrupt or send an interrupt over the SAB. The amount of memory that is activated for data acquisition may be set from 8 bytes to the full 512 megabytes in steps of 8 bytes.

Segment Size – In Segmented Mode this is the number of samples that will be taken each time a valid trigger signal is detected.

Pretrigger Samples – In Single Shot or Segmented Modes, this is the number of samples that will be recorded into RAM that occurred before the trigger.

Delayed Trigger – This sets a delay between the actual applied trigger and the effective trigger for the board. The delay range is from 0 to 64k digitizer clock cycles. In Pretrigger Samples mode the delayed trigger setting establishes the number of post-trigger samples that will be recorded.

Time Stamps

In Segmented Mode "time stamps" allow for storing the time relationship between the memory segments. Time Stamps are 32 bit timer values with a clock resolution of 7.5 nanoseconds. Up to 2048 time stamps are accumulated in memory separate from the data. Time stamps are read after the acquisition is completed.

SAB Operation

The PDA14 can perform SAB data transfers at 64 bits, or at 32 bits over either the high (SABH) or low (SABL) bus ports. This provides flexibility when multiple boards are incorporated into a system. At 64 bits the maximum transfer rate is 500 MB/s.

PCI Operation

The PDA14 is capable of sustaining a long-term data-transfer rate, over the PCI bus, of 250 megabytes per second when installed in a 64 bit PCI slot. It can also be installed in traditional 32-bit slots in which case the maximum rate is about 125 megabytes per second.

SOFTWARE, SYSTEM, AND PERFORMANCE DETAILS

Software

The PDA14 is supplied with the following software:

- Windows NT/2000/XP and Linux Drivers
- C Function Library with source code
- PDA14 Class (C++) that wraps the Library functions and handles the tedious initializations and basic operations.
- Software manual that describes how to use the available library of functions to create larger applications or systems.
- A board diagnostics self test program.
- Multiple Coding examples
 - ✓ Acquire with DMA transfer to PC
 - ✓ Digital oscilloscope interface software
 - ✓ Using multiple PDA14 boards (with Master/Slave support)
 - ✓ SAB data transfer examples

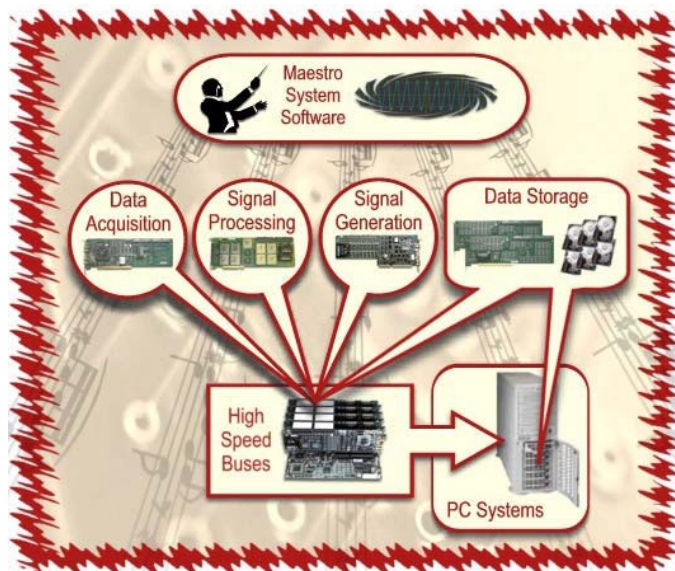
Maestro support for the PDA14

- Menu driven board settings for multiple PDA14 boards
- Using the PDA14 as a high-speed recording system
- Using the PDA14 as a high-speed recording system with the PMP8A (or any other SAB Active Slave capable device)
- Data Analysis tools for poking/peeking onboard memory

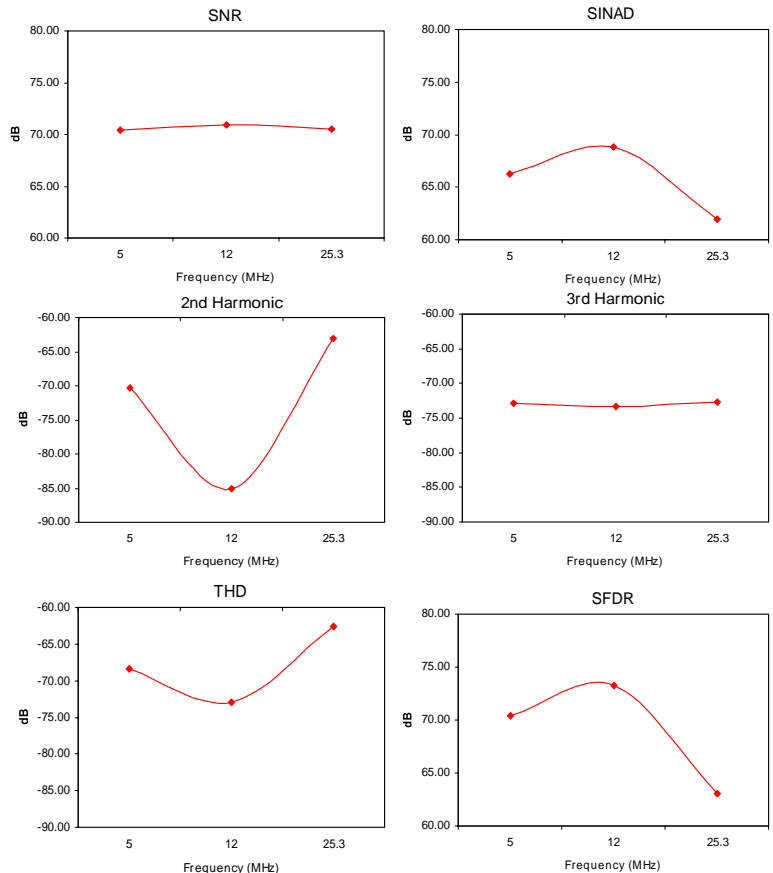
System Capabilities

The system solution offered by Signatec is based on supplying a comprehensive range of products incorporating the Signatec Auxiliary Bus. This 64-bit bus provides transfer rates up to 500 MB/s. SAB boards act as modular building blocks for constructing high performance systems that mechanize a wide variety of applications. Shown in the figure below are elements of such a system. Many systems can be constructed using standard desktop PC's. For demanding applications Signatec can supply a total turnkey system utilizing one of our industrial computer systems.

Devices connected to the SAB may communicate via SAB interrupt and control lines. This allows the boards to accomplish multiple acquisition, transfer, and processing cycles under control of the signal processor device, without PC intervention. Bypassing the host bus and operating system can significantly improve system performance.



Typical Performance (dB versus Frequency in MHz)



DEFINITION OF TERMS

SINAD: Signal to Noise and Distortion: The ratio of the fundamental sinusoidal signal power to the total noise and distortion component power. In other words this is the ratio of the fundamental signal power to the measured power from the remainder of the detectable spectrum from dc to 50 MHz.

SNR: Signal to Noise Ratio: The ratio of the fundamental sinusoidal signal power to the noise power. For this data sheet noise is considered to be the power from all spectral components except for the fundamental signal, the first harmonic, and the second harmonic.

THD: Total Harmonic Distortion: The ratio of the total power of the second and third harmonics to the fundamental sinusoidal power.

Second Harmonic Distortion: The ratio of the power at twice the fundamental frequency to the power of the fundamental sinusoid.

