



Proyecto Fin de Carrera

TDR-LAB 2.0 Improved TDR Software for soil water content and electrical conductivity measurements.

Autor

Enrique Fatás Clavero

Directores

José Vicente Ruiz
(U.S. Food and Drug Administration)

David Moret-Fernández
(Dpto. de Suelo y Agua. Estación Experimental de Aula Dei, CSIC)

Ponente

Víctor Viñals Yúfera
(Dpto. Informática e Ingeniería de Sistemas)

Escuela de Ingeniería y Arquitectura
2013

TDR-LAB 2.0 improved TDR software for soil water content and electrical conductivity measurements.

RESUMEN

En la actualidad, la técnica “Reflectometría de Dominio Temporal” (TDR por sus siglas en inglés) proporciona a los grupos de investigación en agronomía, hidrología o ciencias del suelo, un método fiable para la medida de la humedad (θ) y la conductividad eléctrica (σ) del suelo, factores que regulan el crecimiento y desarrollo de las plantas. La existencia de aplicaciones informáticas que manejen TDR es escasa, por lo que se plantea la necesidad de desarrollar aplicaciones compatibles con los diferentes equipos TDR disponibles en el mercado que aporten productividad y fiabilidad en la medida de θ y σ . En la actualidad, resulta inviable para un organismo investigador el coste extra del software de control necesario para cada nuevo dispositivo TDR. Para reducir costes y complejidad de manejo, surgió la versión inicial de la aplicación TDR-LAB que, al comienzo del proyecto se encontraba en su versión 1.2.3.

Este trabajo presenta una versión mejorada del software (TDR-LAB V2.0) que permite estimar θ y σ utilizando diferentes tipos de análisis de onda. El proyecto ha sido desarrollado en C# con el entorno de programación Visual Studio 2010. Respecto a la gestión de datos, se han desarrollado dos versiones diferentes y compatibles: (a) una que utiliza una base de datos centralizada SQL-Server 2008; y (b) una segunda versión simplificada para ordenadores portátiles de bajo coste que utiliza un sistema de archivos en formato XML. La aplicación permite además exportar información a ficheros de valores separados por coma (CSV), lo que permite analizar resultados utilizando software de terceros. TDR-LAB V2.0 presenta una interfaz de fácil manejo, puede conectarse a distintos ecómetros TDR (Tektronixs 1502C, TDR100 Campbell y TRASE), e incluye un sistema de análisis de ondas TDR para la estimación de θ y σ mediante cuatro métodos diferentes: manual; gráfico por tangentes; gráfico a partir del análisis de la derivada de las ondas TDR; y numérico por análisis inverso de la onda TDR.

Además, se ha implementado un nuevo modelo de configuración de sondas que permite el uso de un componente multiplexor modelo SDMX50 (Campbell Sci.). El sistema permite una configuración anidada de hasta tres niveles, con capacidad de conectar hasta 512 sondas TDR, en vez de la única sonda actual. Mientras que el puerto serie se utiliza para controlar los TDR y recoger las medidas, el control del multiplexor se realiza a través del puerto paralelo.

TDR-LAB V2.0 permite al usuario almacenar la información mediante un sistema virtual de ficheros y proyectos análogo al explorador de Windows. De esta manera, se facilita la organización del trabajo al usuario. Además, el sistema es capaz de detectar si el usuario está usando un sistema de datos obsoleto a la versión del programa y, de manera automática, realiza los cambios necesarios para actualizarse y funcionar de forma correcta.

En resumen, el proyecto ha conseguido actualizar el modelo de datos para optimizar el rendimiento de la versión previa de la aplicación, cambiar la interfaz gráfica para facilitar el proceso de gestión y análisis de los resultados obtenidos, así como incorporar nuevos dispositivos TDR con la posibilidad de usar el nuevo sistema de configuraciones multiplexadas. Este proyecto, de carácter interno para el desarrollo de investigaciones en el Consejo Superior de Investigaciones Científicas (CSIC), representa un gran avance para la medida de propiedades del suelo, y sus conclusiones y posibilidades han sido presentadas en un congreso internacional (Anexo F), referencia de los investigadores en estas temáticas.

Índice General

PARTE I MEMORIA	8
1. Introducción	10
1.1. Técnica TDR: antecedentes.....	10
1.2. Lenguajes y bases de datos usados.....	11
1.3. Responsables del proyecto.....	11
1.4. Objetivos del proyecto.....	12
1.5. Estructura del documento.	13
1.6. Introducción al funcionamiento de la aplicación.....	13
2. Metodología	16
2.1. Análisis de rendimiento de la versión 1.2.3.....	16
2.2. Introducción al sistema de multiplexado de sondas.	17
2.3. Nuevo ecómetro TDR (TRASE).	19
2.4. Incorporación de nuevos métodos de análisis.	19
2.5. Diseño de una nueva interfaz gráfica.....	22
3. Resultados	23
3.1. TDR-LAB Lite.	23
3.2. Incorporación del dispositivo multiplexor de sondas Campbell SDMX50.	26
3.3. Incorporación dispositivo ecómetro TRASE de Soil Moisture Equip. Corp.....	27
3.4. Nuevas medidas.	28
3.5. Implementación de la nueva interfaz gráfica.	28
4. Discusión	31
4.1. Resultados obtenidos.....	31
4.2. Desarrollo futuro.	31
5. Bibliografía	32
PARTE II ANEXOS	34
A. Entorno de trabajo.	35
A.1 Visual Studio 2010.	35
A.2 SQL Server 2008 y archivos XML.....	35
A.3 Repositorio proyecto y metodología de trabajo.	37

B. Diseño estructural de la aplicación.....	38
B.1 Organización capas de información.	38
B.2 Capa "DataAccesLayer".	40
B.3 Capa "BusinessLogicalLayer".....	44
B.4 Capa "GUI".	46
B.5 Capa "Análisis".....	48
C. Diseño de flujos.....	50
C.1 Diagramas de secuencia.....	50
C.2 Diagrama de distribución.....	51
D. Dispositivo multiplexor Campbell y su control	53
E. Capturas de pantalla de la aplicación.....	55
E.1 Formulario Splash.....	55
E.2 Formulario Acerca De.....	55
E.3 Licencia de uso de la aplicación	56
E.4 Licencia de uso de los métodos Analíticos	56
E.5 Formulario principal de gestión de proyectos	57
E.6 Detalle de modificación de propiedades	57
E.7 Configuración de un nuevo proyecto	58
E.8 Forma de generar configuraciones cable-sondas	58
E.9 Formulario para configuración de multiplexor.....	59
E.10 Detalle configurado de un proyecto	59
E.11 Repositorio de cables del proyecto.....	60
E.12 Menú para importar/exportar datos.....	60
E.13 Formulario detalle de importación/exportación.....	61
E.14 Análisis automático de medidas	61
E.15 Obtención automática de ondas.	62
F. Conference on Monitoring and Modeling Soil-Plant-Atmosphere Processes, Naples, 2013.....	63
G. Resumen Tablas utilizadas para calcular rendimientos de la aplicación	72
H. Diagrama de GANTT de los tiempos empleados en el PFC.....	74

PARTE I MEMORIA

Capítulo 1

1. Introducción

1.1. Técnica TDR: antecedentes.

En la actualidad, el uso de técnicas como la “Reflectometría de Dominio Temporal” proporciona a los grupos de investigación que trabajan en el área de la agronomía, hidrología y ciencias del suelo un método fiable para la medida de propiedades físicas del suelo. La Reflectometría de Dominio Temporal (TDR por sus siglas en inglés) es una técnica que permite realizar, de forma no destructiva, medidas simultáneas e instantáneas de la permitividad aparente (ϵ_a), la cual está relacionada con el contenido de humedad volumétrica del suelo (θ) (Topp et al., 1980), y la conductividad eléctrica de la matriz del suelo (σ_a) (Topp et al., 1988). Otras aplicaciones en el ámbito de las ciencias del suelo son, por ejemplo, medidas de niveles de agua en tubos de Mariotte aplicada la técnica de infiltrometría de tensión (Moret et al., 2004), el potencial mátrico (Or and Wraith, 1999) o la conductividad eléctrica de la solución del suelo (σ_w) (Moret-Fernández et al., 2012). En general, las estimaciones de θ se realizan a partir del análisis gráfico de ondas TDR utilizando el método de las tangentes (Herkelrath et al., 1991) o el método de las derivadas (Timlin and Pachepsky, 1996). El valor de σ_a se estima principalmente a partir del análisis de la amplitud de la onda TDR a tiempos largos, según procedimiento expuesto por Lin et al. (2008). El modelado de ondas TDR por inversión numérica resulta una alternativa interesante para la estimación de θ y σ_a (Oswald et al. , 2003; Greco, 2006; Heimovaara et al., 2004).

Hasta la fecha, el diseño de equipos TDR específicos para la medida de θ y σ_a ha mostrado un vertiginoso aumento, prueba de ello son las numerosas empresas que fabrican y comercializan estos equipos: Adcon, IMKO, Streat Instruments Ltd., Campbell Scientific, Global Water Instrumentation, Inc., Environmental Sensors, Inc. Automata Inc., Meteolabor AG, Dynamax, Soil Moisture Equipment Corporation. El primer equipo utilizado para medidas de humedad del suelo fue el modelo 1502C Metallic, fabricado por Tektronix (Beaverton, Oregon). Este equipo, que incorpora una pequeña pantalla LCD que muestra las ondas TDR, permitía únicamente un análisis manual de las ondas TDR, hecho que limitaba su uso. Esta limitación fue inicialmente resuelta por Evett (2000), con el desarrollo del programa TACQ (<http://www.cprl.ars.usda.gov/programs/>), que trabajando con una versión DOS y compatible con el ecómetro Tektronix 1502C y los multiplexores SDMX50 y Dynamax, permitía medidas directas de θ y σ_a , y multiplexado de varias sondas TDR. Posteriormente, en 1993, el Soil Physics Group de Utah State University, Logan, Utah, USA, creó el programa WINTDR, el cual, compatible con el ecómetro Tektronix 1502C y desarrollado para MS Windows 95 y XP, presentaba una interfaz más amigable y medidas eficientes de θ y σ_a (<http://soilphysics.usu.edu/wintdr/download.htm>). Sin embargo, esta versión solo llegó a funcionar correctamente para MS Windows 2000, mostrando muchos problemas para versiones superiores de Windows. Otros software's TDR disponibles son el WinTrase, compatible únicamente con el ecómetro de la casa Soil Moisture TDR, y el programa PC-TDR, compatible con el equipo TDR-100 (Campbell Sci.). Más recientemente, Moret-Fernández et al. (2010) desarrollaron un nuevo software TDR, TDR-Lab 1.0, para la estimación de θ y σ_a por TDR. Este nuevo programa supuso un nuevo avance con respecto a sus predecesores ya que, siendo compatible con dos equipos TDR diferentes, incluía una interfaz muy amigable para análisis y manejo de ondas TDR. Sin embargo, esta versión, que quedó incompleta, no incluye la opción de multiplexación y utiliza una base de datos centralizada muy pesada que impide su uso en ordenadores de bajo coste.

A pesar de la considerable expansión de la técnica TDR para la medida de θ y σ_a , la existencia de programas gratuitos compatibles con diferentes equipos TDR es aún muy limitada. A su vez, la versatilidad del software disponible para el análisis y almacenamiento de ondas TDR es a su vez muy restringida. Este trabajo presenta una nueva versión del software TDR-Lab 1.0 (Moret-Fernández et al., 2011), el TDR-LAB V2.0, que siendo compatible con diferentes ecómetros TDR (Tektronixs 1502C, TDR100 Campbell y TRASE), supone una significativa mejora con respecto a la versión existente. El nuevo software incluye dos variantes, una de laboratorio (TDR-Lab) y otra de campo más ligera (TDR-Lab Lite), presenta una interfaz de fácil manejo, incluye un sistema automático de medidas y análisis de ondas TDR y permite la multiplexación de numerosas sondas TDR.

1.2. Lenguajes y bases de datos usados.

En el desarrollo de este proyecto se ha usado el lenguaje orientado a objetos C#, puesto que ya se había usado con éxito en la versión inicial 1.2.3. Además ofrece una serie de ventajas tanto al usuario como al programador:

- a) La primera ventaja reside en la reutilización de controles y componentes que de forma automática se adapta a la versión del sistema operativo para no desentonar con el conjunto de aplicaciones ya instaladas en el ordenador. Desde el punto de vista del programador, permite el desarrollo de aplicaciones de escritorio con fáciles e intuitivas pestañas.
- b) La segunda es la posibilidad de controlar puertos del ordenador con total facilidad. Permite al programador utilizar las clases que ofrece el entorno para manipular la información de entrada/salida del ordenador.

En el proyecto se han desarrollado dos soluciones complementarias para implementar las bases de datos, SQL-Server2008 y un sistema de archivos XML. La necesidad de este esfuerzo se justifica en el apartado 2.1, donde se analizan los problemas de eficiencia, especialmente en ordenadores de bajas prestaciones.

Para completar este apartado, hay que explicar que para poder controlar los dispositivos de los distintos fabricantes, se ha recopilado, en el caso de TDR100, el conjunto de instrucciones del SDK, mientras que para los otros dos dispositivos se han implementado las comunicaciones desde cero.

1.3. Responsables del proyecto.

El proyecto se realiza directamente bajo la tutela de tres responsables.

El director principal del proyecto es D. José Vicente-Ruiz (Máster en Ingeniería de Sistemas e Informática), autor de la versión TDR-Lab 1.2.3., y que se encargará de tutelar en los aspectos de ingeniería de software

Colabora en la dirección el Doctor David Moret-Fernández (PhD en Ingeniería Agrónoma, científico titular del CSIC). Se encargará de todos los aspectos técnico y científicos del proyecto. Será el encargado de decidir qué aspectos gráficos se mejoran para adaptarlos a las nuevas exigencias y de validar los resultados teóricos que muestra la aplicación en sus formularios.

El Doctor Víctor Viñals se encarga de la tutela académica, supervisando además el desarrollo de la aplicación, y proporcionando parte del material necesario para el desarrollo del proyecto.

1.4. Objetivos del proyecto.

El proyecto tiene como objetivo el diseño y la implementación de una nueva versión de la aplicación TDR-LAB (Moret-Fernández, et al., 2010) que ofrezca una funcionalidad completa y automatizada para la medida de las propiedades del suelo, y su posterior instalación, validación y prueba en el Departamento de Suelo y Agua de la Estación Experimental de Aula Dei (CSIC), Zaragoza.

Los objetivos del proyecto se enumeran a continuación, representando los agrupados bajo nuevas funcionalidades alrededor del 75% del tiempo de desarrollo. El anexo H representa el Diagrama de Gantt de los tiempos establecidos a cada tarea.

1. Mejora de prestaciones:
 - a. Nueva capa de acceso a datos que optimice el rendimiento de la aplicación dependiendo de las prestaciones del ordenador en donde se ejecute el programa.
 - b. Solución de errores que se identifiquen en la versión 1.2.3.
 - c. Nueva interfaz gráfica que facilite el proceso de configuración de plantillas para el trabajo científico.
 - d. Mejorar el algoritmo que implementa el método de tangentes con un nuevo cálculo más preciso.
2. Nuevas funcionalidades
 - a. Implementación de un nuevo sistema de control de multiplexores que aumenta el número de sondas que se pueden utilizar de forma automatizada. Pasaremos de la lista de sondas inicial (multiplexado manual) a las 512 sondas que potencialmente podemos conectar y dirigir mediante el dispositivo multiplexor de Campbell SDMX50.
 - b. Incorporación del control de un nuevo sistema ecómetro el dispositivo TDR denominado TRASE, de la compañía Soil-moisture Equipment Corp. (Figura 1)



Figura 1. Ecómetro TRASE de la compañía Soil-moisture Equipment Corp

1.5. Estructura del documento.

Las siguientes secciones de este documento tratarán de abordar las diferentes fases que se han seguido a lo largo del proyecto junto con las complicaciones a las que se ha tenido que hacer frente durante las mismas.

Durante el capítulo segundo se especificarán las distintas metodologías empleadas para la resolución de cada uno de los objetivos planteados para este proyecto. Se abarcará el proceso de detección y localización de las limitaciones que ofrece el sistema y el proceso seguido para lograr su resolución.

En el tercer capítulo se describen los resultados obtenidos a partir de la aplicación de la metodología descrita en el capítulo anterior. Por lo que, nos limitaremos únicamente a la explicación del resultado, de forma que para profundizar en la implementación se adjuntan los anexos correspondientes.

Para finalizar, se describirá un capítulo para el debate y reflexión sobre el proyecto realizado, limitaciones y futuros desarrollos.

1.6. Introducción al funcionamiento de la aplicación.

En este apartado se describe una introducción a la aplicación desarrollada, así como los componentes involucrados.

Para aclarar los conceptos que se describen en esta memoria se detallan a continuación los componentes necesarios para realizar una obtención y análisis de una onda.

Para poder utilizar esta tecnología se necesitan los siguientes componentes:

- Ordenador.
- Ecómetro TDR (modelo TDR100, Tektronixs 1502C, TRASE).



Figura 2. Ecómetros disponibles en el desarrollo del proyecto. De izquierda a derecha, TRASE, Tektronixs 1502C Y TDR100.

- Cables coaxiales.



Figura 3. Detalle del cable coaxial utilizado, para conectar los ecómetros TDR a las sondas.

- Cables (USB-RS232, USB-Paralelo con adaptador).



Figura 4. Detalle de un adaptador USB-Paralelo con el conector utilizado para el control de los multiplexores.

- Sondas Trifilares.



Figura 5. Sonda utilizada para la adquisición de ondas

- Dispositivo Multiplexor Campbell SDMX50.



Figura 6. Multiplexor utilizado en el desarrollo del proyecto, modelo Campbell SDMX50.

El formato de conexión de todo el conjunto se realiza inicialmente desde el ordenador. El ecómetro TDR se conecta y controla desde el ordenador a partir del cable RS232. Si el ordenador no dispone de salida para dicho puerto, se necesitará un adaptador específico. En la versión 1.2.3. se conectaba directamente la sonda a utilizar al ecómetro TDR mediante el cable coaxial. En esta nueva versión 2.0, se permite conectar el ecómetro TDR a un multiplexor, y desde éste se conectan las diferentes sondas mediante cables coaxiales. La comunicación entre el TDR y los multiplexores se realiza también con cables coaxiales. Para establecer el control de los diferentes multiplexores se establece un cableado desde la salida del puerto paralelo del ordenador hasta los conectores situados en la parte superior del multiplexor.

El proceso de obtención de una onda requiere de varios procesos. En primer lugar, hay que configurar manualmente todos los dispositivos necesarios de forma que se establezca un circuito físico. En segundo lugar, se necesita configurar informáticamente tanto el ecómetro TDR como el resto de cableado incluyendo los multiplexores. En este proceso, el ordenador configura el ecómetro TDR con la sonda que se desea medir y las variables necesarias para su correcta obtención. Posteriormente, el ordenador configura el resto de multiplexores incluidos en la configuración de forma que establece el circuito necesario. Finalmente, una vez que se han configurado todos los dispositivos es cuando se ejecuta la orden de obtener la onda (Figura 7)

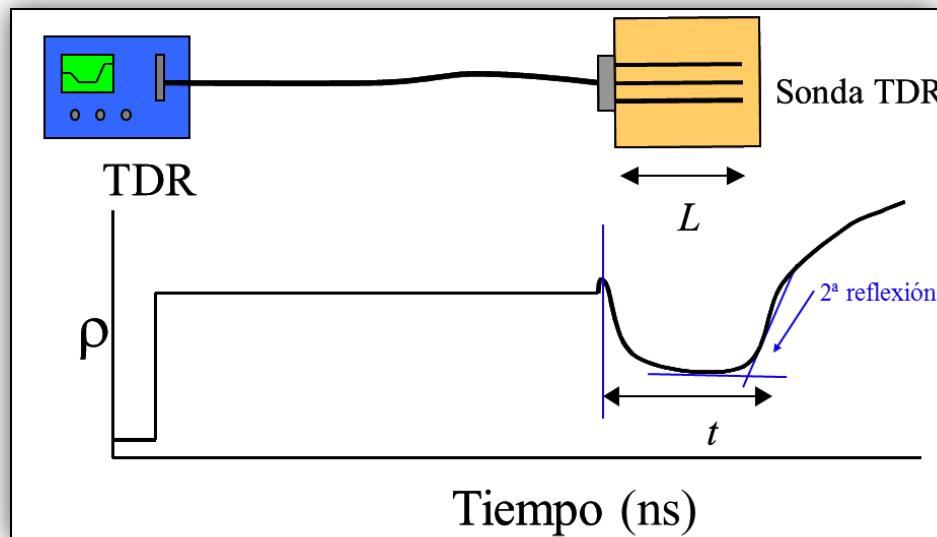


Figura 7. Esquema de la recepción de la onda a su paso por los diferentes componentes.

Como se aprecia en la Figura 7 debido a las diferentes constantes eléctricas que tenemos en el circuito, somos capaces de establecer e identificar todas las variables necesarias para el análisis de la onda obtenida. Dentro de los análisis disponibles tenemos los que ya estaban en la versión 1.2.3 y los nuevos métodos descritos en el apartado 3.5 de este manual.

Capítulo 2

2. Metodología.

En este apartado se describe la metodología empleada para resolver los problemas planteados en los objetivos, a saber:

- Análisis de rendimiento de la versión 1.2.3.
- Introducción al sistema de multiplexación de sondas.
- Nuevo sistema ecómetro TDR.
- Incorporación de nuevas medidas de análisis.
- Diseño de una nueva interfaz gráfica.

2.1. Análisis de rendimiento de la versión 1.2.3.

El primer punto de discusión es el estudio y análisis de las prestaciones ofrecidas por la primera versión existente de la aplicación. Se había detectado por parte de los investigadores del CSIC que, en determinadas circunstancias, se producía una reducción en las prestaciones ofrecidas por el sistema. Por lo tanto, la primera parte del desarrollo del proyecto consiste en identificar dónde y por qué se produce la citada disminución de prestaciones.

Tras sucesivas reuniones con los investigadores, se analiza la percepción mostrada por los investigadores del CSIC de que al realizar la primera carga de los formularios, el tiempo de respuesta de la aplicación superaba el que presentaban las sucesivas cargas de información. Se puede apreciar en la Figura 8 el rendimiento que presenta la aplicación en distintos ordenadores obtenidos a partir de los resultados de las tablas del Anexo G.

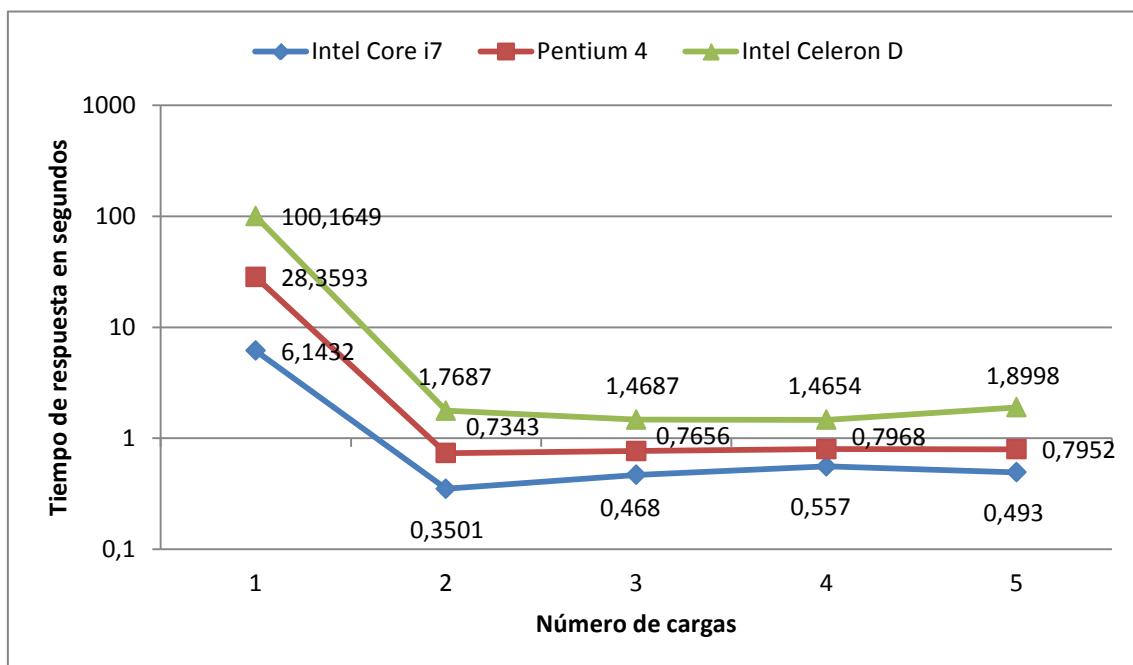


Figura 8. Gráfica con los resultados iniciales de la aplicación TDR V1.2.3.

