ESCUELA SUPERIOR DE INGENIERÍA INFORMÁTICA

INGENIERÍA INFORMÁTICA

Curso Académico 2009/2010

Proyecto de Fin de Carrera

SISTEMA PARA TRANSFORMADORES DE TENSIÓN

Autor: Alberto Blasco Molina-Prados
Tutor: Rafael Capilla Sevilla
Cotutor: Eduardo Suárez Vallina
Índice

Resumen ..................................................................................................................................... 7
Capítulo 1: Introducción............................................................................................................ 8
  1.1 Motivación ....................................................................................................................... 8
  1.2 Objetivos ........................................................................................................................ 9
  1.3 Método de trabajo ......................................................................................................... 9
Capítulo 2: Estado del arte ...................................................................................................... 10
  2.1 Plataforma .NET .......................................................................................................... 10
  2.1.1 Introducción a .NET ................................................................................................ 10
  2.1.2 Lenguajes C# y ASP ............................................................................................... 11
  2.2 DCOM .......................................................................................................................... 13
  2.3 SQL Server .................................................................................................................... 14
  2.4 Hibernate ...................................................................................................................... 15
  2.5 AJAX ............................................................................................................................. 15
  2.6 Log4Net ........................................................................................................................ 18
  2.7 Aplicaciones para la gestión de recursos ..................................................................... 18
    2.7.1 Herramienta de Ayuda a la Explotación de Equipos de la red de Unión Fenosa 18
    2.7.2 Gestión Vial de Madrid ......................................................................................... 19
Capítulo 3: Descripción Informática ....................................................................................... 21
  3.1 Descripción del Problema ............................................................................................ 21
  3.2 Especificación de requisitos ........................................................................................ 23
  3.3 Análisis .......................................................................................................................... 27
  3.4 Diseño ............................................................................................................................. 32
    3.4.1 Interfaz de usuario .................................................................................................. 32
    3.4.2 Base de datos .......................................................................................................... 36
    3.4.3 Arquitectura software ............................................................................................. 49
  3.5 Implementación .......................................................................................................... 55
    3.5.1 Aplicación Web ....................................................................................................... 55
  4.1 Pruebas ......................................................................................................................... 65
Capítulo 4: Conclusiones .......................................................................................................... 76
Bibliografía ............................................................................................................................... 78
ANEXO I: Detalle Ensayos de Fábrica ............................................................... 80
ANEXO 2: Hardware y Software Utilizado ........................................................................... 90
Índice de Ilustraciones

Ilustración 1: Estructura de .NET ................................................................. 11
Ilustración 2: Esquema estructura AJAX .................................................. 16
Ilustración 3: Esquema funcionamiento AJAX ........................................... 17
Ilustración 4: Introducción de medidas de transformadores aplicación HAE ...... 18
Ilustración 5: Resumen medidas de Interruptores HAE ............................... 19
Ilustración 6: Pantalla de inicio GESVIAM ................................................... 20
Ilustración 7: Caso de Uso Encargado ......................................................... 28
Ilustración 8: Caso de Uso Administrador de usuarios, usuario de consulta .......... 30
Ilustración 9: Plantilla de las páginas ......................................................... 33
Ilustración 10: Detalle acciones menú ......................................................... 34
Ilustración 11: Pantalla principal del transformador .................................... 35
Ilustración 12: Diagrama de tablas de seguridad ......................................... 37
Ilustración 13: Diagrama de tablas transformador ...................................... 42
Ilustración 14: Diagrama de tablas de especificaciones ............................... 43
Ilustración 15: Diagrama de tablas ensayos de fábrica ................................ 44
Ilustración 16: Diagrama ensayos de campo ............................................... 46
Ilustración 17: Ensayos de análisis ......................................................... 47
Ilustración 18: Ensayos de medidas de transformadores .............................. 48
Ilustración 19: Diagrama de despliegue .................................................... 49
Ilustración 20: Diagrama de paquetes ..................................................... 50
Ilustración 21: Diagrama de Clases Capa Lógica del Negocio ....................... 52
Ilustración 22: Diagrama de Clases ......................................................... 53
Ilustración 23: Creación de informes ....................................................... 58
Ilustración 24: Crear proyecto de Instalación ............................................. 61
Ilustración 25: Dependencias de Instalación .............................................. 62
Ilustración 26: Autenticación incorrecta ................................................... 65
Ilustración 27: Perfil incompleto ............................................................ 66
Ilustración 28: Perfil creado correctamente ............................................... 66
Ilustración 29: Eliminar perfiles ............................................................ 66
Ilustración 30: Usuarios no autorizado ...................................................... 67
Ilustración 31: Datos de usuarios incorrectos .............................................. 67
Ilustración 32: Datos de usuarios correctos ................................................. 67
Ilustración 33: Validación consulta transformadores ................................... 68
Ilustración 34: Validación consulta ensayos ............................................. 68
Ilustración 35: Comparativa de transformadores sin resultados ....................... 69
Ilustración 36: Comparativa de transformadores con resultados .................... 69
Ilustración 37: Comparativa de transformadores ....................................... 70
Ilustración 38: Validación de datos de comparativa de ensayos .................... 70
Ilustración 39: Error alta de transformadores Paso 1 ................................... 71
Ilustración 40: Error alta de transformadores Paso 2 ................................... 71
Ilustración 41: Error alta de transformadores Paso 3 ................................... 72
Ilustración 42: Error alta de transformadores Paso 4 ................................... 72
Ilustración 43: Error al modificar los datos del transformador ....................... 73
Ilustración 44: Datos del transformador modificados correctamente ............... 73
Ilustración 45: Generación de informes ................................................................. 74
Ilustración 46: Añadir ensayo ........................................................................... 74
Ilustración 47: Estados antes de cerrar, después de cerrar el ensayo .............. 74
Ilustración 48: Datos erróneos al modificar un ensayo ................................... 75
Ilustración 49: Ensayo de calentamiento .......................................................... 81
Ilustración 50: Ensayo tipo rayo ..................................................................... 82
Ilustración 51: Ensayo de intensidad homopolar ............................................ 82
Ilustración 52: Ensayo de capacidad y tangente delta ..................................... 83
Ilustración 53: Ensayo de pérdidas en vacío .................................................... 83
Ilustración 54: Ensayo tipo maniobra .............................................................. 84
Ilustración 55: Ensayo de pérdidas en vacío .................................................... 84
Ilustración 56: Ensayo de medidas de ruido ..................................................... 85
Ilustración 57: Ensayo de regulador ................................................................. 85
Ilustración 58: Ensayo de pérdidas en carga ................................................... 86
Ilustración 59: Ensayo de resistencia de arrollamientos ............................... 86
Ilustración 60: Ensayo de resistencia de aislamiento ..................................... 87
Ilustración 61: Ensayo de tensión aplicada ...................................................... 87
Ilustración 62: Ensayo de relación de transformación ..................................... 88
Ilustración 63: Ensayo de tensión inducida ..................................................... 89
Ilustración 64: Ensayo de tensión inducida sin DDPP ..................................... 89
Índice de Tablas

Tabla 1: Requisitos Funcionales ........................................................................................................ 23
Tabla 2: Requisitos No Funcionales .................................................................................................. 26
Tabla 3: Requisitos Hardware/Software .............................................................................................. 27
Tabla 4: Tabla Usuarios dados de alta en la DB .............................................................................. 36
Tabla 5: Tabla Perfil .......................................................................................................................... 36
Tabla 6: Tabla Página de acciones .................................................................................................... 36
Tabla 7: Tabla relación perfil página ................................................................................................. 37
Tabla 8: Tabla transformador ............................................................................................................ 38
Tabla 9: Tabla tipo de transformador ................................................................................................ 39
Tabla 10: Tabla fabricantes de equipos .............................................................................................. 39
Tabla 11: Tabla de ubicaciones .......................................................................................................... 39
Tabla 12: Tabla relación de fabricantes con ubicaciones ................................................................. 40
Tabla 13: Tabla devanados ................................................................................................................ 40
Tabla 14: Tabla borna ...................................................................................................................... 40
Tabla 15: Tabla terminal ................................................................................................................... 41
Tabla 16: Tabla regulador .................................................................................................................. 41
Tabla 17: Tabla regulación ................................................................................................................ 41
Tabla 18: Tabla especificación .......................................................................................................... 43
Tabla 19: Tabla ensayos .................................................................................................................... 44
Tabla 20: Tabla ensayos campo ......................................................................................................... 45
Tabla 21: Medición transformador ................................................................................................... 45
Tabla 22: Tabla análisis ...................................................................................................................... 46
Resumen

Debido a la gran cantidad de información, datos y equipos que mueven las grandes empresas eléctricas, y las empresas en general, surge la necesidad de desarrollar e implantar aplicaciones de gestión y de mantenimiento para estos equipos.

Muchos de estos equipos son críticos para el buen funcionamiento de las estructuras e instalaciones de estas empresas, por lo que es necesario desarrollar procedimientos para poder poner en marcha estas unidades. Asimismo, parte de estos procedimientos requieren que se realicen, una serie de medidas a los equipos, para cumplir unos requisitos mínimos para que sean válidos.

Este trabajo pretende mejorar la gestión y el mantenimiento de este tipo de equipos aplicando una serie de soluciones informáticas. En concreto se mejora proceso de entrada de nuevos transformadores de potencia, para la empresa eléctrica Unión Fenosa. Para ello propone el desarrollo de una herramienta informática para agilizar todo el proceso de entrada de nuevos transformadores de potencia a la red de Unión Fenosa.
Capítulo 1: Introducción

1.1 Motivación

Debido a la gran importancia de los transformadores de potencia dentro de la red eléctrica, es necesario realizar sobre ellos una serie de medidas, con el fin de validar su funcionamiento en función de las características de cada uno de estos equipos. Además, debe existir una gestión de estos recursos para poder integrarlos dentro de la red eléctrica de forma adecuada.

Por estos motivos resulta necesario introducir una herramienta informática dentro de todo el proceso de validación y comprobación del buen funcionamiento de los transformadores de potencia, de manera que todo este proceso quede recogido en una base de datos, proporcionando a los usuarios mayor agilidad en el instante de realizar las labores necesarias para que un transformador pueda dar servicio. Esta herramienta debe permitir un acceso concurrente de los usuarios del sistema.

Por otro lado y debido a que la fabricación de estos equipos no se realiza en una misma ubicación y a la imposibilidad de acceso al interior de la red corporativa, es necesario generar una forma para poder trabajar sobre los transformadores de una manera que de aquí en adelante se denominará OFFLINE. Así, cualquier usuario podrá realizar los trabajos como si se encontrara dentro de la red corporativa, pudiendo posteriormente volcar toda la información al servidor principal. Esto permitirá a los usuarios de la aplicación la posibilidad de trabajar con la herramienta en cualquier lugar independientemente de si se tiene acceso o no a al interior de la red. En este sentido resulta necesaria la existencia de una herramienta que permita a estos usuarios el trabajo de una manera transparente, dando la sensación que siempre están conectados al servidor central de la herramienta.
1.2 Objetivos

Una vez comentado los motivos anteriores, los objetivos que se pretenden en este proyecto son los siguientes:

- Estudio de las tecnologías enfocadas a las aplicaciones Web, así como todas las tecnologías necesarias para el desarrollo de las características especiales de este proyecto.
- Análisis e implementación de una aplicación Web que permita, solucionar los problemas expuestos en el punto anterior, que son los siguientes:
  - Acceso concurrente de diversos usuarios a la aplicación.
  - Realizar todas las acciones necesarias para poner en servicio un transformador de potencia.
  - Permitir a los usuarios trabajar si una conexión en la red interna de Unión Fenosa.

1.3 Método de trabajo

La metodología de trabajo que se va a seguir va a ser el modelo de ciclo de vida clásico en cascada que consta de las siguientes etapas.

- Captura de requisitos.
- Análisis y especificación de requisitos.
- Diseño de la arquitectura software, base de datos e interfaz.
- Implementación.
- Pruebas.
Capítulo 2: Estado del arte

El objetivo de esta sección es realizar un análisis previo de las distintas tecnologías que han sido utilizadas en el desarrollo del proyecto que se describe en esta memoria. Asimismo, se analizarán las aplicaciones existentes que puedan tener una cierta similitud con la aplicación descrita.

2.1 Plataforma .NET

2.1.1 Introducción a .NET

La plataforma .NET de Microsoft es un componente software que puede ser añadido al sistema operativo Windows. Provee un extenso conjunto de soluciones predefinidas para necesidades generales de la programación de aplicaciones, y administra la ejecución de los programas escritos específicamente con la plataforma. Esta solución es el producto principal en la oferta de Microsoft, y pretende ser utilizada por la mayoría de las aplicaciones creadas para la plataforma Windows.

Un framework, es una estructura conceptual y tecnológica definida, normalmente con artefactos o módulos de software concretos, en base a la cual otro proyecto de software puede ser organizado y desarrollado. El framework de .NET incluye soluciones en áreas como: la interfaz de usuario, acceso a datos, conectividad a bases de datos, criptografía, desarrollo de aplicaciones web, algoritmos numéricos y comunicación de redes. La ilustración 1 muestra la estructura de la plataforma .NET que explicamos a continuación.
Ilustración 1: Estructura de .NET

a) **Common Language Runtime (CLR):** Administra servicios en tiempo de ejecución como por ejemplo la seguridad, la administración de la memoria y la integración de los lenguajes. Es el corazón de la ejecución de una aplicación en .NET. Se encarga entre otros de la ejecución de las aplicaciones.

b) **La librería de clases (BCL):** Conjunto de librerías que usan los lenguajes de programación .NET. Proveen de código reusable para las tareas más comunes como por ejemplo: acceso a datos, desarrollo de Servicios Web y formularios Web o de Windows. Estas clases pueden ser extendidas por los desarrolladores ya que estos pueden agregar sus propias clases.

c) **ADO.NET:** Son clases que proveen de acceso a los datos.

d) **ASP.NET:** Son clases que permiten construir los Servicios Web y la interfaz de usuario de las aplicaciones Web. La plataforma .NET provee de herramientas para su creación, prueba y distribución.

e) **Interfaz del Usuario:** La plataforma soporta tres interfaces de usuario: consola de comando, formularios Windows y formularios Web.

f) **Lenguajes:** Cualquier lenguaje que sea compatible con la Especificación de Lenguaje Común puede ser utilizado en la plataforma .NET. Además de los lenguajes nativos de Visual Studio hay muchos otros, por ejemplo PERL, Cobol o RPG.

### 2.1.2 Lenguajes C# y ASP

Para el desarrollo de la aplicación de este proyecto se han utilizado básicamente dos lenguajes, en primer lugar ASP, que va a ser en encargado de mostrar toda la parte visual de la aplicación web y por el otro C# que es el encargado de implementar toda la lógica producida.
**Lenguaje ASP:** Active Server Pages, también conocido como ASP es una tecnología de Microsoft desarrollada para generar web dinámicamente. Además es parte del Internet Information Server (IIS) desde la versión 3.0 y es una tecnología de páginas activas que permite el uso de diferentes scripts y componentes en conjunto con el tradicional HTML para mostrar páginas generadas dinámicamente, traduciendo la definición de Microsoft: “Las Active Server Pages son un ambiente de aplicación abierto y gratuito en el que se puede combinar código HTML, scripts y componentes ActiveX del servidor para crear soluciones dinámicas y poderosas para el web”.

La tecnología ASP intenta ser una solución para desarrollar aplicaciones web de una manera rápida. Además permite la utilización de diversos componentes ya desarrollados como son algunos controles ActiveX así como componentes del lado servidor. ASP .NET integra lenguajes compilados de código intermedio, como son Visual Basic, C# o cualquier otro lenguaje de programación que esté soportado en la plataforma .NET. De esta manera es posible desarrollar código directamente en la parte servidora de la aplicación web sin la necesidad de integrar librerías externas.