Third Harmonic Distortion: The ratio of the power at three times the fundamental frequency to the power of the fundamental sinusoid.

SFDR: Spurious Free Dynamic Range: The ratio of the fundamental sinusoidal power to the power of the next highest spurious signal. Normally the highest spurious signal is the second or third harmonic.

TEST METHOD

A filtered sine wave signal is applied to the channel 1 and channel 2 inputs. Test frequencies used are 5.0, 12.0, and 25.3 MHz. The digitizer clock setting is 100 MHz. The voltage range is 600mV. Signal amplitude is set for 95% of full scale. Performance measurements are made using a 4096 point FFT with a Blackman-Harris window. Signatec uses the first 10 bins to represent the DC term, 64 bins centered around the peak for the fundamental signal power, 9 bins centered at twice the fundamental for the second harmonic and 9 bins centered at three times the fundamental for the third harmonic. All other bins are considered to be noise. (NOTE: large number of bins for the fundamental is necessary so that energy in the side lobes of the window function is not misinterpreted as noise or spurs.)

PDA14 SPECIFICATIONS AND ORDERING INFORMATION

Electrical Specifications

External Signal Connections(SMA)

Analog Input, Channel 1
Analog Input, Channel 2
Clock Input
Trigger Input
Digital Output

Analog Inputs

Full Scale Volt. Ranges: 200mV, 333mV, 600mV, 1.00V, 1.66V, 3.00V
Impedance : 50 ohms
Bandwidth : 50 MHz
Equivalent Noise : 0.5 lsb RMS (typical)
Coupling : AC or DC¹

External Trigger

Impedance : 1k ohms
Trigger Level : ± 1.75 Volts
Adjustment Method : via 12 bit DAC
Bandwidth : 50 MHz
Coupling : DC

External Clock

Signal Type : sine wave or square wave
Coupling : AC
Impedance : 50 ohms
Frequency : 30 MHz to 100 MHz
Amplitude : 100 mV p-p to 2.0 V p-p

Digital Output

Type : TTL Logic Level
Max. Frequency : 100 MHz
Suggested Load : 1k ohms
Amplitude : TTL

DC Offset Voltage

Resolution: : 12 bit DAC
Range : $\pm 1.1 \times$ FS ADC input

Digitizer

Voltage Range : 2.0V p-p full scale
Resolution : 14 bits
Linearity, Integral : ± 0.5 lsb max.
Linearity, Differential : ± 0.75 lsb max.
Aperture Jitter : < 0.5 pS typical

Internal Clock

Available Rates : 100 MHz down to 97.6 kHz in factors of 2
62.5 MHz down to 61.0 kHz in factors of 2
Accuracy : $\pm 0.01\%$

Trigger Modes

Post Trigger : single start trigger fills active memory
Pre-trigger : single trigger stops acquisition
Segmented : start trigger for each memory segment

Trigger Options

Pre-trigger Samples : samples prior to trigger are stored; Single Channel: 8k max.; Dual Channel: 4k max per channel
Delayed Trigger : delay from trigger to data storage; Up to 64k digitizer clock cycles

Memory

Active Size : Up to 256 MegaSamples
Segment Size : Up to 128 Megasamples
Start Address Setting : Anywhere in memory
Segment re-arm time² : 150 nanoseconds
Addressing : DMA transfer from starting address
Memory Address (PC) : Plug and Play selected

I/O Addressing

PCI Controller Address : 64 bytes, Plug and Play selected
Control/Status Registers: 32 bytes, Plug and Play selected

Signatec Auxiliary Bus

Data Transfer Modes : Block or Packet
Data Transfer Rates : 500 MB/s max @ 64 bits
Data Direction : output only

Power Requirements

+12V : 400mV Amps max.
+5V : 1.5 Amps max.
+3.3V : 2.3 Amps max.

Absolute Maximum Ratings

Analog Inputs : ± 5 volts
Trigger Input : ± 5 volts
Clock Input : 5 volts peak to peak
Ambient Temperature : 0 to 50°C

Ordering Information

PDA14 Board

Part Number: PDA14

SAB Cables

Refer to the "SAB Cable Assembly Ordering Guide" to select and order the appropriate cable assemblies.

Master-Slave Cables

The PDA14 may be software configured to operate as a Master or a Slave in a multiple board system. In order to operate in a Master/Slave configuration a 20-pin ribbon cable is required to connect the boards. This cable is ordered using the basic part number PDA14MS-X where X is the total number of boards connected together. Master/Slave boards must occupy adjacent slots. The maximum number of boards to be connected is one master and three slaves.

Documentation & Accessories

The PDA14 is supplied with a comprehensive operator's manual, which thoroughly describes the operation of both the hardware and the software. Also supplied are two four-foot coaxial cables with BNC to SMA connectors. Extra cables may be purchased from Signatec. Supplied software disks contain a function library for Microsoft Visual C/C++, example programs, and all source code to libraries and examples.

Customer Support

The Signatec Web Site: www.signatec.com is the primary access point for software updates, documentation updates, or technical support. For the best technical support it is very important to follow the instructions on the technical support page.

Product Warranty

All Signatec products carry a full 3-year warranty. During the warranty period, Signatec will repair or replace any defective product at no cost to the customer. This warranty does not cover customer misuse or abuse of the products or physical damage not reported within 15 days of the time of shipment by Signatec.

Notes:

1. Selected via dip switch.
2. In segmented mode, time from the end of a segment until a trigger will be accepted to begin another segment acquisition.

Signatec reserves the right to make changes in this specification at any time without notice. The information furnished herein is believed to be accurate, however no responsibility is assumed for its use. Data Sheet Date 02-25-2004



R15I-AST Sensor

Integral Preamplifier

Acoustic Emission Sensor

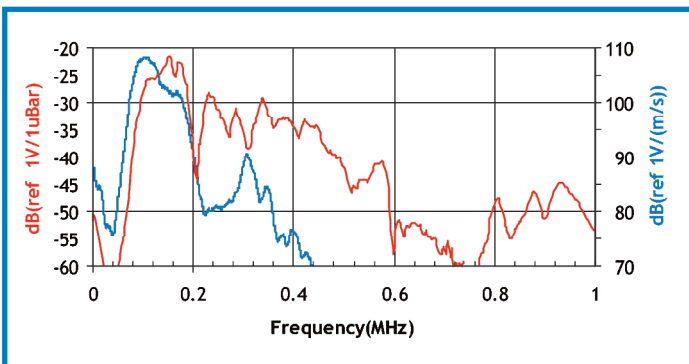
Description and Features

PAC's integral preamp sensors were specifically engineered to attain high sensitivity and have the capability to drive long cables without the need for a separate preamplifier. Incorporating a low-noise input, 40 dB preamplifier and a filter all inside the sensor housing, these transducers are completely enclosed in metal stainless steel (or aluminum) housings that are treated to minimize RFI/EMI interference. Care has also been taken to thermally isolate the critical input stage of the preamplifier in order to provide excellent temperature stability over the range of -35° to 75° C.

Their integrated Auto Sensor Test (AST*) capability allows these sensors to pulse as well as receive. This feature lets you verify the sensor coupling and performance at any time throughout the test.

Applications

This general purpose sensor provides a good mix of high sensitivity and high low frequency rejection. These properties make it very useful for monitoring common structures such as pipelines, vessels, bridges, and storage tanks in petroleum, refineries, chemical plants, offshore platforms, as well as factory and process monitoring applications. It is PAC's most popular and highest volume selling sensor.