Comparación de resultados de la aplicación en distintos procesadores, al recuperar información de la base de datos para analizar el problema de rendimiento de la primera carga de datos.

Una vez comprobado que el problema residía en la primera carga de la información en los formularios, el siguiente paso fue chequear, mediante los paneles de propiedades existentes en nuestro entorno de trabajo, que la configuración de la aplicación es la correcta. Tras haber validado esta configuración y verificar que es correcta, buscamos si el error o la limitación venían de la configuración del servicio de bases de datos en nuestro sistema operativo Windows. Llegados a este punto, nos encontramos que el sistema operativo por defecto pospone el arranque del servicio de bases de datos SQL hasta que una primera aplicación lo requiere, aunque se fuerce su inicialización al arranque del sistema operativo.

Detectado y localizado el problema se proponen dos alternativas para solucionarlo:

- a. Sustituir el actual servicio de bases de datos por otro que cumpla con las mismas funcionalidades, pero mejorando la productividad de la aplicación.
- b. Desarrollo de un motor de bases de datos mediante un sistema de archivos con formato XML.

Con el responsable científico del proyecto se analizan los requisitos que se pretenden para esta nueva versión de la aplicación. Entre esos requisitos figura el de ser una aplicación que funcione con los mismo niveles de productividad, independientemente de las prestaciones del ordenador.

Una ventaja de usar un sistema de archivos XML como alternativa a la base de datos actual reside en que es un sistema compatible con cualquier dispositivo, independientemente de la implementación. No es necesario instalar ningún complemento extra en el programa TDR-Lab para hacer funcionar esta alternativa. Por el contrario, intentar sustituir el actual sistema SQL por algún otro equivalente suponía un riesgo de tiempo, ya que no había garantías de que fuera la solución que necesitábamos.

Como se describirá en el capítulo 3.1, mediante la alternativa implementada de ficheros XML logramos dar una aproximación al problema de rendimiento de la primera carga en ordenadores de prestaciones reducidas.

2.2. Introducción al sistema de multiplexado de sondas.

Otra limitación existente en la versión 1.2.3. reside en que, aunque permite configurar varias sondas, en la práctica el investigador necesita su conexión y desconexión manual al ecómetro TDR. Con este nuevo sistema de multiplexación se pretende ahorrar tiempo en la configuración física de los distintos dispositivos.

Para la realización de este objetivo se utilizará el dispositivo multiplexor desarrollado por Campbell (SDMX50). Dicho dispositivo está formado por un sistema interno de relés configurables eléctricamente, que permite multiplexar hasta 8 salidas. Por diseño del fabricante del dispositivo, se pueden controlar hasta 512 sondas mediante un máximo de tres niveles diferentes de multiplexación.

La forma de conectar este nuevo dispositivo con el resto de hardware es a partir de dos tipos de comunicación (Figura 9). Una primera comunicación para el flujo de datos y una segunda para el control del dispositivo. Para establecer el correcto funcionamiento del flujo de datos usaremos cables coaxiales de longitud variable que nos permitirán conectar el multiplexor al ecómetro TDR o a otro multiplexor.

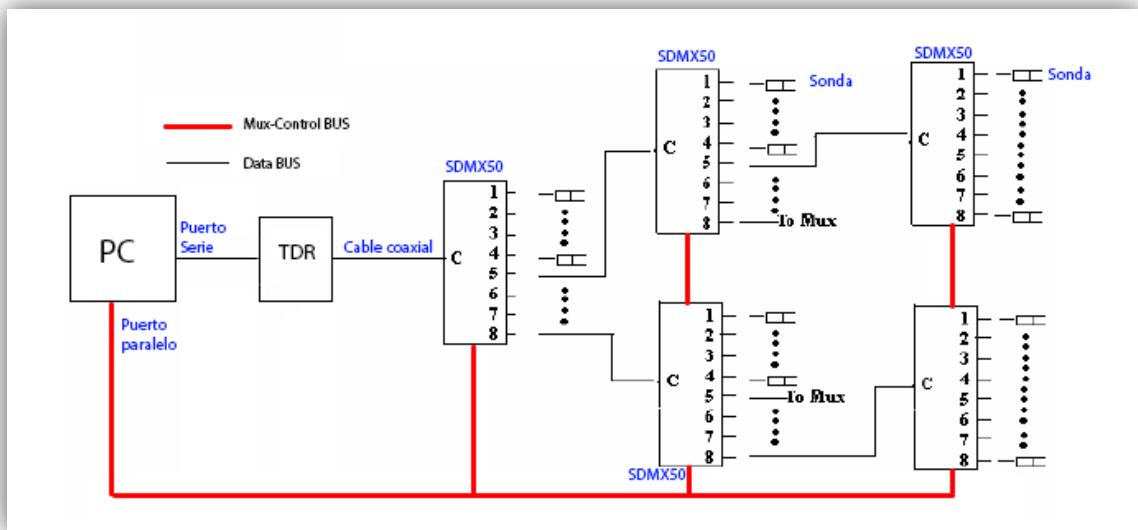


Figura 9. Esquema de los buses de datos y control.

Esquema de buses de datos (línea negra) y control (línea roja) para la conexión de los dispositivos necesarios en la tecnología TDR.

El control de los multiplexores se realiza a través de un bus de control digital de 3 cables que se conectarán al puerto paralelo. El software implementará el protocolo de control del SDMX50 para direccionar, mediante el puerto paralelo, la sonda apropiada. Una vez seleccionada la sonda, el software realizará la medida a través del dispositivo TDR conectado al puerto serie.

Se decide usar el puerto paralelo del ordenador para controlar el bus de control porque nos ofrece la posibilidad, de una forma sencilla y directa, de obtener de forma separada los cables necesarios para el control de multiplexores. Además, existen librerías de software libre desarrolladas en C# para la escritura/lectura del puerto paralelo.

Una vez definido el bus de control, el siguiente paso es describir el protocolo a implementar para la comunicación con los multiplexores (Anexo D). En el interior de cada multiplexor nos encontramos con unas pestañas que sirven para codificar el nivel de multiplexión. Esas pestañas (Figura 11) se codifican a partir de los valores de la Figura 10.

TDR100 Thumbwheel Switch	Level 1 SDMX50 Mux (TDR100+1)		Level 2 SDMX50 Muxs (TDR100+2)		Level 3 SDMX50 Muxs (TDR100+3)	
	MSD	LSD	MSD	LSD	MSD	LSD
0	0	1	0	2	0	3
1	0	2	0	3	1	0
2	0	3	1	0	1	1
3	1	0	1	1	1	2
4	1	1	1	2	1	3
5	1	2	1	3	2	0
6	1	3	2	0	2	1
7	2	0	2	1	2	2
8	2	1	2	2	2	3
9	2	2	2	3	3	0
A	2	3	3	0	3	1
B	3	0	3	1	3	2
C	3	1	3	2	3	3
D	3	2	3	3	not available	
E	3	3	not available		not available	
F	not available		not available		not available	

Figura 10. Tabla de configuración manual de los multiplexores.

Posición que se debe configurar manualmente en los multiplexores para el correcto funcionamiento.

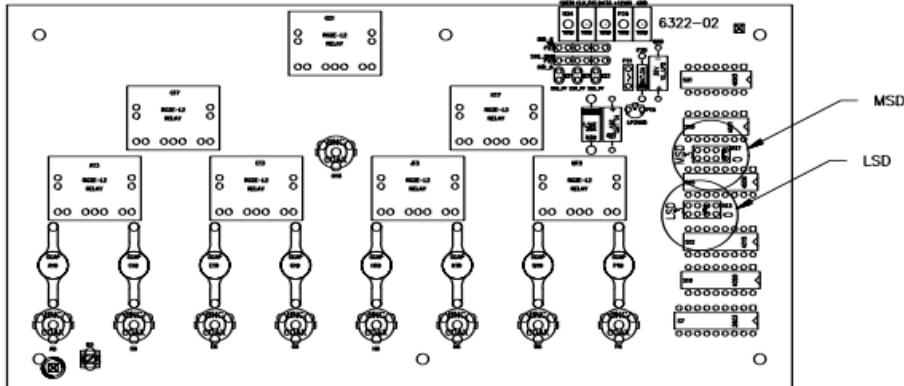


Figura 11. Detalle de los relés que hay que modificar.

2.3. Nuevo ecómetro TDR (TRASE).

Este punto describe la metodología empleada para incorporar al sistema existe un nuevo ecómetro TDR, TRASE (Figura 1), de la casa Soil Moisture Equipment Corporation. Al igual que el resto de ecómetros utilizados en el TDR-LAB, el equipo TRASE es utilizado en numerosos centros de investigación a nivel mundial para realizar medidas de θ y σ_a . Su amplia utilización nos ha llevado a incluirlo en la nueva aplicación TDR-LAB 2.0. Para tal fin, se colaborará con el investigador Carlos Regalado de Instituto Canario de Investigaciones Agrarias (Tenerife), quien dispone de un equipo TRASE en su laboratorio. Para introducir este dispositivo en nuestro sistema utilizaremos una aplicación desarrollada en otro lenguaje de programación para el ecómetro TRASE, elaborado por el Instituto Canario de Investigaciones Agrarias en colaboración en el Servicio de Investigación y Tecnología Agraria de la Junta de Castilla y León.

Al tener acceso a dicha aplicación, se ha podido analizar el funcionamiento de la aplicación, detectar sus limitaciones y las similitudes con nuestro sistema. El primer paso era estudiar el proceso de comunicación con el dispositivo, de forma que pudiera ser compatible con el uso de conectores RS-232 empleados ya en los demás ecómetros del TDR-LAB.

Lo primero que hay que hacer es configurar el ecómetro, por lo que se ejecuta en el programa la orden de configurar el TDR con los valores que mediante los formularios de configuración hemos establecido. El siguiente paso ya es obtener la medida. Para realizar este procedimiento se actúa de la misma manera que con los otros TDR ya soportados. En primer lugar, se lanza la orden de obtener la onda y, cuando el ecómetro tiene registrada la onda, le comunica al software que ha terminado, el cual, procede a su almacenamiento en la base de datos.

2.4. Incorporación de nuevos métodos de análisis.

Dado que uno de los objetivos de esta nueva versión TDR-LAB es aumentar sus prestaciones y versatilidad, se ha considerado importante incorporar nuevas utilidades de la tecnología TDR, entre las que cabe destacar, a parte de la medida de θ y σ_a , la medida de niveles de agua en tubos de Mariotte aplicada a la técnica de infiltrometría de tensión (Moret et al., 2004), el potencial mátrico (Or and Wraith, 1999) y la conductividad eléctrica de la

solución del suelo (σ_w) (Moret-Fernández et al., 2012). Así pues, en una primera fase se estudió los nuevos parámetros de suelo a incorporar, seguido por una segunda fase en la que se analizó la forma de incluir estas nuevas medidas.

Medida de nivel de agua por TDR

Utilizando una sonda TDR vertical de longitud L inmersa en agua, el nivel de agua (L-X) se calcula según (Moret et al, 2004)

$$x = L \frac{\sqrt{\varepsilon_{TDR}} - \sqrt{\varepsilon_w}}{\sqrt{\varepsilon_{air}} - \sqrt{\varepsilon_w}} \quad (1)$$

donde x es la longitud de la sonda TDR por encima del nivel de agua, ε_{TDR} es la permitividad aparente medida por el TDR, y ε_{air} y ε_w son las permitividades aparentes del agua y aire respectivamente. Para calcular el nivel de agua los valores de ε_{air} , ε_w y L deben ser previamente introducidos en el programa TDR-Lab.

Potencial mátrico y conductividad eléctrica de la solución del suelo

El potencial mátrico (ψ) y la conductividad eléctrica de la solución del suelo (σ_w) se miden con sensores cerámicos TDR (Or and Wraith, 1999). El valor de ψ se calcula a partir de una relación $\theta-\psi$ predeterminada para cada sensor (Or and Wraith, 1999). Para el caso del TDR-LAB 2.0, se utilizó la función unimodal de Van Genutchten (1980) según

$$\theta = \left[(\theta_{sat} - \theta_r) \left[\frac{1}{1 + (\alpha\psi)^n} \right]^m \right] + \theta_r \quad (2)$$

donde n representa la distribución de tamaños de poros del suelo, $m = 1-(1/n)$, α [kPa] es un factor de escala y θ_{sat} y θ_r son la humedad a saturación y residual de las placas cerámicas. Estos parámetros deben ser previamente introducidos en la aplicación TDR-LAB

La conductividad eléctrica de la solución del suelo corregida a 25 °C ($\sigma_w/25$) se calcula según (Moret-Fernández et al. 2012)

$$\sigma_{w/25} = \sigma_w f \quad (3)$$

donde f es un factor empírico expresado como (US Salinity Laboratory Staff, 1954)

$$f = 1 - 0.20346(T) + 0.03822(T^2) - 0.00555(T^3) \quad (4)$$

y $T = (T \text{ °C} - 25)/10$.

El valor de σ_w se calcula según

$$\sigma_w = \frac{\sigma_a}{\theta_{sat}^r \left(\frac{\theta}{\theta_{sat}} \right)^\beta} - \sigma_{a-s} \quad (5)$$

donde σ_{a-s} es la conductividad eléctrica de la fase sólida de las placas cerámicas, β es un factor que depende de la porosidad transmisora de las placas porosas y τ es un coeficiente de transmisión de la placa porosa saturada (Mualem and Friedman, 1991). Los valores de σ_a y θ son estimados por TDR, y σ_{a-s} (despreciable), θ_{sat} , β y τ , que son obtenidos tras previa calibración de las placas porosas, deben ser introducidos en la aplicación TDR-LAB.

Una vez definida la teoría que queremos aplicar en nuestro desarrollo, procedemos a comprobar las diferencias con el modelo actual del análisis de las ondas. Actualmente, todas las ondas establecidas dentro de un mismo proyecto tienen una misma configuración común. De forma que, en el caso de querer establecer mediciones automáticas, nos encontramos con una limitación. Como se explica en el punto 3.5, una alternativa para dar solución a esta limitación es enlazar estas configuraciones a cada una de las sondas y no al proyecto en general.

Además, la inclusión de estas nuevas medidas puede permitir al usuario configurar el número de medidas que se calculan para cada sonda, permitiendo así una escalabilidad en las mediciones requeridas para cada zona de la configuración del proyecto.

Otro punto a destacar dentro del análisis que se realiza sobre las nuevas medidas es la modificación de uno de los métodos ya implementados. El análisis de la onda por medio del método de las tangentes, en determinadas configuraciones, calcula un valor erróneo de los puntos críticos necesarios para la obtención de la tangente. El problema reside en que, cuando analiza la onda y obtiene el primer punto mínimo de la onda, no sabe precisar dónde se encuentra el siguiente punto de interés de la función. Tras analizar con el responsable científico del proyecto distintas ondas realizadas mediante un ecómetro TDR, obtenemos como conclusión que, el segundo punto de interés que se tiene que calcular debe estar comprendido en la mitad derecha de nuestra gráfica de representación. Para implementar dicha solución se desarrolla un método basado en una doble iteración. Mientras que, la primera iteración nos sirve para calcular el punto mínimo de la gráfica, la segunda iteración nos permite dividir la zona restante a la derecha en secciones, de forma que acotamos la búsqueda del segundo punto crítico, optimizando con ello los valores resultantes con el método.

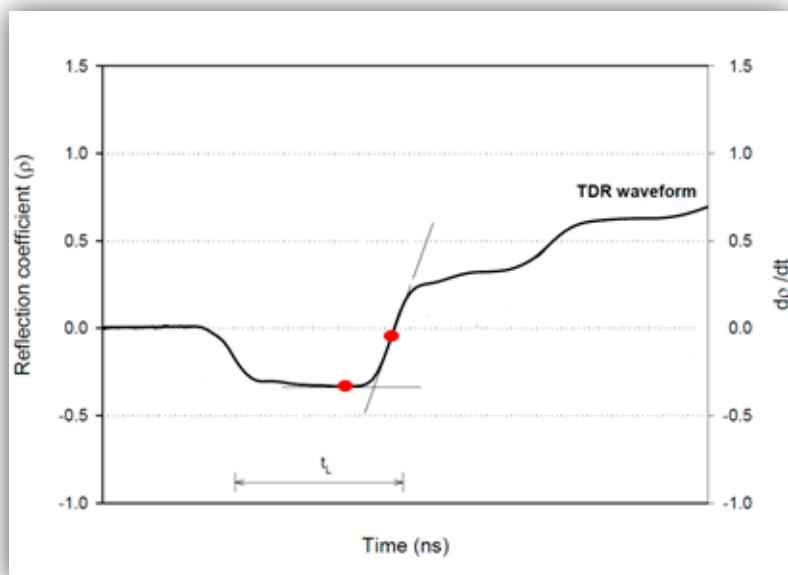


Figura 12. Detalle de la onda obtenida con los puntos críticos necesarios marcados en rojo.

2.5. Diseño de una nueva interfaz gráfica.

Con el diseño de la nueva interfaz se pretende facilitar el proceso de configuración de los distintos componentes de la aplicación, de forma que permita al usuario un manejo más intuitivo del programa. Tras varias reuniones con los usuarios y el director científico, se identificaron los formularios que resultaban demasiado complejos o de difícil comprensión, abordando un rediseño basado en dos acciones principales:

- Complementar los menús con barras de herramientas.
- Simplificar el número de pestañas en los menús de configuración.

En concreto, se actualizaron los formularios para la creación de cables-sondas y para la creación de proyectos. Además, se trabajó con la misma filosofía gráfica en todos los nuevos componentes, desarrollando los formularios de configuración de los multiplexores o la gestión del ecómetro TRASE.

Capítulo 3

3. Resultados.

En este capítulo se exponen los resultados obtenidos, resumiendo los siguientes aspectos: la nueva base de datos con ficheros XML, la gestión de los dispositivos añadidos de multiplexado y ecometría, las nuevas medidas sobre la onda capturada y, finalmente, la nueva interfaz gráfica.

3.1. TDR-LAB Lite.

TDR-LAB Lite es una versión completa de todo la aplicación, casi con la misma funcionalidad que TDR_LAB v2.0, pero pensada para ordenadores de prestaciones reducidas. Una diferencia a resaltar es la imposibilidad de calcular la humedad por inversión numérica de las ondas TDR. A diferencia de la versión completa, con la nueva capa de datos implementada se soluciona el problema de rendimiento.

Antes de comenzar a describir como se ha implementado esta solución es necesario describir la organización modular que implementa la aplicación. La forma en la que se organiza la aplicación tiene su base en el modelo de capas separadas. Estas capas únicamente tienen la funcionalidad dentro de su nivel de aplicación.

La Figura 13 muestra el esquema de las distintas capas que intervienen en este proyecto.

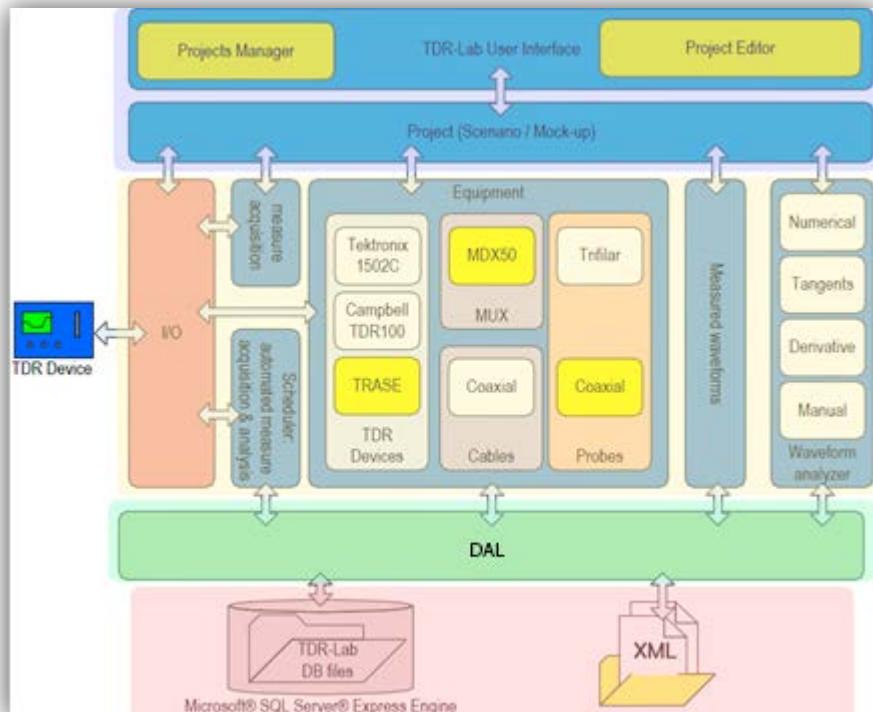


Figura 13. Estructura en capas de los componentes TDR-LAB.

Esquema de componentes que se utilizan para esta nueva versión de la aplicación.

En la imagen superior se puede apreciar cuatro zonas diferenciadas. De arriba hacia abajo tenemos una zona azul que se correspondería a los componentes de carácter más gráfico que queremos que sean los que muestren la información.

Luego nos encontramos con una zona ocre que hace relación a todos los dispositivos hardware que se necesitan para el funcionamiento de la tecnología TDR, además de las técnicas científicas que se aplican en el software.

Para finalizar, tenemos primero una zona verde que hace relación a la capa "DAL" o comúnmente llamada "capa de acceso a datos". Esta capa es la encargada de preparar las consultas para la obtención de la información así como de almacenar la información.

Abajo del todo vemos una zona en rojo que es la que propiamente almacena la información. A continuación vamos a hablar de la capa de datos, que ha sido nuestro objetivo de optimización mejorar el rendimiento de la aplicación.

La nueva programación de la base de datos se ha configurado en torno a una configuración de ficheros XML. Para decidir sobre la configuración correcta de ficheros se realizó un estudio sobre cuál sería la relación perfecta de tablas de la base de datos en cada fichero.

A partir de los resultados obtenidos en la Tabla 4 (Anexo G), observamos que la opción más recomendable es la de crear un único fichero XML para cada tabla. Esta técnica es eficiente ya que, para momentos puntuales de la aplicación en la que sólo debemos buscar una mínima información, no debemos mover todos los registros, si no solo la parte correspondiente a la tabla.

Los resultados obtenidos, tal y como se muestran en las Tablas 5, 6 y 7 (Anexo G), muestran que, aunque en términos generales, los resultados son peores que en los casos en los cuales el servicio de SQL ya está activo, la mejora sustancial se produce en la primera iteración. Conseguimos en el mejor de los casos pasar de 120 segundos a 5 segundos cuando la base de datos tiene bastante cantidad de documentos y en torno al segundo y medio cuando la base de datos está establecida por defecto.

Para poder realizar este experimento, una vez diseñada todas las estructuras necesarias en la cadena de capas, se procedió a habilitar un pequeño script que en una iteración previa generaba los ficheros XML necesarios para su utilización posterior en las pruebas. Una vez que disponíamos de los ficheros ya se podría trabajar de forma convencional con la prueba.

La Figura 14 muestra la representación gráfica de los rendimientos iniciales en comparación a los rendimientos desarrollados con la nueva alternativa. El eje vertical está en escala logarítmica para poder apreciar realmente el cambio de prestaciones entre las versiones.