Una de las grandes ventajas que proporciona ASP .NET es, que gran parte de su código es ejecutado en el lado del servidor lo que permite que cualquier página desarrollada con este lenguaje, pueda ser visualizada por cualquier navegador. ASP.NET introduce el concepto que no había sido utilizado hasta el momento, code-behind, por el que una misma página está compuesta de dos ficheros: el de la interfaz de usuario (que no deja de ser código HTML) y el que contiene el código. Con ello se facilita la programación de aplicaciones en múltiples capas, lo que en definitiva se traduce en la total separación entre lo que el usuario ve y lo que la base de datos tiene almacenado. Por tanto, cualquier cambio drástico de especificaciones minimiza los cambios en la aplicación y maximiza la facilidad de mantenimiento.

A continuación se enumeran algunas de las ventajas que produce ASP .NET respecto a otras tecnologías web:

- Permite almacenar en la caché del servidor tanto páginas enteras, como controles personalizados o simples variables. En páginas críticas con mucha carga de base de datos es muy útil almacenar datos de la base de datos en la caché, reduciendo enormemente el consumo de recursos.
- Carpetas especializadas, como por ejemplo app_code que compila automáticamente las clases que se alojan en él.
- Los archivos de configuración Web.config permite realizar operaciones de configuración en ficheros que hasta ahora había que realizar en el servidor. De esta manera se pueden modificar parámetros que será utilizados en la aplicación, sin que haya que volver a generar ninguna parte interna del código.
- La adaptación automática del código devuelto a los dispositivos que le acceden. Lo que permite el funcionamiento de la web desarrollada en cualquier navegador.
- La eliminación total de la necesidad de frames con la introducción de las masterpages (estructura de página web que será común en todas las páginas de una aplicación web). De esta manera es posible, configura una plantilla que sea utilizada para todas las páginas definidas en la web.
- Compatible con XML y servicios Web.
- Multitud de controles Web que permiten mucha funcionalidad con poco código.
Lenguaje C#: es un lenguaje orientado a objetos que ha sido desarrollado y estandarizado por Microsoft como parte de su plataforma .NET. Aunque .NET es capaz de trabajar con diferentes lenguajes de programación, Microsoft desarrolló un lenguaje que no cuenta con elementos heredados de versiones anteriores e innecesarios en esta plataforma y que por tanto sea lo más sencillo posible para programar aprovechando toda su potencia y versatilidad.

Este lenguaje combina los mejores elementos de múltiples lenguajes de amplia difusión como C++, Java, Visual Basic o Delphi. La idea principal detrás del lenguaje es combinar la potencia de lenguajes como C++ con la sencillez de lenguajes como Visual Basic. A continuación se enumeran algunas de las mejoras introducidas por C#, con respecto a otros lenguajes:

- **Respecto a C y C++:**
  - Recolección de basura automática.
  - Eliminación del uso de punteros.
  - No importa el orden en el que hayan sido definidas las clases ni las funciones.
  - No existen dependencias circulares.
  - Compila a código intermedio, (CIL) que es independiente del lenguaje en que haya sido escrita la aplicación e independiente de la máquina donde vaya a ejecutarse.

- **Respecto a Java:**
  - Concepto formalizado de los métodos get y set, ya que se incluyen dentro de una misma propiedad lo que hace que sea más legible.
  - El rendimiento por lo general es mucho mejor.
  - Soporta más tipos primitivos.
  - El lenguaje intermedio que soporta (CIL) está estandarizado, al contrario que los bytecodes de java.
  - Soporta sobrecarga de operadores.

Además de C#, Microsoft proporciona Visual Studio.NET, la nueva versión de su entorno de desarrollo adaptada a la plataforma .NET y que ofrece una interfaz común para trabajar de manera cómoda y visual con cualquiera de los lenguajes de la plataforma .NET (por defecto, C++, C#, Visual Basic.NET y JScript.NET, aunque pueden añadirse nuevos lenguajes mediante los plugins que proporcionen sus fabricantes).

### 2.2 DCOM

En primer lugar definimos la tecnología COM, puesto que DCOM es una extensión directa de COM.

COM, también conocido como Component Object Model es una plataforma de Microsoft para componentes de software introducida por dicha empresa en 1993. Esta plataforma es utilizada para permitir la comunicación entre procesos y la creación dinámica de objetos, en cualquier lenguaje de programación que soporte dicha tecnología. El término COM es a menudo utilizado para abarcar las tecnologías OLE, OLE Automation, Active X, COM+ y DCOM. COM es una manera de implementar objetos neutrales con respecto al lenguaje, de manera que pueden ser usados en entornos distintos de aquel en que fueron creados, a través de
fronteras entre máquinas. COM permite la reutilización de objetos sin conocimiento de su implementación interna, porque fuerza a los implementadores de componentes a proveer interfaces bien definidas que están separadas de la implementación.

La versión distribuida de COM se denomina DCOM, Distributed Component Object Model, que consiste una tecnología propietaria de Microsoft para desarrollar componentes software distribuidos sobre varios ordenadores y que se comunican entre sí. Extiende el modelo COM de Microsoft, como se ha visto anteriormente y proporciona el sustrato de comunicación entre la infraestructura del servidor de aplicaciones COM+ de Microsoft.

En términos de las extensiones que añade a COM, DCOM resuelve problemas como:

- Aplanamiento - Serializar y deserializar los argumentos y valores de retorno de las llamadas a los métodos "sobre el cable".
- Recolección de basura distribuida, asegurándose que las referencias mantenidas por clientes de las interfaces sean liberadas cuando, por ejemplo, el proceso cliente ha caído o la conexión de red se pierde.

Uno de los factores clave para resolver estos problemas es el uso de DCE/RPC, que es un sistema que permite que el software trabaje a través de múltiples ordenadores, como si todo el funcionamiento se estuviera produciendo en la misma máquina. Este sistema permite que los programadores escriban software distribuido sin tener que preocuparse del código subyacente de la red. DCE/RPC define reglas estrictas en cuanto al aplanamiento y a quién es responsable de liberar la memoria.

Los defensores de la tecnología DCOM sostenían que algún día serían el modelo de código y servicios sobre Internet. Sin embargo, las dificultades que suponía conseguir que estas tecnologías funcionasen a través de cortafuegos y sobre máquinas inseguras o desconocidas, significó que las peticiones HTTP normales, combinadas con los navegadores web les ganasen la partida. Microsoft, en su momento intentó y fracasó anticiparse a esto añadiendo un transporte extra HTTP a DCE/RCP denominado "ncacn_http" (Connection-based, over HTTP).

2.3 SQL Server

Como motor de base de datos para este proyecto, se ha utilizado SQL Server, que es un sistema para la gestión de base de datos generado por Microsoft y basado en el modelo relacional. Las principales características por las que utilizar SQL Server dentro de este proyecto son las siguientes:

a) Soporta transacciones, lo que será indispensable para esta aplicación ya que el acceso va a ser concurrente por distintos usuarios.
b) Permite gestionar la seguridad de una manera sencilla, lo que permitirá conectarse a ella de diversas formas.
c) Incorpora la posibilidad de generar procedimientos almacenados.
d) Incluye también un potente entorno gráfico de administración, que permite el uso de comandos DDL y DML gráficamente.
e) Es muy estable y escalable.
f) Permite trabajar en modo cliente-servidor donde la información y datos se alojan en el servidor y las terminales o clientes de la red sólo acceden a la información.

g) Además permite administrar información de otros servidores de datos.

h) Soporta diferentes conexiones al mismo tiempo.

Además de las de SQL Server, otra de las razones por las que utilizar esta tecnología es la perfecta integración que produce con la herramienta en la que se desarrolla el proyecto, Visual Studio 2008. Además existe una versión más reducida de este motor de base de datos “SQL Express Edition”, que se distribuye de manera gratuita, y que es perfecto para las necesidades del proyecto.

2.4 Hibernate

Hibernate es una tecnología que se utiliza para poder conectar diversas aplicaciones, ya sean Web o de cualquier otro tipo con una base de datos, lo que proporciona una capa intermedia, abstrayendo a las clases que son utilizadas por la aplicación, de cómo están estructuradas en la base de datos. Hibernate, permite realizar una mapeo objeto relacional, para así de esta forma poder transformar tablas de una base de datos a un modelo de objetos definido en la aplicación. Hibernate está disponible para Java y .NET.

Hibernate permite a la aplicación manipular los datos, que se encuentra en la base de datos, operando sobre ellos como si se trataran de objetos, con todas las características de la programación orientada a objetos. Hibernate convierte los datos entre los tipos utilizados por C# y los definidos por SQL. Hibernate genera las sentencias SQL, de inserción, modificación y consulta de datos, proporcionando libertad al desarrollador del manejo manual de los datos que resultan de la ejecución de dichas sentencias, manteniendo la portabilidad entre todos los motores de bases de datos con un ligero incremento en el tiempo de ejecución.

Con Hibernate, se pueden realizar todas las acciones típicas, que se realizan con una base de datos, como puede ser insertar, eliminar, y actualizar datos. Además, las restricciones que se puedan definir en la base de datos, como por ejemplo relaciones entre tablas, resultaran independientes en el mapeo realizado por los programadores a través de los ficheros XML.

Siempre que se realice una acción a través de Hibernate que no cumpla alguna restricción, definida en la base de datos, producirá error el cual deberá ser tratado en la aplicación.

Hibernate ofrece un lenguaje de consulta de datos denominado HQL (Hibernate Query Language), al mismo tiempo que una API para construir las consultas programáticamente (conocida como "criteria").

2.5 AJAX

AJAX, son las iniciales de Asynchronous JavaScript And XML que es una forma de desarrollo Web para crear aplicaciones interactivas. AJAX no constituye una tecnología en sí, sino que es un término que engloba a un grupo de éstas que trabajan conjuntamente. AJAX combina las tecnologías:
• XHTML y hojas de estilo en cascada (CSS) para el diseño que acompaña a la información.
• Document Object Model (DOM) accedido con un lenguaje de scripting por parte del usuario, especialmente implementaciones ECMAScript como JavaScript y JScript, para mostrar e interactuar dinámicamente con la información presentada.
• Utiliza el objeto XMLHttpRequest para intercambiar datos asincrónicamente con el servidor web. En algunos frameworks y en algunas situaciones concretas, se usa un objeto iframe en lugar del XMLHttpRequest para realizar dichos intercambios.

Las aplicaciones en AJAX se ejecutan en el cliente y mantienen una comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre la misma página sin necesidad de recargarla. Un esquema del funcionamiento de AJAX podemos observarlo en la ilustración 2.

Como puede verse en la ilustración 2, las aplicaciones que utilizan AJAX, no necesariamente tiene que ir al servidor para mostrar datos nuevos, o validar información introducida en la página, si no que estas aplicaciones que utilizan AJAX, cargan en una capa intermedia la información que modificará en la pantalla a través de Java Script. Por otro lado en la parte derecha de la imagen, se ve como se comportaría si el sitio Web no contara con AJAX, cada petición que se realice tiene que pasar previamente por el servidor para mostrar información nueva en la página.
Una aplicación AJAX reduce el tiempo de recarga de la página mediante el motor AJAX que se sitúa entre el cliente y el servidor. En vez de cargar un pagina Web, al inicio de la sesión, el navegador carga al motor AJAX que es responsable de renderizar la interfaz que el usuario ve y de comunicarse con el servidor en nombre del usuario. El motor AJAX permite que la interacción del usuario con la aplicación se ejecute de forma asíncrona, de esta manera se consigue que el usuario casi no note como la página con la que está interactuando refresque, produciendo así una mayor velocidad en todas las acciones que se realizan con dicha página.

Ilustración 3: Esquema funcionamiento AJAX

Como se observa en la ilustración 3 cada acción de un usuario, que normalmente generaría una petición HTTP, toma la forma de una llamada Java Script al motor AJAX, en lugar de realizar una petición completa. Cualquier respuesta a una acción del usuario que no requiera un viaje de vuelta al servidor es manejada por el motor AJAX.

Si el motor de AJAX necesita algo del servidor para responder, hace esas solicitudes asincrónicamente, lo que permite no frenar la interacción del usuario con la aplicación.
2.6 Log4Net

Para aplicaciones de gran tamaño, es recomendable el poder contar con alguna librería o software que permita trazar los eventos o posibles errores que se producen en ellas con el fin de mejorar y optimizar los sistemas, o bien solventar problemas internos que no son mostrados a los usuarios con el fin de no alterar la interacción con dichos usuarios. Por este motivo, existe una librería para .NET, denominada Log4Net, la cual permite realizar Logs, gestionándolos de una manera automática. Esta librería permite realizar Logs de diversas maneras como pueden ser ficheros de texto, MS SQL, Oracle, SQL Lite, SMTP, etc.

Log4net, permite abstraer al desarrollador que está utilizando dicha librería y permite introducir mensajes de diversos motivos, como pueden ser, entradas de errores, informativos, fatales, etc. Además automáticamente es capaz de generar copias de estos ficheros con un determinado tamaño dividiendo las informaciones en periodos de tiempo establecidos por los desarrolladores.

2.7 Aplicaciones para la gestión de recursos

2.7.1 Herramienta de Ayuda a la Explotación de Equipos de la red de Unión Fenosa

En primer lugar, se muestra la aplicación denominada Herramienta de ayuda a la Explotación (HAE), cuyo objetivo principal es el de la gestión de activos dentro de Unión Fenosa. Esta aplicación prolonga la vida del proyecto SPARTA, ya que además de otras opciones, da la posibilidad de realizar medidas sobre los transformadores de potencia una vez hayan sido puestos en servicio. De esta manera las medidas realizadas en SPARTA valdrán como primera medida para los transformadores activos que existen dentro de la red de Unión Fenosa.

![Ilustración 4: Introducción de medidas de transformadores aplicación HAE](image)
Además de las medidas de transformadores (ilustración 4) la herramienta HAE permite realizar mediciones sobre otros elementos que también forman parte de la red eléctrica, como pueden ser interruptores, reguladores, etc. Esta herramienta, tiene un enfoque más administrativo ya que el objetivo de estas medidas es el de realizar correcciones sobre estos equipos dando una probabilidad de fallo antes de que se produzcan.

En este caso los equipos no son dados de alta manualmente en el sistema si no que se realizan a partir de una importación de ficheros Excel. Es una aplicación Web, realizada con el Framework 2.0 en su versión inicial, aunque posteriormente fue migrada al 3.5. Utiliza Hibernate, para comunicarse con un motor de base de datos SQL Server.

### 2.7.2 Gestión Vial de Madrid

La otra aplicación a la cual se quiere hacer referencia, está enfocada en su totalidad a la gestión de incidencias producidas en las carreteras dependientes de la Comunidad de Madrid. Al contrario de lo que sucedía con la aplicación anterior, esta está realizada íntegramente con el Framework 3.5 de .NET, utilizando además Hibernate y SQL Server como motor de base de datos y Log4net para la gestión de LOGS.

Esta aplicación, es utilizada básicamente, para registrar toda la información que se produce en las distintas carreteras gestionadas por la Comunidad de Madrid, ya sea por accidentes, retenciones, inclemencias meteorológicas, o incluso por las llamadas de usuarios pidiendo información del estado de una carretera.
Toda esta recopilación de información, permite disponer en todo momento de los sucesos que se han producido, y que se están produciendo en tiempo real, en diversos lugares, de tal manera que los usuarios pueden estar informados de la situación de las carreteras sin encontrarse en el centro de control.

Las dos aplicaciones mostradas en estos puntos, no cubren todas las necesidades de la aplicación que se define en esta memoria. Esto es debido a lo específico del sistema ya que trata de transformadores de potencia. Por este motivo surge la necesidad de realizar un sistema nuevo, para cubrir todas las necesidades que se verán en los puntos sucesivos.
Capítulo 3: Descripción Informática

Tras haber finalizado un estudio previo del estado del arte y analizado las distintas opciones en cuestión de tecnologías y entornos de desarrollo que han sido utilizados en el desarrollo de este proyecto, se pasa a describir el problema en cuestión y la solución técnica aportada con el fin de cubrir los objetivos marcados, en el momento del comienzo del proyecto.