Frequency response of the R15I-AST. Calibration based on ASTM E1106; Calibration based on ASTM E976.

Operating Specifications

Dynamic

Peak Sensitivity, Ref V/(m/s)	109 dB
Peak Sensitivity, Ref V/ μ bar	-22 dB
Operating Frequency Range	80-200 kHz
Resonant Frequency, Ref V/(m/s).....	75 kHz
Resonant Frequency, Ref V/ μ bar	150 kHz
Directionality.....	+/-1.5 dB

Environmental

Temperature Range	-35 to 75°C
Shock Limit	500 g
Completely shielded crystal for maximum RFI/EMI immunity	

Physical

Dimensions.....	1.13" diameter x 1.23" h(29 x 31 mm)
Weight.....	70
Case Material.....	Stainless Steel (304)
Face Material.....	Ceramic
Connector.....	BNC
Connector Locations	Side

Ordering Information and Accessories

R15I	R15I-AST
Cable (specify cable length)	1234 - X
Magnetic Hold-Down	MHR15I
Amplifier	AE2A

Sensors include

NIST Calibration Certificate & Warranty

* AST – Auto Sensor Testing feature allows AE systems to control the sensor as a pulser and a receiver at the same time. It can therefore characterize its own condition as well as send out a simulated acoustic emission wave that other sensors can detect, so the condition of the nearby sensors also can be tested.

Product Data

Hydrophones — Types 8103, 8104, 8105 and 8106

USES:

- Calibration reference standard
- Ultrasonic measurements in liquids
- Cavitation measurements
- Laboratory and industrial measurements in liquids and gases
- As underwater projectors (8103, 8104 and 8105)
- Noise measurements in humid and polluted atmospheres

FEATURES:

- Frequency ranges from 0.1 Hz to 180 kHz
- Individually calibrated; traceable to NIST
- Flat frequency response over wide range
- Omnidirectional over wide frequency range
- Working pressures up to 9.8×10^6 Pa (100 atm. [1000 m])
- Shielded-element construction
- Highly corrosion resistant

ADDITIONAL FEATURES 8103:

- Very small size (50×9.5 mm)
- -3 dB limit in air at 15 kHz
- Double-shielded low-noise integral cable

ADDITIONAL FEATURES 8105:

- Omnidirectional over full frequency range
- No metallic parts exposed
- -3 dB limit in air at 7 kHz

ADDITIONAL FEATURES 8106:

- High sensitivity: -174 dB re 1 V/ μ Pa
- Built-in preamplifier with provision for insert-voltage calibration
- Durable construction
- Equivalent noise level well below sea-state zero

The Brüel & Kjær range of hydrophones is a range of individually calibrated waterborne-sound transducers which have a flat frequency response and are omnidirectional over a wide frequency range. Their construction is such that they are absolutely waterproof and have good corrosion resistance. There are four types of Brüel & Kjær Hydrophone:

Type **8103** is suitable for laboratory and industrial use and particularly for the acoustic study of marine animals or for cavitation measurements.

Type **8104** is ideal for calibration purposes.

Type **8105** is a robust spherical hydrophone usable down to 1000 m ocean depth with excellent directional characteristics, being omnidirectional over 270° in the axial plane and 360° in the radial plane.

Type **8106** has a built-in amplifier which gives a signal suitable for transmission over long underwater cables. It is usable down to 1000 m ocean depth.



The Brüel & Kjær range of water-borne-sound transducers consists of the following:

Hydrophone Type 8103. A small-size, high-sensitivity transducer for making absolute sound measurements over the frequency range 0.1 Hz to 180 kHz with a receiving sensitivity of -211 dB re 1 V/ μ Pa. It has a high sensitivity relative to its size and good all-round characteristics which make it generally applicable to laboratory, industrial and educational use. The 8103's high-frequency response is especially valuable when making acoustic investigations of marine animals and in the measurement of the pressure-distribution patterns in ultrasonic-cleaning baths. It is also useful for cavitation measurements. Fig. 1 indicates the major features of the 8103.

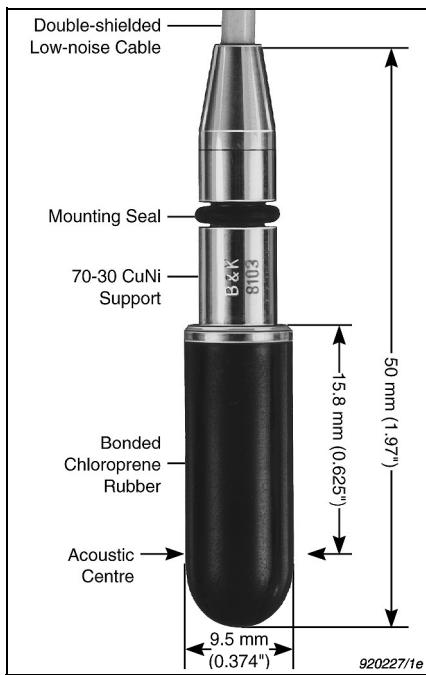


Fig. 1 Hydrophone Type 8103

Hydrophone Type 8104. A wide-range standard measuring transducer for making absolute sound measurements over the frequency range 0.1 Hz to 120 kHz with a receiving sensitivity of -205 dB re 1 V/ μ Pa. It can also be used as a sound transmitter (projector) which makes it ideal for calibration purposes by the reciprocity, calibrated-projector and comparison methods. The main features of the 8104 are shown in Fig. 2.

Hydrophone Type 8105. A small, spherical transducer for making absolute sound measurements over the frequency range 0.1 Hz to 160 kHz with a receiving sensitivity of -205 dB

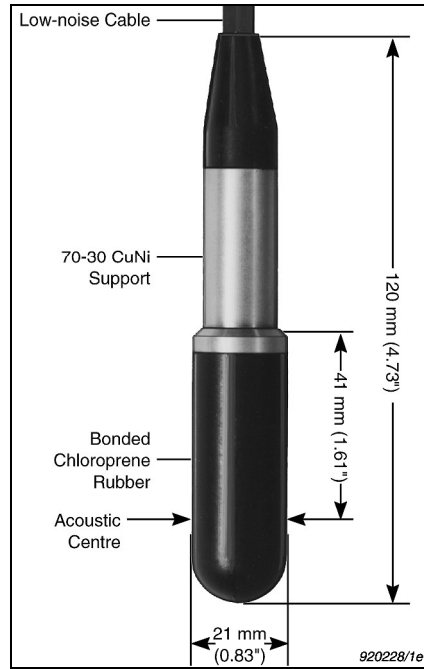


Fig. 2 Hydrophone Type 8104

re 1 V/ μ Pa. It is rugged, being capable of withstanding pressures of up to 10^7 Pa (100 atm.; 1000 m (3250 ft.) ocean depth). This hydrophone has excellent directional characteristics: at 100 kHz, it is omnidirectional over 360° in the x - y (radial) plane and 270° in the x - z (axial) plane. The Type 8105 is illustrated in Fig. 3.

Hydrophone Type 8106. A wide-range, general-purpose transducer for making absolute sound measurements over the frequency range 7 Hz to 80 kHz with a receiving sensitivity of -174 dB re 1 V/ μ Pa. The hydrophone is capable of withstanding high static pressure, the operational upper limit being 10^7 Pa (100 atm.; 1000 m (3250 ft.) ocean depth). A built-in high-quality, thick-film, low-noise, 10 dB preamplifier provides signal conditioning for transmission over long underwater cables. The preamplifier features a 7 Hz high-pass filter and an insert-voltage calibration facility, but does not allow the hydrophone to be used as a projector. An integrated watertight connector allows quick disconnection of the cable and makes replacements and storage very easy. Type 8106 is shown in Fig. 4.