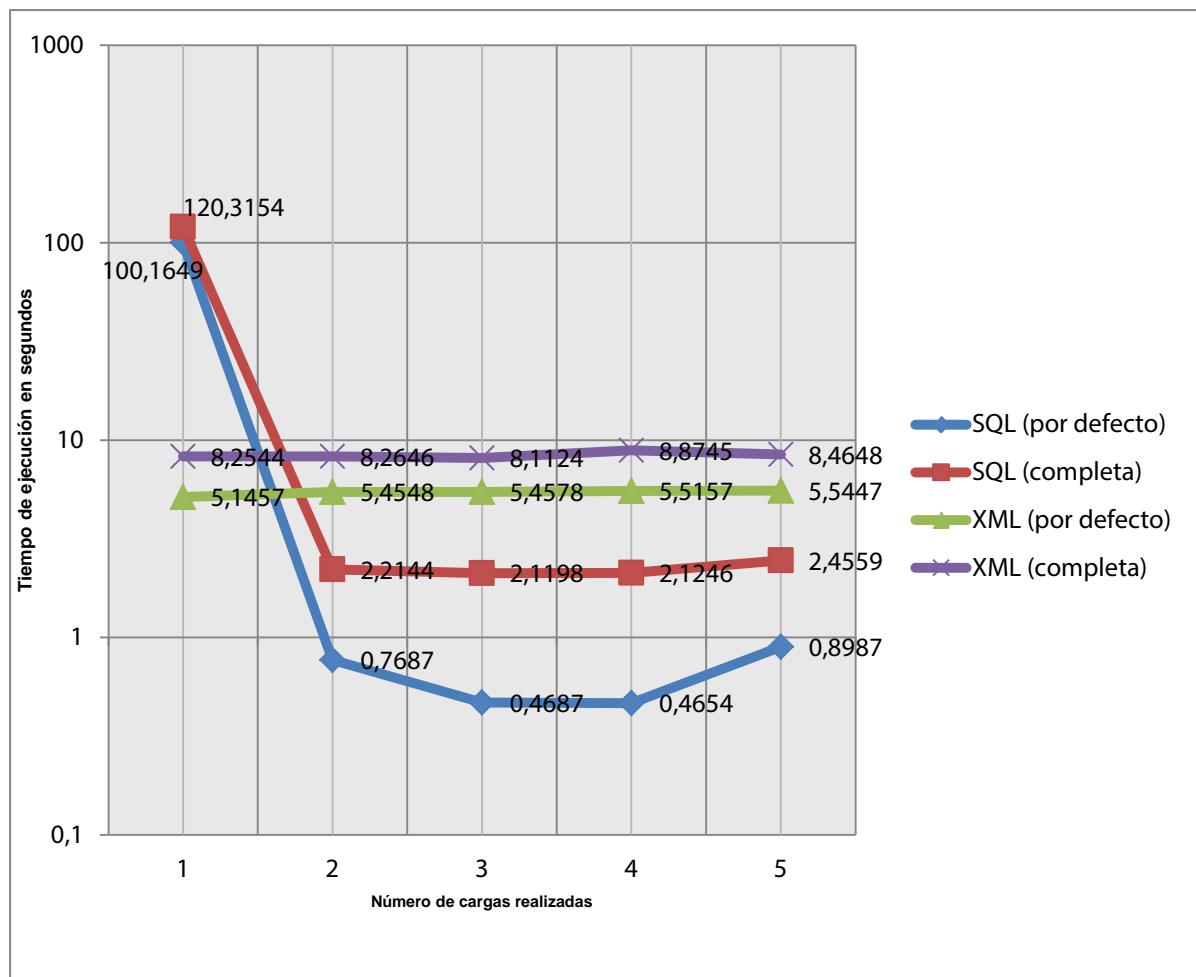


Figura 14. Gráfica de rendimiento de la alternativa frente al formato original.

Comparativa de tiempo de respuesta de la alternativa desarrollada frente al sistema inicial.

Para que las dos versiones TDR-LAB sean interoperables se han desarrollado los mecanismos de importación/exportación necesarios. Estos mecanismos permiten a los usuarios, por ejemplo, crear en laboratorio un escenario de ensayos en TDR-LAB v2.0, exportarlo a formato XML y, finalmente, cargarlo en TDR-LAB Lite. En la Figura 52 del anexo E, se puede apreciar el formulario necesario para esta implementación de modo que permite importaciones a nivel de proyecto o a nivel de proyectos contenidos en una carpeta. Para poder importar/exportar de forma correcta, el proceso hace uso de identificadores positivos y negativos. Cuando al importar una información reconoce un identificador negativo, detecta que es un elemento que se ha creado en la versión TDR-LAB Lite.

3.2. Incorporación del dispositivo multiplexor de sondas Campbell SDMX50.

Dada la modularidad del sistema, para añadir el dispositivo multiplexor SDMX50 se han introducido modificaciones en todas las capas de diseño:

- Bases de datos

Se han añadido dos nuevas tablas. Una tabla hace referencia al modelo de multiplexor y la segunda tabla al conjunto de multiplexores disponibles, de forma que un multiplexor sólo puede pertenecer a un modelo, mientras que un modelo puede tener varios dispositivos. En la práctica se ha implementado dicho comportamiento mediante la propagación de los identificadores únicos de las tablas.

En la versión 1.2.3 existe una tabla definida para almacenar las distintas duplas sondable que se configuran en el proyecto. En la nueva versión 2.0 se añade a esa dupla, un nuevo valor que identifica (si corresponde) el multiplexor utilizado en la configuración del proyecto.

- Capa DAL

En esta capa crearemos las dos nuevas clases que nos hacen falta para poder controlar el proceso de modificación de la base de datos, tal y como se ha explicado en los puntos anteriores. Como en este punto del desarrollo ya tenemos las dos versiones de la aplicación, tenemos que crear las dos interfaces para la capa.

- Capa GUI

Además de los básicos cambios en la interfaz, debido a que ahora se permiten más configuraciones, se establece un nuevo configurador de cable sondas específico para la configuración de multiplexores.

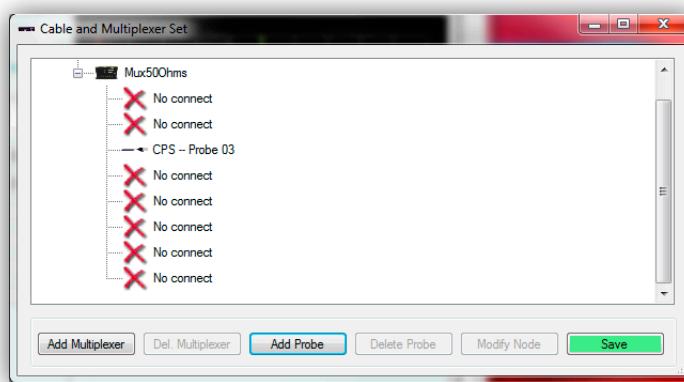


Figura 15. Formulario configuración de multiplexores.

Formulario que permite la configuración de sondas a partir de los multiplexores.

Como se aprecia en la Figura 15, el formulario se compone básicamente de dos zonas. Una zona central en donde dibuja la configuración actual que se dispone y, una zona inferior en donde se presentan todos los botones con los que el usuario puede interaccionar dependiendo del nodo que tenga seleccionado de la parte superior.

Cuando se quiere añadir información relacionada con configuraciones cable-sonda o cable-multiplexor aparece una ventana como la siguiente (Figura 16):

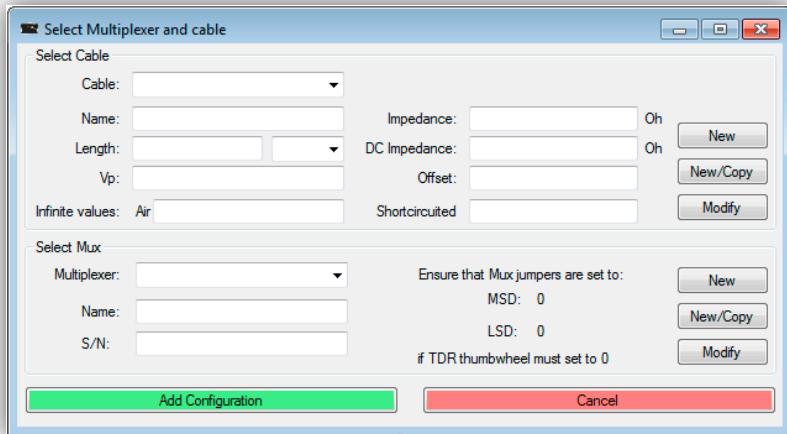


Figura 16. Detalle configuración cable-multiplexor.

Formulario que permite la configuración de particular de cables y multiplexores.

- Capa Análisis

En esta capa lo que creamos son las clases para la virtualización de los dispositivos que luego son usados tanto la manipulación de información en la GUI como del proceso de control físico de los dispositivos. Creamos la clase “Multiplexion” que da soporte virtual a toda la información situada en las capas inferiores.

3.3. Incorporación dispositivo ecómetro TRASE de Soil Moisture Equip. Corp.

Aplicando la metodología descrita en el apartado 2.3, hemos conseguido un software capaz de comunicarse con el sistema ecómetro TRASE. Dada la modularidad y la abstracción de las capas desarrolladas, el proceso de control y manipulación de este ecómetro es equivalente al de los otros dispositivos ecómetros ya implementados.

Este nivel de programación nos permite que, una vez conocido el protocolo de comunicación necesario para ese dispositivo en particular, el proceso de adición de nuevos dispositivos se produzca de una forma eficaz y eficiente. Esto es así debido a que, el tiempo empleado en analizar el protocolo de comunicación, en relación al tiempo para crear la estructura necesaria en el código, ronda el 80% para analizar y 20% para implementar.

En el desarrollo se ha tenido en cuenta la compatibilidad con la configuración de múltiples sondas posible en la arquitectura de multiplexado. En este punto, se ha abstraído el proceso de comunicación de los multiplexores, independientemente de la configuración del ecómetro, para que físicamente detecte la sonda que está en el extremo, pero no detecte el camino que hay hasta la sonda.

Gráficamente, para crear un proyecto formado por el ecómetro TRASE, los pasos necesarios serán:

- Creación de un proyecto a partir del formulario listado de carpetas y proyectos.
- En las configuraciones del proyecto, seleccionar de la lista de dispositivos disponibles, el dispositivo TRASE.

- Abrir el proyecto una vez configurado.
- Configurar la estructura de sondas.
- Obtener la onda y/o ejecutar el análisis de la misma.

3.4. Nuevas medidas.

Como la versión 1.2.3 ya permitía el uso de medidas de análisis para el estudio científico, el primer paso es extender la clase genérica de análisis de forma que permita analizar y realizar las nuevas medidas a implementar. Dentro del formulario en donde se recogen las sondas se llama a éstos nuevos métodos en los que analiza mediante los resultados y se propaga la información hacia las capas inferiores como se hacía hasta el momento, con la diferencia que antes las medidas iban configuradas a partir de un proyecto y ahora lo que interesa es que podamos tener distintas sondas con distintas configuraciones. Para realizar este proceso se configura la base de datos para que en la tabla de análisis se añada el identificador al que hace referencia.

A nivel de interfaz gráfica, se establece un nuevo formulario que permite configurar para cada sonda en particular que nuevas medidas se desean calcular y los parámetros variables que necesita la ecuación, como se aprecia en el siguiente dibujo (Figura 17).

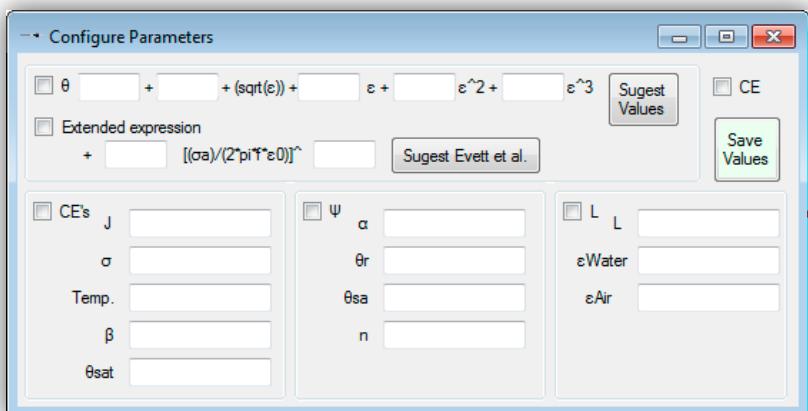


Figura 17. Detalle configuración nuevas medidas.

Formulario que permite al usuario configurar las nuevas medidas que se pueden analizar.

3.5. Implementación de la nueva interfaz gráfica.

En este apartado se resume la nueva organización de los formularios desde el punto de vista del usuario. Inicialmente se presenta un reducido “splash” que muestra la información de la versión de la aplicación entre otros logotipos. Más tarde se produce la carga del primer formulario.

Durante la aparición de esta ventana de bienvenida se produce la localización de las bases de datos, de forma que en caso de ocurrir cualquier incidencia se soluciona. En el caso de no encontrar la base de datos, se copian las bases de datos por defecto en la ubicación correspondiente.

Tras mostrar el anterior panel, se carga (de no haberlo registrado antes) el acuerdo de licencia de usuario con la licencia de uso. Finalmente, aparece el primer listado correspondiente a la carga de carpetas virtuales y los proyectos almacenados como se aprecia en la siguiente ilustración (Figura 18).

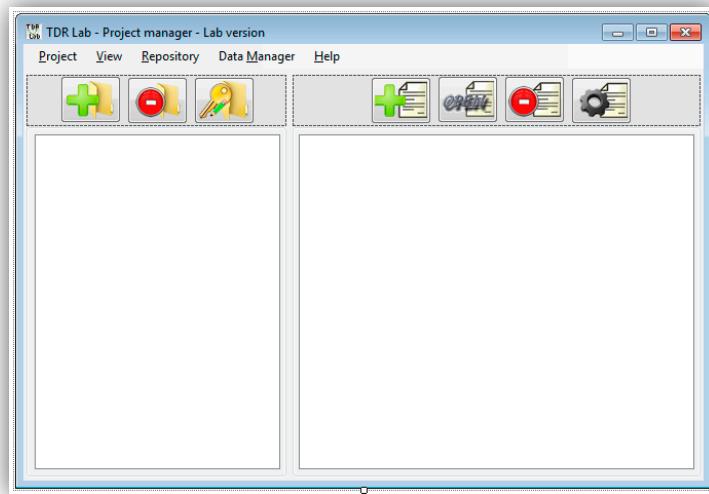


Figura 18. Formulario listado de proyectos.

Formulario que detalle al usuario configurar y crear proyectos

Dentro de la aplicación se diferencian dos tipos de formularios. El formulario de configuración y el formulario de interacción. Aunque los dos formularios en sí son interaccionables con el usuario, la clasificación hace referencia a cuestiones meramente de utilización. El criterio de utilización consiste en si la finalidad del formulario está en configurar algún apartado del proyecto o si, por el contrario, hay que validar o analizar los resultados mostrados.

En este apartado vamos a centrarnos en los formularios interactivos, ya que todos en su totalidad, comparten las mismas zonas.

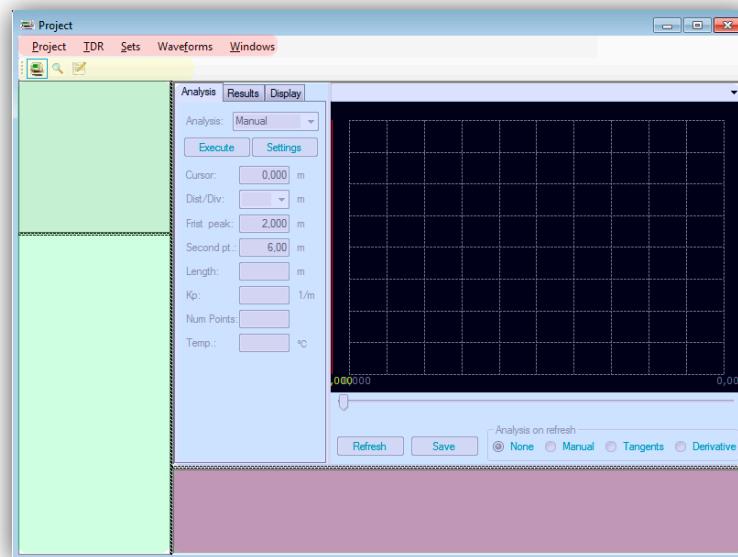


Figura 19. Estructura de la ejecución de un proyecto.

Detalle de cómo se configura un panel que permite al usuario interaccionar con el ecómetro a la vez que visualiza los resultados.

En la Figura 19 se pueden apreciar cinco zonas diferenciadas. La primera parte situada en la zona superior, sombreada en color rojo, se corresponde a un menú tradicional de gestión y mantenimiento de los componentes utilizables en el formulario.

Debajo de esta zona se presenta otra que representa algunas de las funciones del menú de arriba pero con un formato más accesible e iconográfico. La siguiente zona se divide a su vez en otras dos. Estas dos nuevas zonas son las verde situada a la izquierda y la azul situada a la derecha.

La verde se corresponde a un espacio reservado para las configuraciones que se han aplicado, ya sea el listado de proyectos disponibles o si, por el contrario, el listado de sondas utilizadas. La zona azul de la derecha se corresponde al espacio en donde se mostraran los datos o la información. La parte violeta del inferior es una zona residual en donde se visualizan otros paneles con los que el usuario pueda interaccionar.

Los formularios del formato detalle configuración contienen dos zonas básicas (Figura 20). Una primera zona resaltada en color verde que muestra los detalle para poder configurar los nuevos objetos mientras que, la segunda zona, resalta los botones de aceptar/cancelar necesarios en color azul.

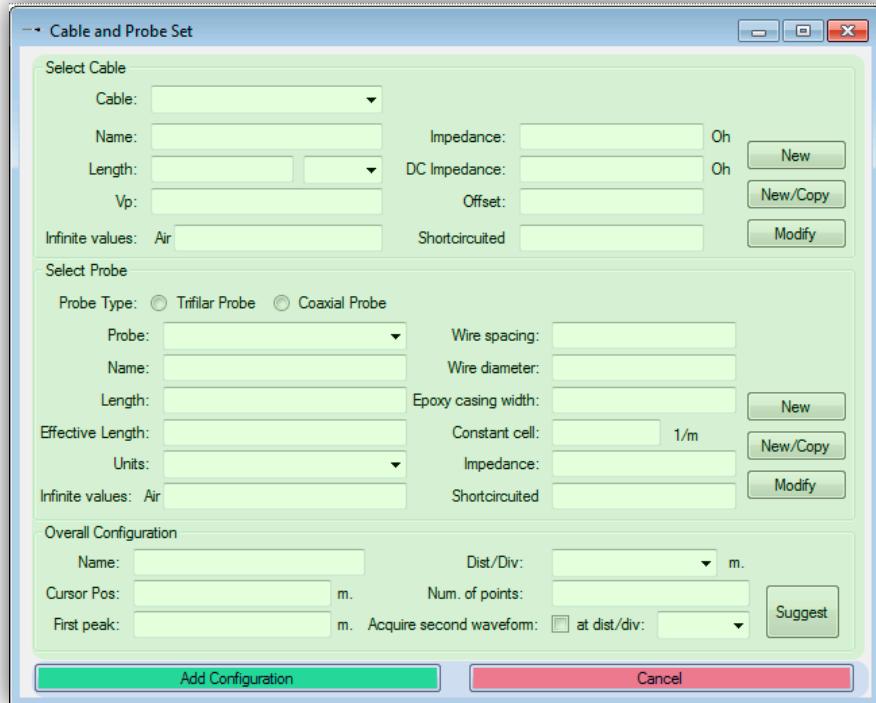


Figura 20. Estructura formulario de la aplicación.

Capítulo 4

4. Discusión

En este último capítulo vamos a comentar las conclusiones que, una vez terminado el proyecto, se pueden sacar de su realización.

Como conclusión personal, la realización de este proyecto ha supuesto un reto personal y profesional al tener que abordar todas las fases del desarrollo. He podido poner en práctica parte de los conocimientos que durante los años de carrera he aprendido.

También desde estas líneas agradecer el apoyo, tanto técnico como personal, que me han prestado personas que durante los meses de realización del proyecto me han estado ayudando.

4.1. Resultados obtenidos.

Como resultado se presenta una tecnología totalmente autosuficiente capaz de gestionar y controlar los diferentes retos planteados en la memoria. Se han desarrollado dos versiones. Una versión completa con base de datos orientada a ordenadores de prestaciones más elevadas y otra versión, más reducida, optimizada para ordenadores de bajas prestaciones. La intención por parte de los responsables del proyecto es que cualquier usuario con autorización del CSIC pueda descargarse la aplicación desde la página web de forma que a modo colaborativo puedan sugerir mejoras o la posibilidad de incorporar nuevos dispositivos, etc.

Además de las aplicaciones, un resultado que ha surgido de la realización del proyecto ha sido la redacción y presentación de la comunicación científica en el International Conference on Monitoring and Modeling Soil-Plant-Atmosphere Processes, Nápoles-2013 (Fatás et al., 2013) en formato poster. En los anexos se puede leer el *Abstrac*, el documento desarrollado y el fichero pdf que sirvió de presentación. Paralelamente, se tiene previsto preparar un artículo científico para ser enviado para su revisión y posterior publicación a la revista *Computers and Electronics in Agriculture*.

4.2. Desarrollo futuro.

El proyecto no está cerrado. Las posibilidades que ofrece este software son bastantes amplias. El siguiente desarrollo podría consistir en la sustitución del ordenador portátil en los experimentos de campo (equipado con TDR-LAB Lite) por una tarjeta industrial de prestaciones y consumo reducido, sin disco duro, de forma que se pudiera conectar a baterías o a alimentación solar y pudiera recoger muestras durante largos períodos de tiempo de forma autónoma y muy barata. Si se permite el uso de tecnología 3G o similar, se podría incluso controlar la placa a distancia y así los investigadores no tendrían ni que desplazarse hasta el campo de pruebas.

Capítulo 5

5. Bibliografía

Evett, S. R., 2000. The TACQ computer program for automatic time domain reflectometry measurements: Waveform interpretation methods. *Transactions ASAE* 43: 1947-1956.

Greco, R., Feb. 2006. Soil water content inverse profiling from single TDR waveforms. *J. Hydrol.* 317: 325-339.

Fatás, E., Latorre, B., Lera, F. Viñals, V., López, M.V., Blanco, N., Peña, C., González-Cebollada, C., Moret-Fernández, D. 2013. TDR-LAB 2.0 Improved TDR Software for soil water content and electrical conductivity measurements. International Conference on Monitoring and Modeling Soil-Plant-Atmosphere Processes. *Procedia Environmental Sciences*. DOI: 10.1016/j.proenv.2013.06.054

Heimovaara, T. J., Huisman, J. A., Vrugt, J. A., Bouten, W., Nov. 2004. Obtaining the spatial distribution of water content along a TDR probe using the scem-ua bayesian inverse modeling scheme. *Vadose Zone J.* 3: 1128-1145.

Herkelrath, W. N., Hamburg, S. P., Murphy, F., 1991. Automatic, real-time monitoring of soil moisture in a remote field area with time domain reflectometry. *Water Resour. Res.* 27: 857-864.

Lin, C.-P., Chung, C.-C., Huisman, J. A., Tang, S.-H., 2008. Clarification and calibration of reflection coefficient for electrical conductivity measurement by time domain reflectometry. *Soil Sci. Soc. Am. J.* 72: 1033-1040.

Moret, D., López, M. V., Arrué, J. L., 2004. TDR application for automated water level measurement from mariotte reservoirs in tension disc infiltrometers. *J. Hydrol.* 297: 229 -235.

Moret-Fernández, D., Vicente, J., Aragüés, R., Peña, C., López, M.V. 2012. A new TDR probe for measurements of soil solution electrical conductivity. *Journal of Hydrology* 448-449, 73-79.

Moret-Fernández, D., Vicente, J., Lera, F., Latorre, B., López, M.V., Blanco, N., González-Cebollada, C., Arrué, J.L., Gracia, R., Salvador, M.J., Bielsa, A. 2010. TDR-Lab Version 1.0 Users Guide (<http://digital.csic.es/handle/10261/35790>).

Mualem, Y., and S.P. Friedman. 1991. Theoretical prediction of electrical conductivity in saturated and unsaturated soil. *Water Resou. Res.* 27, 2771-2777.

Or, D., Wraith, J. M., 1999. A new soil metric potential sensor based on time domain reflectometry. *Water Resour. Res.* 35: 3399-3407.

Oswald, B., Benedickter, H. R., Bächtold, W., Flühler, H., Dec. 2003. Spatially resolved water content profiles from inverted time domain reflectometry signals. *Water Resour. Res.* DOI: 10.1029/2002WR001890

Timlin, D. J., Pachepsky, Y. A., 1996. Comparison of three methods to obtain the apparent dielectric constant from time domain reflectometry wave traces. URL <https://www.agronomy.org/publications/ssaj/abstracts/60/4/970>

Topp, G. C., Davis, J. L., Annan, A. P., 1980. Electromagnetic determination of soil water content: Measurements in coaxial transmission lines. *Water Resour. Res.* 16: 574-582.

Topp, G. C., Yanuka, M., Zebchuk, W. D., Zegelin, S., 1988. Determination of electrical conductivity using time domain reflectometry: Soil and water experiments in coaxial lines. *Water Resour. Res.* 24: 945-952.

US Salinity Laboratory Staff. 1954. Diagnosis and improvement of saline and alkali soils. USDA Handbook 60, U.S. Government Printing Office, Washington, D. C.

van Genuchten, M.T., 1980. A closed-form equation for predicting the hydraulic properties of unsaturated soils. *Soil Sci. Soc. Am. J.* 44, 892-898.

PARTE II ANEXOS

Anexo A

A. Entorno de trabajo.

En esta sección hablaremos sobre los aspectos que hacen relación con la configuración del puesto físico de trabajo, así como las herramientas empleadas para su desarrollo, terminando con los temas relacionados con el método de trabajo.

En primer lugar, debido a que el proyecto se compagina con el último año de ingeniería informática, se establece el puesto de trabajo en la propia EINA, en el edificio Ada Byron, concretamente en el laboratorio 2.08.

Como herramientas de trabajo se ha utilizado un ordenador del mismo laboratorio en el cual se instala inicialmente el entorno de programación Visual Studio 2010. Posteriormente, conforme se van avanzando etapas en el desarrollo de la aplicación, se van instalando aplicaciones como el motor de bases de datos SQL-Server o el framework 4.0.