3.1 Descripción del Problema

La aplicación que se pretende desarrollar trata del alta de equipos de transformadores de potencia de una manera controlada y eficiente de tal forma que estos datos sean accesibles en el futuro y sirvan de experiencia para futuras altas de equipos. Esta información únicamente podrá ser visualizada o modificada por personas que tengan permiso para hacerlo.

La aplicación debe permitir realizar una serie de medidas que son obligatorias, para poder poner en marcha un transformador de potencia, de manera que cumpla los requisitos establecidos por diferentes normativas vigentes. Estas medidas que se realizan, son muy dependientes, tanto en forma, como en límite, de los distintos parámetros del tipo de transformador al que se realizan, que van en función de las necesidades que tienen que cubrir, por este motivo, los límites para que las medidas puedan considerarse correcta son muy variables de unos modelos a otros. En base a esta situación, la aplicación debe realizar un “diagnóstico”, definiendo si el resultado obtenido es o no el correcto. Estas medidas además serán de distinta manera en función si son realizadas en el lugar de fabricación del transformador, o si son realizada en la ubicación final en la que se ponga en marcha el equipo medido.

Existirá una sección de consultas que permitan al usuario obtener los datos referentes a los componentes y medidas asociados a cualquier transformador dado de alta en el sistema. Esta consulta podrá realizarse por los datos del propio transformador, o por datos de cualquiera de las medidas realizadas en el mismo.

Por otro lado, las personas encargadas de analizar los resultados obtenidos en las medidas, deben dar validez a los resultados basándose en su experiencia, aparte de saber si los
 datos obtenidos son correctos en función de los límites obtenidos por la norma. Para poder dar esta validez se basan en experiencias pasadas de otros transformadores que por su arquitectura son parecidos. Para este motivo, la aplicación debe implementar una sección de comparativas en las que poder comparar diferentes transformadores de una manera sencilla.

Para la implementación de la sección de comparativas se tendrá que generar una página en la que se muestren los datos que se desean comparar, ya sea de transformadores o de medidas.

La aplicación permitirá generar informes, que contendrán todos los datos del transformador, ya sean datos del propio equipo o datos de las medidas realizadas sobre este, de manera que todas las acciones realizadas al transformador puedan ser presentadas, bien a personas que no tengan acceso a la herramienta o bien que quieran ser impresas por algún motivo.

Asimismo, es necesario disponer de una versión “OFFLINE”, que permita trabajar sin tener conexión a la red de Unión Fenosa, pudiendo tener todos los datos del sistema disponibles y dando la posibilidad de trabajar igual que si se estuviera trabajando con la versión “ONLINE”. Una vez introducidos estos datos en la versión OFFLINE, bastará con transferirlos a la aplicación que se encuentra dentro de la red.

Por último y como se ha dicho al principio, la aplicación únicamente será accesible por el personal que este designado a las distintas acciones que han sido descritas, por lo que se asignarán perfiles a los distintos usuarios que formen parte de la aplicación, y que hayan sido dados de alta previamente. Estos perfiles serán definidos en la propia aplicación asumiendo las diferentes acciones de las que se disponen en el sistema.
3.2 Especificación de requisitos

Una vez descrito el problema los requisitos que hemos obtenido son los siguientes:

Tabla 1: Requisitos Funcionales

<table>
<thead>
<tr>
<th>Número de Requisito</th>
<th>Nombre de Requisito</th>
<th>Descripción</th>
</tr>
</thead>
<tbody>
<tr>
<td>RF001</td>
<td>Autentificación</td>
<td>El sistema únicamente podrá ser accedido por aquellas personas que estén autorizadas para ello. Por este motivo, antes de empezar a utilizarlo deberá introducir un nombre de usuario y una contraseña.</td>
</tr>
</tbody>
</table>
| RF002               | Perfiles de Usuarios    | Se definen perfiles, los cuales contendrán los permisos, para realizar acciones, que dispone cada usuario. Las acciones serán:  
  - Administrador del sistema (permite modificar datos después de cerrar un transformador)  
  - Administrador de usuarios (permite la gestión de los usuarios)  
  - Administrador de Exportación (permite exportar datos del sistema)  
  - Administrador de Importación (permite importar datos al sistema)  
  - Dar de alta transformadores (permite dar de alta transformadores)  
  - Visualizar (permite visualizar todos los datos)  
  - Visualizar ensayos de campo (permite visualizar ensayos de campo)  
  - Visualizar Histórico (permite visualizar las repeticiones de los ensayos.) |
<p>| RF002.1             | Alta de Perfiles        | Se darán de alta perfiles, para poder asignárselos a los distintos usuarios dados de alta en la aplicación. |
| RF002.2             | Modificar Perfiles      | En cualquier momento se podrá modificar un perfil concreto.                    |
| RF002.3             | Eliminar Perfiles       | Se podrán eliminar los perfiles creados.                                      |
| RF003               | Gestión de Usuarios     | Para poder acceder a la aplicación se puede dar de alta al usuario.          |</p>
<table>
<thead>
<tr>
<th>Número de Requisito</th>
<th>Nombre de Requisito</th>
<th>Descripción</th>
</tr>
</thead>
<tbody>
<tr>
<td>RF003.1</td>
<td>Alta de usuarios</td>
<td>Para dar de alta un usuario, se introducirá un identificador que será único en el sistema, una contraseña, nombre y apellidos y el perfil que posee.</td>
</tr>
<tr>
<td>RF003.2</td>
<td>Modificar usuarios</td>
<td>Todos los datos de usuario, identificador, contraseña, nombre, apellidos y perfil podrán ser modificados.</td>
</tr>
<tr>
<td>RF003.3</td>
<td>Eliminar usuario</td>
<td>En cualquier momento se podrán eliminar usuarios, para impedir que accedan a la aplicación.</td>
</tr>
<tr>
<td>RF004</td>
<td>Alta de Transformadores</td>
<td>Los usuarios podrán dar de alta transformadores de potencia en la aplicación, para ello deberán llenar todos los campos obligatorios, para que quede definido dentro del sistema. Los campos obligatorios son: número de serie, fabricante, ubicación, potencia, tensión, posee regulador, refrigeración y arrollamientos.</td>
</tr>
<tr>
<td>RF005</td>
<td>Modificar datos de transformadores</td>
<td>Se podrán modificar los datos de los transformadores una vez dados de alta en la aplicación. Así como los datos de componentes del mismo. Los componentes son, regulador, bornas y especificaciones.</td>
</tr>
<tr>
<td>RF006</td>
<td>Cambiar estado de los transformadores</td>
<td>Los transformadores podrán estar en diferentes estados, en función de la situación en la que se encuentre, los estados son los siguientes:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• En fábrica</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Terminado en fábrica</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• En campo</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Terminado en campo</td>
</tr>
<tr>
<td>RF007</td>
<td>Introducir medidas de transformadores</td>
<td>Se permite introducir medidas a los transformadores dados de alta en el sistema.</td>
</tr>
<tr>
<td>RF007.1</td>
<td>Medidas de Fabrica</td>
<td>Si el transformador se encuentra en estado en fábrica, o en estado terminado en fábrica se podrán introducir medidas de este tipo. Las medidas podrán ser de los siguientes tipos:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Medida de resistencia de aislamiento</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Medida de relación de transformación</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Medida de resistencia de arrollamientos</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Ensayo de impedancia homopolar</td>
</tr>
<tr>
<td>Número de Requisito</td>
<td>Nombre de Requisito</td>
<td>Descripción</td>
</tr>
<tr>
<td>---------------------</td>
<td>---------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| RF007.2             | Medidas de Campo          | Si el transformador se encuentra en estado en campo, o en estado terminado en campo se podrán introducir medidas de este tipo. Estas medias podrán ser introducidas de manera manual o importando un fichero que contiene los datos. La manera en la que introducir los datos dependerá del tipo de medida. Las medidas podrán ser de los siguientes tipos:  
  • Análisis de Gases  
  • Análisis Físico-Químico  
  • Análisis de PCBS  
  • Resistencia Arrollamientos en Campo  
  • Medida de capacidad/tg delta  
  • Medida de capacidad en bornas capacitivas  
  • Medida de test de excitación  
  • Medida de relación de transformación |
| RF007.3             | Modificar Medidas de Fábrica y de Campo | Se podrán modificar los campos introducidos en las medidas así como, modificar su estado.  
  • Abierto  
  • Cerrado |
<p>| RF007.4             | Eliminar Medidas          | Se podrán eliminar medidas asociadas al transformador, siempre que se tenga permiso para realizarlo. |
| RF007.5             | Generar Gráficas          | En algunos ensayos será posible generar una serie de gráficas par visualizar los datos introducidos de una manera gráfica. |
| RF008               | Generar Informes          | Se debe generar un informe del transformador desde el mismo momento |</p>
<table>
<thead>
<tr>
<th>Número de Requisito</th>
<th>Nombre de Requisito</th>
<th>Descripción</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>en el que esté sea dado de alta en el sistema. Que contendrá todos los datos asociados al transformador.</td>
</tr>
<tr>
<td>RF009</td>
<td>Consultas de datos de transformadores y ensayos</td>
<td>Se podrán generar las consultas que se deseen, tanto de transformadores como de medidas realizadas sobre dichos equipos. Para ello se deberán seleccionar unos parámetros para así poder acotar la consulta.</td>
</tr>
<tr>
<td>RF010</td>
<td>Comparativas de datos de transformadores y ensayos</td>
<td>Se pueden realizar comparativas, o bien por transformadores, o bien por ensayos.</td>
</tr>
</tbody>
</table>

**Tabla 2: Requisitos No Funcionales**

<table>
<thead>
<tr>
<th>Número de Requisito</th>
<th>Nombre de Requisito</th>
<th>Descripción</th>
</tr>
</thead>
<tbody>
<tr>
<td>RNF001</td>
<td>Disponibilidad</td>
<td>El sistema tiene que estar disponible en todo momento debido al tipo de información que contiene. Para ello se tiene que contar con una versión cuando no se disponga de conexión a la intranet de Unión Fenosa.</td>
</tr>
<tr>
<td>RNF002</td>
<td>Documentación</td>
<td>El sistema tendrá que estar documentado, con un manual de usuario, ya que dispone de muchas alternativas.</td>
</tr>
<tr>
<td>RNF003</td>
<td>Seguridad</td>
<td>Únicamente los usuarios que posean permiso, podrán acceder al sistema.</td>
</tr>
<tr>
<td>RNF004</td>
<td>Extensibilidad</td>
<td>El sistema estará preparado para ser ampliado en caso, de que se introdujeran nuevas necesidades por parte de los usuarios.</td>
</tr>
<tr>
<td>RNF005</td>
<td>Usabilidad de la Interfaz Web</td>
<td>La interfaz Web deberá ser sencilla e intuitiva.</td>
</tr>
<tr>
<td>RNF005.1</td>
<td>Navegabilidad</td>
<td>El usuario podrá en todo momento acceder a toda aquella parte de la aplicación para la que tenga permisos, rápidamente y sin perderse en las opciones disponibles.</td>
</tr>
<tr>
<td>RNF005.2</td>
<td>Botones y colores</td>
<td>Tanto los botones como los colores que se utilicen en el sistema, deberán ser agradables para el usuario y que dejen claro las acciones que se realizarán tras utilizarlos.</td>
</tr>
<tr>
<td>RNF005.3</td>
<td>Mensajes y nombres descriptivos</td>
<td>Los mensajes que se muestren en la aplicación, deberán ser claros definiendo exactamente lo que desean exponer al usuario.</td>
</tr>
</tbody>
</table>
Tabla 3: Requisitos Hardware/Software

<table>
<thead>
<tr>
<th>Número de Requisito</th>
<th>Nombre de Requisito</th>
<th>Descripción</th>
</tr>
</thead>
<tbody>
<tr>
<td>RNF005.4</td>
<td>Situación</td>
<td>El usuario deberá saber en todo momento en la página de la aplicación en la que se encuentra. Para ello todas las pantallas deberán contener el título a la funcionalidad que implementa la página.</td>
</tr>
<tr>
<td>RH001</td>
<td>Servidor</td>
<td>Se dispondrá de un servidor que contará con un disco duro en espejo. Tendrá instalado una licencia de Windows XP.</td>
</tr>
<tr>
<td>RH002</td>
<td>Base de datos</td>
<td>Se dispondrá dentro del servidor de un motor de Base de Datos SQL Server.</td>
</tr>
<tr>
<td>RH003</td>
<td>Servidor Web</td>
<td>Se contará con un IIS el cual alojará la aplicación Web.</td>
</tr>
<tr>
<td>RH004</td>
<td>Red</td>
<td>Existirán dos bocas de Ethernet las cuales están asociadas a dos IPs diferentes para dar una mayor accesibilidad al sistema. Estas IPs deberán ser accesibles desde dentro de la Intranet de UF.</td>
</tr>
</tbody>
</table>

3.3 Análisis

Una vez especificados los requisitos que debe cumplir el sistema, se pasa a detallar la parte de análisis con la descripción de los casos de uso. Los actores principales del sistema van a ser los siguientes:

- **El encargado de realizar las medidas de los transformadores**, que es la persona encargada de dar de alta los transformadores, introducir las medidas realizadas y realizar la validación de los transformadores para poder ponerlos en funcionamiento.
- **El usuario de consulta**, será aquella persona que únicamente entre al sistema a consultar datos ya introducidos y validados, únicamente podrá realizar acciones de consulta y visualización.
- **Por último se encuentra el administrador de usuarios**, que será aquella persona encargada de dar los permisos correspondientes a cada uno de los usuarios dados de alta en la aplicación.

Un diagrama de casos de uso muestra la relación entre los actores y los casos de uso del sistema. Representa la funcionalidad que ofrece el sistema en lo que se refiere a su interacción externa. A continuación se muestran, en las ilustraciones 7 y 8 los diagramas de uso definidos para los actores mostrados anteriormente.
- **Dar de alta transformador:** El encargado podrá dar de alta un transformador dentro del sistema, para ello deberá seleccionar todos los campos necesarios, que será todos los campos de los que está compuesto un transformador, como pueden ser:
  - Tipo de transformadora, lo que influirá en todos los demás aspectos.
  - Número de serie.
  - Año de fabricación.
  - Lugar de fabricación.
  - Fabricante.
  - Datos bases de ese tipo de transformador.
  - Potencia.
  - Tipo de refrigeración.
  - Si posee o no regulación.
  - Datos de los devanados de los que está compuesto el transformador.

- **Modificar transformador:** Una vez dado de alta el transformador, se podrá modificar parte de estos datos introducidos anteriormente, además existirá la posibilidad de modificar otros parámetros del transformador, que no son requeridos en el momento del alta como son las bornas que posee y los datos específicos del regulador.

- **Cambiar el estado del transformador:** El encargado de gestionar los transformadores, modificará el estado del transformador en función de en qué proceso de fabricación se encuentre, los estados podrán ser en fábrica, terminado en fábrica, en campo y terminado en campo.

- **Informes de transformador:** En cualquier momento el encargado podrá realizar un informe en el cual muestre todos los datos del transformador y las medidas que crea necesarias.
Medidas de transformadores: Dependiendo el estado en el que se encuentre el transformador se pueden introducir medidas de distinto tipo:

- **En fábrica o terminado en fábrica:**
  - Medida de resistencia de aislamiento
  - Medida de relación de transformación
  - Medida de resistencia de arrollamientos
  - Ensayo de impedancia homopolar
  - Ensayo Regulador
  - Ensayo de medida de ruido
  - Ensayo de Tensión Aplicada
  - Ensayo de Tensión Inducida sin DDPP
  - Ensayo de Tensión Inducida
  - Ensayo de vacío
  - Medida de pérdidas en carga
  - Ensayo de tipo maniobra
  - Ensayo de impulso tipo rayo
  - Ensayo de calentamiento
  - Medida de capacidad/tg delta

- **En campo o terminado en campo:**
  - Análisis de Gases
  - Análisis Fisico-Quimico
  - Análisis de PCBS
  - Resistencia Arrollamientos en Campo
  - Medida de capacidad/tg delta
  - Medida de capacidad en bornas capacitivas
  - Medida de test de excitación
  - Medida de relación de transformación

Cada una de estas medidas contendrá una serie de campos u otros en función de los parámetros que hayan sido definidos en el transformador. Además los límites de estas medidas también tendrán variaciones dependiendo de parámetros, como el tipo de transformador y la refrigeración del mismo.