Construction

The four Brüel & Kjær Hydrophones are piezoelectric transducers, i.e., they use piezoelectric ceramics as

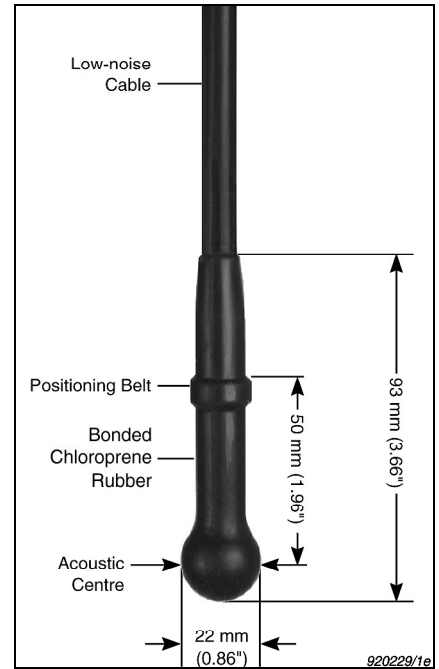


Fig. 3 Hydrophone Type 8105

sensing elements. The piezoelectric sensing element and its internal supporting structure are permanently bonded into sound-transparent polychloroprene rubber.

The support body of Hydrophones Types 8103 and 8104 is made from a 70-30 Copper-Nickel alloy. Type

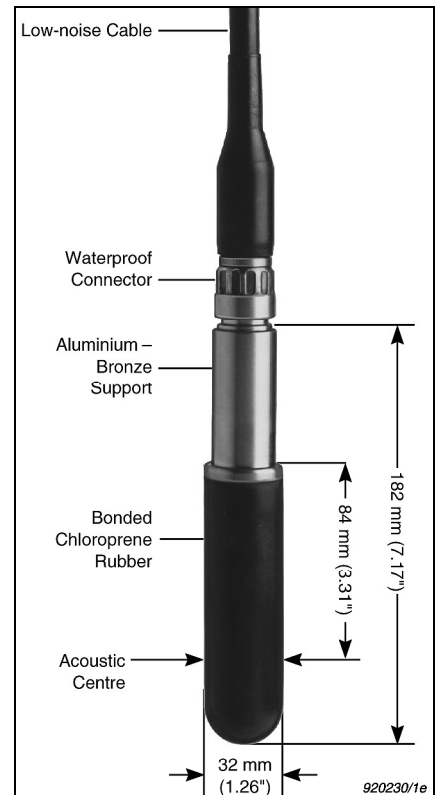


Fig. 4 Hydrophone Type 8106

8106's support body is made of Aluminium-Bronze alloy $\text{CuAl}_{10}\text{Ni}_5\text{Fe}_4$ (ISO 428). Both alloys have extremely high corrosion resistance to virtually all hostile environments, and very good anti-fouling properties when immersed in seawater. Type 8105 has no metal parts exposed.

The internal support is mechanically and electrically isolated from the metal housing, being coupled only by synthetic rubber. This provides vibration isolation of the sensing element. Except for Type 8106, the hydrophones are equipped with an integral cable, the shield of which is connected to the internal support/housing, thus providing electrical shielding for the sensing element. Note that Type 8103 has its internal support and metal housing connected at the plug end of the cable.

Cables and Connectors

Special care has been taken in the production of Brüel & Kjær Hydrophone Cables in order to obtain good electrical shielding. This enables the hydrophones to be used in air and in water tanks where the electrical potential of the water is different from the ground potential, as well as in cases of high electromagnetic interference.

The miniature Hydrophone Type 8103 is fitted with 6 m integral double-shielded cable terminated with a Brüel & Kjær Miniature Plug JP 0056. In cases of high electromagnetic interference, a metal screen can be clamped onto the metal support. This support is connected internally to the outer shield of the cable which is in turn connected to the inner shield of the terminating plug end of the cable.

Hydrophones Types 8104 and 8105 have integral 10 m cables (AC 0034) fulfilling MIL-C-915. These are fitted with BNC plugs. For use at greater depths (up to 1000 m with the 8105), MIL-C-915 standard extension cables of up to 300 m are available to order. These can be fitted to the hydrophone's integral cable with Brüel & Kjær Waterproof Connectors JP 0415 (male) and JJ 0415 (female), which are suitable for use with Hydrophones Types 8104 and 8105. Pictorial assembly instructions are provided with each male or female connector set. We recommend that the male connector (JP 0415) is fitted to the cable on the hydrophone.

Hydrophone Type 8106 is supplied without a cable. A 10 m Cable

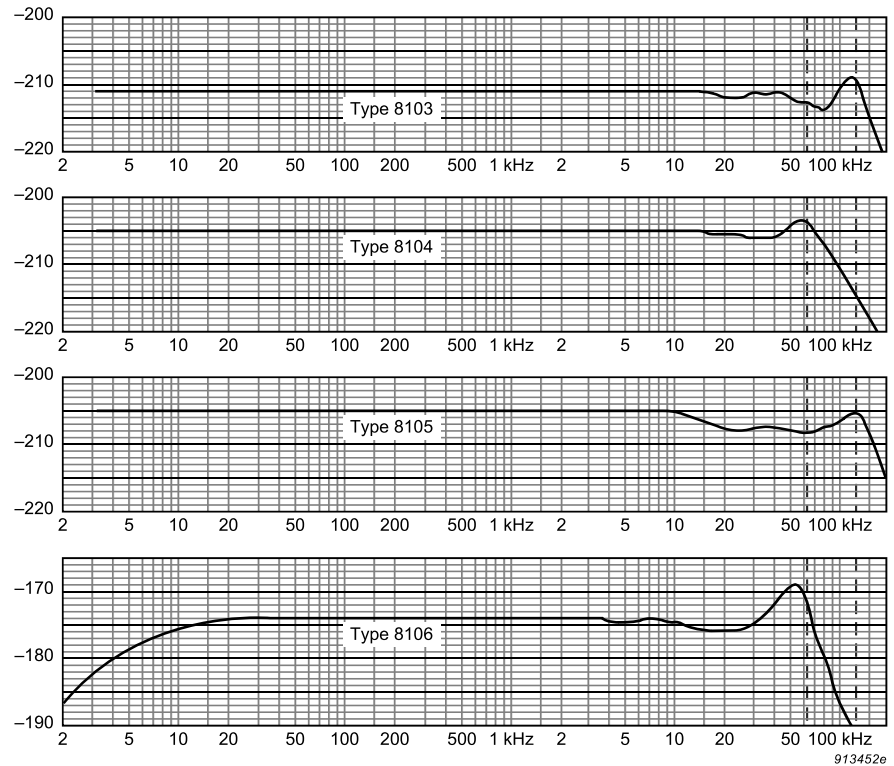


Fig. 5 Typical receiving frequency characteristics of Hydrophones Types 8103, 8104, 8105 and 8106 (dB re 1V/ μPa)

AO 0390 (a 10 m length of AC 0101: see below) mounted with a watertight Connecting Plug JP 0735 and a Brüel & Kjær 7-pin Plug JP 0717 is available for direct connection to the Preamp. socket of Brüel & Kjær Measuring Amplifiers and Frequency Analyzers. The Connecting Plug JP 0735 fits the 7-pin female connector (corresponding to the Jupiter® connector series 10M) which is an integrated part of the hydrophone's bronze support.