A.1 Visual Studio 2010.

En este punto explicaremos por qué se ha elegido este entorno de programación. Como ya se ha adelantado en la introducción y en parte del resumen, la elección de este entorno viene determinada en parte porque había sido el target de la versión anterior. Además ofrece otros beneficios, como la sencillez con la que trabaja con los archivos en formato XML.

Hay que destacar y comentar que, para conseguir que Visual Studio 2010 en conjunto con C# sea nuestro target, se necesita que el usuario tenga instalado en su ordenador el framework 4.0 para entornos C#. Este framework es el encargado de dar soporte a todas las funciones, botones, componentes en general, que se pueden usar con Visual Studio. Dicho Framework no presenta un inconveniente extra al usuario ya que, en el propio instalable que presenta la aplicación, está marcado como requisito de instalación. Es decir, si la primera vez que instalamos el programa no disponemos del entorno instalado, no debemos preocuparnos en buscarlo e instalarlo.

Una gran utilidad es el desarrollado “Common Languaje Runtime” (CLR), con el que se desempeñan funciones de gran utilidad tanto en el proceso de desarrollo como en la ejecución. Cuando el componente se está ejecutando, el CLR es responsable de administrar la asignación de memoria, iniciar y detener subprocesos y procesos, y hacer cumplir la directiva de seguridad, así como satisfacer las posibles dependencias del componente sobre otros componentes.

A.2 SQL Server 2008 y archivos XML.

Una parte importante, aunque no la fundamental de la aplicación, es el uso que se realiza de la gestión de bases de datos. Por esa razón el motor de bases de datos, que en su primera versión suponía un cuello de botella en el rendimiento, ha necesitado de una revisión.

Inicialmente, la versión contenía un motor de SQL-SERVER gratuito que se podía marcar como requisito en el instalable de la aplicación y que el propio Visual Studio incorporaba de base.

Algunas de las razones por las que se ha escogido usar el motor de base de datos de SQL-SERVER:

- Herramienta ya incorporada e integrada con Visual Studio.
- Posibilidad de importar/exportar/gestionar bases de datos con la aplicación "SQL-Management Studio".
- Realizar transacciones tanto a nivel de bases de datos como a nivel de aplicación, de forma que si algún método provoca una interrupción no controlada, rehace la base de datos para configurarla en un estado estable.
- Soporta procedimientos almacenados. Podemos configurar métodos de inserción en base de datos de forma diferente a los convencionales.
- Permite al programador establecer un modo cliente-servidor, de forma que todos los datos se comparten a partir de un único y centralizado sistema.

También se podrá almacenar la información en unos ficheros correctamente formados con etiquetas. Esos ficheros son los llamados XML. En la siguiente imagen podemos ver un esquema de cómo es un fichero XML (Figura 21).

```
<?xml version="1.0" standalone="yes"?>
<TDRDB xmlns="http://tempuri.org/TDRDB.xsd">
  <Project>
    <Id>124</Id>
    <Name>TDR Measures</Name>
    <CreationDate>18/06/2012</CreationDate>
    <LastModifiedDate>19/06/2012</LastModifiedDate>
    <Notes>Proyecto de medidas usadas en campo</Notes>
    <FolderId>514</FolderId>
  </Project>
</TDRDB>
```

Figura 21. Formato fichero XML.

Existe una serie de etiquetas que sirven para abrir o cerrar nodos, de forma que con cada nodo podemos guardar la información relativa a cada registro. XML se diseña con el fin de servir de estándar de comunicación entre plataformas, de forma que se pueda almacenar información de una forma estructurada para su posterior manipulación.

El siguiente paso consiste en implementar que la capa que extrae los datos de la base de datos, aprenda también a extraer la información desde los ficheros XML. De ahí surge la idea de desarrollar dos versiones de la misma aplicación. Una aplicación con la capa de datos en SQL para ordenadores con más capacidad de cálculo llamada directamente TDR-LAB 2.0 y una segunda versión optimizada para ordenadores de menos potencia que trabaja con el sistema de archivos en XML. De esta forma el usuario puede decidir que aplicación instalar y controlar el rendimiento.

Un requisito por parte de los investigadores del CSIC es que se pueda utilizar en los ordenadores de bajas prestaciones usados en el campo, por lo que surge la necesidad de esta nueva versión optimizada que permite reducir el tiempo de arranque (i.e. activar el

servicio de SQL puede costar entre 5 y 10'). Al introducir este requisito se llega al acuerdo de que, como mínimo tiene que haber algún ordenador con el motor de base de datos SQL, de forma que puedan sincronizar los datos almacenados en la versión de campo.

A.3 Repositorio proyecto y metodología de trabajo.

Durante el tiempo de realización del proyecto, la curva de aprendizaje ha sido lenta, pero se han alcanzado los objetivos propuestos. Unos de los motivos que han influido en la curva ha sido que la aplicación existente estaba escrita en un lenguaje de programación que era desconocido para mí. Aunque no se diferencia mucho de otros lenguajes orientados a objetos, al principio es necesario tiempo para acostumbrarse a la forma de llamar a las funciones, de declarar variables o incluso de configurar el proyecto para su desarrollo.

El segundo problema ha sido que la aplicación no partía desde cero, si no que ya había una versión en producción y, por lo tanto, había que entender que era lo que el anterior programador había desarrollado, ya que como suele pasar como norma, en principio los códigos nunca están lo suficientemente comentados y preparados para que otro programador pueda continuarlos. En este punto también he dedicado esfuerzo en comentar y justificar los códigos fuente de la versión 1.2.3. antes de comenzar la programación de la nueva versión.

Pasando a hablar ahora de la metodología, al inicio del proyecto se dedicó un tiempo a entender y asimilar el código y las explicaciones del programador anterior. Con el tiempo esta metodología cambió, ya que al empezar a desarrollar versiones de la aplicación necesitaba disponer de reuniones donde validar los resultados. En esta última parte del proyecto es cuando más reuniones se han producido con los investigadores del CSIC, ya que como se aprecia en los anexos, este trabajo ha sido presentado en el congreso celebrado en la ciudad de Nápoles (Italia) en las fechas del 19 al 21 de Junio de 2013 con el título "FOUR DECADES OF PROGRESS IN MONITORING AND MODELING OF PROCESSES IN THE SOIL-PLANT-ATMOSPHERE SYSTEM: APPLICATIONS AND CHALLENGES"

Para finalizar esta sección queda hablar únicamente del sistema de seguridad que he estado usando para guardar la información y poder recuperarla en cualquier momento. El formato elegido ha sido el de utilizar un repositorio con una política de revisiones con base en una cuenta del servidor danae-cifs, creada por el administrador de sistemas del departamento. De forma que yo, personalmente, podía llevar la relación de las versiones que iba generando y a la par se creaba una copia de seguridad que se guardaba en los servidores del departamento. Destacar que no se ha tenido que usar esta alternativa como método de recuperación de información.

Anexo B

B. Diseño estructural de la aplicación.

En esta sección abordaremos todo lo relacionado con el diseño de la aplicación. Haremos un repaso desde las entidades más simples hasta los diagramas complejos que explican el flujo de información en la aplicación. El propósito de esta sección es dar a comprender la filosofía de diseño que se ha empleado en la realización del proyecto.

Desde los inicios, tras analizar las necesidades variables de los científicos del CSIC, se diseñó una aplicación modular que acoplara perfectamente todas las capas de forma que en caso de añadir nuevas funcionalidades, resultara el cambio de forma más eficaz y natural.

B.1 Organización capas de información.

La forma en la que se organiza la aplicación tiene su base en el modelo de capas separadas. Estas capas únicamente tienen la funcionalidad dentro de su nivel de aplicación de forma que la capa que se encargue de la base de datos, las únicas funciones serán la de almacenar la información y recuperarla en base a unas consultas que se generan en otro nivel.

Esas consultas son las que dicen qué datos se van a recuperar, pero no hace relación al proceso de obtención. De la misma manera, tiene sentido que exista una serie de clases o métodos que estén almacenados en otro nivel de forma que la única función sea la de mostrar la información por pantalla de forma correcta. La analogía es idéntica para el proceso de análisis de la información. A la aplicación en esa capa no le interesa saber cómo se muestran los datos o como se almacenan.

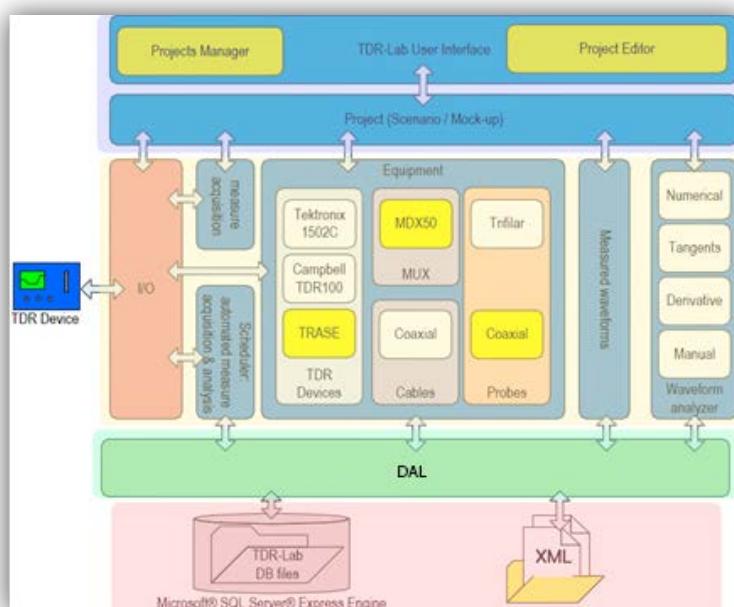


Figura 22. Diseño de capas de la información.

Organización de los componentes que interaccionan en la versión TDR-Lab V2.0

Inicialmente, partiendo de un esquema de los componentes que era preciso que dispusiera la nueva versión de la aplicación, se diseña el sistema. En la Figura 22 se aprecian cuatro zonas diferentes. De arriba hacia abajo tenemos una zona azul que se correspondería a los componentes de carácter más gráfico que queremos que sean los que muestren la información. Luego nos encontramos con una zona amarilla que hace relación a todos los dispositivos hardware que se necesitan para el funcionamiento de la tecnología TDR, además de las técnicas científicas que se aplican en el software. Para finalizar tenemos primero una zona verde que hace relación a la capa "DAL" o comúnmente llamada "Capa de acceso a datos". Esta capa es la encargada de preparar las consultas para la obtención de la información así como de almacenar la información. Abajo del todo vemos una zona en rojo que es la que propiamente almacena la información. Hablamos de la capa de datos y que, como ya se ha mencionado, es la que se ha optimizado para mejorar el rendimiento de la aplicación.

Para conseguir que esta organización teórica tenga sentido a la hora de programación en Visual Studio 2010, se realiza mediante la configuración de proyectos y soluciones. Se establece la solución como conjunto de todos los proyectos involucrados de forma que cuando se compila y se genera la aplicación reléase, todos los archivos necesarios para la ejecución de la aplicación se encuentran en la misma ubicación.

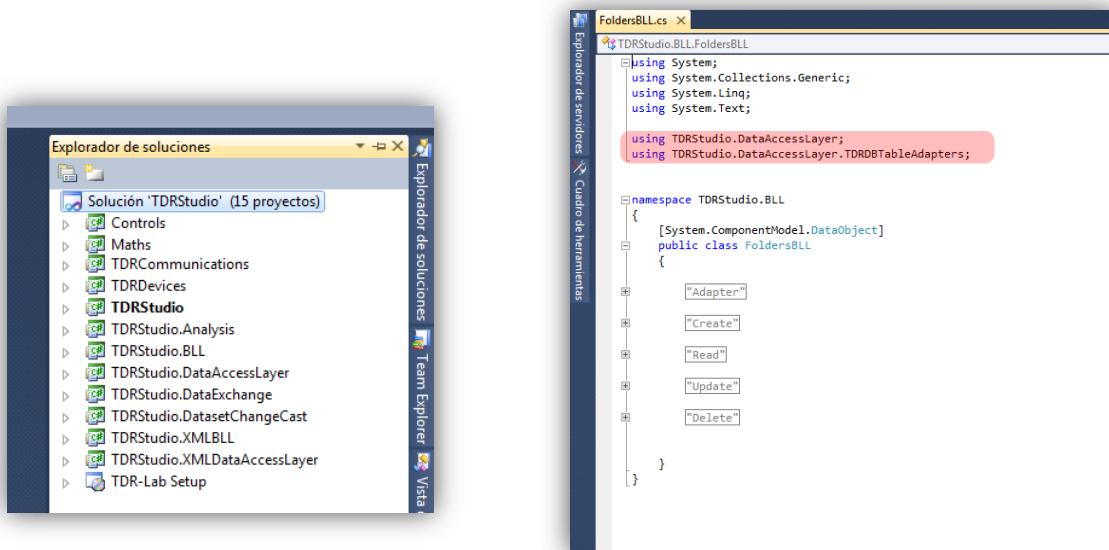


Figura 23. Detalle configuración proyectos en Visual Studio 2010.

Relación de configuración de los diferentes proyectos en Visual Studio 2010 y el proceso para añadir las dependencias en el código.

Lo que hace que se separen las capas es el espacio de nombres que se le asigna a cada proyecto, de forma que lo único que debemos hacer en el código es instanciar al espacio de nombres que queramos y así tenemos acceso a todo el contenido de forma encapsulada.

B.2 Capa “DataAccesLayer”.

Esta capa como su nombre indica hace referencia a la capa correspondiente de la base de datos. Es el nivel más bajo en la pirámide de capas y, como se ha mencionado ya en capitulo anteriores y como se desarrollará en el punto cuarto, presenta dos alternativas. En SQL-Server y archivos XML. Esta sería la descripción a nivel de implementación pero antes de adelantar temas, vamos a proceder a describir analíticamente las funciones que debe aportar esta capa al conjunto de la aplicación.

Las características y funciones deseadas en este nivel son las siguientes:

- Permita almacenar y recuperar la información almacenada previamente en la base de datos.
- Permita mantener un orden lógico de los elementos.
- Ofrezca unas garantías de que la base de datos sea robusta, fiable y persistente ante errores.
- Debe ofrecer una facilidad de control y manipulación por parte del desarrollador del proyecto o por la continuación del desarrollo de la aplicación.

A partir de los requisitos de la aplicación, el usuario será capaz de poder almacenar información sobre los distintos dispositivos TDR disponibles, así como información relativa al tipo de dispositivo. Además, almacenará los tipos y configuraciones de los cables y las sondas que se necesitan para poder tomar mediciones. Para ofrecer un sistema de carpetas y proyectos como la jerarquía de carpetas de Windows se almacenará información carpetas virtuales y configuración de proyectos.

Para completar la información se deberá añadir las entidades de análisis y resultados de los mismos, así como los distintos tipos de análisis soportados en la versión actual. Como mejoras realizadas dentro de este proyecto final de carrera se incorporará información sobre los distintos modelos de multiplexión, configuraciones y formas de conexión disponibles.

En la siguiente imagen (Figura 24) se puede apreciar un esquema entidad/relación minimizado del resultado de análisis de los puntos descrito anteriormente.

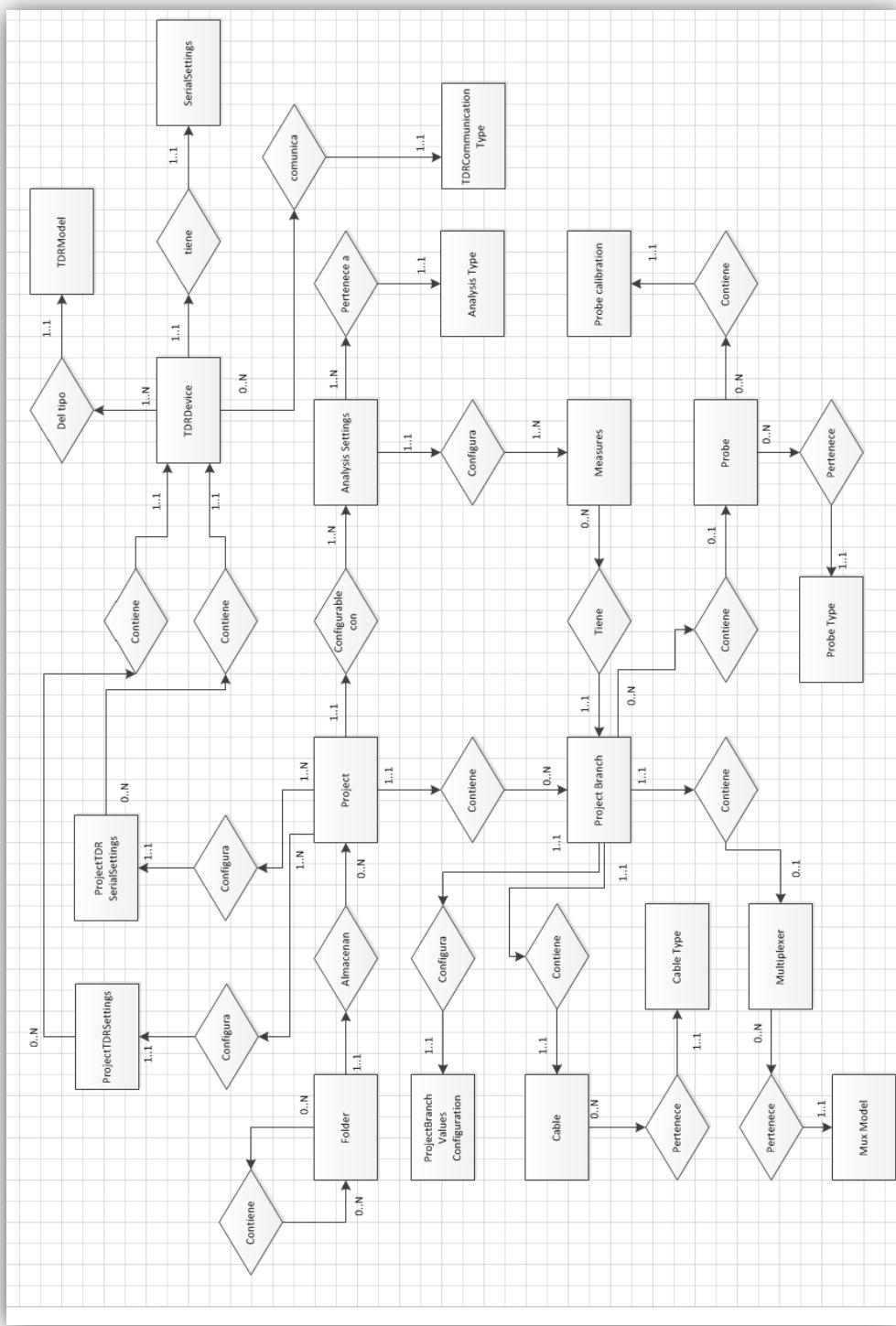


Figura 24. Esquema base de datos relacional.

Detalle del esquema empleado para la confección de la nueva versión de la aplicación, mejorando los conceptos de la versión anterior.

Una vez completado en un primer análisis de la información necesaria, procedemos a traducir dicho esquema a otros modelos de los cuales podamos obtener realmente el esquema de tablas que usaremos en la base de datos (Figura 25).

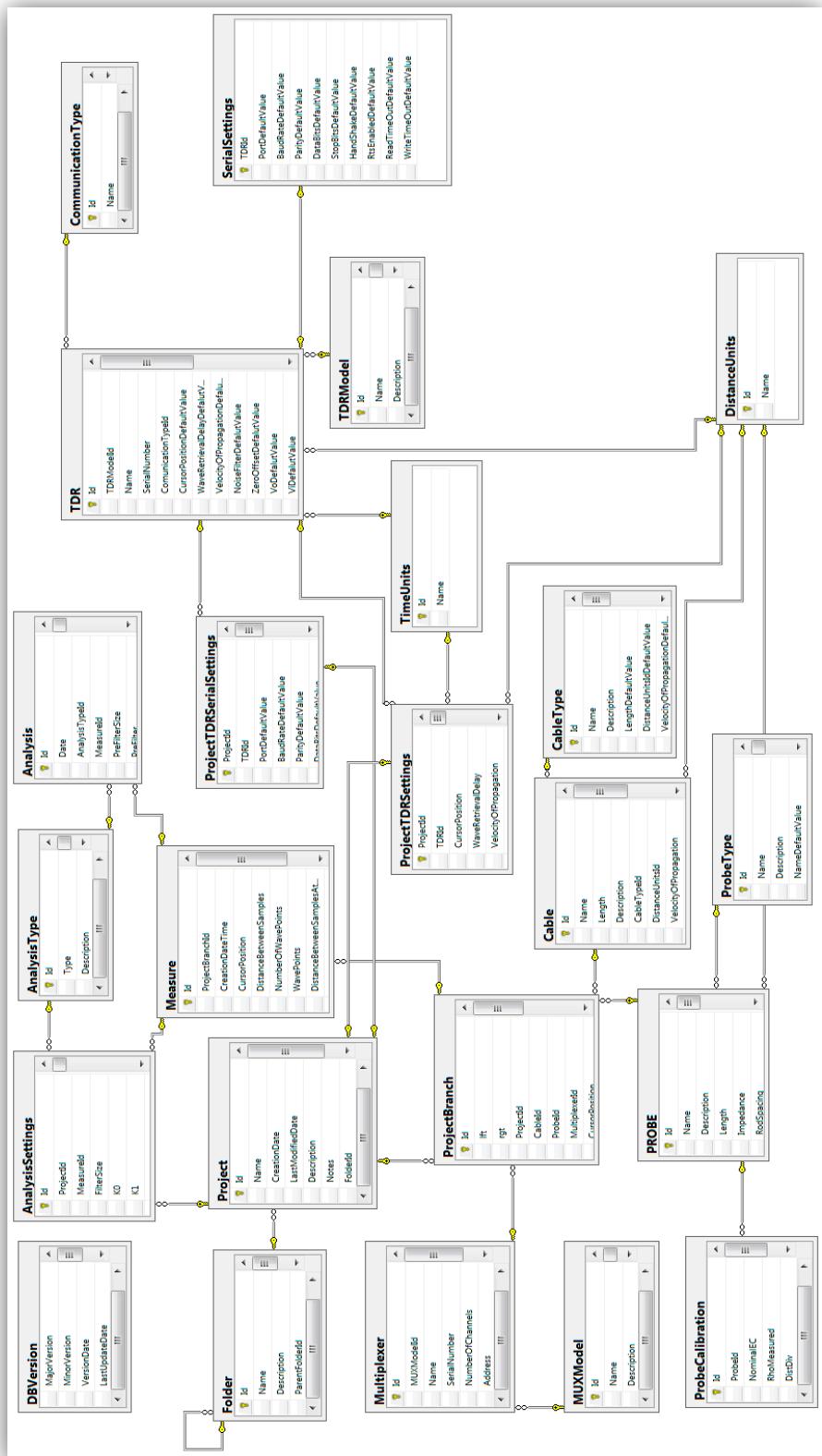


Figura 25. Implementación base de datos.

Esquema de cómo se implementa en Visual Studio 2010 el diagrama de bases de datos de la Figura 24.

Podemos observar, en primer lugar, el editor de bases de datos que aporta Visual Studio con la información necesaria para dar soporte a los distintos tipos de configuración de cada proyecto (Figura 26).

Nombre de columna	Tipo de datos	Permitir val...
Id	int	<input type="checkbox"/>
Name	nvarchar(255)	<input type="checkbox"/>
CreationDate	datetime	<input checked="" type="checkbox"/>
LastModifiedDate	datetime	<input checked="" type="checkbox"/>
Description	nvarchar(512)	<input checked="" type="checkbox"/>
Notes	text	<input checked="" type="checkbox"/>
FolderId	int	<input type="checkbox"/>

Figura 26. Composición de tablas en Visual Studio 2010.

Permite configurar de una forma gráfica el contenido de las tablas de la base de datos.

En la siguiente composición vemos tanto el proyecto de base de datos dentro del explorador de soluciones como el listado organizado de la base de datos con todas las tablas creadas. En este punto, además, configuramos para que la base de datos de nuestra aplicación esté situada en una ubicación preestablecida, de forma que si se detecta que no existe o no se corresponde con la versión correspondiente procede a copiar una base de datos vacía con los valores por defecto o a actualizar las tablas con los comandos necesarios (Alter table...etc.)(Figura 27).

Figura 27. Relación de ficheros de base de datos.

Esquema de cómo se ha configurado el fichero de creación de la base de datos y a la izquierda las tablas que se han creado.

Para terminar esta parte del capítulo queda hablar de unas herramientas propias del framework y Visual Studio para el manejo de bases de datos. Una de las ventajas de usar esta combinación de entorno de programación y lenguaje objetivo reside en las clases DataSet, DataTable y DataRow. Cuando se usan estas tres clase de forma conjunta con un modelo de bases de datos descrito antes, se consigue que de forma automática el entorno de programación cree un mapeo de los atributos de la base de datos como clases ya definidas. Es decir, si nosotros a partir del ejemplo anterior tenemos la clase DataSet definida como TDRDB.xsd, tenemos todos los atributos directamente establecidos, ya que no es necesario tener que duplicar configuraciones manualmente. La información se propaga de forma inmediata.