Las medidas podrán ser dadas de alta en el sistema, modificadas o eliminadas. También será posible modificar el estado, dando la posibilidad de que una medida se encuentre abierta o cerrada, si ya se han realizado todas las acciones correspondientes sobre ella.
Ilustración 8: Caso de Uso Administrador de usuarios, usuario de consulta.

- **Perfiles**: El administrador de usuarios llevará a cabo la gestión de perfiles dentro del sistema, más en concreto, podrá crear un nuevo perfil, modificar o eliminar uno ya existente.

- **Usuarios**: El igual que sucede con los perfiles, el administrador, tendrá la posibilidad de crear un nuevo usuario para que acceda al sistema, modificar o eliminar usuarios existentes.

- **Autenticación**: Para que cualquier usuario acceda a la aplicación web deberá previamente logarse introduciendo su nombre de usuario y contraseña, para poder...
realizar esto el administrador de usuarios, deberá haber dado de alta este usuarios dentro del sistema.

- **Consulta:** Los usuarios de que accedan para realizar consultas en el sistema podrán hacerlo o bien para consultar transformadores o bien para realizar consultas sobre las medidas realizadas a estos equipos. Para poder realizarlas deberán seleccionar previamente una serie de parámetros que les facilitarán la búsqueda.

- **Comparativa:** Cuando un usuario acceda al sistema podrá realizar comparativas sobre transformadores introducidos en el sistema o sobre medidas de de un mismo transformador.
3.4 Diseño

A continuación realizará el diseño del sistema que incluirá tanto la interfaz de usuario, el diseño de la base de datos y la estructura software utilizada.

3.4.1 Interfaz de usuario

El diseño de la interfaz de este tipo de aplicaciones, hará que los usuarios que la utilizan puedan realizar su trabajo de una forma más rápida ya que facilitará de una manera considerable sus funciones. Para ello se ofrecerá al usuario una forma óptima de realizar todas las interactuaciones con la aplicación, haciendo fácil la adaptación de cualquiera de ellos al funcionamiento de la misma, haciendo además que la utilización de esta aplicación sea plenamente satisfactoria. Por todos estos motivos la aplicación tiene que cumplir los siguientes objetivos:

a) **Simplicidad**: Los usuarios que utilizan este tipo de aplicaciones, que son complicadas por su finalidad en sí, buscan una herramienta que no incremente esta complejidad. Por lo tanto, esperan aplicaciones sencillas orientadas a las tareas que deben realizar pero que sean sencillas de utilizar. Si una aplicación presenta muchos menús, vistas y controles, puede confundir y frustrar al usuario que desee realizar una tarea con rapidez. Este punto soluciona el requisito no funcional RNF001 Usabilidad de la interfaz.

b) **Coherencia**: Consigue que una interfaz sea familiar y predecible, reduce la confusión y aumenta la productividad. Nuestra aplicación debe tener coherencia por varios motivos:
   i. **Facilidad de aprendizaje**: Los usuarios en un primer vistazo deben de ser capaces de saber la funcionalidad de la que disponen al utilizar esta aplicación.
   ii. **Mayor comodidad y confianza**: Los usuarios valoran la sencillez y la familiaridad en la ejecución de las tareas.
   iii. **Mayor productividad**: Con un tiempo de aprendizaje más breve, los usuarios pasan menos tiempo intentando aprender a realizar las tareas y se ponen a trabajar enseguida.

Este punto soluciona el requisito no funcional RNF005.1 Navegabilidad.

c) **Estética**: La estética, aunque subjetiva y difícil de medir, es casi tan importante como la ejecución de las funciones propias de una aplicación. El aspecto de la interfaz puede afectar a la correcta ejecución de las tareas por parte del usuario, con independencia de la eficacia de la propia aplicación. Si un usuario no está satisfecho con el aspecto o la interacción con una aplicación, se sentirá incómodo con ella, con independencia de lo buena que ésta sea en las pruebas de prestaciones. Este punto soluciona el requisito no funcional RNF005.2 Botones y colores.

d) **Comentarios**: Los usuarios necesitan información clara e inmediata sobre sus acciones, que deben recibir cuanto antes. Esta información les ayuda a saber qué ocurre con la aplicación, sus datos, el dispositivo y el resultado de sus acciones, sean éstas deliberadas o accidentales. Este punto soluciona el requisito no funcional RNF005.3 Mensajes y nombres descriptivos.
e) **Situación del usuarios:** En aplicaciones con gran número de pantallas y algunas de ellas bastante parecidas, es necesario que el usuario pueda ubicarse en todo momento en que sección de ella se encuentra. Esto permitirá al usuario cometer menos errores y por lo tanto poder realizar las tareas determinadas en un menor tiempo. Para solucionar este problema, en todo momento existirá un título que contendrá la sección de la aplicación en la que se encuentra el usuario. Este punto soluciona el requisito no funcional RNF005.4 Situación.

Una vez vistas las características que debe cumplir la interfaz gráfica del sistema, se muestran las secciones de la interfaz más interesantes desde el punto de vista de diseño, en las cuales se puede observar cómo se cumplen los puntos desarrollados anteriormente. No se van a mostrar todas las pantallas realizadas en el sistema, si no que se van a mostrar aquellas más representativas.

En la ilustración 9 se muestra la plantilla básica que seguirá todas las pantallas de la aplicación, y que será a su vez común para todos los usuarios.

![Ilustración 9: Plantilla de las páginas](image)

<table>
<thead>
<tr>
<th>Código</th>
<th>Fabricante</th>
</tr>
</thead>
<tbody>
<tr>
<td>201</td>
<td>A.B &amp; C</td>
</tr>
<tr>
<td>202</td>
<td>D.E &amp; F</td>
</tr>
<tr>
<td>203</td>
<td>G.H &amp; I</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Código</th>
<th>Fabricante</th>
</tr>
</thead>
<tbody>
<tr>
<td>201</td>
<td>A.B &amp; C</td>
</tr>
<tr>
<td>202</td>
<td>D.E &amp; F</td>
</tr>
<tr>
<td>203</td>
<td>G.H &amp; I</td>
</tr>
</tbody>
</table>

Sección de cabecera de las páginas en la que se puede observar el nombre del usuario, si se encuentra en la aplicación, ONLINE/OFFLINE, y la fecha actual.

Además puede observarse el nombre de la aplicación y el logo de Unión Fenosa.
En la ilustración 10 se puede ver el detalle del menú, a partir del cual se podrá navegar a todas las secciones de la aplicación. Estará siempre visible y permitirá al usuario moverse por las diferentes páginas en cualquier momento. El menú se muestra dividido en los bloques principales, que están definidos y explicado en la ilustración 9.

Como puede verse en la imagen de la parte superior derecha, el menú está dividido en tres grandes bloques, el primero de ellos denominado General, contendrá información general del usuario y el manual de usuario. La segunda sección, denominada Administración, hará referencia a la seguridad de la aplicación, como puede verse con más detalle en la imagen de la parte inferior derecha. Por último, la sección denominada SPARTA, contendrá toda la funcionalidad que posee la herramienta.

Ilustración 10: Detalle acciones menú

Por último se muestra la pantalla principal del transformador (ilustración 11) en la que se pueden modificar todos los datos del transformador, añadir medidas sobre el mismo, navegar hacia las medidas ya introducidas, etc.

La pantalla del transformador, que puede verse en la ilustración 11, está organizada en las diferentes secciones por las que está compuesto el transformador, que son, datos generales, arrollamientos, regulador, bornas y especificaciones. Además en la parte inferior está la sección en la que se introducen todos los ensayos que pueden asociarse al transformador. Por otro lado, en la parte superior de la pantalla se encuentran todos los botones para realizar acciones sobre el transformador.
Ilustración 11: Pantalla principal del transformador
3.4.2 Base de datos

El motor de base de datos utilizado ha sido SQL Server, el cual dispone de diversas distribuciones, pero en el caso de este proyecto se decidió utilizar la distribución libre denominada SQL Server Express. La base de datos se ha denominado GETUF_DB. A continuación, debido al gran número de tablas que existen en el sistema, únicamente se describen aquellas que tienen mayor importación dentro del mismo. El resto de tablas se mostrarán gráficamente en el diagrama de tablas posterior.

**Tabla Usuario:** almacena los datos relativos al personal de Unión Fenosa, es decir, los usuarios que pueden tener acceso al sistema.

<table>
<thead>
<tr>
<th>Campo</th>
<th>Tipo</th>
<th>Nulo</th>
</tr>
</thead>
<tbody>
<tr>
<td>id_usu(PK)</td>
<td>nvarchar(20)</td>
<td>No</td>
</tr>
<tr>
<td>id_perfil(FK)</td>
<td>nvarchar(20)</td>
<td>No</td>
</tr>
<tr>
<td>nombre</td>
<td>nvarchar(50)</td>
<td>No</td>
</tr>
<tr>
<td>apellido1</td>
<td>nvarchar(50)</td>
<td>No</td>
</tr>
<tr>
<td>apellido2</td>
<td>nvarchar(50)</td>
<td>Sí</td>
</tr>
<tr>
<td>nif</td>
<td>nvarchar(20)</td>
<td>No</td>
</tr>
<tr>
<td>teléfono</td>
<td>nvarchar(20)</td>
<td>Sí</td>
</tr>
<tr>
<td>email</td>
<td>nvarchar(256)</td>
<td>No</td>
</tr>
<tr>
<td>cargo</td>
<td>nvarchar(50)</td>
<td>Sí</td>
</tr>
<tr>
<td>password</td>
<td>nvarchar(100)</td>
<td>No</td>
</tr>
<tr>
<td>id_jefe(FK)</td>
<td>nvarchar(20)</td>
<td>Sí</td>
</tr>
<tr>
<td>id_jefe2(FK)</td>
<td>nvarchar(20)</td>
<td>Sí</td>
</tr>
<tr>
<td>id_jefe3(FK)</td>
<td>nvarchar(20)</td>
<td>Sí</td>
</tr>
<tr>
<td>activo</td>
<td>bit</td>
<td>No</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Campo</th>
<th>Tipo</th>
<th>Nulo</th>
</tr>
</thead>
<tbody>
<tr>
<td>id_perf(PK)</td>
<td>nvarchar(20)</td>
<td>No</td>
</tr>
<tr>
<td>nombre</td>
<td>nvarchar(50)</td>
<td>No</td>
</tr>
<tr>
<td>descripción</td>
<td>text</td>
<td>No</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Campo</th>
<th>Tipo</th>
<th>Nulo</th>
</tr>
</thead>
<tbody>
<tr>
<td>id_jefe</td>
<td>nvarchar(20)</td>
<td>No</td>
</tr>
</tbody>
</table>

**Tabla Perfil:** almacena los perfiles que se van a asignar a cada usuario.

**Tabla Página:** almacena las acciones que van a poder realizarse dentro de la aplicación. Estas acciones serán definidas previamente y no serán editables desde la propia aplicación.
Tabla Perfil Página: Almacena las asociaciones que se definen en el momento de dar de alta un perfil.

Tabla 7: Tabla relación perfil página

<table>
<thead>
<tr>
<th>Campo</th>
<th>Tipo</th>
<th>Nulo</th>
</tr>
</thead>
<tbody>
<tr>
<td>id_pag(FK)</td>
<td>nvarchar(10)</td>
<td>No</td>
</tr>
<tr>
<td>id_perf(FK)</td>
<td>nvarchar(20)</td>
<td>No</td>
</tr>
</tbody>
</table>

A continuación se muestra el diagrama de tablas que define la seguridad de la aplicación:

Ilustración 12: Diagrama de tablas de seguridad
Una vez vistas las tablas y el diagrama de tabla que gestionaran a la seguridad y los usuarios de la aplicación, paso a mostrar las tablas referentes a los transformadores de potencia.

**Tabla Trafo:** es la tabla “principal” de la aplicación, sobre ella cuelga el resto de elementos que se describen de aquí en adelante. En ella estarán contenidos todos los datos principales que forman parte de este equipo, dichos campos son características que serán determinantes en el momento de determinar su funcionamiento. (Debido al gran número de campos que forman parte de esta tabla únicamente se describen los más importantes, el resto podrá verse en el diagrama)

<table>
<thead>
<tr>
<th>Campo</th>
<th>Tipo</th>
<th>Nulo</th>
<th>Descripción</th>
</tr>
</thead>
<tbody>
<tr>
<td>id(PK)</td>
<td>int</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>idGrupoConexion(FK)</td>
<td>int</td>
<td>No</td>
<td>Estructura interna del trafo</td>
</tr>
<tr>
<td>idFabricante(FK)</td>
<td>int</td>
<td>No</td>
<td>Quien construye el trafo</td>
</tr>
<tr>
<td>NSerie</td>
<td>nvarchar(50)</td>
<td>No</td>
<td>Indica el identificador que va a tener el trafo a lo largo de toda su vida.</td>
</tr>
<tr>
<td>Destino</td>
<td>nvarchar(50)</td>
<td>Si</td>
<td></td>
</tr>
<tr>
<td>Ubicación(FK)</td>
<td>int</td>
<td>No</td>
<td>Donde se construye el transformador</td>
</tr>
<tr>
<td>Potencia(FK)</td>
<td>decimal(18,2)</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>AnioFabricacion</td>
<td>datetime</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>TipoRefrigeracion</td>
<td>nvarchar(10)</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>Trifasico</td>
<td>bit</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>Estrella</td>
<td>bit</td>
<td>No</td>
<td>Forma en la que está dispuesta la estructura del trafo</td>
</tr>
<tr>
<td>UbicRegulador</td>
<td>nvarchar(50)</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>EscalonInicial</td>
<td>int</td>
<td>Si</td>
<td></td>
</tr>
<tr>
<td>Estado</td>
<td>nvarchar(50)</td>
<td>No</td>
<td>Estado en el que se encuentra el trafo</td>
</tr>
<tr>
<td>dtAlta</td>
<td>datetime</td>
<td>No</td>
<td>Fecha de alta del transformador, contará con una por cada estado del trafo</td>
</tr>
<tr>
<td>IdUsrAlta(FK)</td>
<td>nvarchar(20)</td>
<td>No</td>
<td>Usuario que dio de alta el transformador, existe un campo por cada estado del trafo</td>
</tr>
<tr>
<td>PersonalPresente</td>
<td>ntext</td>
<td>Si</td>
<td></td>
</tr>
<tr>
<td>PersonalMedida</td>
<td>ntext</td>
<td>Si</td>
<td></td>
</tr>
<tr>
<td>UltimaModificacion</td>
<td>datetime</td>
<td>No</td>
<td>Ultima modificación del trafo, ya sea de los datos propios o de las medidas</td>
</tr>
<tr>
<td>IdEspecificacion(FK)</td>
<td>int</td>
<td>Si</td>
<td></td>
</tr>
<tr>
<td>UltimaModificacionOnline</td>
<td>datetime</td>
<td>No</td>
<td>Permite saber si se pueden sobrescribir datos en la importación</td>
</tr>
<tr>
<td>Publicado</td>
<td>int</td>
<td>No</td>
<td>Indica si se ha publicado la información para que otras aplicaciones importen sus datos</td>
</tr>
</tbody>
</table>
Tabla Tipo de Transformador: contiene los tipos de transformador que pueden definirse en la aplicación, cada uno de los campos, describen la estructura que posee el transformador, y que será determinante para saber las características de este equipo. Los tipos de transformadores están definidos previamente y no podrán darse de alta dinámicamente en la aplicación.