For use of Type 8106 at depths greater than 10 m, Hydrophone Cable AC 0101 can be supplied in continuous lengths of up to 300 m. This cable is a screened, 4-core cable with built-in KEVLAR® reinforcement. AC 0101 can tolerate high loads – 1300 N in service, with a breaking load of 2600 N. This makes the cable strong enough to be self-supporting for measurements at ocean depths of 1000 m.

Frequency Response and Directivity Patterns

Typical frequency responses of the hydrophones are shown in Fig. 5. These were measured in a water tank in free-field conditions achieved by means of pulse techniques using a

factory calibration Gating System Type 4440.

The hydrophones have very good omnidirectional characteristics. Typical directivity patterns of the hydrophones in water are shown in Fig. 6. These polar directivity patterns were measured in free-field conditions achieved by means of gating techniques in a water tank. This method requires a standard hydrophone as a projector and the unknown hydrophone as the receiver whose polar directivity pattern is to be determined.

Calibration System

Fig. 7 shows a system which can be used for measurement of the frequency response of hydrophones and for calibration of hydrophone systems. With additional components, the directivity pattern is also included. The PULSE system Type 3560 is a PC-based system including a front-end with generator and input modules, Power Amplifier Type 2713, Turntable Type 3922, Turntable Interface Type 5997, Current Probe WB 1490 and a PC with software and Windows operating system. The principle is the well-known gating technique, where a tone burst is sent out from the generator, amplified in the power amplifier and sent to the transmitter (in

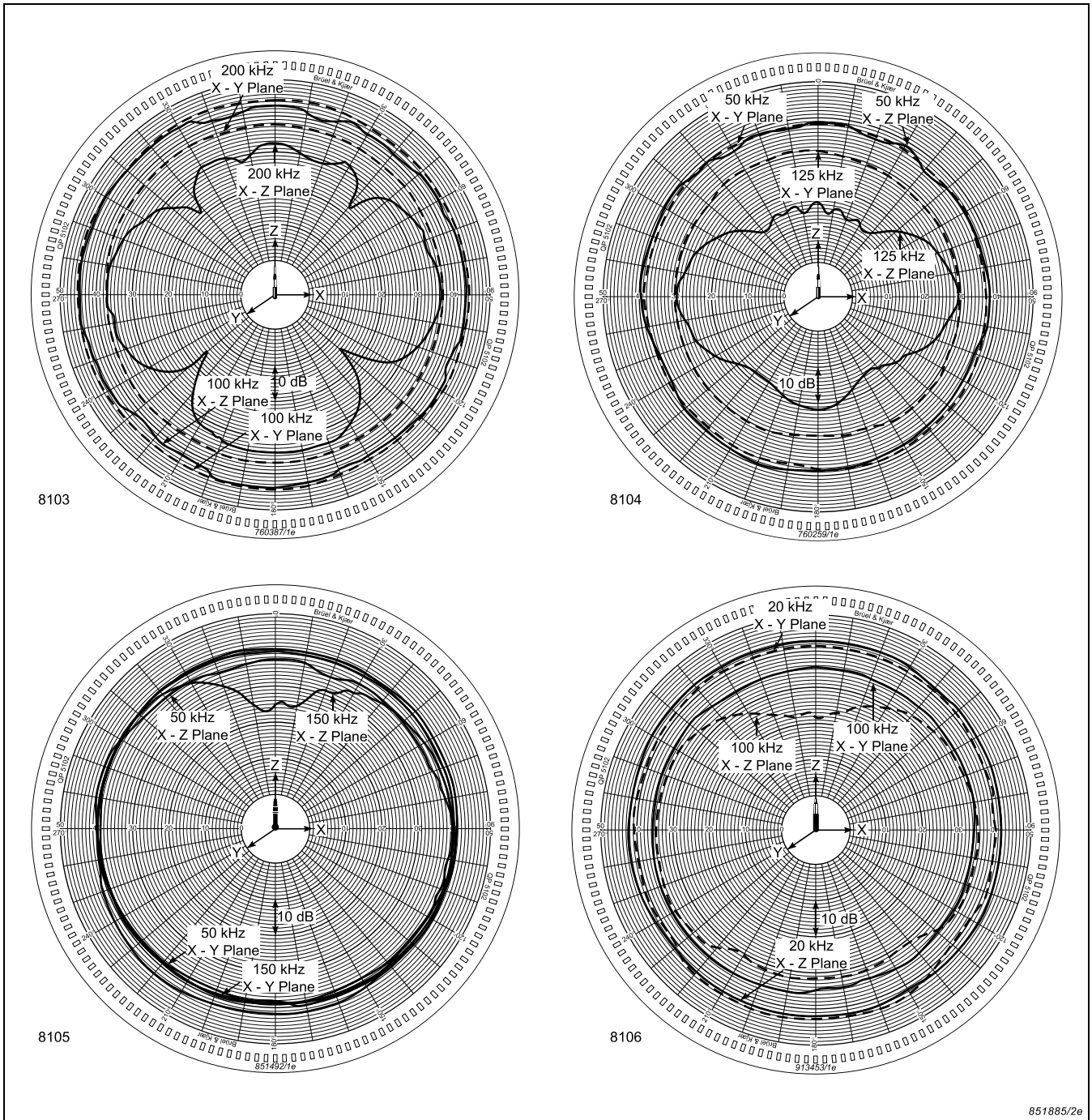


Fig.6 Typical directivity patterns of Hydrophones Types 8103, 8104, 8105 and 8106

this case a Hydrophone Type 8104). The signal received from another hydrophone is amplified and the time signal is shown on the PC monitor. A delay due to the distance between transmitter and receiver and a gating period depending on frequency range and tank dimensions is set up in the analyzer in order to analyze on a stable direct signal without transients or reflections from walls. The measurements are performed at specified frequencies and a complete report is

available. Directivity patterns can be found with software for the turntable. All responses include amplitude and phase. This system can be used for reciprocity, comparison and impedance measurements.

Calibration

Each Brüel & Kjær Hydrophone is submitted to an extensive ageing and temperature stabilizing procedure be-

fore being individually calibrated. Individual calibration data and frequency response curves are supplied with each hydrophone. The receiving sensitivity calibration of Brüel & Kjær Hydrophones is traceable to NIST.