Además crea en el caso de la tabla "Project" la clase "ProjectDataTable" y "ProjectDataRow" que son especializaciones de las genéricas DataTable y Dataset. Estas clases proporcionan un nivel de abstracción de forma que el programador no debe modificar la clase cada vez que añada campos en la base de datos, ya que estos datos se propagan de forma automática.

Otra utilidad, entre las muchas que ofrece, es la posibilidad de incorporar un método que se encarga de escribir la información de la clase en un fichero XML. Esta herramienta será de gran utilidad a la hora de implementar la nueva capa de datos.

B.3 Capa "BusinessLogicalLayer".

En esta sección analizaremos el papel de la capa que está justo un nivel por encima de la capa de base de datos, de forma que nos sirve de puente lo que tenemos almacenado y lo que mostramos por pantalla. Es la capa encargada de "transformar" la información al formato correcto para ser guardada de la forma correcta.

Esta capa se divide en dos implementaciones. Tenemos una implementación o interfaz para trabajar con la base de datos de SQL, realizando las consultas y recuperando información, mientras que existe otra interfaz para la versión de la capa de datos que funciona con el sistema de archivos XML.

Todas las clases, independientemente de la interfaz, tienen una estructura común a todas ellas. En primer lugar, hay una sección donde se añaden las referencias que esa clase va a necesitar para poder funcionar correctamente. Luego, en la parte siguiente, nos encontramos con las referencias al desarrollo de la base de datos con la que va a trabajar la clase. Puede hacer referencia al proyecto de DataAccesLayer o al proyecto XMLDataAccesLayer

En este punto es donde realmente comienza el contenido de la clase (Figura 28). Está organizado en regiones de forma que la comprensión del código resulta más asequible. Tenemos cinco regiones (Adapter, Create, Read, Update, Delete).

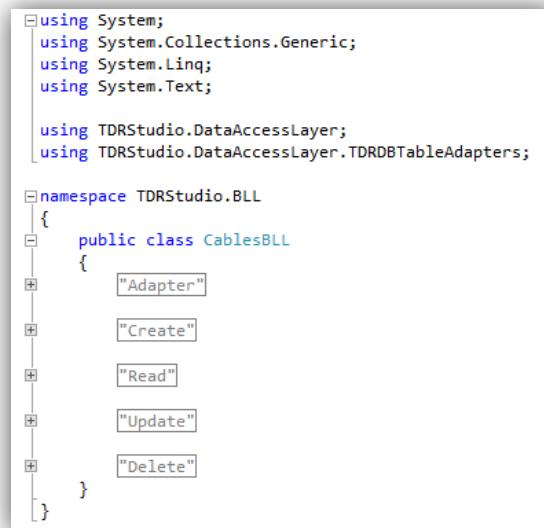
La región Adapter mapea la clase DataTable que hemos descrito en la sección anterior para enlazar con el sistema de bases de datos.

Create lo que hace es añadir un nuevo registro a la base de datos, haciendo los controles necesarios para que los datos sean coherentes y consistentes.

Dentro de la región Read, se incluyen todos los métodos de obtención de datos, pasando desde los más genéricos de obtención de toda la información hasta los más complejos con búsquedas filtradas.

La región Update es la encargada de modificar la información que ya contiene la base de datos por lo que realizará las comprobaciones necesarias.

Por último está la región Delete, que se encargará de borrar el contenido de la base de datos. Esta función dependiendo de cómo hayamos configurado la base de datos procederá a realizar borrados en cascada si fuera necesario.



```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

using TDRStudio.DataAccessLayer;
using TDRStudio.DataAccessLayer.TDRDBTableAdapters;

namespace TDRStudio.BLL
{
    public class CablesBLL
    {
        public "Adapter"
        public "Create"
        public "Read"
        public "Update"
        public "Delete"
    }
}
```

Figura 28. Estructura de una clase “DAL”.

Se muestra la estructura que compone una clase genérica de la capa de acceso a datos.

Para terminar con este punto de la sección, la siguiente ilustración muestra para el método de leer crear el adaptador de la base de datos para las dos interfaces. En primer lugar la interfaz de SQL (Figura 29) .



```
#region "Adapter"

private CableTableAdapter _cableAdapter = null;
protected CableTableAdapter Adapter
{
    get
    {
        if (_cableAdapter == null)
            _cableAdapter = new CableTableAdapter();

        return _cableAdapter;
    }
}

#endregion
```

Figura 29. “REGION ADAPTER” para SQL.

Detalle de cómo se implementa el conector con la base de datos en SQL

En esta imagen tenemos el modelo para la interfaz de XML (Figura 30). Se puede apreciar que para poder leer los datos en primer lugar tenemos que tenerlos cargados en memoria.

```
#region Atributos
private string pathToUse = Utiles.pathlocal;
private XMLDataSet _data = null;
protected XMLDataSet data
{
    get
    {
        if (_data == null)
        {
            _data = new XMLDataSet();
            _data.ReadXml(this.obtenerpath());
        }
        return _data;
    }
}
#endregion
```

Figura 30. “REGION ADAPTER” para XML.

[Detalle de cómo se implementa el conector con la base de datos en SQL](#)

B.4 Capa “GUI”.

En este nivel haremos una presentación de cómo está organizado el sistema de visualización de la aplicación de forma que quede clara la separación entre las capas.

Esta capa se llama GUI porque es el acrónimo del inglés Graphic User Interface. Como su nombre indica, la función de esta capa es la de presentar los datos que han sido obtenidos por las capas anteriores al usuario, permitiéndole interaccionar con los resultados. Mediante esta capa el usuario es capaz de añadir al sistema su propia colección de cables, sondas, multiplexores. También le permite gestionar la estructura de carpetas virtuales de la aplicación con los proyectos y sus configuraciones.

Par la realización de esta capa se ha usado la técnica de crear todas las ventanas con formularios personalizados de forma que cada formulario contiene los controles que en otro proyecto se han editado y personalizado para la aplicación. Es muy común, en el desarrollo de aplicaciones de esta índole, el uso de formulario que no hacen sino referencias a controles con funciones y métodos previamente definidos por el desarrollador.

Los controles creados para esta aplicación tienen distintos comportamientos. Desde el control complejo de visualización de resultados llamado DataDisplay (Figura 31), pasando por controles de configuración de dispositivos, todos ellos tienen un conjunto de atributos para cargar el contenido que se quiere mostrar y luego una serie de funciones para poder interaccionar con el formulario que los ha instanciado.

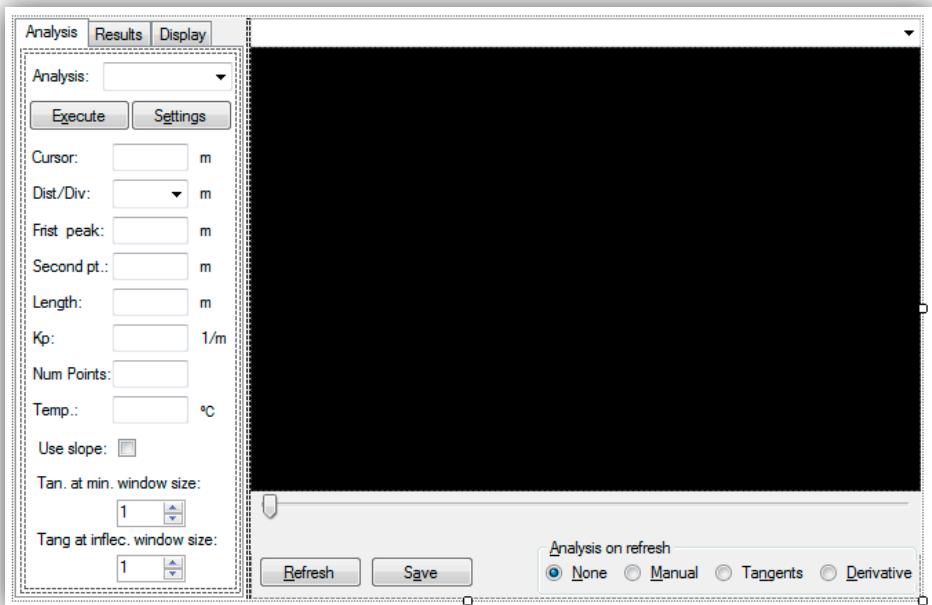


Figura 31. Detalle control “DATADISPLAY”.

Dentro del entorno de Visual Studio dicho comportamiento se ha implementado creando dos proyectos. Un proyecto que además de ser el proyecto principal, contiene todos los formularios que usa la aplicación llamado “TDRStudio” y luego existe otro proyecto con su espacio de nombres diferenciado del anterior llamado “Controls”. Este último proyecto es el contenedor de los controles usados en la aplicación (Figura 32).

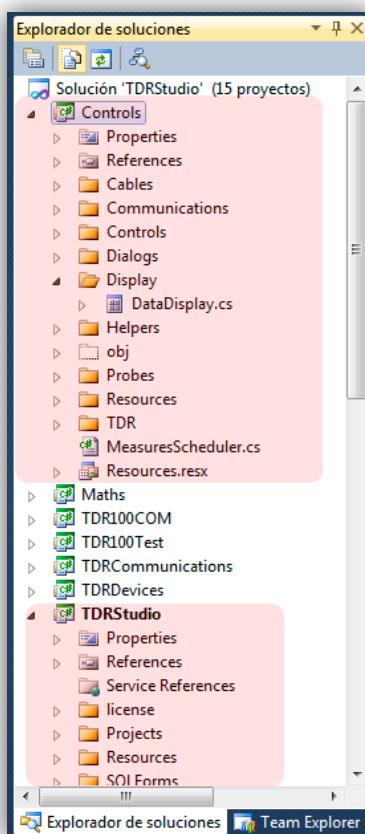


Figura 32. Explorador de solución de la capa “GUI”.

B.5 Capa “Análisis”.

En esta capa agrupamos los distintos componentes que no son ni interfaz ni tienen relación con las bases de datos. Nos referimos a la capa de “Análisis”. Dentro de esta capa nos encontramos tanto las clases necesarias para realizar los análisis necesarios para el tratamiento de la información como de las clases que virtualizan los dispositivos físicos y de los cuales guardamos la información.

Por ese motivo, este apartado lo vamos a dividir en dos partes. Una primera parte dedicada a la parte de la capa dedicada al manejo y control de los dispositivos y una segunda parte relacionada con el análisis propio de resultados.

Para la parte de control de dispositivos se ha diseñado de forma que tenemos dos proyectos en la solución encargados con esta tarea. Un proyecto llamado “TDRDevices” contiene las clases que son capaces de mantener en el momento de ejecución los atributos y métodos que habría que ejecutar sobre el dispositivo real de forma que controla toda la circuitería.

En nuestro caso, hemos dispuesto de una clase abstracta llamada “TDR-Generic” que contiene los atributos y métodos mínimos que comparten los dispositivos. Ejemplos de métodos que están definidos en esta clase son:

- Connect.
- Disconnect
- NumberOfWavePoints.
- OutputImpedance
- Etc...

Como se aprecia son funciones básicas para conectar, desconectar y enviar información hacia el dispositivo. Luego cada tipo de dispositivo mantiene una clase que deriva de la genérica donde se re-escriben los distintos métodos y atributos para adaptarlo al dispositivo concreto (Figura 33).

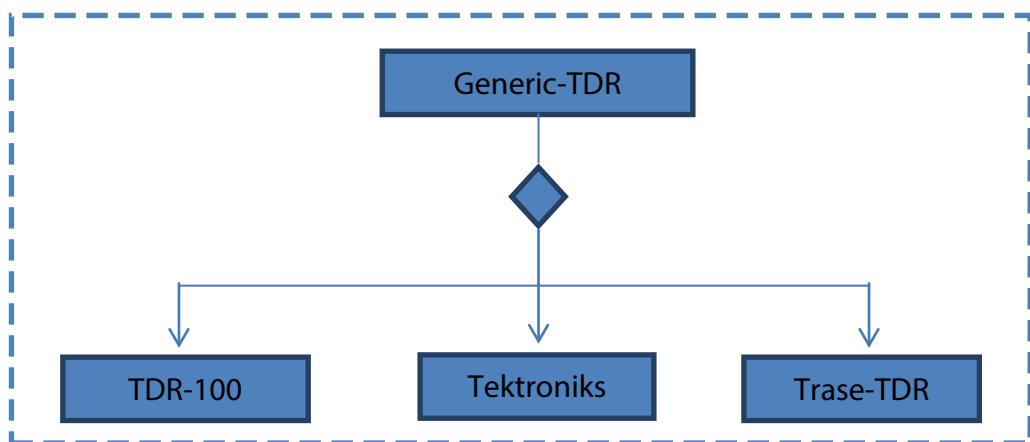


Figura 33. Estructura de las clases TDR.

Como forma de relacionar esta capa de análisis con la capa anterior de interfaz gráfica, el modelo usado para visualizar e interactuar con el dispositivo es mediante un atributo establecido en el propio formulario que es del tipo de la clase genérica de esta capa. Al ser genérico, cualquier objeto derivado de esa clase es aceptado en ejecución, por lo que se produce la interacción mínima y necesaria entre las capas.

Por otra parte, existe un segundo proyecto que se encarga de las comunicaciones. Es importante este proyecto ya que, en nuestra configuración del dispositivo, somos capaces de controlar el ecómetro TDR mediante el puerto serie y la comunicación con los multiplexores se realiza a través del puerto paralelo.

Por lo tanto, en un proyecto llamado TDRCommunications se establecen todas las clases que son necesarias para el proceso de comunicación por el cable físico (Cable serie o paralelo).

Para las conexiones mediante el puerto serie, el framework utilizado incorpora una serie de clases nativas para el manejo de estas comunicaciones, por lo que en principio sólo nos basta con extender su funcionamiento de forma mínima. Distinto es para las comunicaciones mediante el puerto paralelo. Aunque es verdad que existen librerías que son capaces de trasmitir bits por el puerto paralelo, ha sido necesario crear una clase que utiliza dichas librerías y que además ejecuta el protocolo necesario para la comunicación con los multiplexores (Figura 34).

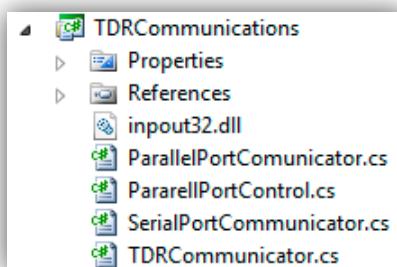


Figura 34. Contenido proyecto de “COMMUNICACION”.

La segunda parte de clases que tienen relación con esta capa hacen referencia al sistema de análisis. De la misma forma que lo descrito en la parte anterior para el sistema de comunicación, el formato de diseño de los análisis es idéntico. Se establece una clase genérica que contenga los atributos y métodos que se puedan aplicar y luego cada tipo de análisis ejecuta sus funciones de forma determinada. Como se puede apreciar en la siguiente (Figura 35) imagen existe un proyecto único para análisis que de forma organizada contiene todas las clases que se necesitan.

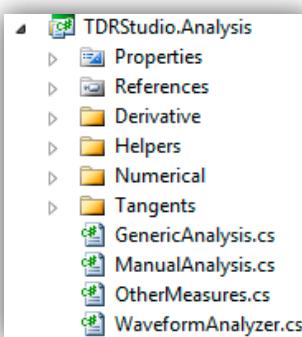


Figura 35. Contenido proyecto de “ANALYSIS”.

Anexo C

C. Diseño de flujos.

Después de haber analizado la estructura de la aplicación desde la perspectiva de usuario y desde la visión que ofrece la programación modular, vamos a describir mediante diagramas el proceso de gestión que ofrece la aplicación.

En un primer lugar, se encuentran los diagramas de secuencia que nos ofrecen una visión general de los procedimientos que se tienen que cumplir para el correcto funcionamiento, mientras que, en un segundo lugar, tenemos un diagrama de distribución que nos muestra el modelo de comunicación de los componentes.

C.1 Diagramas de secuencia.

El primer diagrama de secuencia (Figura 36) que se presenta en este apartado representa el modelo de creación y configuración de proyectos existentes en el sistema.

El diseño abarca la utilización de tres componentes. El usuario, el sistema gráfico y, en último lugar, la base de datos. El usuario inicia las acciones pidiendo abrir la aplicación para posteriormente proceder a la creación del proyecto y su configuración. A cada proceso ejecutado por el usuario se desencadena una serie de acciones en el sistema gráfico y éste a su vez en la base de datos.

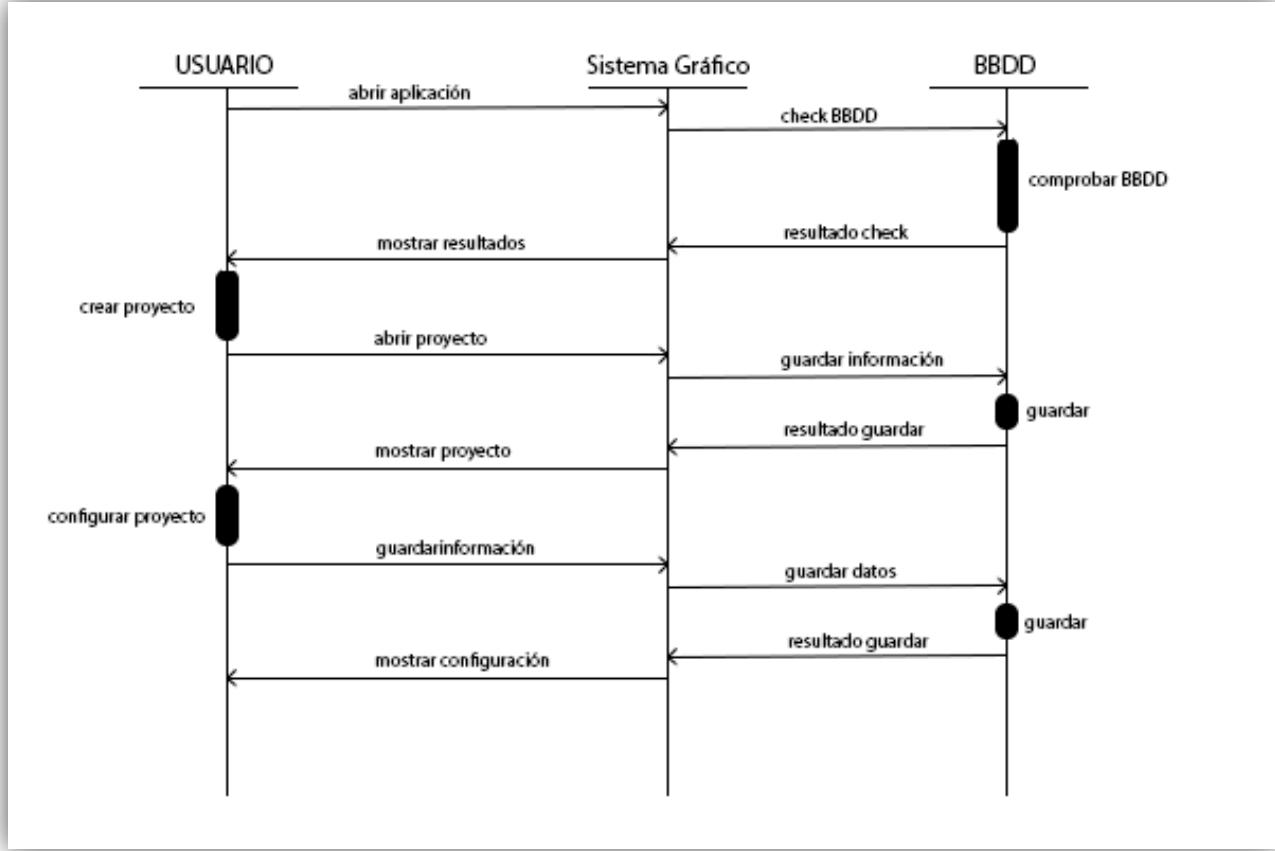


Figura 36. Secuencia de configuración de un proyecto.

En el siguiente diagrama (Figura 37) se muestra la secuencia para poder realizar análisis de una medida obtenida con el ecómetro TDR. El usuario solicita la adquisición de la onda al TDR que procesa la solicitud. Cuando ha terminado avisa al usuario mostrándole la onda. El usuario en ese punto decide el método que desea utilizar y procede.

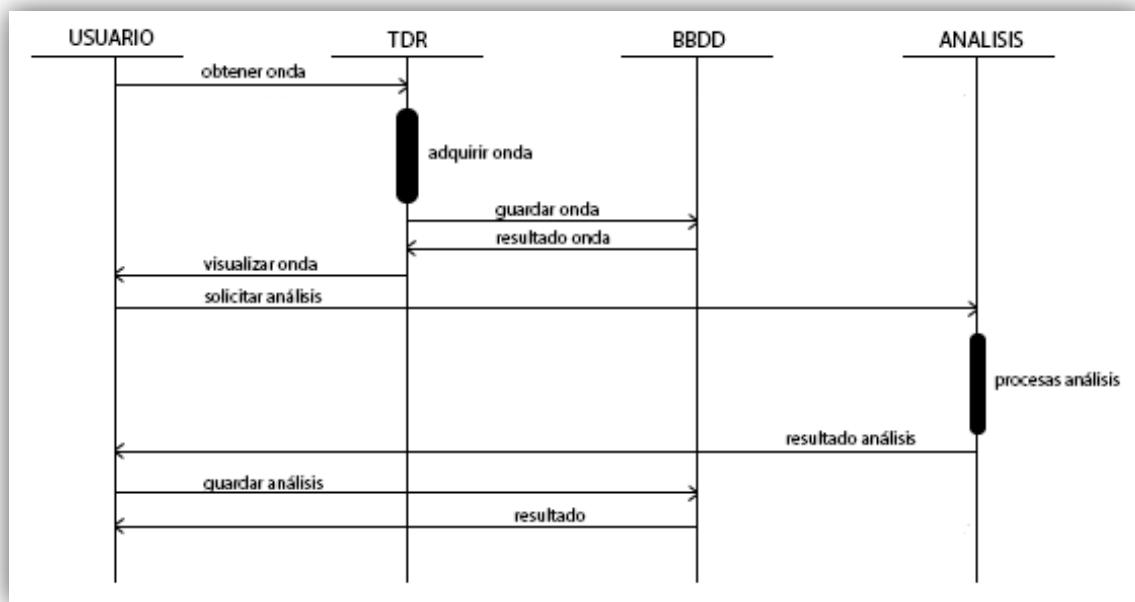


Figura 37. Secuencia de obtención y análisis de una onda.

C.2 Diagrama de distribución.

Mediante este diagrama (Figura 38) se describe el modo en el que los distintos componentes de la aplicación se conectan entre ellos. En la primera ilustración se puede ver mediante el diagrama de componentes como se conectan los componentes de forma informática, mientras que en la segunda ilustración observamos gráficamente el sistema cableado.

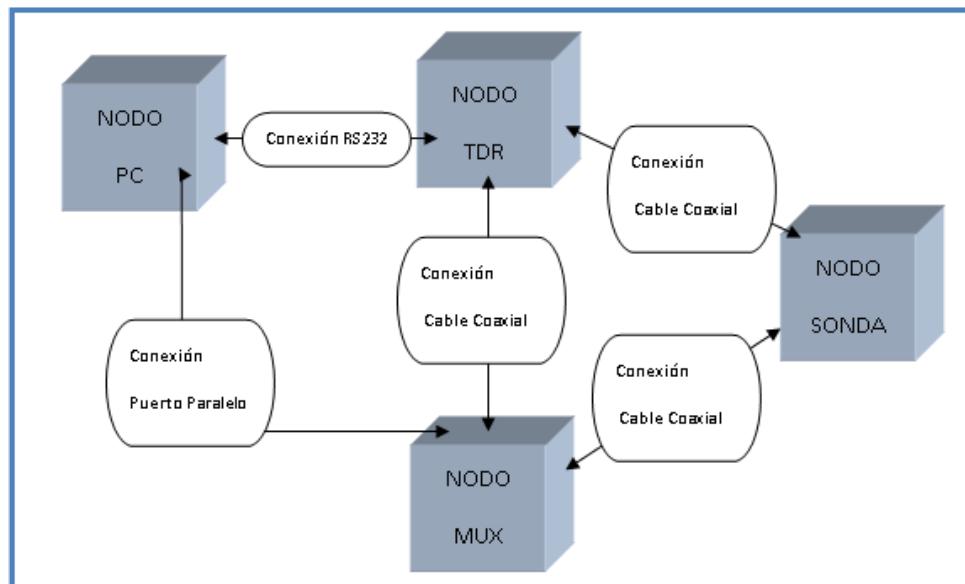


Figura 38. Comunicaciones entre componentes.