<table>
<thead>
<tr>
<th>Campo</th>
<th>Tipo</th>
<th>Nulo</th>
</tr>
</thead>
<tbody>
<tr>
<td>id(PK)</td>
<td>id</td>
<td>No</td>
</tr>
<tr>
<td>TipoTrafo</td>
<td>nvarchar(50)</td>
<td>No</td>
</tr>
<tr>
<td>Observ</td>
<td>nvarchar(50)</td>
<td>Sí</td>
</tr>
<tr>
<td>NumAT</td>
<td>int</td>
<td>No</td>
</tr>
<tr>
<td>NumBT</td>
<td>int</td>
<td>No</td>
</tr>
<tr>
<td>NumTer</td>
<td>int</td>
<td>No</td>
</tr>
<tr>
<td>NeutroAT</td>
<td>bit</td>
<td>No</td>
</tr>
<tr>
<td>NeutroBT</td>
<td>bit</td>
<td>No</td>
</tr>
<tr>
<td>NumTermTer</td>
<td>int</td>
<td>No</td>
</tr>
<tr>
<td>orden</td>
<td>int</td>
<td>No</td>
</tr>
</tbody>
</table>

Tabla Fabricante: con esta tabla, se almacenarán los datos básicos de los fabricantes de los equipo, tanto de los transformadores como de bornas y reguladores.

<table>
<thead>
<tr>
<th>Campo</th>
<th>Tipo</th>
<th>Nulo</th>
</tr>
</thead>
<tbody>
<tr>
<td>id(PK)</td>
<td>int</td>
<td>No</td>
</tr>
<tr>
<td>Nombre</td>
<td>nvarchar(100)</td>
<td>No</td>
</tr>
<tr>
<td>Descripción</td>
<td>nvarchar(100)</td>
<td>Sí</td>
</tr>
</tbody>
</table>

Tabla Ubicación: se almacenan las ubicaciones de los equipos que se vayan asociando, a los fabricantes por un lado y a los transformadores por el otro.

<table>
<thead>
<tr>
<th>Campo</th>
<th>Tipo</th>
<th>Nulo</th>
</tr>
</thead>
<tbody>
<tr>
<td>id(PK)</td>
<td>int</td>
<td>No</td>
</tr>
<tr>
<td>Localización</td>
<td>nvarchar(30)</td>
<td>No</td>
</tr>
<tr>
<td>Activa</td>
<td>bit</td>
<td>No</td>
</tr>
</tbody>
</table>

Tabla Relación Fabricante Ubicación: relaciona, los fabricantes con las ubicaciones. Un fabricante va asociado a una determinada ubicación, de manera que cuando un transformador esté fabricado por un fabricante, el transformador únicamente podrá tener las ubicaciones en las que se encuentre el fabricante.
Tabla 12: Tabla relación de fabricantes con ubicaciones

<table>
<thead>
<tr>
<th>Campo</th>
<th>Tipo</th>
<th>Nulo</th>
</tr>
</thead>
<tbody>
<tr>
<td>id(PK)</td>
<td>int</td>
<td>No</td>
</tr>
<tr>
<td>Id_fabricante(FK)</td>
<td>int</td>
<td>No</td>
</tr>
<tr>
<td>Id_ubicacion(FK)</td>
<td>int</td>
<td>No</td>
</tr>
</tbody>
</table>

Tabla Devanado: componente del transformador, almacenará una entrada por cada devanado del que esté compuesto el transformador. Existirán 3 tipos de devanados AT/BT/TER, el resto de campos del devanado son configurables en función del transformador.

Tabla 13: Tabla devanados

<table>
<thead>
<tr>
<th>Campo</th>
<th>Tipo</th>
<th>Nulo</th>
</tr>
</thead>
<tbody>
<tr>
<td>id(PK)</td>
<td>int</td>
<td>No</td>
</tr>
<tr>
<td>IdTrafo(FK)</td>
<td>int</td>
<td>No</td>
</tr>
<tr>
<td>Tipo</td>
<td>nvarchar(50)</td>
<td>No</td>
</tr>
<tr>
<td>Nombre</td>
<td>nvarchar(50)</td>
<td>No</td>
</tr>
<tr>
<td>Um</td>
<td>decimal(18,2)</td>
<td>Si</td>
</tr>
<tr>
<td>Pot_Onan</td>
<td>decimal(18,2)</td>
<td>Si</td>
</tr>
<tr>
<td>Pot_Odaf</td>
<td>decimal(18,2)</td>
<td>Si</td>
</tr>
<tr>
<td>Pot_Ofaf</td>
<td>decimal(18,2)</td>
<td>Si</td>
</tr>
<tr>
<td>Pot_An</td>
<td>decimal(18,2)</td>
<td>Si</td>
</tr>
</tbody>
</table>

Tabla Borna: componente de los devanados, el número de bornas de un transformador, estará definido por el número de devanados que posea y del tipo del transformador. Los datos de las bornas serán introducidos manualmente, por el contrario el número de bornas, será siempre el mismo respecto los devanados y el tipo de transformador, tal y como se ha dicho antes.

Tabla 14: Tabla borna

<table>
<thead>
<tr>
<th>Campo</th>
<th>Tipo</th>
<th>Nulo</th>
</tr>
</thead>
<tbody>
<tr>
<td>id(PK)</td>
<td>int</td>
<td>No</td>
</tr>
<tr>
<td>IdDevanado(FK)</td>
<td>int</td>
<td>No</td>
</tr>
<tr>
<td>Tipo</td>
<td>nvarchar(50)</td>
<td>No</td>
</tr>
<tr>
<td>IdFabricante(FK)</td>
<td>int</td>
<td>Si</td>
</tr>
<tr>
<td>CorrienteNominal</td>
<td>decimal(18,2)</td>
<td>Si</td>
</tr>
<tr>
<td>Modelo</td>
<td>nvarchar(50)</td>
<td>Si</td>
</tr>
<tr>
<td>Tensión</td>
<td>decimal(18,2)</td>
<td>Si</td>
</tr>
</tbody>
</table>

Tabla Terminal: dependiendo del tipo de transformador, cada borna tendrá un determinado número de terminales, las cuales definirán las fases en las que está dividida la borna. Esta tabla contendrá las características de la los terminales de las bornas del transformador.
Tabla 15: Tabla terminal

<table>
<thead>
<tr>
<th>Campo</th>
<th>Tipo</th>
<th>Nulo</th>
</tr>
</thead>
<tbody>
<tr>
<td>id(PK)</td>
<td>int</td>
<td>No</td>
</tr>
<tr>
<td>IdBorna(FK)</td>
<td>int</td>
<td>No</td>
</tr>
<tr>
<td>NSerie</td>
<td>nvarchar(50)</td>
<td>Si</td>
</tr>
<tr>
<td>Fase</td>
<td>nvarchar(50)</td>
<td>Si</td>
</tr>
<tr>
<td>C1</td>
<td>decimal(18,2)</td>
<td>Si</td>
</tr>
<tr>
<td>Tgd1</td>
<td>decimal(18,2)</td>
<td>Si</td>
</tr>
<tr>
<td>C2</td>
<td>decimal(18,2)</td>
<td>Si</td>
</tr>
<tr>
<td>Tgd2</td>
<td>decimal(18,2)</td>
<td>Si</td>
</tr>
<tr>
<td>Anio</td>
<td>datetime</td>
<td>Si</td>
</tr>
</tbody>
</table>

Tabla Regulador: otro componente del transformador, al contrario que sucedía con el resto de componentes que todos los transformadores los poseían, el regulador no tiene porque estar contenido en un transformador, pudiendo ser prescindible.

Tabla 16: Tabla regulador

<table>
<thead>
<tr>
<th>Campo</th>
<th>Tipo</th>
<th>Nulo</th>
</tr>
</thead>
<tbody>
<tr>
<td>id(PK)</td>
<td>int</td>
<td>No</td>
</tr>
<tr>
<td>IdFabricante(FK)</td>
<td>int</td>
<td>Si</td>
</tr>
<tr>
<td>NSerie</td>
<td>nvarchar(50)</td>
<td>Si</td>
</tr>
<tr>
<td>Modelo</td>
<td>nvarchar(50)</td>
<td>Si</td>
</tr>
<tr>
<td>Anio</td>
<td>datetime</td>
<td>Si</td>
</tr>
<tr>
<td>Lu</td>
<td>decimal(18,2)</td>
<td>Si</td>
</tr>
<tr>
<td>RTrans</td>
<td>decimal(18,2)</td>
<td>Si</td>
</tr>
<tr>
<td>Um</td>
<td>decimal(18,2)</td>
<td>Si</td>
</tr>
<tr>
<td>Ui</td>
<td>decimal(18,2)</td>
<td>Si</td>
</tr>
<tr>
<td>MaxEscalon</td>
<td>int</td>
<td>Si</td>
</tr>
</tbody>
</table>

Tabla Regulación: contiene la información adicional a los devanados, si además el transformador tiene regulación, existirá más de una entrada por devanado.

Tabla 17: Tabla regulación

<table>
<thead>
<tr>
<th>Campo</th>
<th>Tipo</th>
<th>Nulo</th>
</tr>
</thead>
<tbody>
<tr>
<td>id(PK)</td>
<td>int</td>
<td>Si</td>
</tr>
<tr>
<td>IdDevanado(FK)</td>
<td>int</td>
<td>Si</td>
</tr>
<tr>
<td>IdRegualdor</td>
<td>int</td>
<td>Si</td>
</tr>
<tr>
<td>Posicion</td>
<td>nvarchar(50)</td>
<td>Si</td>
</tr>
<tr>
<td>Tension</td>
<td>float</td>
<td>No</td>
</tr>
<tr>
<td>Intensidad_Onaf</td>
<td>float</td>
<td>No</td>
</tr>
<tr>
<td>Intensidad_Onan</td>
<td>float</td>
<td>No</td>
</tr>
<tr>
<td>Tipo</td>
<td>float</td>
<td>No</td>
</tr>
<tr>
<td>Intensidad_Ofaf</td>
<td>float</td>
<td>No</td>
</tr>
<tr>
<td>Intensidad_Odaf</td>
<td>float</td>
<td>No</td>
</tr>
<tr>
<td>Intensidad_An</td>
<td>float</td>
<td>No</td>
</tr>
</tbody>
</table>

Una vez vistas todas las tablas que componen el transformador, se muestra el diagrama de tablas:
Ilustración 13: Diagrama de tablas transformador
Sistema Para Transformadores de Tensión

A continuación se muestra el diagrama de tablas (ilustración 14) y la tabla de especificaciones (tabla 18) que están definidos para los transformadores (esta parte del modelo, no se describe entera, debido a la poca importancia dentro del sistema).

**Tabla Especificación:** todo tipo de transformador, tienen un serie de características definidas por los fabricantes y que serán iguales para el tipo de transformador definido. Esta información será importante para posteriores medidas que se realicen sobre el transformador.

<table>
<thead>
<tr>
<th>Campo</th>
<th>Tipo</th>
<th>Nulo</th>
</tr>
</thead>
<tbody>
<tr>
<td>id(PK)</td>
<td>int</td>
<td>No</td>
</tr>
<tr>
<td>IdTrafo(FK)</td>
<td>int</td>
<td>No</td>
</tr>
<tr>
<td>Descripcion</td>
<td>varchar(200)</td>
<td>Si</td>
</tr>
<tr>
<td>Corriente100</td>
<td>float</td>
<td>Si</td>
</tr>
<tr>
<td>Corriente110</td>
<td>float</td>
<td>Si</td>
</tr>
<tr>
<td>Perdidas100</td>
<td>float</td>
<td>Si</td>
</tr>
<tr>
<td>Perdidas110</td>
<td>float</td>
<td>Si</td>
</tr>
<tr>
<td>NivelRuido1</td>
<td>float</td>
<td>Si</td>
</tr>
<tr>
<td>NivelRuido2</td>
<td>float</td>
<td>Si</td>
</tr>
<tr>
<td>Calentamientos</td>
<td>float</td>
<td>Si</td>
</tr>
</tbody>
</table>

El diagrama de tablas es el siguiente:

Ilustración 14: Diagrama de tablas de especificaciones

Tras definir todos los elementos que forman parte del transformador o que tiene que ver con la estructura de la aplicación en sí, se pasa a ver la estructura principal y todas las medidas que pueden asociarse a los transformadores.
Tabla Ensayo: Es la tabla principal de los ensayos de fábrica, contiene todos los datos generales. Los ensayos que forman parte de un transformador y que se realizan en fábrica tienen una tabla para poder ampliar la información del ensayo, para así detallar más su contenido. Esta tabla es común para todos los ensayos.

<table>
<thead>
<tr>
<th>Campo</th>
<th>Tipo</th>
<th>Nulo</th>
</tr>
</thead>
<tbody>
<tr>
<td>id(PK)</td>
<td>int</td>
<td>No</td>
</tr>
<tr>
<td>idTrafo(FK)</td>
<td>int</td>
<td>No</td>
</tr>
<tr>
<td>Fecha</td>
<td>datetime</td>
<td>No</td>
</tr>
<tr>
<td>Tmpra_Aceite</td>
<td>decimal(4,2)</td>
<td>Si</td>
</tr>
<tr>
<td>Observaciones</td>
<td>nText</td>
<td>Si</td>
</tr>
<tr>
<td>ObservacionesCortas</td>
<td>nvarchar(50)</td>
<td>Si</td>
</tr>
<tr>
<td>Resultado</td>
<td>nvarchar(50)</td>
<td>Si</td>
</tr>
<tr>
<td>Estado</td>
<td>nvarchar(50)</td>
<td>Si</td>
</tr>
<tr>
<td>Borrado</td>
<td>bit</td>
<td>No</td>
</tr>
<tr>
<td>dtAlta</td>
<td>datetime</td>
<td>No</td>
</tr>
<tr>
<td>IdUsrAlta</td>
<td>nvarchar(50)</td>
<td>No</td>
</tr>
<tr>
<td>dtCerrar</td>
<td>datetime</td>
<td>Si</td>
</tr>
<tr>
<td>IdUsrCerrar</td>
<td>nvarchar(50)</td>
<td>Si</td>
</tr>
</tbody>
</table>

Ilustración 15: Diagrama de tablas ensayos de fábrica
Tras describir la tabla principal de los ensayos de fábrica (ilustración 15) y sus relaciones con el resto de los ensayos, se puede ver la estructura y relaciones de cada uno de ellos en el Anexo 1 “Detalle Ensayos de Fábrica” definido al final de este documento.

Además de los ensayos de fábrica existen los ensayos de campo. De esta manera se termina con toda la estructura de base de datos que ha sido definida en el sistema. A continuación se describen este tipo de ensayos.

La estructura que disponen los ensayos de campos, está separada en dos partes, por un lado, análisis y por otro lado medidas. De cada una de estas partes colgarán el resto de ensayos. Se pasa a ver la estructura de la tabla ensayos de campo y de estas dos partes.

**Tabla Ensayo Campo:** Es la tabla principal de los ensayos de fábrica, contiene todos los datos generales, los ensayos que forman parte de un transformador y que se realizan en campo tiene una tabla para poder ampliar la información del ensayo, para así detallar más su contenido. Esta tabla será común para todos los ensayos de este tipo.

**Tabla 20: Tabla ensayos campo**

<table>
<thead>
<tr>
<th>Campo</th>
<th>Tipo</th>
<th>Nulo</th>
</tr>
</thead>
<tbody>
<tr>
<td>id(PK)</td>
<td>int</td>
<td>No</td>
</tr>
<tr>
<td>IdTrafo(FK)</td>
<td>int</td>
<td>No</td>
</tr>
<tr>
<td>IdTipoEnsayo</td>
<td>int</td>
<td>No</td>
</tr>
<tr>
<td>Fecha</td>
<td>datetime</td>
<td>No</td>
</tr>
<tr>
<td>Tmpra_Aceite</td>
<td>decimal(4,2)</td>
<td>Si</td>
</tr>
<tr>
<td>Observaciones</td>
<td>nText</td>
<td>Si</td>
</tr>
<tr>
<td>ObservacionesCortas</td>
<td>nvarchar(50)</td>
<td>Si</td>
</tr>
<tr>
<td>Resultado</td>
<td>nvarchar(50)</td>
<td>Si</td>
</tr>
<tr>
<td>Estado</td>
<td>nvarchar(50)</td>
<td>Si</td>
</tr>
<tr>
<td>Valido</td>
<td>bit</td>
<td>Si</td>
</tr>
<tr>
<td>Borrado</td>
<td>bit</td>
<td>No</td>
</tr>
<tr>
<td>dtAlta</td>
<td>datetime</td>
<td>No</td>
</tr>
<tr>
<td>IdUsrAlta</td>
<td>nvarchar(50)</td>
<td>No</td>
</tr>
<tr>
<td>dtCerrar</td>
<td>datetime</td>
<td>Si</td>
</tr>
<tr>
<td>IdUsrCerrar</td>
<td>nvarchar(50)</td>
<td>Si</td>
</tr>
</tbody>
</table>

**Tabla Medición Trafos:** Es la tabla principal de uno de los conjuntos de ensayos de campo, en ella se recoge información que será común para todos los ensayos de este tipo.