A convenient calibration check at low frequencies can be performed using Brüel & Kjær Hydrophone Calibrator Type 4229 (the sensitivity of a hydrophone is the same in air as in water for low frequencies). This pro-

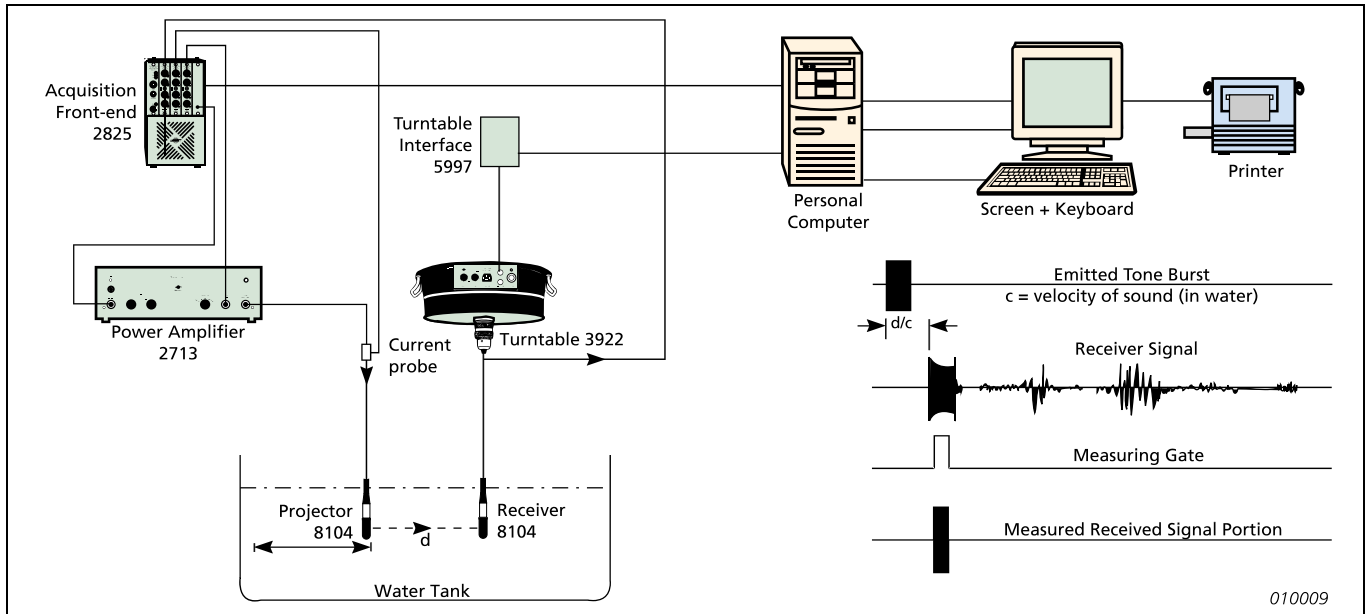


Fig. 7 Hydrophone Calibration System: Set-up for Calibration in the 4 kHz to 100 kHz range using PULSE Multi-analyzer System Type 3560

vides a rapid and easy method for air calibration of sound measuring systems terminating in Brüel & Kjær Hydrophones. Type 4229 is battery powered and can be used both in the laboratory and in the field.

The principle of operation of the 4229 is the production of a sound pressure in the coupler cavity by four pistons which oscillate back and forth in phase. A frequency of 251.2 Hz is produced which is electronically maintained within $\pm 0.1\%$.

Fig. 8 shows sectional drawings of the Calibrator. The calibrator couplers are shown fitted with their respective Brüel & Kjær Hydrophones. The typical sound pressure levels produced in the coupler volumes are 166, 162, 151.5 and 152 dB re $1 \mu\text{Pa}$ for the 8103, 8104, 8105 and 8106, respectively. Note that the sound pressure level in the coupler volume can be monitored with a $1/2''$ microphones (see inset of Fig. 8), thus enabling the calibration to be traceable to NIST.

Note: The 8106 coupler, WA 0658, is available separately. The other couplers are supplied with Type 4229.

A barometer is supplied with the 4229 giving the atmospheric pressure correction in dB in the range 0.65×10^5 to 1.08×10^5 Pa (650 to 1080 hPa), i.e. -3.85 to $+0.55$ dB.

Measurements

Before measurements are taken, the high-impedance output signals from Hydrophones Types 8103, 8104 and 8105 should be routed through a signal-conditioning amplifier. Although the entire Brüel & Kjær range of accelerometer preamplifiers is suitable, Conditioning Amplifier Type 2635 or Measuring Amplifier Type 2525 is recommended. These amplifiers feature sensitivity adjustment to give convenient output values (for example 0.1 or 1 V/Unit) and have a wide frequency range, and adjustable low-pass and high-pass filters. Note that Type 2635 is battery powered and portable. For technical details of these preamplifiers, the reader is referred to their respective product data sheets.

When using Type 8104 or 8105 with Measuring Amplifier Type 2525, BNC to TNC Adaptor JP 0226 must be used. The built-in preamplifier of Hydrophone Type 8106 provides an output signal which does not require further conditioning. The 8106 can be directly plugged into the Preamplifier socket of the Brüel & Kjær range of Measuring Amplifiers and Frequency Analyzers, with 7-pin Brüel & Kjær sockets, which supply the necessary voltages. Dual Microphone Supply Type 5935 must be ordered with modification WH 3125 and requires an external supply. Microphone

Multiplexer Type 2822 must be ordered with modification WH 3127.

Use as Projectors

The piezoelectric effect of the sensing element is reversible, i.e., mechanical excitation causes an electrical output, and conversely, an applied alternating voltage causes corresponding mechanical excitation. Therefore, Hydrophones Types 8103, 8104 and 8105 (but not Hydrophone Type 8106 because it also includes a built-in preamplifier), can be used as sound transmitters (projectors), for both measurement and reciprocity calibration. Transmitting responses to voltage for Hydrophones Types 8103, 8104 and 8105 in water are shown in Fig. 9. Note the 12 dB/octave slope which is typical for piezoelectric hydrophones driven with constant voltage input.

When hydrophones are used as projectors, the driving signal requires power amplification. Brüel & Kjær Power Amplifier Type 2713 has been especially designed for driving Hydrophones Types 8103, 8104 and 8105 as projectors, but is also useful for driving other highly reactive loads. Type 2713 is a low-noise 100 VA power amplifier with a frequency range extending from 10 Hz up to 200 kHz, and a gain continuously adjustable between 0 and 60 dB