Esquema detallado del hardware utilizado en el proyecto (Figura 39).

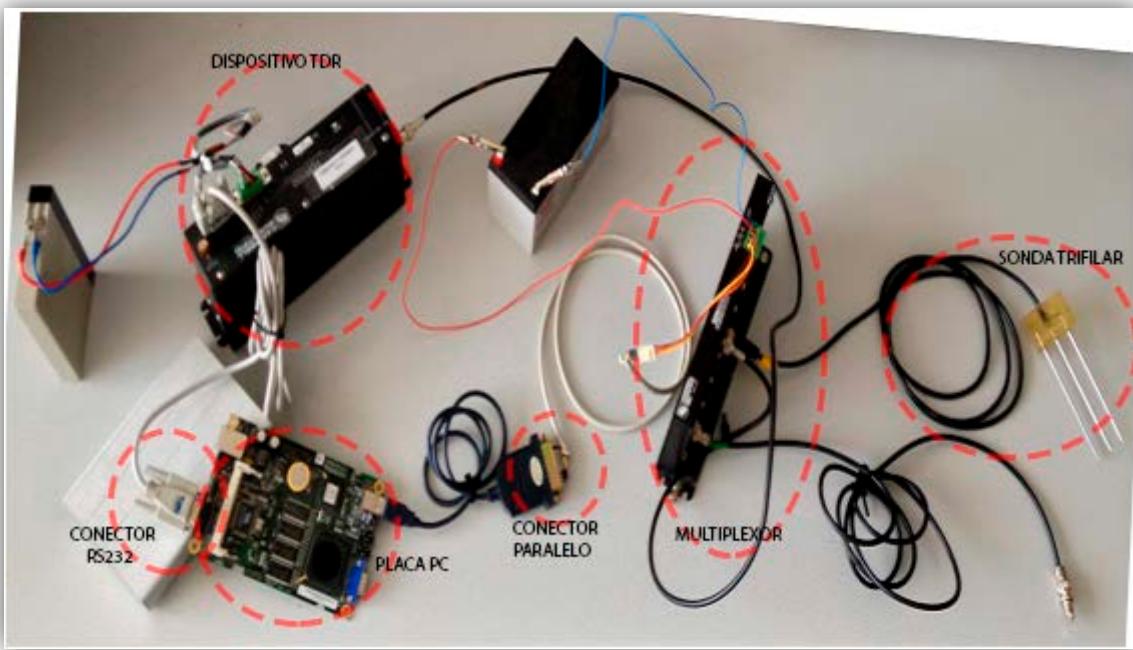


Figura 39. Dispositivos hardware de la tecnología TDR.

Anexo D

D. Dispositivo multiplexor Campbell y su control

El dispositivo hardware empleado es el dispositivo Multiplexor fabricado por Campbell. Concretamente el modelo “SDMX50 Coaxial Multiplexer” (Figura 40). Dicho dispositivo es capaz mediante un interno sistemas de relés multiplexar 8 salidas.



Figura 40. Multiplexor “CAMPBELL”.

Como se puede apreciar en la imagen, el dispositivo consta en la parte superior de un sistema de entrada para líneas de cable. Estas entradas se corresponden a las señales de control que permiten la manipulación del dispositivo.

En la parte central nos encontramos con 9 conectores para cables coaxiales. El conector que se encuentra desalineado de los demás es el encargado de ser el conector de entrada desde otros dispositivos mientras que los 8 conectores restantes representan cada una de las salidas que ofrece el dispositivo.

El último elemento a destacar del multiplexor es una rueda configurable que en el apartado de configuración del multiplexor (apartado 4.2.3) se explicará su funcionamiento. Antes de pasar a explicar el sistema de control, hay que explicar el cableado que ha sido necesario para su control. En primer lugar, la forma más fácil de poder controlar los multiplexores por parte del ordenador es mediante el puerto paralelo (ya que el puerto serie lo usamos ya para controlar el ecómetro) por lo que, dado que sólo necesitamos 5 pines del total del puerto paralelo, deberemos crear un adaptador que con empalme las salidas que nos sean disponibles.

En la siguiente composición (Figura 41) observamos los empalmes que deben realizarse tanto en el extremo del puerto paralelo como en el extremo del multiplexor.

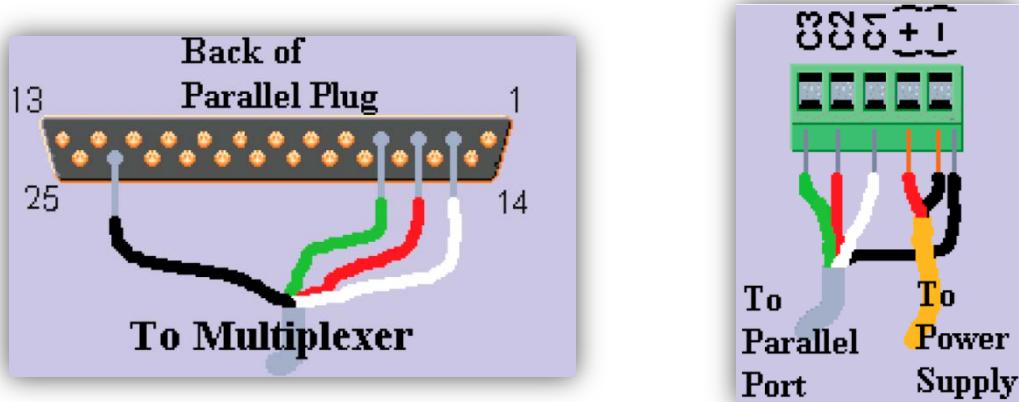


Figura 41. Esquema puerto paralelo-multiplexor.

Para realizar el control de los multiplexores se ha implementado en lenguaje C# un protocolo admitido para los multiplexores de forma que permite controlarlos y hacerles modificar el funcionamiento internos de los relés. Con este protocolo se permite configurar por etapas todos los niveles admitidos por el sistema. Debido a que cuantos más niveles son añadidos al proyecto menos calidad tenemos a la hora de recoger la información, se ha limitado hasta 3 niveles de multiplexión, de forma que pasamos de la posibilidad de trabajo con 1 sonda a las 512 que potencialmente pueden modificarse.

La forma de controlar los distintos multiplexores se produce mediante el envío por el bus de control de una ristra de señales que tienen la siguiente estructura:

- Secuencia para comunicar a los multiplexores que se produce una transmisión.
- Bit con valor alto.
- Secuencia de tres bits que indican la salida del multiplexor que se desea cortocircuitar.
- Secuencia de cuatro bits que indican el nivel de multiplexor que deseamos configurar. Esta secuencia representa la codificación del número hexadecimal de cada multiplexor, tal y como se ha explicado anteriormente.
- Secuencia para comunicar a los multiplexores que se cierra la transmisión.

Como cada multiplexor tiene configurado un identificador de nivel mediante el relé situada en la parte superior, puede detectar si el mensaje de configuración le corresponde o si, por el contrario, tiene que desestimar la petición de configuración. En el caso de tener tres niveles de multiplexión sería necesario lanzar hasta tres veces la orden de configurar los multiplexores, una para cada nivel.

Anexo E

E. Capturas de pantalla de la aplicación.

E.1 Formulario Splash

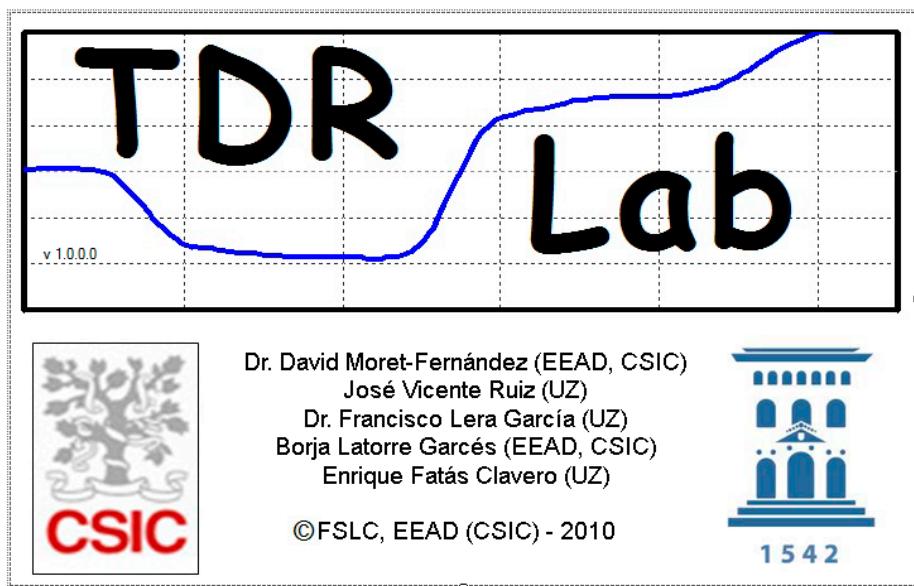


Figura 42. Formulario de bienvenida.

E.2 Formulario Acerca De

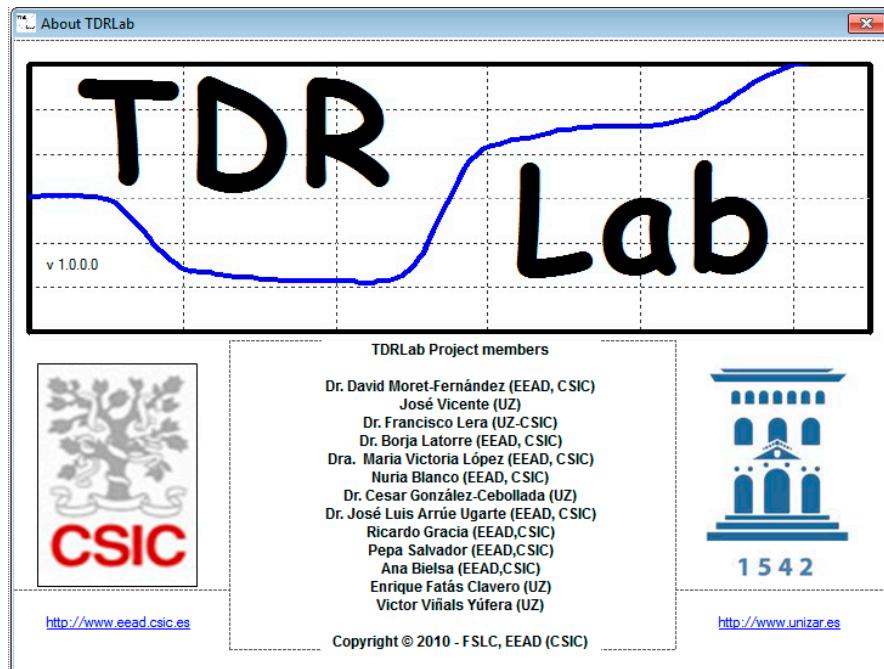


Figura 43. Formulario de "Acerca de".

E.3 Licencia de uso de la aplicación

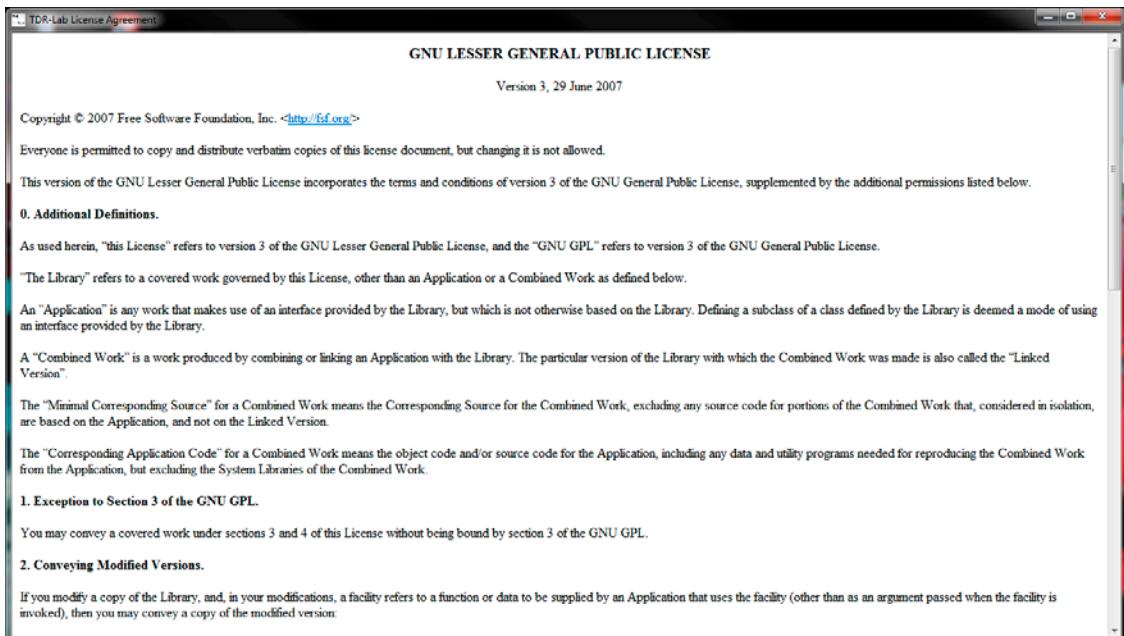


Figura 44. Formulario de licencia de la aplicación.

E.4 Licencia de uso de los métodos Analíticos

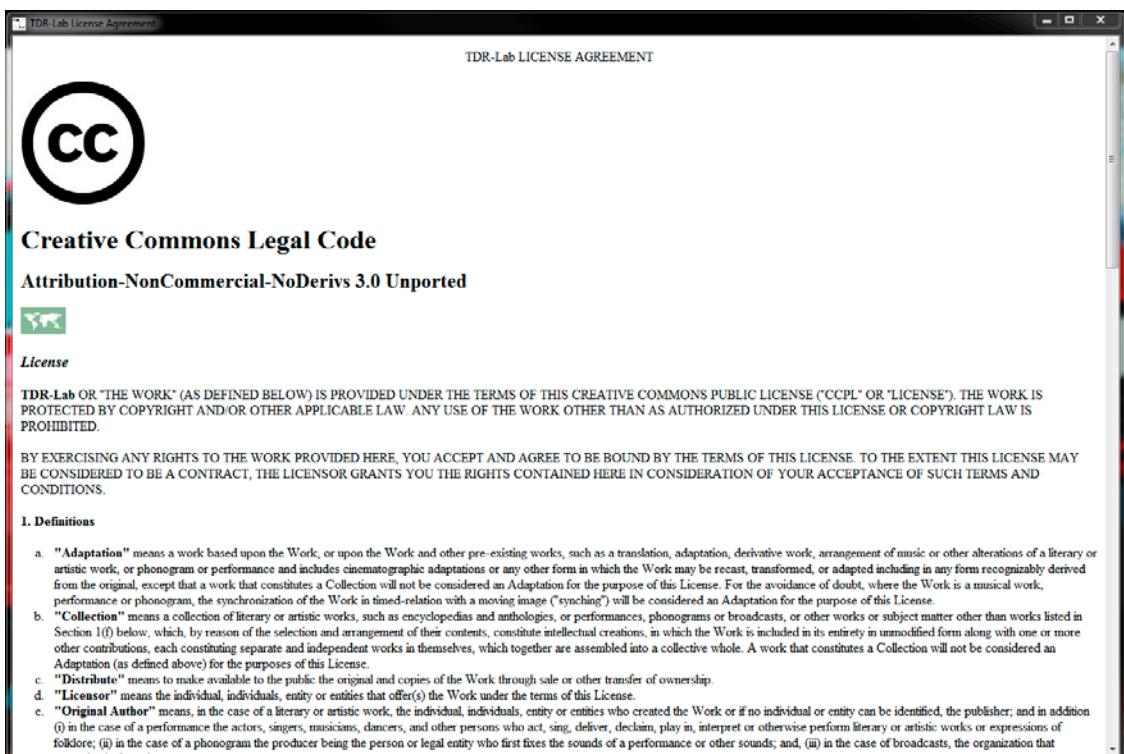


Figura 45. Formulario de licencia de métodos analíticos.

E.5 Formulario principal de gestión de proyectos

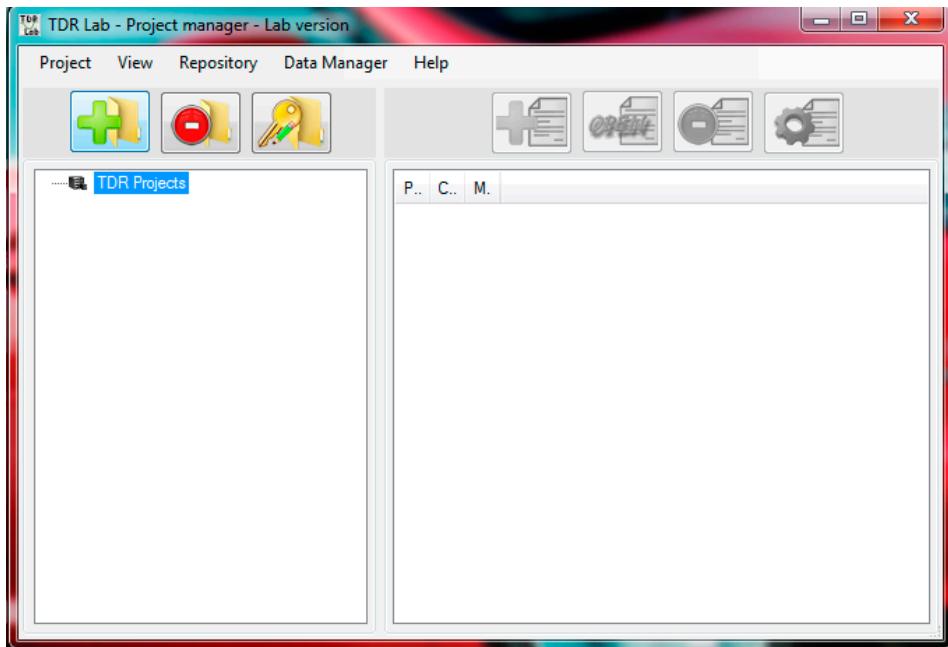


Figura 46. Formulario principal para la gestión de proyectos.

E.6 Detalle de modificación de propiedades

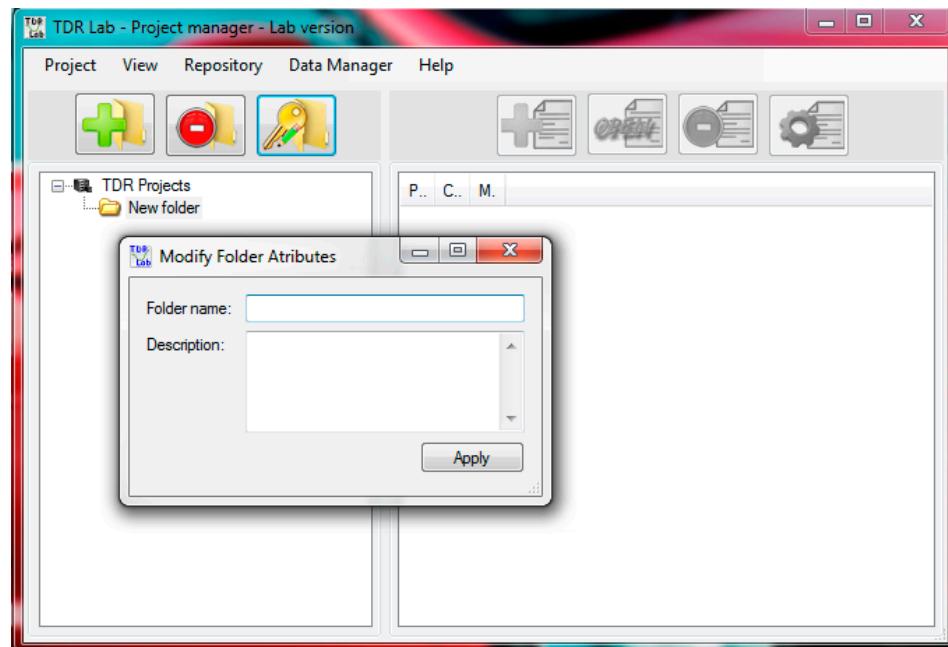


Figura 47. Detalle de modificación de propiedades.

E.7 Configuración de un nuevo proyecto

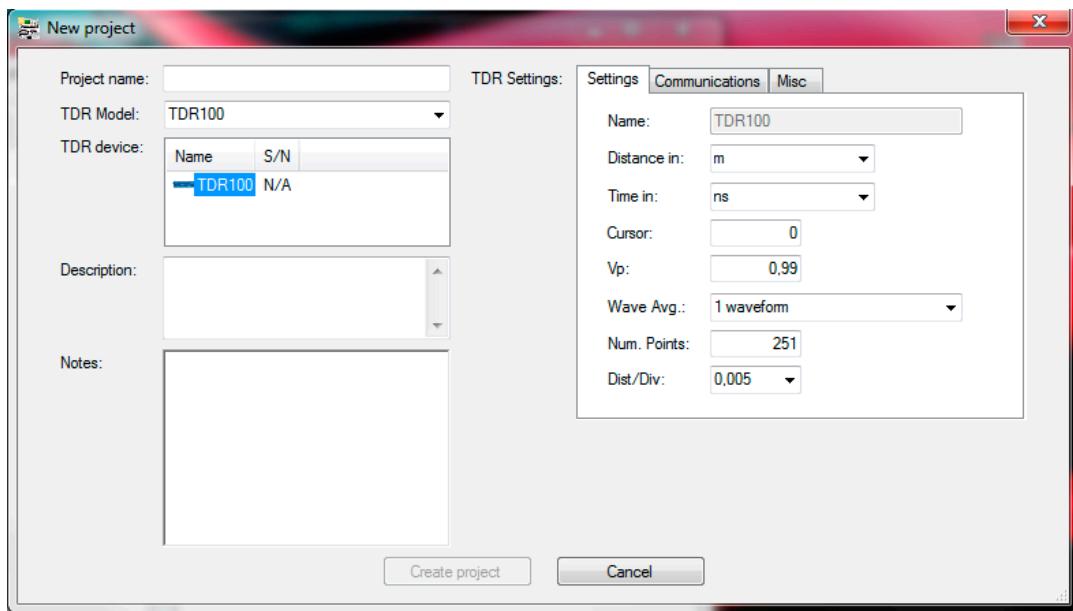


Figura 48. Configuración de un nuevo proyecto.

E.8 Forma de generar configuraciones cable-sondas

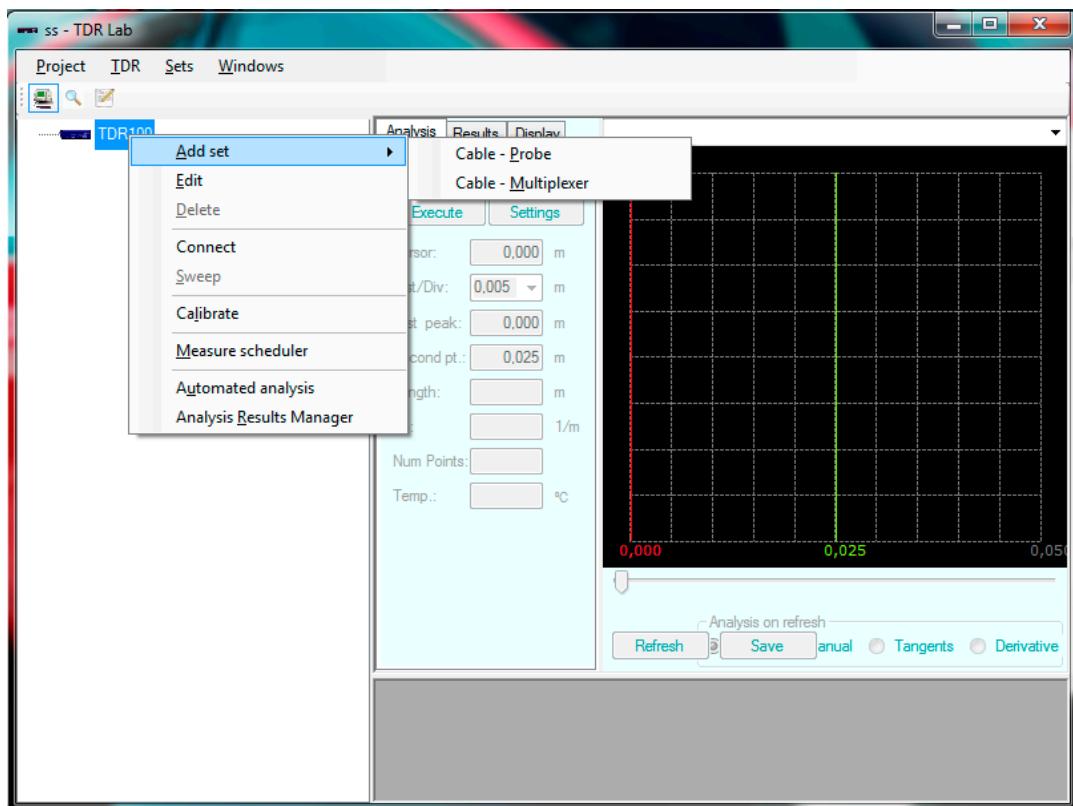


Figura 49. Vista para añadir configuraciones.