**Tabla 21: Medición transformador**

<table>
<thead>
<tr>
<th>Campo</th>
<th>Tipo</th>
<th>Nulo</th>
</tr>
</thead>
<tbody>
<tr>
<td>id(PK)</td>
<td>int</td>
<td>No</td>
</tr>
<tr>
<td>id_medidas_trafos(FK)</td>
<td>int</td>
<td>No</td>
</tr>
</tbody>
</table>
### Tabla Análisis

Tabla principal del otro conjunto de ensayos de campo, en ella se recoge información que será común para todos los ensayos de este tipo.

#### Tabla 22: Tabla análisis

<table>
<thead>
<tr>
<th>Campo</th>
<th>Tipo</th>
<th>Nulo</th>
</tr>
</thead>
<tbody>
<tr>
<td>id_análisis (PK)</td>
<td>int</td>
<td>No</td>
</tr>
<tr>
<td>id_ensayo_campo (FK)</td>
<td>int</td>
<td>No</td>
</tr>
<tr>
<td>antecedente</td>
<td>int</td>
<td>No</td>
</tr>
<tr>
<td>toma_muestra</td>
<td>Int</td>
<td>Si</td>
</tr>
<tr>
<td>dt_toma_muestra</td>
<td>datetime</td>
<td>Si</td>
</tr>
<tr>
<td>dt_análisis</td>
<td>datetime</td>
<td>Si</td>
</tr>
<tr>
<td>temperatura</td>
<td>float</td>
<td>Si</td>
</tr>
<tr>
<td>comentario</td>
<td>text</td>
<td>Si</td>
</tr>
</tbody>
</table>

Una vez vista toda la estructura básica de los ensayos de campos, se mostrarán los diagramas que defienden la base de datos, para una mejor comprensión se dividirán por un lado en la estructura principal y por el otro, se mostrarán los dos conjuntos de manera separada.

![Diagrama ensayos de campo](image)

**Ilustración 16: Diagrama ensayos de campo**
La ilustración 16 muestra la estructura principal de los ensayos, que como se observa, está compuesta por una tabla principal “t_Ensayo_Campo”, y dos tablas que se relacionan con ella. Estas dos tablas con “t_analisis” y “t_medicion_trafos”.

Ilustración 17: Ensayos de análisis

Como puede verse en la ilustración 17, se observa la tabla t_analisis, que es la principal de este grupo de relaciones, la cual se relaciona con el resto ampliando la información que contiene, pudiendo detallar cada uno de los análisis de una manera amplia.

Por último en la ilustración 18, se muestra las relaciones existentes entre las medidas de campos definidas por la tabla principal t_medicion_trafos.
Ilustración 18: Ensayos de medidas de transformadores
3.4.3 Arquitectura software

En este punto se describe la arquitectura software que ha sido utilizada en el sistema, para ello se utilizarán tres tipos de diagramas: diagrama de despliegue, diagrama de paquetes y diagrama de clases.

En primer lugar se muestra y describe el diagrama de despliegue definido para la aplicación, que puede verse en la ilustración 19.

La descripción de los nodos en el diagrama de despliegue es la siguiente:

- **Estaciones de trabajo (PC1, PC2):** Se trata de ordenadores portátiles, o un ordenador de sobremesa, que estén equipados con un navegador web y que tengan acceso a la red de Unión Fenosa, o bien que tengan instalada la versión OFFLINE de la aplicación. Serán utilizados por los usuarios de la aplicación.
- **Servidor Web:** Aloja el servidor web que albergará la pagina Web que será utilizada por los usuarios que accedan a ella. Estará conectado a la intranet.
- **Servidor de base de datos:** Alberga el motor de base de datos que será utilizado por la aplicación Web.

A continuación, se muestra el diagrama paquetes, ilustración 20. Nuestro sistema estará organizado en tres capas: la capa de interfaz de usuario, la capa lógica de negocio y la de datos.
Ilustración 20: Diagrama de paquetes
**La capa de la interfaz de usuario.** Agrupa los elementos que están presentes para el usuario, como son los menús que integran la aplicación. Además es la encargada de mostrar los distintos formularios que se definen en la aplicación, y a través de los cuales se muestran los datos almacenados o se introducen los nuevos datos. En esta sección se encuentran todos los formularios y controles necesarios para mostrar todas las secciones la aplicación.

**La capa lógica de negocio.** La integran los elementos que realizan el procesamiento del trabajo. Esta capa es la responsable de recibir y devolver la información a la capa de la interfaz y de interactuar con la capa de datos para solicitar o almacenar datos. Contiene toda la información referente al funcionamiento de la aplicación en sí. Las clases principales que forman parte de ellas son las siguientes (no se muestran todas las clases ni todos los métodos implicados en esta capa por motivos de espacio):

- **CUsuarioSrv:** Realiza todas las acciones necesarias sobre el objetos CUsuario, como obtener los datos, almacenarlos, eliminarlos y comprobar el acceso a la aplicación.
- **CPerfilSrv:** Comprueba si los usuarios tiene permisos sobre las acciones que se van a realizar. Además obtiene los datos de la clase CPerfil y los almacena, y elimina.
- **CTransformadorSrv:** Obtiene todos los datos referente al transformador. Almacena la información introducida. Permite realizar consultas avanzadas al igual que genera informes.
- **CEnsayosSrv:** Con esta clase se almacenan, crean y eliminan los ensayos de este tipo. Por otro lado permite obtener los ensayos de un determinado transformador, y de realizar consultas avanzadas.
- **CEnsayosCampoSrv:** Con esta clase se almacenan, crean y eliminan los ensayos de campo. Por otro lado permite obtener los ensayos de un determinado transformador, y de realizar consultas avanzadas.
- **CAnalisisSrv:** Se utiliza para almacenar y eliminar objetos CAnalisis, además de dar la posibilidad de consultar los datos de estos objetos por transformador. Por último permite calcular un diagnóstico.
- **CMedidaSrv:** Da la posibilidad de modificar y eliminar medidas. Permite obtener el diagnóstico de la medida e importar el XML en el que se contienen estas medidas. Por último genera las gráficas correspondientes.
- **CECLSrv:** El comportamiento de esta clase es igual al resto de las que extienden de CEnsayo, por este motivo únicamente se muestra esta. En este conjunto de clases, es posible almacenar y eliminar los datos. Por otro lado es posible obtener el diagnostico y gráficas asociados a este ensayo.

En la ilustración 21 se muestra el diagrama de clases correspondiente a la lógica de negocio. En el diagrama correspondiente a la lógica de negocio se muestra más en detalle cómo está realizada la estructura con respecto a lo mostrado en la ilustración 20, aún así como sucedía en la descripción de esta capa, no se muestra todas las clases que toman partido de este diagrama, por cuestiones de espacio.
Ilustración 21: Diagrama de Clases Capa Lógica del Negocio

**Capa de datos.** Contiene toda la información referente a la estructura de datos de la aplicación, es la encargada de transformar todos los datos, contenidos en la base de datos, en clases estáticas, para su posterior instanciación de manera que estos datos puedan ser tratados como objetos por parte de la aplicación. En esta capa se realiza el mapeo objeto-relacional (ORM), para convertir los datos alojados en la base de datos en objetos, para que puedan ser tratados por la aplicación.
Una vez vista como está dividida la aplicación en paquetes, se pasa a mostrar el diagrama de clases que define la capa de datos de la aplicación, de una manera más extensa a como se vio en la ilustración 20. Por motivos de espacio únicamente se muestra las clases que forman parte de este diagrama, dejando sin mostrar las propiedades y métodos de dichas clases, así como el detalle de cada uno de los ensayos.
Como puede verse en la estructura de la ilustración 22, es una conversión del modelo de datos generado en la base de datos con el fin de poder realizar la transformación a objetos en la aplicación. A continuación se explican cada uno de los grupos de clases que se ha definido, y que son equivalentes a los paquetes definidos en la ilustración 20.

**Seguridad**: en este primer grupo de clases se encuentran aquellas que son las encargadas de albergar todos los datos referentes a los usuarios, como nombres, identificadores, email, etc. Además de contener la información básica, tiene la relación con los perfiles. Las clases que agrupan estos datos son, CUsuario, CPerfil y CPagina.

**Datos de los transformadores**: contiene toda la información de los transformadores, así como de todos los elementos de los que está compuesto. Las clases que componen este grupo son, CTrafo, CUbicación, CFabricante, CPotencia, CEspecificación, CGrupoConexión, CRegulador, CRegulación, CTerminal, CBorna y CDevanado. A través de estas clases se obtiene toda la información del transformador.

**Ensayos**: en este grupo de clases nos encontramos todos los datos relacionados a los ensayos de fábrica. Con la clases principal, CEnsayo, existe una relación de herencia con todos los datos específicos de cada uno de los ensayos, lo que permite tratar a todos estos ensayos como si de un mismo tipo de se tratara, a través del polimorfismo.

**Ensayos de campo**: es el último grupo de clases, el cual está compuesto por la clase principal CEnsayoCampo, que es la clase padre, lo que nos permite tratar a todos los ensayos más específicos como CEnsayoCampo, a través del polimorfismo. Además cada una de las clases que heredan de CEnsayoCampo, tiene a su vez herencia con clases que especifican aún más los distintos tipos de ensayos que contienen.
3.5 Implementación

En esta sección se describen todos aquellos apartados que han tenido que ver en la implementación del sistema, ya sea por el hardware utilizado en el desarrollo del mismo, como por fragmentos de código y configuración que mayor dificultad han ocasionado en todo el proceso de desarrollo.

3.5.1 Aplicación Web

A continuación se detallan las partes más relevantes del código desarrollado para la aplicación Web que ha sido descrita en este proyecto.

Conexión Base de Datos

La primera acción que se ha definido ha sido la comunicación con la base de datos, para así poder, a través de Hibernate, tratar los datos de la base de datos de una manera sencilla, pudiendo así definir el modelo de datos asociado a la base de datos. Para realizar esto es necesario en primer lugar, generar la cadena de conexión con el servidor de base de datos. Esta información de conexión estará alojada en un fichero de configuración el cual estará disponible para todos los elementos de la aplicación. El fichero de configuración que utilizan los proyectos Web en .NET se denomina “web.config”, y contendrán toda la información de configuración que vaya a ser utilizada.

A continuación se muestra la sección del “web.config” a partir de la cual se establecen los datos necesarios para realizar la conexión con la base de datos a través de la aplicación. En los datos de configuración se puede observar que se establece por un lado, la dirección en la que estará alojado el servidor de base de datos, el nombre de la base de datos que va a ser utilizada, y la autenticación que gestionará los permisos dentro del SQL Server. Estos parámetros pueden verse en el atributo "connection.connection_string". Otro aspecto que llama la atención dentro de esta sección de configuración es que se tiene que establecer el ensamblado que será el encargado de transformar las tablas y columnas de la base de datos en clases y atributos dentro de la aplicación, esta información se define en la sección “mapping”.

```xml
<hibernate-configuration xmlns="urn:nhibernate-configuration-2.2">
  <session-factory>
    <property name="connection.provider">Nhibernate.Connection.DriverConnectionProvider</property>
    <property name="dialect">Nhibernate.Dialect.MsSql2005Dialect</property>
    <!--CONEXIÓN INSTALACION-->
    <property name="connection.connection_string">Server=10.105.3.212\SQLEXPRESS; Database=GETUF_DB;User Id=*******;Password=*******</property>
    <property name="show_sql">true</property>
    <property name="set quoted_identifier">off</property>
    <mapping assembly="GETUF.DAL"/>
  </session-factory>
</hibernate-configuration>
```

Una vez establecida la cadena de conexión, se tiene que poder realizar consultas en la base de datos para obtener los datos dentro de la aplicación, y así convertir los datos en objetos para ser tratados desde la aplicación.
Todas las clases que se encarguen de realizar acciones con la base de datos, ya sean consultas/inscripciones/actualizaciones, extenderán de esta clase, que será la encargada de establecer la conexión con la base de datos. Para realizar la conexión de una manera correcta, por defecto al ejecutar el método “BuildSessionFactory()”, obtiene los datos de la cadena de conexión del fichero de configuración “web.config”, que se mostró anteriormente.

**Tratamiento de ficheros XML**

En el siguiente fragmento de código se muestra como se tratan los ficheros XML, de los cuales se obtienen datos para integrar medidas automáticamente a los ensayos. En el primero de los métodos, se puede comprobar cómo se inicia el tratamiento para obtener todas las medidas, y en el otro como se comprueba si el fichero es correcto, para el transformador seleccionado.

```csharp
public class CServicios
{
    private static readonly ISessionFactory sessionFactory;
    //Constructor que crea la Session Factory al iniciar la aplicación.
    static CServicios()
    {
        sessionFactory = new Configuration().Configure().BuildSessionFactory();
    }

    private ISession m_oNHibernateSession;
    public CServicios()
    {
        //Cada vez que se instancia un servicio, se abre una nueva sesión
        m_oNHibernateSession = sessionFactory.OpenSession();
    }

    public ISession prNHibernateSession
    {
        get { return m_oNHibernateSession; }
    }
}

public string STratarXml(string sPath, CTrafo oTrafo)
{
    //Abrimos el documento
    XmlDocument xmldoc = new XmlDocument();
    xmldoc.Load(sPath);

    //Obtenemos el grupo de nodos de las diferentes medidas
    XmlNodeList aMedidas = xmldoc.GetElementsByTagName("dta_data");

    //Obtenemos cada uno de los diferentes tipos de medidas
    NObtenerCapacidadTangente(aMedidas, oTrafo);
    NObtenerTomaCapacitiva(xmldoc, aMedidas, oTrafo);
    NObtenerCollarCaliente(aMedidas, oTrafo);
    NObtenerTestExcitacion(aMedidas, oTrafo);
    NObtenerRelacionTransformacion(aMedidas, oTrafo);
}

public string coincideNumSerie(string sPath, string sNumSerie)
{
    if (Directory.Exists(sPath))
    {
        string[] asFicheros = Directory.GetFiles(sPath);
        for (int i = 0; i < asFicheros.Length; i++)
        {
            //Abrimos el documento
            XmlDocument xmldoc = new XmlDocument();
            xmldoc.Load(asFicheros[i]);
        }
    }
}
```
Tratamiento hojas de cálculo

Otro apartado digno de mención en el proyecto, es la manera en la que se manejan las hojas de cálculo para la representación de gráficas. Para ello se generan una serie de plantillas definidas de antemano, en formato Excel.

Tras realizar las plantillas y una vez obtenidos los datos necesarios para la generación de las gráficas, se escriben los datos para cumplimentar las gráficas, y se modifican las opciones de configuración necesarias para visualizarlas correctamente. Además de estas gráficas se obtiene unos polinomios, los cuales son útiles para el diagnóstico de los equipos.