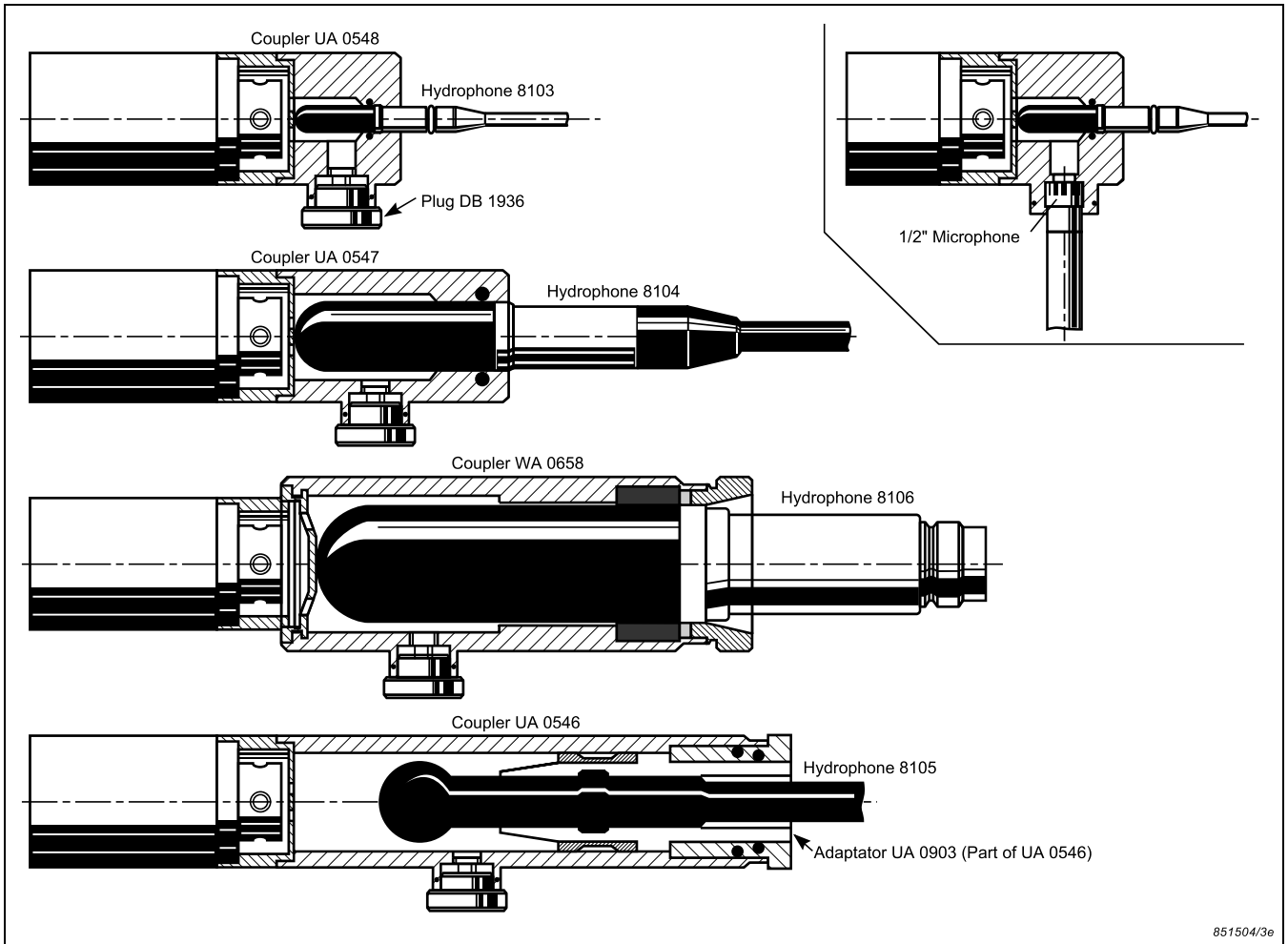


Fig. 8 Mounting of Brüel & Kjær Hydrophones Types 8103, 8104, 8105 and 8106 on their respective couplers with Calibrator Type 4229. The inset shows a 1/2" microphone inserted into a coupler for monitoring the sound pressure level

over six 10 dB ranges. It also features selectable maximum output voltage limits, as well as extensive protective functions to prevent damage to both the amplifier and the transducer.

A feature of the Sine Generator 1051 is that its amplitude memory can be programmed with the inverse

of the projector hydrophone's transmitting response (Fig. 9). The combination of drive signal and projector characteristic then results in constant SPL with frequency. The set-up for this method of calibration is the same as that shown in Fig. 7.

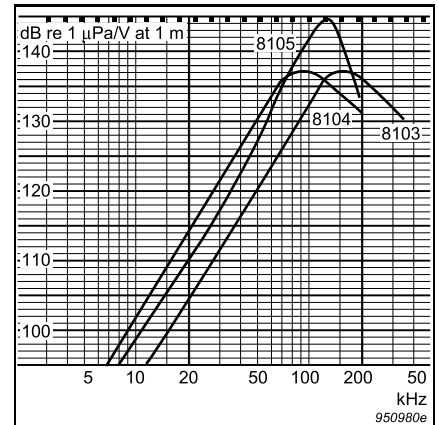


Fig. 9 Typical transmitting response to voltage of Types 8103, 8104 and 8105

Specifications 8103, 8104, 8105 and 8106

Type	8103	8104	8105	8106
Voltage sensitivity*: (with cable) at 20°C	30 μV/Pa ±8 μV (-211dB re 1 V/μPa ±2dB)	56 μV/Pa ±15 μV (-205dB re 1 V/μPa ±2dB)		2000 μV/Pa ±500 μV (-174dB re 1 V/μPa ±3dB)
Charge sensitivity*	0.12 pC/Pa	0.44 pC/Pa	0.42 pC/Pa	
Capacitance*: (with integral cable)	3850 pF	7800 pF	7500 pF	
Frequency range* (re 250 Hz):	(+1.5 dB) 0.1 Hz to 100 kHz (-6.0 dB) (+3.5 dB) 0.1 Hz to 180 kHz (-12.5 dB)	(±4.0 dB) 0.1 Hz to 80 kHz (+4.0 dB) 0.1 Hz to 120 kHz (-12.0 dB)	(+1.0 dB) 0.1 Hz to 100 kHz (-6.5 dB) (+3.5 dB) 0.1 Hz to 160 kHz (-10.0 dB)	(+0.5 dB) 10 Hz to 10 kHz (-3.0 dB) (+0.5 dB) 7 Hz to 30 kHz (-8.0 dB)
Horizontal directivity:† (radial xy plane)	±2 dB at 100 kHz			±2 dB at 20 kHz
Vertical directivity: (axial xz plane)	±4 dB at 100 kHz	±2 dB at 50 kHz	±2 dB over 270° at 100 kHz	±3 dB at 20 kHz
Leakage resistance*: (at 20°C)	>2500 MΩ			
Operating temperature range Short-term: Continuous:	-40°C to +120°C -40°C to +80°C			-10°C to +60°C
Sensitivity change with temperature Charge: Voltage:	0 to +0.03 dB/°C 0 to -0.03 dB/°C	0 to +0.03 dB/°C 0 to -0.04 dB/°C	0 to +0.03 dB/°C 0 to -0.03 dB/°C	- 0 to +0.01 dB/°C
Max. operating static pressure:	252 dB = 4 × 10 ⁶ Pa = 40 atm. = 400 m ocean depth		260 dB = 9.8 × 10 ⁶ Pa = 100 atm. = 1000 m ocean depth	
Sensitivity change with static pressure:	0 to -3 × 10 ⁻⁷ dB/Pa (0 to -0.03 dB/atm.)			0 to 1 × 10 ⁻⁷ dB/Pa 0 to 0.01 dB/atm.
Allowable total radiation dose:	5 × 10 ⁷ Rad.			
Dimensions: Length: Body dia:	50 mm (1.97") 9.5 mm (0.37")	120 mm (4.73") 21 mm (0.83")	93 mm (3.66") 22 mm (0.87")	182 mm (7.17") 32 mm (1.26")
Weight: (including integral cable)	170 g (0.37 lb)	1.6 kg (3.5 lb)		382 g (0.84 lb)
Integral cable:	6 m waterproof low-noise double-shielded teflon cable with standard miniature coaxial plug	10 m waterblocked low-noise shielded cable to MIL-C-915 with BNC plug		