E.9 Formulario para configuración de multiplexor.

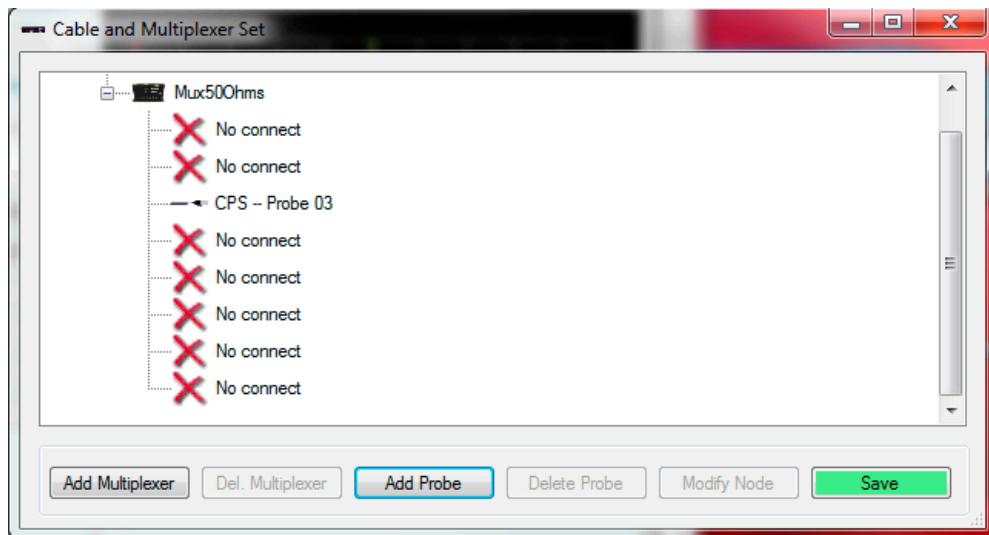


Figura 50. Esquema de configuración de multiplexación completa.

E.10 Detalle configurado de un proyecto

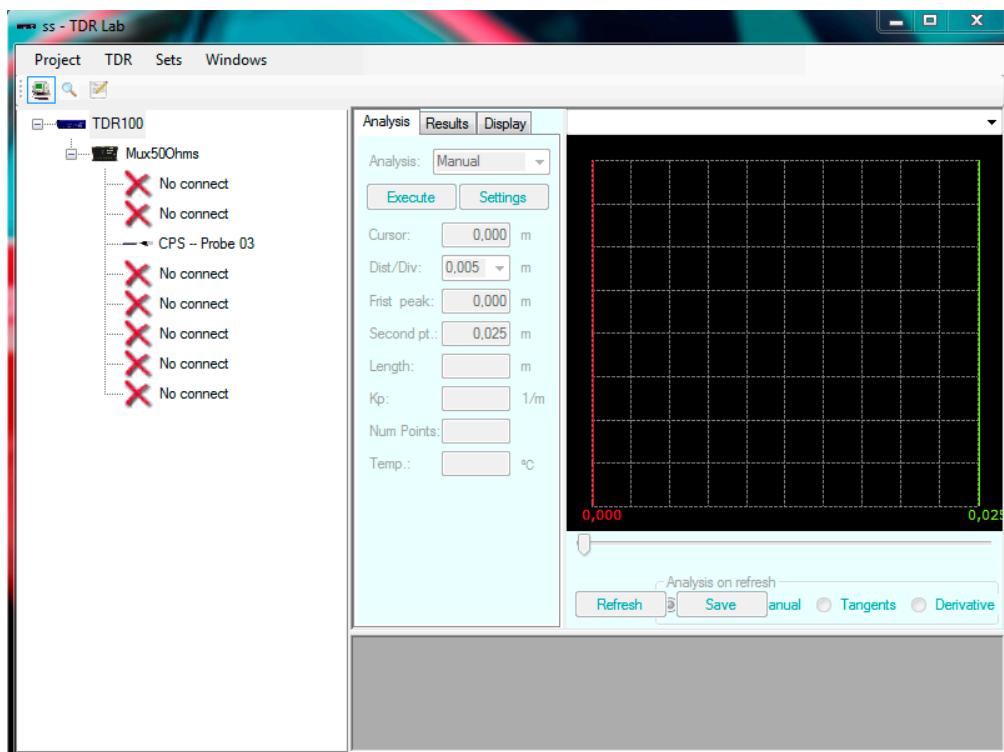
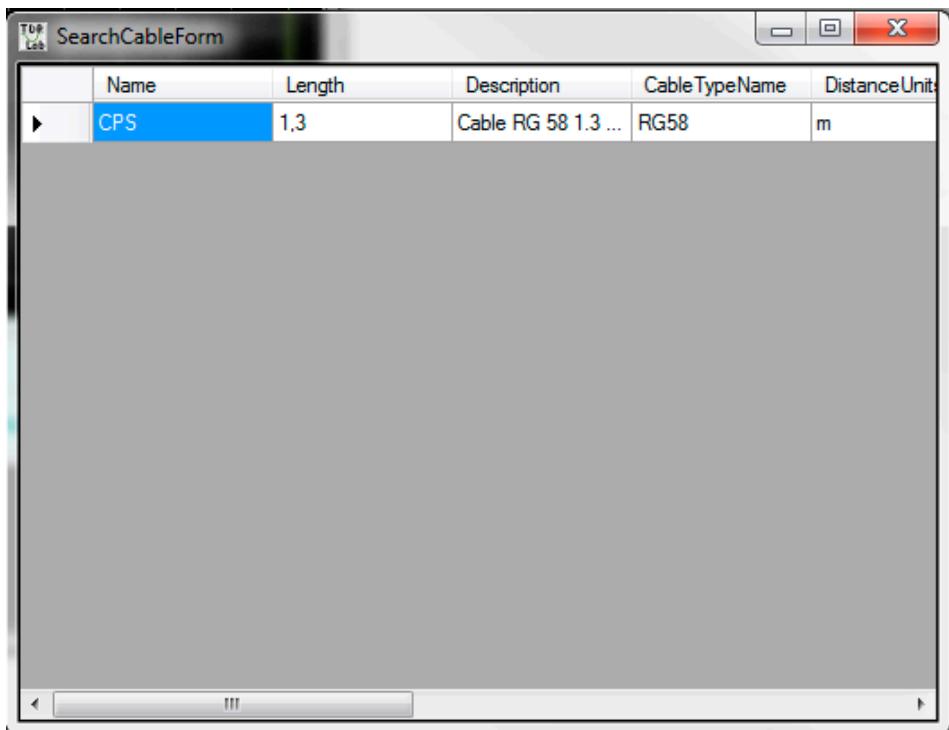


Figura 51. Formulario "DISPLAY" configurado.

E.11 Repositorio de cables del proyecto



The screenshot shows a Windows application window titled 'SearchCableForm'. The window has a standard title bar with minimize, maximize, and close buttons. The main content is a table with the following columns: Name, Length, Description, CableTypeName, and DistanceUnit. One row is visible, showing 'CPS' in the Name column, '1.3' in the Length column, 'Cable RG 58 1.3 ...' in the Description column, 'RG58' in the CableTypeName column, and 'm' in the DistanceUnit column. The table has a light gray background and dark gray header rows. The application window is surrounded by a dark gray border.

	Name	Length	Description	CableTypeName	DistanceUnit
▶	CPS	1.3	Cable RG 58 1.3 ...	RG58	m

Figura 52. Repositorio de cables del proyecto.

E.12 Menú para importar/exportar datos

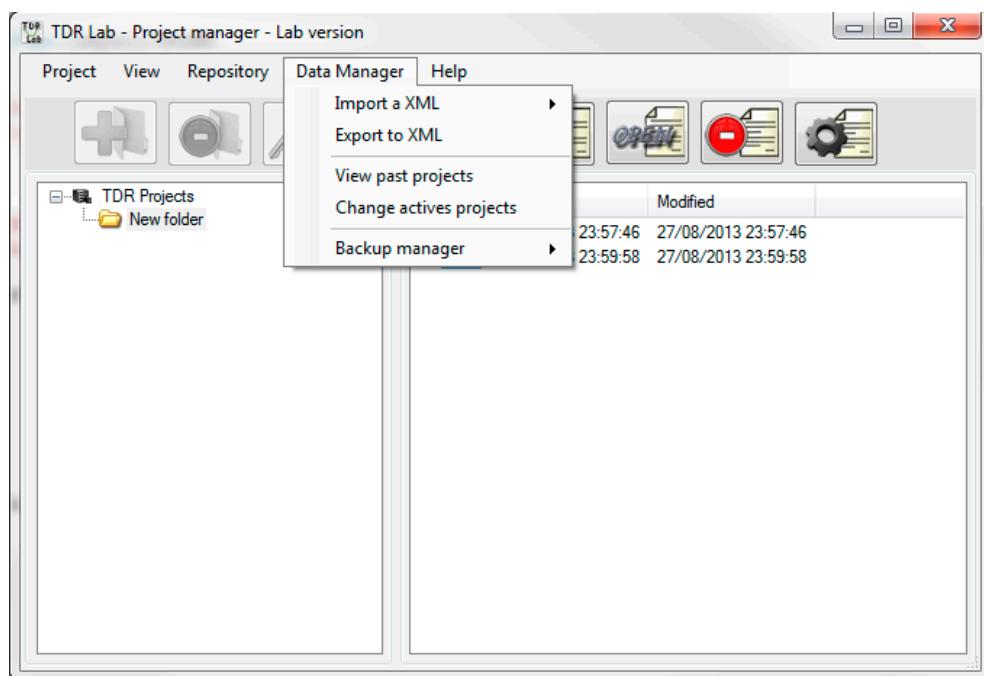


Figura 53. Menú para importar/exportar proyectos.

E.13 Formulario detalle de importación/exportación

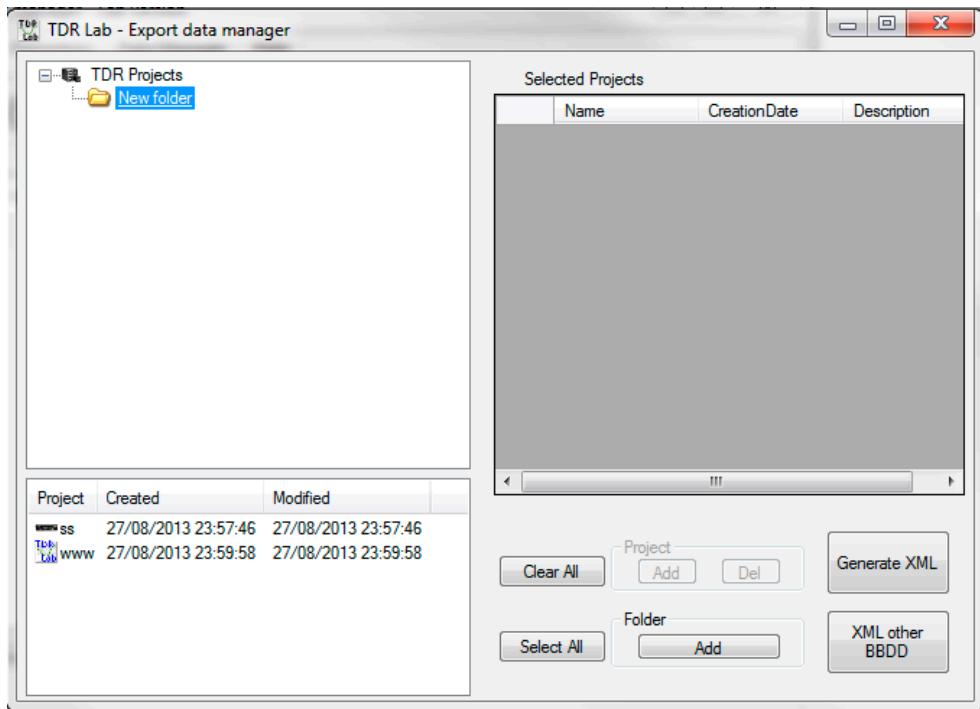


Figura 54. Detalle de la importación/exportación.

E.14 Análisis automático de medidas

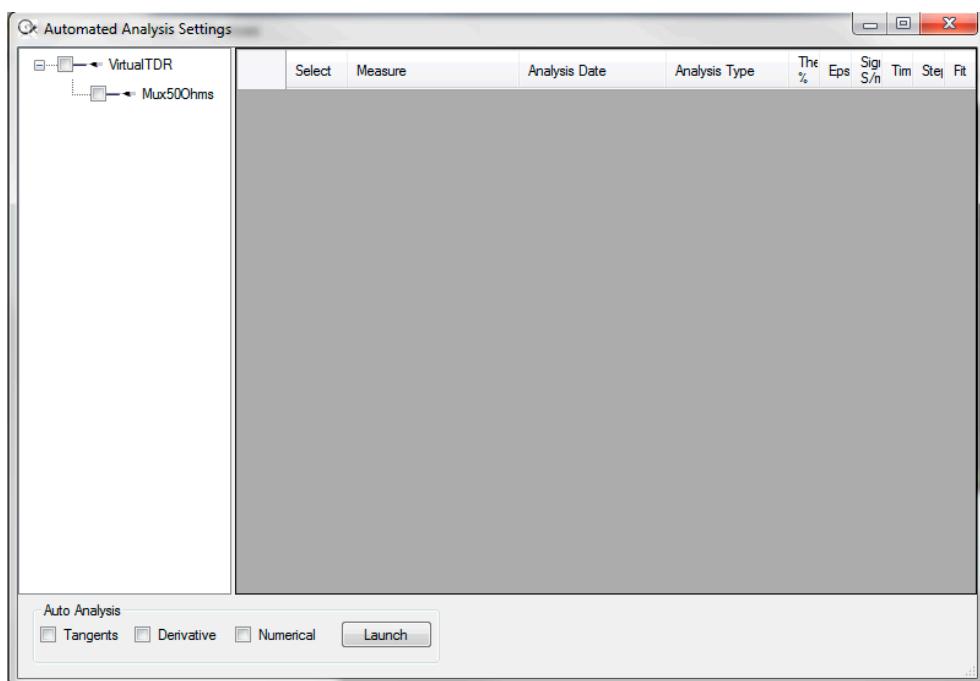


Figura 55. Formulario de análisis automático de ondas.

E.15 Obtención automática de ondas.

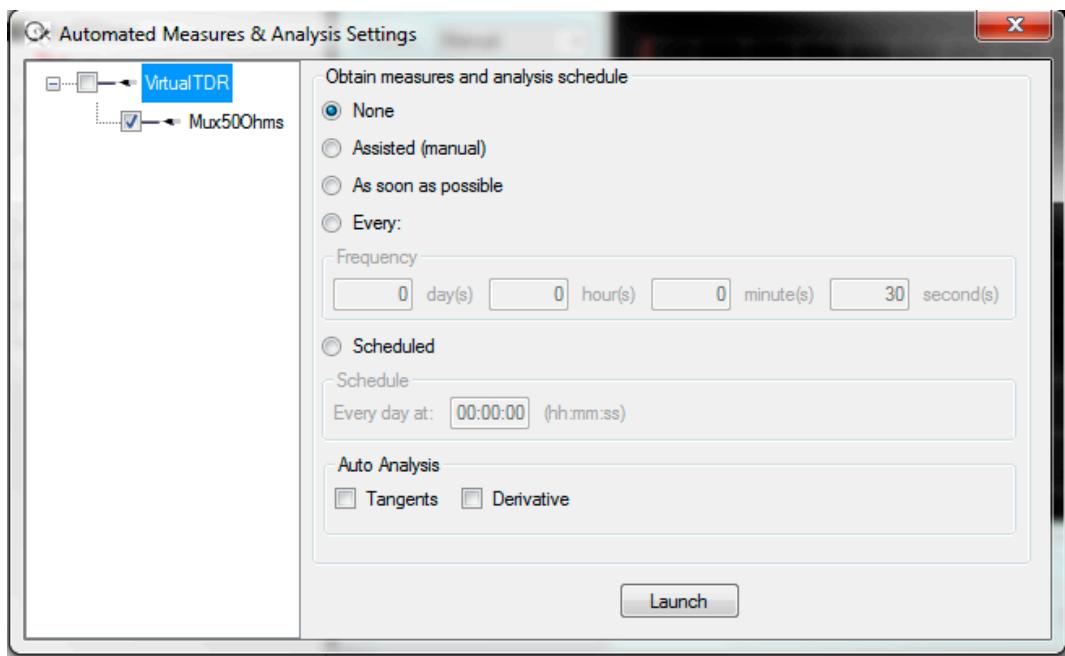


Figura 56. Obtención automática de ondas.

Anexo F

F. Conference on Monitoring and Modeling Soil-Plant-Atmosphere Processes, Naples, 2013

Four Decades of Progress in Monitoring and Modeling of Processes in the Soil-Plant-Atmosphere System: Applications and Challenges

TDR-LAB 2.0 Improved TDR Software for soil water content and electrical conductivity measurements

E. Fatás¹, J. Vicente³, B. Latorre⁴, F. Lera², V. Viñals¹, M.V. López², N. Blanco², C. Peña², C. González-Cebollada⁵
and D. Moret-Fernández⁴

(1) Departamento de Informática e Ingeniería de Sistemas- Universidad de Zaragoza C/María de Luna 1, 50018, Zaragoza, (Spain)

(2) Departamento de Ingeniería Electrónica y Comunicaciones. ICMA-CSIC - Universidad de Zaragoza C/María de Luna 1, 50018, Zaragoza, (Spain)

(3) Division of Physics, Office of Science and Engineering Laboratories, Center for Devices and Radiological Health, U.S. Food and Drug Administration, Silver Spring, Maryland, USA

(4) Departamento de Suelo y Agua. Estación Experimental de Aula Dei, CSIC PO Box 202, 50080, Zaragoza (Spain)

(5) Área de Mecánica de Fluidos. Universidad de Zaragoza. Huesca (Spain)

Abstract

Time Domain Reflectometry is a technique widely used in hydrology and agronomy that allows real time estimation of soil volumetric water content (θ), which is related to soil's apparent permittivity (ϵ_a) and bulk electrical conductivity (σ). This work presents an enhanced release of TDR-Lab, software which controls instrumentation for field measurements of θ and σ , enabling a convenient recording and retrieving of data.

TDR-Lab 2.0 supports Tektronics 1502C, TDR-100 Campbell Sci. and TRASE Soil-moisture Equipment Corp., which can be connected to a multiplexing system (SDMX50, Campbell Sci) allowing automated scheduled measurements from up to 251 different probes. This new release, when connected to TDR-100, allows increasing the waveform accuracy up to 2048 points/waveform. Multiple comparison of TDR waveform within the same TDR screen is now allowed. Graphical or numerical methods can be used for the estimation of θ and σ from soil measurements coming taken with different probes. Additional features to carry out water-surface-level measurements such as matric potential and soil solution electrical conductivity are also available when measurements are made using the corresponding specific probes. Two different versions, a lite and a full release, for field and laboratory applications have been developed. The light version with a reduced set of features (TDR-Lab Lite) was designed to run in low-end ultraportable devices. TDR-Lab Lite works with XML-files instead of the SQL database engine of the extended TDR-Lab, and has lower system requirements and a faster boot-up time. A robust import/export graphical user interface (GUI) facilitates transferring projects between the centralized SQL database and XML files. A new project manager window has been implemented, where the bar menu has been complemented with a useful set of icons. The display system projects has been improved and simplified. Within the project, a new friendly configuration manager for cables and TDR probes has been developed. Finally, an improved calibration procedure for TDR probes has been implemented. TDR Lab 2.0 supports more devices and provides additional estimated features from soil's measurements. Besides, the low hardware requirement of TDR-Lab Lite facilitates faster in-the-field tests and measurements.

© 2013 The Authors. Published by Elsevier B.V.

Selection and/or peer-review under responsibility of the Scientific Committee of the conference.

Keywords: Time Domain Reflectometry; Software; Waveform;

Introduction

Knowledge of soil water content and its distribution in the vadose zone is of paramount importance in many soil-related disciplines such as soil science, agriculture, forestry and hydrology. The Time Domain Reflectometry (TDR) has become a popular method for the accurate, quick, and non-destructive estimation of the apparent permittivity (ϵ_a), which is related to the volumetric soil

water content (θ) and the bulk electrical conductivity (σ) (Topp and Ferré, 2002). Other TDR applications for soil science have been focused, for instance, on measurements of water levels in Mariotte tubes (Moret et al., 2004), estimation of the soil matric water potential (Or and Wraith, 1999) or the soil water solution electrical conductivity (Moret-Fernández et al., 2012). Estimations of the apparent permittivity by TDR, and consequently water content, are generally based on a graphical interpretation of the reflected TDR waveform from the probe length using either the double-tangent waveform analysis (Herkelrath et al., 1991). The bulk electrical conductivity, however, is mainly estimated by analyzing the amplitude of the long-time TDR signal according to the procedure by Lin et al. (2008). Modelling of TDR signals by numerical inversion of the TDR waveform is becoming a robust alternative to the classical methods to estimate both water content and soil bulk electrical conductivity (Greco 2006; Heimovaara et al. 2004).

To date, the design of specific instruments to use the TDR technology in measuring soil water content has experienced a fast development, as shown by numerous companies that market TDR instruments for soil applications: Adcon, IMKO, Streat Instruments Ltd., Campbell Scientific, Global Water Instrumentation, Inc., Environmental Sensors, Inc. Automata Inc., Meteolabor AG, Dynamax, Soil Moisture Equipment Corporation. The earliest instrument commonly used for field investigations was the model 1502, Metallic Cable Tester, manufactured by Tektronix of Beaverton, Oregon. This instrument, only allowed to the users a manual determination of θ . This manual procedure was time consuming and inaccurate. These problems were solved by designing specific software that allowed automatic analysis of the TDR waveform. This is the case, for instance, of the TACQ program (Evett, 2000) (<http://www.cprl.ars.usda.gov/programs/>) developed in the early 1990s. This program allowed the users a complete control over multiplexer, individual settings for probe length, window width, averaging, distance to each probe, gain, and type of data acquired. Following this incipient project, the Soil Physics Group at Utah State University, Logan Utah, USA, created in the 1993 the WINTDR program, which being only compatible with the TDR cable testers Tektronix 1502, incorporated an easy and friendly Windows interface for accurate and efficient analysis under a variety of conditions (<http://soilphysics.usu.edu/wintdr/download.htm>). Other software also developed for water content estimations using TDR systems are, for instance, the WinTrase software from Soil Moisture that works only with the Soil Moisture TDR products and runs under MS Windows operating systems, or the free available PC-TDR software developed by Campbell Scientific Cop. developed exclusively for the TDR-100 Campbell Sci. cable tester. More recently, Moret-Fernández et al. (2010) developed new TDR software, TDR-Lab 1.0, for estimates of θ and σ . This software supposed a new advance regarding to the former TDR applications since, being compatible to two different TDR cable testers, it included a user-friendly and dynamic file format to show and save the TDR waveforms and different methods of TDR waveform analysis. However, this version, that resulted incomplete, did not include multiplexers and was developed on a heavy central data base that restricted its use on low capacity computers.

In spite of the large expansion of the TDR technology for θ and σ estimations, the public release of software to date which are compatible with different TDR instruments are quite limited. On the other hand, the versatility for the analysis and storage of the recorded TDR waveform of the different free available TDR software is currently quite restricted. The objective of this paper is to present a new and improved version of the TDR-Lab 1.0 software (TDR-Lab 2.0) which, being compatible with three different TDR cable testers, has significant improvements regarding to the former version. Two different versions for laboratory (TDR-Lab) and field (TDR-Lab Lite) measurements have been developed. TDR-Lab Lite includes a reduced set of features and was designed to run on low-end ultraportable devices.

TDR-LAB FEATURES: TDR-Lab AND TDR-Lab LITE

Two different linked versions for laboratory (TDR-Lab) and field (TDR-Lab Lite), with an easy Windows user interface, have been developed.

The TDR-Lab Lite, which works with XML-files, has been designed to run on low-end ultraportable devices. The main advances regarding to the former TDR-Lab 1.0 version are: compatibility with the Tektronix 1502C Metallic TDR, the TDR100 TDR Campbell Scientist and the TRASE (Soil-moisture Equipment Corp.) cable testers; compatible to the multiplexer system SDMX50 (Campbell Sci.); high resolution waveforms (up to 2048 samples) when connected with TDR100; self-calibration with graphical methods, and a more intuitive configuration manager for cables and TDR probes; an open interface to make important data visible, allowing that recorded and stored TDR waveforms to be

combined; multiple waveforms can be superimposed for on screen comparison; three different methods for water content estimations (manual, derivative, and tangent methods) and a single procedure for bulk electrical conductivity determinations; additional features to compute water-surface-level measurements, matric potential or soil solution electrical conductivity when using the specific probes; automated waveform readings; a new dynamic file format to show and save the TDR waveforms and analysis results and a simplified and improved display system; and a reliable import/export graphical interface (GUI) to allow transferring projects between TDR-Lab centralized SQL database and XML files.