Todo este tratamiento de hojas de cálculo se realiza a través de objetos Excel, proporcionados por Microsoft.

```csharp
private string GenerarGrafica() {
    Excel.Application oXL;
    Excel.Workbook oWB;
    Excel.Worksheet oSheet;
    try {
        oXL = new Excel.Application();
        //Abrimos el documento que queremos
        string workbookPath = CGraficas.ObtenerNombreFichero(prRutaGraficas, Session.SessionID, "epv1.xls");
        oWB = oXL.Workbooks.Open(workbookPath, 0, false, 5, "", "", Excel.XlPlatform.xlWindows, "", true, false, 0, true, true, true);

        //Escribimos en la hoja 2
        oSheet = (Excel.Worksheet)oWB.Sheets[2];

        //Añadimos los datos
        int i = 1;
        foreach (CEPV_Med oMedidas in prViewStateListaMedidas) {
            if (!oMedidas.prTeor.Equals(115)) {
                oSheet.Cells[i, 1] = oMedidas.prUn;
                oSheet.Cells[i, 2] = oMedidas.prPcorr;
                i++;
            }
        }

        Excel.Worksheet oSheetGrafica = (Excel.Worksheet)oWB.Sheets[1];
    } catch {
        return "C";
    }
    return "";
}
```
Como se muestra en el fragmento de código, “Generar Grafica1”, en primer lugar se carga la plantilla que ha sido generada previamente en el Excel. Una vez hecho esto se introducen los datos que van a ser mostrados en la gráfica y a su vez se modifican los parámetros específicos de las gráficas en cuestión, para así adaptarla a las necesidades.

### Generación de informes

En la aplicación también es posible realizar informes para generar ficheros PDF, de manera que se obtengan todos los datos del ensayo y puedan mostrarse en un único documento. Para ello se utiliza “ReportViewer” una herramienta que está incorporada con Visual Studio, de esta manera y con un editor gráfico (como se observa en la ilustración 23), se pueden generar las plantillas donde están ubicados los datos de los transformadores.
Este tipo de plantillas, diseñadas con la herramienta “ReportViewer” están estructuradas internamente en XML, el cual podrá ser modificado a través de un editor de texto o gráficamente, esta última opción es la que se ha utilizado para desarrollar esta aplicación. Esta herramienta gráfica puede verse en la ilustración 23.

Estas plantillas deberán ir rellenas con propiedades de listas de objetos que será la fuente de datos para estos informes. Una vez incluida esta lista de objetos se añadirá la estructura de estos objetos para que así el informe conozca que campos pueden ser mostrados en estos documentos.

```csharp
protected void btnExportarCampo_Click(object sender, EventArgs e)
{
    //Almacenamos los ID de los ensayos que tenemos que mostrar
    ArrayList aoIds = new ArrayList();
    try
    {
        DataGridItemCollection items = dtgEnsayosCampo.Items;
        foreach (DataGridItem item in items)
        {
            System.Web.UI.WebControls.CheckBox myCheckbox =
            (System.Web.UI.WebControls.CheckBox)item.Cells[6].Controls[1];
            if (myCheckbox.Checked)
            {
                LinkButton oId = (LinkButton)item.Cells[0].Controls[0];
                string sId = oId.Text;
                aoIds.Add(int.Parse(sId));
            }
        }
    } catch (Exception ex){}
    CTrafo oTrafo = trafoSrv.OSelectByPk(int.Parse(prGetIdTrafo));
    RVInformeCampo.LocalReport.Dispose();
    RVInformeCampo.LocalReport.ReportPath = ".\Informes\InformeCampo.rdlc";
    RVInformeCampo.LocalReport.DataSources.Clear();
    //Construimos los datos generales
    ConstruirTrafo(oTrafo, RVInformeCampo);
    //Construimos los diferentes ensayos
    ConstruirEnsayosCampo(oTrafo, aoIds);
    ExportarPDF(RVInformeCampo, false);
}
```

En el método anterior “btnExportarCampo_Click”, se cargan el informe seleccionado, que habrá sido generado a partir de la plantilla diseñada con “ReportViewer”. En el método que se muestra a continuación “ConstruirTrafo”, se insertan los datos que van a ser mostrados en el informe.

```csharp
private void ConstruirTrafo(CTrafo oTrafo, ReportViewer oReport)
{
    CTransformadorInforme oTrafoInforme = new CTransformadorInforme(oTrafo);
    CReguladorInforme oReguladorInforme = new CReguladorInforme(ObtenerRegulador(oTrafo));
    //Almacenamos los datos en las colecciones
    ArrayList aoDebanados = new ArrayList();
    ArrayList aoTrafo = new ArrayList();
    ArrayList aoRegualdores = new ArrayList();
    ArrayList aoBornas = new ArrayList();
    ArrayList aoEnsayosFabrica = new ArrayList();
    ArrayList aoEnsayosCampo = new ArrayList();
    aoDebanados = ObtenerDebanados(oTrafo);
    aoBornas = ObtenerBornas(oTrafo);
    aoTrafo.Add(oTrafoInforme);
    aoRegualdores.Add(oReguladorInforme);
    aoEnsayosFabrica = ObtenerEnsayosFabrica(oTrafo);
    aoEnsayosCampo = ObtenerEnsayosCampo(oTrafo);
}
Además de generar informes en PDF, la aplicación permite generar un fichero comprimido con las gráficas que han sido realizadas para ese transformador. Para ello se realizarán las gráficas de la misma manera que se vio anteriormente, y una vez se tengan todos los ficheros necesario para generar el informe se utilizará una librería la cual proporciona la funcionalidad de comprimir N ficheros en un .zip. Una vez generado el .zip, para descargarlo desde la aplicación, bastará con escribir el fichero en el response.

```csharp
public string Comprimir(string[] fileNames, CTrafo oTrafo, bool sFabrica)
{
    //Nos creamos la carpeta donde vamos a generar el zip
    string sDir = prDirBase + "\ZIP\" + Session.SessionID;
    if (!Directory.Exists(sDir))
        Directory.CreateDirectory(sDir);

    // comprimir los ficheros del array en el zip indicado
    string sZip = sDir + NombreFichero(true, oTrafo, ".zip", "");
    using (ZipOutputStream strmZipOutputStream = new ZipOutputStream(File.OpenWrite(sZip)))
    {
        foreach (string filePath in fileNames)
        {
            FileStream stream = File.OpenRead(filePath);
            ZipEntry entry = new ZipEntry(Path.GetFileName(filePath));
            strmZipOutputStream.PutNextEntry(entry);
            strmZipOutputStream.Write(new byte[stream.Length], 0, stream.Length);
            stream.Close();
        }
        strmZipOutputStream.Close();
    }
    HttpContext.Current.Response.AddHeader("Content-Disposition", "attachment; filename=" + sZip);
    HttpContext.Current.Response.ContentType = "application/zip";
    HttpContext.Current.Response.WriteFile(sZip);
}
```
Generador del instalable

Otra de las funcionalidades que proporciona esta aplicación, es la de poder ser instalada en cualquier equipo que tenga los requisitos solicitados por la aplicación. Esta instalación será realizada con un instalador generado en Visual Studio.

Para ello se generó un proyecto de instalación de una aplicación Web, como se muestra en la imagen.

Una vez creado el proyecto correspondiente, era necesario marcar las dependencias que iba a necesitar, para que pudiera funcionar en la máquina en la que fuera instalado. En este caso las dependencias son las que se ven en la imagen siguiente. Estas dependencias tienen que ser marcadas en las propiedades del proyecto de instalación creado anteriormente.
Tras generar toda la configuración necesaria para poder realizar el proyecto de instalación, es necesario indicar la salida de proyecto que se va a incluir en el instalador, en nuestro caso, van a ser dos, por un lado el propio proyecto Web, y por el otro todas las instrucciones que se encargarán de crear, restaurar y establecer permisos de la base datos que se instalará de manera local en el equipo del usuario.

Las acciones que se realizan sobre la base de datos, son las de crear la base de datos, crear los permisos, modificar la configuración de SQL para que admita autenticación de SQL y restaurar la base de datos. Además de ejecutar estas acciones sobre la base de datos, es necesario reiniciar el servicio SQL para que así se apliquen todos los cambios.

Hay que comentar que las acciones que se realizan son sentencias SQL que se ejecutan a través de objetos SQL proporcionados por la librería “SqlClient” proporcionada por el Framework. Algunos de los métodos que realizan las acciones que se han mencionado anteriormente se pueden ver a continuación:

```csharp
public void CambiarAutenticacionMixta()
{
    // Inicio de sesion mixto unicamente la 1ª vez que instalamos
    string sBackup = "EXEC xp_instance_regwrite N'HKEY_LOCAL_MACHINE',
                    N'Software\Microsoft\MSSQLServer\MSSQLServer',
                    N'LoginMode', REG_DWORD, 2 "+
                    "EXEC xp_instance_regwrite N'HKEY_LOCAL_MACHINE',
                    N'Software\Microsoft\MSSQLServer\MSSQLServer',
                    N'AuditLevel', REG_DWORD, 2 ";

    SqlConnectionStringBuilder csb = new SqlConnectionStringBuilder();
    csb.DataSource = "localhost\\SQLExpress";
    csb.InitialCatalog = "master";
    csb.IntegratedSecurity = true;

    using (SqlConnection con = new SqlConnection(csb.ConnectionString))
    {
        try
        {
            con.Open();
            SqlCommand cmdBackup = new SqlCommand(sBackup, con);
            cmdBackup.ExecuteNonQuery();
            con.Close();
        }
        catch (Exception ex)
        {
            throw ex;
        }
    }
}
```
El método “CambiarAutenticaciónMixta”, mostrado anteriormente, modifica el modo en el que autentificarse en SQL Server, que por defecto en la instalación, es establecido como autentificación de Windows, lo que no permitiría realizar una conexión a la base de datos a través de un usuario y una contraseña. Para modificar la autentificación, se genera un script que realiza este cambio, y que es ejecutado directamente contra el motor de base de datos.

```csharp
public void CrearDB()
{
    //Creamos la DB únicamente la 1ª vez que instalamos
    string sBackup = "CREATE DATABASE [GETUF_DB] ON PRIMARY " +
    "( NAME = N'GETUF_DB', FILENAME = N'C:\Arquivos de
programa\Microsoft SQL Server\MSSQL.1\MSSQL\DATA\GETUF_DB.mdf' , SIZE = 3072KB ,
FILEGROWTH = 1024KB ) " +
    "LOG ON " +
    "( NAME = N'GETUF_DB_log', FILENAME = N'C:\Arquivos de
programa\Microsoft SQL Server\MSSQL.1\MSSQL\DATA\GETUF_DB_log.ldf' , SIZE = 1024KB ,
FILEGROWTH = 10%) " +
    "EXEC dbo.sp_dbcmptlevel @dbname=N'GETUF_DB', @new_cmptlevel=90 " +
    "IF (1 = FULLTEXTSERVICEPROPERTY('IsFullTextInstalled')) " +
    "begin " +
    "EXEC [GETUF_DB].[dbo].[sp_fulltext_database] @action = 'disable'
    "end ";

SqlConnectionStringBuilder csb = new SqlConnectionStringBuilder();
csb.DataSource = "localhost\SQLEXPRESS";
// Es mejor abrir la conexión con la base Master
csb.InitialCatalog = "master";
csb.IntegratedSecurity = true;

using (SqlConnection con = new SqlConnection(csb.ConnectionString))
{
    try
    {
        con.Open();
        SqlCommand cmdBackUp = new SqlCommand(sBackup, con);
        cmdBackUp.ExecuteNonQuery();
        con.Close();
    }
    catch (Exception ex)
    {
        
    }
}
```

En este caso, en el método “CrearDB”, se genera el script necesario para crear la base de datos a partir de un backup que se realizó previamente y que contiene toda la estructura de base de datos.

**Generador del instalable**

Por último, es destacable en esta sección la manera en la que la aplicación muestra como si el usuario está trabajando conectado al servidor, o por el contrario lo está realizando con una copia en su máquina local.

Esta acción se realiza de la siguiente manera:
Para realizar esta acción el sistema, en la página master obtiene la dirección del cliente a través del request. Una vez obtenida esta dirección se invoca a un método (“offline”) el cual comprobará si la dirección IP del cliente es igual a la dirección IP de la máquina en la que se está ejecutando la aplicación. Para ello comprueba las interfaces de red de las que dispone.

```csharp
ip = Request.UserHostAddress;

public bool offline(String ip)
{
    //Leemos todas las interfaces de red de nuestra máquina
    NetworkInterface[] ni = NetworkInterface.GetAllNetworkInterfaces();

    //Después, las recorremos y las tratamos.
    foreach (NetworkInterface n in ni)
    {
        if (n.GetIPProperties().UnicastAddresses.Count > 0)
        {
            if (ip.Equals(n.GetIPProperties().UnicastAddresses[0].Address.ToString()))
            {
                return true;
            }
        }