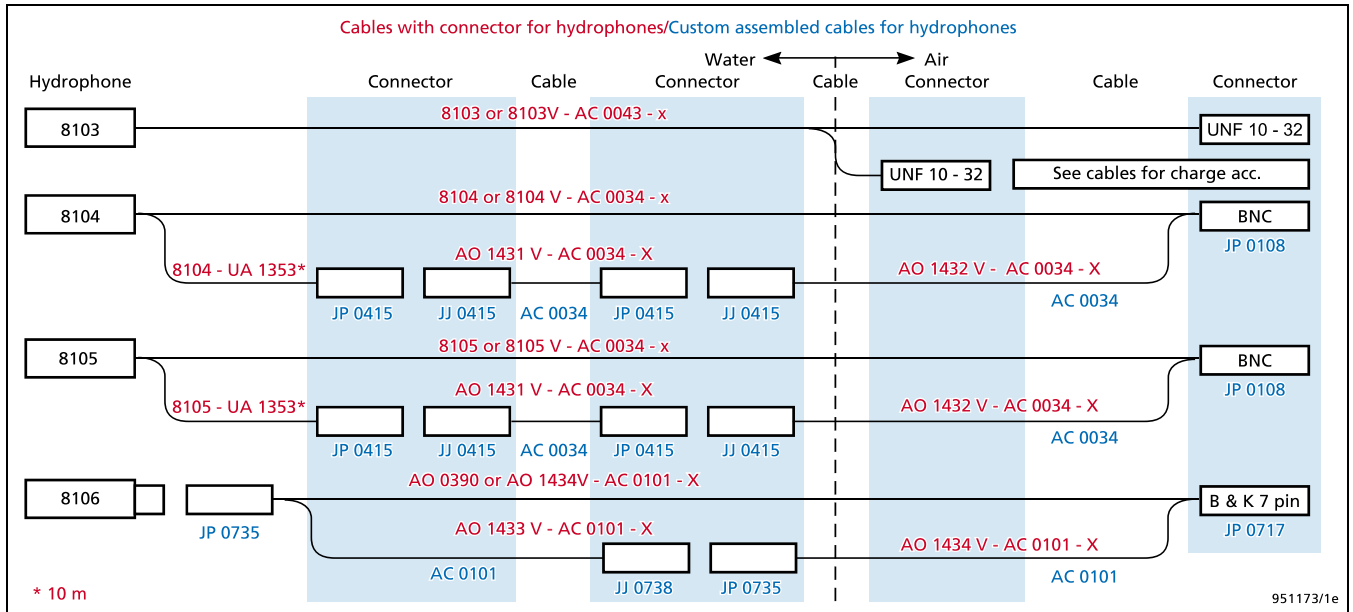
* Nominal value, each hydrophone is supplied with its own calibration data

† See polar directivity given in Fig. 7

Note: All values are typical at 25°C (77°F), unless measurement uncertainty is specified. All uncertainty values are specified at 2σ (i.e. expanded uncertainty using a coverage factor of 2)

Ordering Information

Optional Accessories		Type 8106:	Adaptors:
Types 8104 and 8105:		WB 0850: Insert-voltage Junction Unit	JP 0145: 10 – 32 UNF to BNC
AC 0034:	Waterblocked low-noise shielded extension cable to MIL-C-915 available to any length up to 300 m	AC 0101: Watertight, shielded 4-core cable, Kevlar reinforced. Any length up to 300 m	JP 0226: BNC to TNC
JP 0415	Male underwater connector (for integral cable)	AO 0390: 10 m of AC 0101, with JP 0735 and JP 0717	JP 0162: 10 – 32 UNF to TNC
JJ 0415:	Female underwater connector (for extension cable)	JP 0717: 7-pin Brüel & Kjær input connector	JJ 2617: 10 – 32 UNF to 1/2" mic. amp
JP 0108:	BNC connector for extension cable	JP 0744: Male underwater connector for Extension Cable AC 0038	WL 1260: 7 pin B&K to 7 pin LEMO
		JP 0735: Male underwater connector (for extension cable)	General:
		JJ 0738: Female underwater connector (for underwater extension cable)	4229: Hydrophone Calibrator
			Cables: See next page
			* Type 8106 requires Adaptor WA 0658, available separately



Additional Specifications 8106

MAX. OUTPUT SIGNAL:
 12 V supply: 3.5 V or 28 mA
 24 V supply: 7.0 V or 28 mA

MAX. POWER OUTPUT:
 50 mW

OUTPUT IMPEDANCE:
 <30 Ω

HIGH-PASS FILTER:
 -3 dB at 7 Hz (±2 Hz)

DC RIPPLE REJECTION:
 20 Hz to 20 kHz: 70 dB

OVERLOAD SOUND PRESSURE LEVEL:
 12 V supply: 182 dB re 1 μPa
 24 V supply: 188 dB re 1 μPa

Power Supply

SUPPLY VOLTAGE:
 12 to 24 VDC

POWER CONSUMPTION:
 6 mA without load

Note: All values are typical at 25°C (77°F), unless measurement uncertainty is specified. All uncertainty values are specified at 2σ (i.e. expanded uncertainty using a coverage factor of 2)

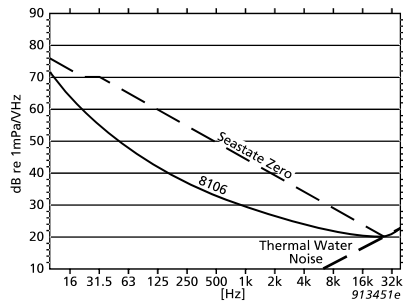


Fig. 10 Typical equivalent noise pressure level of Type 8106

COMPLIANCE WITH STANDARDS:

	CE-mark indicates compliance with: EMC Directive IEC 61010-1
Safety	EN 61010-1 and IEC 1010-1: Safety requirements for electrical equipment for measurement, control and laboratory use.
EMC Emission	EN 50081-1: Generic emission standard. Part 1: Residential, commercial and light industry. EN 50081-2: Generic emission standard. Part 2: Industrial environment. CISPR 22: Radio disturbance characteristics of information technology equipment. Class B Limits. FCC Class B limits.
EMC Immunity	EN 50082-1: Generic immunity standard. Part 1: Residential, commercial and light industry. EN 50082-2: Generic immunity standard. Part 2: Industrial environment. Note 1: The above is guaranteed using accessories listed in this Product Data sheet only. Note 2: Susceptibility when exposed to levels specified in EN 50082-2. Conducted RF: <30 mV Radiated RF: <120 mV
Temperature	IEC 60068-2-1 & IEC 60068-2-2: Environmental Testing. Cold and Dry Heat. Operating Temperature: -10 to +60°C (+14 to +140°F) Storage Temperature: -40 to +80°C (-40 to +176°F)
Humidity	IEC60068-2-3: Damp Heat: 90% RH (non-condensing at 40°C (104°F))
Mechanical	Non-operating: IEC 60068-2-6: Vibration: 0.3 mm, 20 m/s ² , 10-500 Hz IEC 60068-2-27: Shock: 1000 m/s ² IEC60068-2-29: Bump: 3000 bumps at 250 m/s ²
Enclosure	IEC60529: Protection Provided by Enclosures: IP 68

Brüel & Kjær reserves the right to change specifications and accessories without notice

Brüel & Kjær

HEADQUARTERS: DK-2850 Nærum · Denmark · Telephone: +4545800500 · Fax: +4545801405 · <http://www.bksv.com> · e-mail: info@bksv.com
 Australia (02)9450-2066 · Austria 0043-1-8657400 · Brazil (011)5182-8166 · Canada (514)695-8225 · China (86) 1068029906
 Czech Republic 02-67021100 · Finland (09)-755 950 · France (01)69907100 · Germany 06103/733 5-0 · Hong Kong 25487486 · Hungary (1)2158305
 Ireland (01)803 7600 · Italy 02 57 68061 · Japan 03-3779-8671 · Republic of Korea (02)3473-0605 · Netherlands (31)318 559290 · Norway 66771155
 Poland (22)858 9392 · Portugal (1)4711453 · Singapore (65) 377-4512 · Slovak Republic 421 7 544 307 01 · Spain (91)6590820 · Sweden (08)4498600
 Switzerland (0)1 880 70 35 · Taiwan (02)7139303 · United Kingdom (0)1438 739 000 · USA 800 332 2040
 Local representatives and service organisations worldwide