        if (n.GetIPProperties().MulticastAddresses.Count > 0)
        {
            if (ip.Equals(n.GetIPProperties().MulticastAddresses[0].Address.ToString()))
            {
                return true;
            }
        }
    }
    return false;
}
```
4.1 Pruebas

En éste apartado se describen las pruebas realizadas, para lo cual nos guiamos por los casos de uso de las ilustraciones 7 y 8. Por otro lado se encuentra como adjunto un manual de usuario\(^1\) en el que se ven en detalle todos los casos aquí expuestos.

**Autenticación:** En primer lugar para poder acceder a la aplicación, es necesario, introducir los datos necesarios para autenticarse. Los datos solicitados por la aplicación, son un nombre de usuarios, y una contraseña. En caso de que los datos fueran erróneos la aplicación mostrará un mensaje avisando de este hecho, de no ser así la aplicación iniciaría sesión redirigiendo a los usuarios a la bandeja de entrada.

![Imagen de autenticación incorrecta](attachment:image)

**Ilustración 26: Autenticación incorrecta**

En caso de autenticarse correctamente se mostrará, la bandeja de entrada como ya se ha mencionado antes, en la cual podrá acceder a diferentes secciones de la aplicación. Por otro lado aparecerá el menú por el cual se puede navegar a todas las páginas que están disponibles en la aplicación, pulsando sobre cualquiera de los enlaces. Desde una de las secciones del menú es posible acceder al apartado de seguridad, más concretamente a la definición de perfiles, en la cual se puede crear, modificar y eliminar perfiles existentes.

**Nuevo perfil:** Para crear un nuevo perfil se debe pulsar sobre el link del menú, “Crear Perfil”, una vez dentro de esta página, se deberá rellenar de manera obligatoria, el “Id. Perfil”, en caso contrario no se podrá continuar con la creación de los perfiles, por el contrario, si se rellena ese campo correctamente, y se seleccionan algunas de las acciones que se pueden realizar, se creará el perfil elegido.

\(^1\) El manual de usuario puede verse en el documento adjunto al proyecto “ManualUsuarioSparta.pdf”
Una vez visto el funcionamiento de la creación de perfiles, se pasa a validar, la modificación y eliminación de los mismos, para ello se pulsará sobre la opción de menú listado de perfiles, y una vez dentro se seleccionará el perfil que se desea modificar o eliminar.

**Modificar eliminar perfiles:** Al pulsar sobre el perfil seleccionado, se podrán modificar los datos que se deseen, y en función de la opción seleccionada, ya sea la de eliminar o la de modificar, aparecerá un mensaje u otro.
Al igual que desde el menú de la aplicación se puede navegar hasta la creación de perfiles. Para ello se debe tener permiso para realizar acciones sobre los usuarios, de no ser así, al intentar navegar a las pantallas de usuarios aparecerá el siguiente mensaje:

![Ilustración 30: Usuarios no autorizado](image)

**Dar de alta usuarios:** En caso de tener permiso para la creación de usuarios se deben llenar todos los campos necesarios para poder dar de alta un usuario en la aplicación. En caso contrario se muestran mensajes informativos. Los campos obligatorios son, id, nombre, 1º apellido, contraseña, email y perfil.

![Ilustración 31: Datos de usuarios incorrectos](image)

![Ilustración 32: Datos de usuarios correctos](image)
Una vez verificada toda la parte referente a la seguridad se pasa a mostrar, la sección de consultas y comparaciones. Ambas secciones, al no tener casi que introducir resultados, ya que todos los datos son seleccionados con desplegables, es difícil cometer errores por parte del usuario.

**Consultas:** Pueden realizarse por transformadores o por ensayos. En ambos casos, después de seleccionar los datos por los que se desee filtrar, se deberá pulsar sobre el botón buscar.

<table>
<thead>
<tr>
<th>grupo conexión</th>
<th>Transformador</th>
<th>Tensión:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fabricante:</td>
<td>Primario:</td>
<td>y:</td>
</tr>
<tr>
<td>Ubicación:</td>
<td>Secundario:</td>
<td>y:</td>
</tr>
<tr>
<td>Num. Serie:</td>
<td>Potencia entre:</td>
<td>y:</td>
</tr>
<tr>
<td>Refrigeración:</td>
<td>Año entre:</td>
<td>y:</td>
</tr>
<tr>
<td>Asamblea Fábrica:</td>
<td>y:</td>
<td></td>
</tr>
<tr>
<td>Asamblea Campo:</td>
<td>y:</td>
<td></td>
</tr>
</tbody>
</table>

**Ilustración 33: Validación consulta transformadores**

<table>
<thead>
<tr>
<th>Fabricante</th>
<th>N. Serie</th>
<th>Tensión</th>
<th>Potencia</th>
<th>Año</th>
<th>Última Actualización</th>
</tr>
</thead>
<tbody>
<tr>
<td>M40-5</td>
<td>20/20/20/20/20</td>
<td>200.00</td>
<td>2008</td>
<td>02/02/2016 11:43:14</td>
<td></td>
</tr>
<tr>
<td>M40-6</td>
<td>20/20/20/20/20</td>
<td>200.00</td>
<td>2009</td>
<td>04/02/2016 13:30:52</td>
<td></td>
</tr>
<tr>
<td>M40-14</td>
<td>20/20/20/20/20</td>
<td>200.00</td>
<td>2010</td>
<td>05/02/2016 16:27:32</td>
<td></td>
</tr>
<tr>
<td>Prueba</td>
<td>12/12/12/12/12</td>
<td>200.00</td>
<td>2011</td>
<td>13/01/2016 13:44:52</td>
<td></td>
</tr>
<tr>
<td>M40-8</td>
<td>20/20/20/20/20</td>
<td>200.00</td>
<td>2012</td>
<td>06/02/2016 23:28:24</td>
<td></td>
</tr>
</tbody>
</table>

**Comparativas de datos:** Al igual que sucede en el caso de las consultas, las comparativas pueden realizarse por ensayos o por transformadores.

<table>
<thead>
<tr>
<th>tipo de ensayo</th>
<th>fecha</th>
<th>estado</th>
<th>observaciones</th>
<th>resultado</th>
</tr>
</thead>
<tbody>
<tr>
<td>Medida de resistencia de aislamiento</td>
<td>28/12/2009</td>
<td>ABIERTO</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Medida de resistencia de aislamiento</td>
<td>25/12/2009</td>
<td>ABIERTO</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Medida de resistencia de aislamiento</td>
<td>28/04/2010</td>
<td>ABIERTO</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Para la realización de ambas consultas es necesario, seleccionar en primer lugar un transformador, para ello se deberá seleccionar, si así lo desea el usuario, unas características, y al pulsar sobre el botón buscar aparecerán los transformadores que cumplen la búsqueda, si no hay ninguno avisará de que no se obtuvieron resultados.

Ilustración 35: Comparativa de transformadores sin resultados

Ilustración 36: Comparativa de transformadores con resultados

En el caso de realizar comparativas de transformadores, tras seleccionar los transformadores deseados, se pulsará sobre ellos y se abrirá una nueva ventana en la que visualizar todos los datos.
En el caso de comparativas de ensayos, se seleccionará un único transformador, y al pulsarlo sobre los ensayos deseados, aparecerán en la parte inferior de la página.
Una vez comprobado el funcionamiento de todos los “extras” de la aplicación, se pasa a verificar lo relacionado con los transformadores y ensayos asociados con estos, como es el dar de alta transformadores, modificar transformadores, dar de alta ensayos y modificar ensayos.

**Alta de transformadores:** Para dar de alta transformadores dentro de la aplicación, es necesario pinchar sobre el link disponible en el menú, “dar de alta transformador”, una vez realizado esto aparece una página en la cual se deben introducir los datos que van a estar asociados a este equipo (ilustración 39). En caso de no introducir correctamente todos los campos obligatorios la aplicación no dejará pasar al siguiente paso, y mostrará el error. Los campos obligatorios en el primer paso son, grupo de conexión, fabricante, ubicación y número de serie.

**El Grupo de Conexión debe estar obligatoriamente cumplimentado**

<table>
<thead>
<tr>
<th>DAR DE ALTA TRANSFORMADOR: Paso 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Grupo de Conexión:</td>
</tr>
<tr>
<td>Fabricante:</td>
</tr>
<tr>
<td>Ubicación:</td>
</tr>
<tr>
<td>Número de Serie:</td>
</tr>
<tr>
<td>¿Trifásico/Monofásico?:</td>
</tr>
<tr>
<td>Destino Previsto:</td>
</tr>
<tr>
<td>Año de Fabricación:</td>
</tr>
</tbody>
</table>

Ilustración 39: Error alta de transformadores Paso 1

Si se han introducido todos los datos correctamente y se pulsa sobre el botón siguiente se avanzará al paso 2 (ilustración 40) En este paso existe un checkbox que permite copiar datos de otro transformador, si esta marca esta seleccionada, y no se ha seleccionado ningún transformador se avisará de este hecho.

**Debe copiar los datos de un Trafo**

<table>
<thead>
<tr>
<th>DAR DE ALTA TRANSFORMADOR: Paso 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>¿Desea copiar los datos de otro Trafo?</td>
</tr>
<tr>
<td>Fabricante:</td>
</tr>
<tr>
<td>Número de Serie:</td>
</tr>
</tbody>
</table>

Ilustración 40: Error alta de transformadores Paso 2

Si este paso se realiza correctamente al pulsar al siguiente, se mostrará el paso 3 (ilustración 41), en el proceso de alta, si no se introducen los campos obligatorios, la aplicación dará un aviso impidiendo seguir con el proceso.
Ilustración 41: Error alta de transformadores Paso 3

Tras completar el paso 3 correctamente (ilustración 41), y pulsar en siguiente, se muestra la última sección para la conclusión del proceso de alta, en el que hay que introducir una serie de datos numéricos. En caso de que los datos numéricos introducidos no tengan el formato correcto, no se podrá terminar con el proceso. Una vez se introduzcan los datos correctamente, tras pulsar sobre el botón finalizar, se dará por terminada el alta del transformador.

Error recuperando la información de Arrollamientos

Modificar transformador: una vez generado un transformador es posible modificar los datos del mismo, en este caso valdrá con pulsar sobre el botón de guarda de la pantalla “Edición de Transformador”. Si alguno de los campos modificados no es válido, se mostrará un mensaje de error, indicando el lugar en el que se ha producido el error. En caso de que todo sea correcto se muestra un mensaje de cambios almacenados correctamente.
Al transformador, también es posible modificarle el estado, al realizar esta acción, no se mostrará ningún mensaje de aviso, ya que no hay ninguna restricción en el cambio de estado.

También existe la posibilidad de generar informes sobre el transformador, para ello desde la pantalla de datos del transformador, existe un botón, que tras pulsarlo, redirecciona a un formulario en el que se podrá exportar todos los datos del transformador en un documento PDF, y todas las gráficas generadas por los datos del mismo. En este formulario se debe pulsar sobre el botón de generar informe y el navegador permitirá descargar el fichero comprimido que contiene todos los documentos.
Ilustración 45: Generación de informes

**Crear ensayos:** A cada transformador, como se vio en puntos anteriores, se le pueden asociar ensayos, por lo general todos los ensayos se añaden pulsando sobre el botón “Añadir”, y una vez pulsado, tras seleccionar el tipo de ensayo, se añadirá a la lista de ensayos pertinente.

Ilustración 46: Añadir ensayo

Este funcionamiento únicamente varía en un grupo de ensayos, en el que es necesario, importar un fichero, para extraer los ensayos. La aplicación se encarga de comprobar si el fichero seleccionado es del formato correcto y si tiene la estructura adecuada para ser importado. En caso de no ser así mostrará un aviso de error.

**Modificar ensayos:** una vez creados los ensayos, al acceder a ellos se podrá modificar e introducir los datos del ensayo, y modificar el estado del ensayo. En el caso de la modificación del ensayo bastará con pulsar sobre el botón de cerrar, y de esta manera se modificará el estado, no se mostrará ningún mensaje de esta acción.

Ilustración 47: Estados antes de cerrar, después de cerrar el ensayo

Al modificar los datos del ensayo, en el caso en el que todo haya sido correcto, es decir que no se haya introducido ningún dato erróneo, la aplicación no realizará ninguna acción
visible y todos los datos serán almacenados en la base de datos. Si por el contrario se produce algún tipo de error, la aplicación mostrará que se ha producido un error y no almacenará los cambios.

![Imagen de la interfaz de usuario con datos erróneos al modificar un ensayo](image-url)
Capítulo 4: Conclusiones

Una vez finalizado el proceso de desarrollo del sistema, se ha obtenido una aplicación que satisface los objetivos que se perseguían, marcados en la especificación de requisitos. Desde el punto de vista del usuario final se puede decir que se ha construido una aplicación amistosa, de fácil manejo y atractiva, dentro del marco en el que es posible realizar estas características. Las pruebas realizadas han resultado satisfactorias y se cree firmemente que esta aplicación agiliza la gestión y análisis de transformadores, dentro del proceso que todo equipo de este equipo debe satisfacer para dar servicio a los usuarios finales.

Entre las dificultades principales que se han encontrado se pueden mencionar las siguientes:

(a) Para poder generar una versión de la aplicación que permitiera trabajar de manera OFFLINE, existía el problema de cómo poder consultar los datos cuando no se tuviera conexión al servidor, y como servir el contenido de las páginas. Para solucionar esto se genera un instalador para poder ejecutar un servidor de aplicaciones en la máquina local, con una base de datos SQL Server de manera nativa.

(b) En la generación de gráficas, en las primeras fases de su desarrollo, surgieron problemas de permisos en cuanto a la gestión que se debía realizar con el Excel, por este motivo se tuvo que dar permiso al usuario ASP.NET, en el servidor para permitir lanzar procesos relacionados con el Excel.

(c) Surgió la necesidad de mostrar al usuario que versión de la aplicación se estaba utilizando, si la versión OFFLINE, o la versión ONLINE. El problema radicaba en como diferenciar con que versión se estaba trabajando de una manera dinámica. Para solucionar este problema, al cargar cada una de las páginas, se comprueba desde que dirección IP se está accediendo a la aplicación, si ambas son la misma quiere decir que se está utilizando la versión OFFLINE, ya que la dirección del cliente y del servidor son la misma. Si por el contrario con direcciones distintas se está utiliza la versión ONLINE.
(d) Muchas de las secciones de los diferentes formularios, se muestran en distintas pantallas, por ese motivo surgió un problema en cuanto a no tener que repetir código. Para solucionar esto se utilizaron controles de manera que se pudieran incrustar en los diferentes formularios, teniendo que tocar el código únicamente en un lugar en caso de modificación.

(e) El servidor de aplicaciones en el que se encuentra alojada la aplicación, tenía otras aplicaciones Web ya instaladas, de manera que surgieron conflictos en cuanto a las librerías que se utilizaban, para solucionar este problema se crearon directorios virtuales dentro del IIS, para que estos conflictos fueran solventados.

Como trabajos futuros y mejoras se proponen las siguientes:

(a) Para un futuro, sería interesante estudiar y buscar la manera en la que poder realizar las secciones ONLINE y OFFLINE, para el usuario, permitiendo acceder a ambas aplicaciones con la misma URL, evitando tener que importar transformadores y exportando la base de datos del servidor.

(b) Incluir la gestión del resto de equipos que tiene que ser introducidos dentro de la red de Unión Fenosa, ya sean Interruptores, Disyuntores, Bornas, Reguladores, etc.

(c) Adaptar la interfaz a otros tipos de dispositivos móviles de manera que pueda ser visualizada correctamente en todos ellos.
Bibliografía


REFERENCIAS ELECTRÓNICAS

[WEB01] Microsoft Developer Network
Página oficial de soporte de Microsoft .NET
http://msdn.microsoft.com

[WEB02] Microsoft SQL Server
Página oficial de Microsoft para descargar SQL Server
http://www.microsoft.com/spain/sql/downloads/default.mspx

Enciclopedia libre.
http://www.wikipedia.com

[WEB04] ASP .NET
Página oficial ASP .NET
http://www.asp.net

[WEB05] “El Guille”
Información y soporte sobre programación basada en .NET
http://www.elguille.info

[WEB06] Code Project
Ayuda sobre programación con tecnologías de .NET
http://www.codeproject.com
ANEXO I: Detalle
Ensayos de Fábrica
Ilustración 49: Ensayo de calentamiento
Ilustración 50: Ensayo tipo rayo

Ilustración 51: Ensayo de intensidad homopolar
Ilustración 52: Ensayo de capacidad y tangente delta

Ilustración 53: Ensayo de pérdidas en vacío
Ilustración 54: Ensayo tipo maniobra

Ilustración 55: Ensayo de pérdidas en vacío
Ilustración 56: Ensayo de medidas de ruido

Ilustración 57: Ensayo de regulador
Ilustración 58: Ensayo de pérdidas en carga

Ilustración 59: Ensayo de resistencia de arrollamientos
Ilustración 60: Ensayo de resistencia de aislamiento

Ilustración 61: Ensayo de tensión aplicada
Ilustración 62: Ensayo de relación de transformación
Ilustración 63: Ensayo de tensión inducida

Ilustración 64: Ensayo de tensión inducida sin DDPP
ANEXO 2: Hardware y Software Utilizado
**Hardware**

Para la implementación de la aplicación Web he utilizado el Visual Studio 2008, el cual permite realizar la simulación de un servidor de aplicaciones web, como es en este caso, el Internet Information Server. Este sistema permite publicar la aplicación web como si se encontrara realmente alojada en un servidor de aplicaciones Web. Para el desarrollo de todo el sistema, debido a los distintos lugares en los que se ha programado, se han utilizados dos equipos, en primer lugar un portátil de la marca “DELL”, con un procesador Intel Core 2 Duo, a 2.0 Ghz y que contaba con 2GB de memoria RAM. El otro equipo utilizado ha sido un ordenador de sobremesa, también de la marca “DELL”, con procesador Intel Core 2 Duo, a 2.33 Ghz y que contaba con 2GB de memoria RAM. El sistema operativo utilizado en ambos equipos ha sido Windows XP Professional.

**Software**

De las tecnologías software mencionadas en esta memoria se ha elegido la arquitectura .Net para desarrollar nuestra aplicación Web, esta elección ha sido motivada por el potencial, la portabilidad y la facilidad para desarrollar aplicaciones que ofrece esta arquitectura.

El entorno de desarrollo ha sido Microsoft Visual Studio .Net 2008. Se trata de un entorno de desarrollo creado por Microsoft, el cual gracias a su gran comunidad de usuarios, asistentes y sobre todo la documentación MSDN, lo convierte en una plataforma idónea para desarrollar grandes aplicaciones, ganando tiempo, seguridad y eficacia a la hora de programar.

El lenguaje de programación elegido ha sido C#, por ser orientado a objetos, moderno, potente y con seguridad de tipos. Para el desarrollo de las páginas web se ha utilizado el lenguaje ASP.

Respecto a la aplicación Web se ha utilizado el servidor de aplicaciones Web de Microsoft denominado Internet Information Server (IIS), que permite publicar de una manera sencilla dentro de una red, las aplicaciones Web. Además este servidor de aplicaciones Web se puede conseguir de una manera gratuita.

Como gestor de base de datos, se ha decidido utilizar Microsoft SQL Server Express Edition, el cual está perfectamente integrado dentro del entorno de Microsoft Visual Studio, además de dar la posibilidad de realizar de una manera sencilla, todas las tareas necesarias para la realización de esta aplicación. Además esta versión de SQL Server es distribuida de manera gratuita por Microsoft.