The complete version of the TDR-Lab, which works with a SQL database, allows centralizing all data in a same computer. In addition to the features described in the TDR-Lab Lite release, this extended version included: a virtual TDR machine to simulate TDR waveforms; an additional method to estimate of θ and σ_a by numerical modelling of TDR waveform and the corresponding numerical procedure to calibrate TDR probes.

Software description

The TDR-Lab 2.0 is programmed in C# with Microsoft®.Net Framework® 3.5. A new based layers implementation, which includes a module that interacts with the former version programming (TDR-Lab 1.0), has been developed. Three different levels have been defined (Figure 1): (i) the Graphical User Interface (GUI), which allows the user selecting the available operations; (ii) the Business Logical Layard (BLL) layer, that controls the technical management involved in the data collection and data processing; and (iii) the Data Access Layer, where all information coming from the upper layers is stored and processed. A new module implemented in database layer allows simplifying and increasing the application efficiency. This module is compatible with the databases storage features.

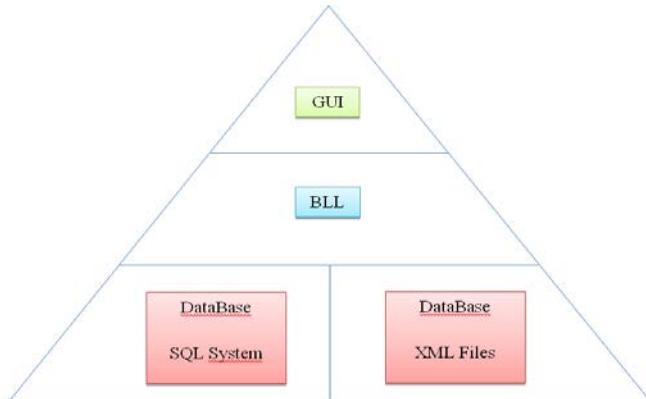


Figure 1. Pyramidal structure layers of the application.

The program architecture consists of five components that provide a unified user interface (UI) to operate different TDR cable testers (Figure 2). These UI components are: (i) project manager and editor windows, (ii) equipment (tdr → probe), (iii) waveform acquisition and measured waveforms, (iv) waveform analysis, and (v) data access. The first step to acquire new TDR waveforms involves, before connecting the cable tester, to select one of the two TDR cable testers available in the program. Once the cable tester is connected, the user should create a new project defining the characteristics of the transmission line.

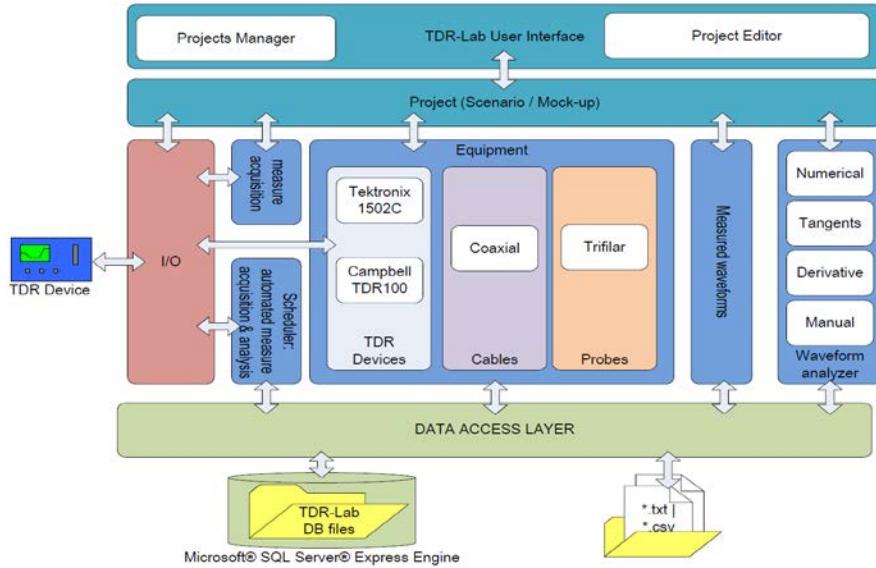


Figure 2. Flowchart of the TDR software.

1.1. Project manager and editor windows

The TDR-Lab data is organized in projects, which are saved in the folders shown in the TDR-Lab Project Manager window (Figure 3). This window is divided in three sections: (i) the folder tree on the left, that shows the set of folders that contains the TDR projects, (ii) the project window on the right, that shows the TDR project within each folder, and (iii) the menu bar that includes a repository application to show all TDR cable and probes and the import/export graphical interface to transfer projects between TDR-Lab centralized SQL database and XML files.

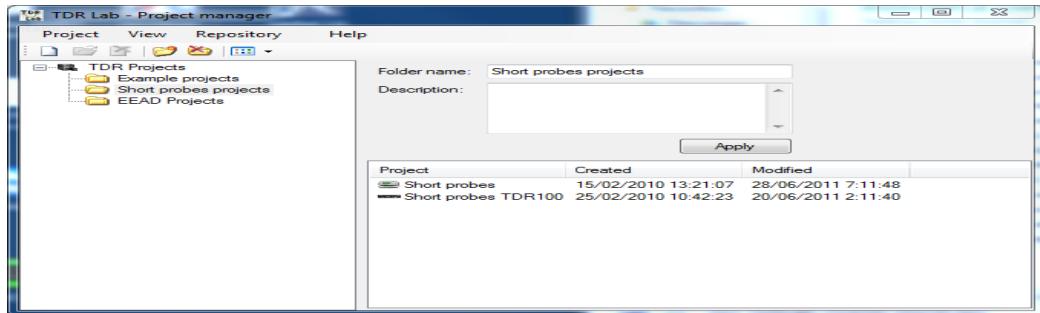


Figure 3. TDR-Lab project Manager window.

The project form storages information about the TDR cable tester related to the settings and communication setup. The Project Editor window (Figure 4) is automatically opened by clicking on a project in the Project Manager window (Figure 3). The Project Editor window is divided is:

- (i) Equipment tree, where the TDR cable tester, and TDR probes and waveforms are defined.
- (ii) TDR Screen, where TDR waveform acquisition, analysis and storage are executed.
- (iii) The menu bar that includes all information about the TDR equipment, the TDR waveforms, cables and TDR probes, calibration and automated analysis options.

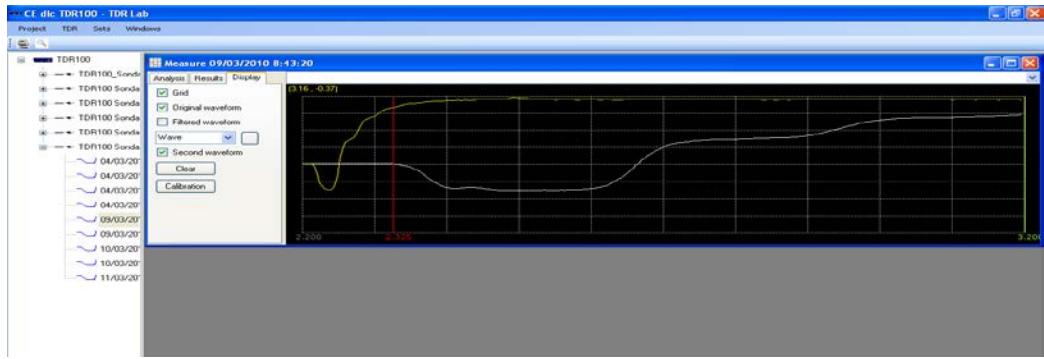


Figure 4. Project editor window and large and small scale TDR waveforms. Red line denotes the first peak of the TDR waveform.

1.2. Equipment

Three different components have been defined (Figure 2): (i) communications component that allows TDR-Lab to send commands and acquire waveforms from the cable tester; (ii) unified cable tester interface that makes possible the acquisition and analysis of the TDR waveforms to be independent of the type of TDR cable tester; and (iii) coaxial cable and TDR probes components that defines the properties of the coaxial cables (propagation velocity, impedance and length) and the probe characteristics. Self-calibration methods for coaxial cables and TDR probes have been included in this component.

1.3. TDR waveform acquisition

Three different forms of TDR waveform acquisition are available: (i) manual or automatically acquisition of the current TDR waveform just recorded by the TDR cable tester; (ii) opening previously saved TDR waveforms which are displayed in the third of the Project Manager windows (Figure 4); and (iii) importing TDR waveforms from an external text file.

Two differently scaled TDR waveforms can be simultaneously acquired. The first one, which is defined on a large scale and allows estimating the soil water content, and a small scale waveform used to estimate of the bulk electrical conductivity from long-time TDR signal (Figure 4).

1.4. Waveform analysis and soil properties measurements

TDR waveform analysis can be performed on current TDR waveforms, on TDR waveforms previously saved in the central database or on imported data. Four different methods of waveform analysis for water content estimations are included: three graphical methods (manual, tangent, derivative) and a numerical inverse analysis of the TDR waveform for three-wire probes immersed in homogeneous media. The soil bulk electrical conductivity can also be estimated by either a graphical or a numerical procedure. While the manual, tangent or derivative methods can be used either on recently recorded or stored waveforms, the numerical method, which is only available in the complete TDR-Lab version, should be used on previously saved TDR signals. An option for automating the analysis of a collection of saved TDR traces has been included in the program.

3.4.1. Estimations of volumetric water content and bulk electrical conductivity

Graphical methods

Estimations of water content using the graphical TDR waveform analysis (manual, derivative or tangent procedure) are based on

$$\varepsilon_a = \left(\frac{ct_L}{2L} \right)^2 \quad (1)$$

where ε_a is the soil bulk dielectric constant of the embedded material, c is the velocity of light (3×10^8 m/s) and t_L (s) is the travel time for the pulse to traverse the length L (m) of the TDR guide. The value t_L is the distance between bump created by the impedance mismatch between cable and TDR probe head (first peak) (Figure 4) and the time when the trace arrives at the end of the TDR probe (second reflection point or ending point). The relationship between ε_a and θ is commonly calculated with a polynomial empirical relationship (Topp and Ferré, 2002).

The bulk electrical conductivity (σ_a) estimated with the long-time analysis of the TDR waveform is calculated according to (Giese and Tiemann, 1975) (Figure 4):

$$\sigma_a = \frac{K_p}{Z_r} \left(\frac{1 - \rho_{\infty, \text{Scale}}}{1 + \rho_{\infty, \text{Scale}}} \right) \quad (2)$$

where Z_r is the output impedance of the TDR cable tester (50Ω), K_p (m^{-1}) is the probe-geometry-dependent cell constant value, and $\rho_{\infty, \text{Scale}}$ is the scaled steady-state reflection coefficient for ideal condition calculated according to (Lin et al., 2008):

$$\rho_{\infty, \text{Scale}} = 2 \frac{(\rho_{\text{air}} - \rho_{\text{SC}})(\rho - \rho_{\text{air}})}{(1 + \rho_{\text{SC}})(\rho - \rho_{\text{air}}) + (\rho_{\text{air}} - \rho_{\text{SC}})(1 + \rho_{\text{air}})} + 1 \quad (3)$$

where ρ , ρ_{air} and ρ_{SC} are the long-time reflection coefficient measured in the studied medium, in air and in a short-circuited probe, respectively.

Numerical TDR waveform analysis

The soil θ and σ_a are numerically estimated by an inverse analysis of the TDR waveform (Heimovaara et al. 2004). The transmission line used in the model is driven by a step source voltage $V_s(t)$ of height V_{s0} with a source impedance R_s (usually 50Ω) and ends in an open termination with $Z_L = \infty$. The cable and probe are modelled as lossy transmission lines in the frequency domain. Fourier analysis is used (Heimovaara et al., 2004; Huebner and Kupfer, 2007) with direct and inverse FFT algorithms for switching from the time to frequency domain and vice-versa. The frequency domain transfer function of the soil-probe-cable set is that of a voltage divider constituted by R_s (nominally 50Ω) and the frequency-dependent input impedance of the cable-probe-soil set (Z). The transmission lines are characterized with four parameters (Ramo et al., 1984): capacitance C (F m^{-1}), inductance L (H m^{-1}) conductance G (S/m) and resistance R (Ωm^{-1}). The ε_c is estimated by computing the frequency-dependent permittivity of pure water $\varepsilon_w(\omega)$ at a given temperature (currently 25°C) (Meissner and Wentz, 2004). For a given θ we obtain $\varepsilon_a(\theta)$ with a polynomial (Topp- and Ferré, 2002) formula (Eq. 2) and finally

$$\varepsilon_c = \varepsilon_{a0} + \frac{\varepsilon_a - \varepsilon_{a0}}{\varepsilon_{a1} - \varepsilon_{a0}} (\varepsilon_w - \varepsilon_{a0}) \quad (4)$$

where $\varepsilon_{a0} = \varepsilon_a(\theta = 0)$ and $\varepsilon_{a1} = \varepsilon_a(\theta = 1)$. The estimation of the soil parameters (θ and σ_a) is achieved by the golden-section search technique (Kiefer, 1953), after minimizing the root mean square (RMSE) from a comparison of the measured and modelled TDR waveforms. This procedure requires a previous calibration process to determine the effective length (l_{eff}) of the TDR probe and the initial time (t_0) at which the electromagnetic pulse enters the TDR probe.

3.4.2. Water level estimations with TDR

Using a vertical coated TDR probe of length L immersed in a water column, the water level ($L-x$) can be calculated according to (Moret et al., 2004)

$$x = L \frac{\sqrt{\varepsilon_{\text{TDR}}} - \sqrt{\varepsilon_w}}{\sqrt{\varepsilon_{\text{air}}} - \sqrt{\varepsilon_w}} \quad (5)$$

where x is the probe length above water level, ε_{TDR} is the apparent dielectric constant measured by the TDR cable tester, and ε_{air} and ε_w are the relative dielectric constants of air and water previously measured with the same probe, respectively. To compute water level measurements, values of ε_{air} and ε_w and L should be previously introduced in the TDR-Lab application.

3.4.3. Soil matric potential and soil solution electrical conductivity estimation

The soil matric potential (ψ) or the soil solution electrical conductivity can be estimated from the σ_a and θ values measured with a ceramic-TDR sensor. This consists on a set of commercially available porous ceramics plates arranged along the axis of a TDR probe (Or and Wraith, 1999). For matric potential estimation a θ - ψ relationship for each sensor should be previously established (Or and Wraith, 1999). In TDR-Lab 2.0 case, a simple Van Genutchten (1980) water retention function relating θ and ψ is so far available,

$$\theta = \left[(\theta_{sat} - \theta_r) \left[\frac{1}{1 + (\alpha\psi)^n} \right]^m \right] + \theta_r \quad (6)$$

where n is the pore-size distribution parameter, $m = 1 - (1/n)$, α [kPa] is the scale factor, and θ_{sat} and θ_r are the saturated and residual volumetric water contents of the ceramic plates, respectively. These parameters should be obtained from previous calibration experiments and introduced in the TDR-Lab.

The soil solution electrical conductivity corrected at 25 °C ($\sigma_{w/25}$) is estimated according to (Moret-Fernández et al., 2012)

$$\sigma_{w/25} = \sigma_w f \quad (7)$$

where f is an empirical factor expressed as (US Salinity Laboratory Staff, 1954)

$$f = 1 - 0.20346(T) + 0.03822(T^2) - 0.00555(T^3) \quad (8)$$

and $T = (T_{\circ C} - 25)/10$. The σ_w is the measured soil solution electrical conductivity expressed as

$$\sigma_w = \frac{\sigma_a}{\theta_{sat}^{\tau} \left(\frac{\theta}{\theta_{sat}} \right)^{\beta}} - \sigma_{a-s} \quad (9)$$

where σ_{a-s} is the bulk electrical conductivity of the solid phase of the dry ceramic plates, β and τ are a factor depending on the ceramic plate's water transmission porosity and a transmission coefficient of the saturated ceramic plates (Mualem and Friedman, 1991). The σ_a and θ are estimated by TDR and the σ_{a-s} (negligible), θ_{sat} , β and τ values, which are obtained from previous calibration experiments, should be introduced in the TDR-Lab.

1.5. Data access

All data of the complete TDR-Lab version are saved in a centralized data base. This includes project information, TDR devices, TDR settings, the repository of cables and probes, saved waveforms and analysis results. The results obtained from the TDR waveform analysis are organized in the Analysis Results Manager window (Figure 5). These data include the following items: the date and time of the TDR waveform analysis and storage, type of TDR waveform analysis, the values of the volumetric water content, dielectric permeability, bulk electrical conductivity and details of the numerical analysis (time-analysis), number of iterations and fitting value. Selected data from the centralized data base can be exported to a .csv format file that contains the following information: the name of the TDR probe, settings of the TDR waveforms, pairs of points of travel time and reflection coefficients for the different TDR waveforms, the results and method used to calculate the water content, dielectric constant and bulk electrical conductivity, if estimated.

Select	Measure	Analysis Date	Analysis Type	Theta %	Epsilon	Sigma S/m	Time	Step	Fit
<input type="checkbox"/>	25/02/2010 13:16:06	11/01/2011 11:15:52	Numerical	24.22	137.0...	0.0781	40.4297	21	0.0241
<input type="checkbox"/>	26/02/2010 10:49:35	11/01/2011 11:15:58	Numerical	15.11	137.0...	0.0781	40.4297	21	0.0184
<input type="checkbox"/>	26/02/2010 12:40:57	11/01/2011 11:16:04	Numerical	94.07	137.0...	0.0992	40.4297	21	0.0280
<input type="checkbox"/>	01/03/2010 8:45:06	11/01/2011 11:16:09	Numerical	0.78	137.0...	0.0781	40.4297	21	0.0220
<input type="checkbox"/>	01/03/2010 9:17:49	11/01/2011 11:16:17	Numerical	41.41	137.0...	2.1943	40.4297	21	0.0138
<input type="checkbox"/>	02/03/2010 8:37:31	11/01/2011 11:16:20	Numerical	36.43	137.0...	1.8947	40.4297	6	0.0167
<input type="checkbox"/>	03/03/2010 8:49:26	11/01/2011 11:16:26	Numerical	20.81	137.0...	0.4270	40.4297	21	0.0145
<input type="checkbox"/>	04/03/2010 9:44:57	11/01/2011 11:16:32	Numerical	13.67	137.0...	0.0781	40.4297	21	0.0289
<input type="checkbox"/>	04/03/2010 9:44:57	11/01/2011 11:33:28	Numerical	12.50	137.0...	0.3100	40.4297	4	0.0178
<input type="checkbox"/>	04/03/2010 9:44:57	11/01/2011 11:38:38	Tangents	-4.87	0.1467	0.3161			
<input type="checkbox"/>	08/03/2010 9:13:45	11/01/2011 11:16:35	Numerical	0.39	137.0...	0.0781	40.4297	21	0.1386
<input type="checkbox"/>	08/03/2010 10:15:01	11/01/2011 11:18:44	Numerical	39.22	137.0...	2.2248	40.4297	21	0.0124
<input type="checkbox"/>	09/03/2010 8:45:01	11/01/2011 11:18:53	Numerical	18.24	137.0...	0.5931	40.4297	21	0.0128
<input type="checkbox"/>	10/03/2010 8:44:13	11/01/2011 11:19:00	Numerical	13.55	137.0...	0.3993	40.4297	21	0.0130
<input type="checkbox"/>	12/03/2010 9:06:24	11/01/2011 11:19:02	Numerical	0.39	137.0...	0.0781	40.4297	21	0.1313

Figure 5. Results manager window

Conclusions

This paper presents new software which, being compatible with three different TDR cable testers, allows soil water content and bulk electrical conductivity to be estimated using different methods of analysis of the TDR waveforms (the manual, derivative, tangent and numerical methods). The software also allows estimations of water level, matric potential or soil solution electrical conductivity when specific probes are used. Two different versions for laboratory (TDR-Lab) and field (TDR-Lab Lite) measurements have been developed. TDR-Lab Lite includes a reduced set of features and was designed to run on low-end ultraportable devices. It includes multiplexer options, and the recorded TDR waveforms are stored in a central database and can easily be recovered for subsequent analysis. The output data are also saved in the central database and are displayed in a friendly Excel-type format. Taking into account the above-mentioned characteristics, the program results presented here, compared to previously available TDR software, represent a significant advance for TDR waveform management and analysis. The TDR-Lab software presented here is free and can be downloaded, after a requested registration, from <http://digital.csic.es/handle/10261/35790>.

Acknowledgements

This research was supported by the Ministerio de Educación y Ciencia of Spain (grant AGL2010-22050-C03-02) and DGA- Obra social La Caixa (Grants: 2012/ GA LC 074). The authors are grateful to M.V. López, A. Bielsa, J. Salvador and R. Gracia for their technical help in several aspects of this study.

References

- [1] Evett, S. R., 2000. The TACQ computer program for automatic time domain reflectometry measurements: Waveform interpretation methods. *Transactions ASAE* 43: 1947-1956.
- [2] Giese, K., Tiemann, R., 1975. Determination of the complex permittivity from thin-sample time domain reflectometry: improved analysis of the step waveform. *Adv. Mol. Relax. Proc.* 7: 45-49.
- [3] Greco, R., Feb. 2006. Soil water content inverse profiling from single TDR waveforms. *J. Hydrol.* 317: 325-339.
- [4] Heimovaara, T. J., Huisman, J. A., Vrugt, J. A., Bouting, W., Nov. 2004. Obtaining the spatial distribution of water content along a TDR probe using the scem-ua bayesian inverse modeling scheme. *Vadose Zone J.* 3: 1128-1145.

- [5] Herkelrath, W. N., Hamburg, S. P., Murphy, F., 1991. Automatic, real-time monitoring of soil moisture in a remote field area with time domain reflectometry. *Water Resour. Res.* 27: 857-864.
- [6] Huebner, C., Kupfer, K., 2007. Modelling of electromagnetic wave propagation along transmission lines in inhomogeneous media. *Measurement Sci. Technol.* 18: 1147-1154.
- [7] Kiefer, J., 1953. Sequential minimax search for a maximum. *Proc. Amer. Math. Soc.* 4: 502-506.
- [8] Lin, C.-P., Chung, C.-C., Huisman, J. A., Tang, S.-H., 2008. Clarification and calibration of reflection coefficient for electrical conductivity measurement by time domain reflectometry. *Soil Sci. Soc. Am. J.* 72: 1033-1040.
- [9] Meissner, T., Wentz, F. J., Sept. 2004. The complex dielectric constant of pure and sea water from microwave satellite observations. *Geoscie. Remote Sensing, IEEE Transactions.* 42: 1836-1849.
- [10] Moret, D., López, M. V., Arrué, J. L., 2004. TDR application for automated water level measurement from mariotte reservoirs in tension disc infiltrometers. *J. Hydrol.* 297: 229 -235.
- [11] Moret-Fernández, D., Vicente, J., Lera, F., Latorre, B., López, M.V., Blanco, N., González-Cebollada, C., Arrué, J.L., Gracia, R., Salvador, M.J., Bielsa, A., 2010. TDR-Lab Version 1.0 Users Guide (<http://digital.csic.es/handle/10261/35790>).
- [12] Moret-Fernández, D., Vicente, J., Aragüés, R., Peña, C., López, M.V. 2012. A new TDR probe for measurements of soil solution electrical conductivity. *Journal of Hydrology* 448-449, 73-79.
- [13] Or, D., Wraith, J. M., 1999. A new soil metric potential sensor based on time domain reflectometry. *Water Resour. Res.* 35: 3399-3407.
- [14] Ramo, S., Whinnery, J., Van Duzer, T., 1984. *Fields and waves in communication electronics*. John Wiley and Sons, New York.
- [15] Topp, G.C., and T.P.A. Ferre, 2002. Water content, In, *Methods of Soil Analysis*. Part 4. (Ed. J.H. Dane and G.C. Topp), SSSA Book Series No. 5. *Soil Science Society of America*, Madison WI.

Anexo G

G. Resumen Tablas utilizadas para calcular rendimientos de la aplicación

	1 carga	2 carga	3 carga	4 carga	5 carga
BBDD original	4.4432541	0.2530145	0.2550146	0.2860163	0.2980170
BBDD completa	6.1432942	0.3501339	0.4680269	0.5570319	0.4930282

Tabla 1. Resultados obtenidos con un Intel Core i7-2600, 3.40GHz, 8GB RAM

	1 carga	2 carga	3 carga	4 carga	5 carga
BBDD original	28.3593269	0.7343750	0.7656237	0.7968124	0.7952112
BBDD completa	92.1718784	2.7954127	2.2489657	2.5691457	2.9157954

Tabla 2. Resultados obtenidos con un Pentium4, 3GHz, 1GB RAM

	1 carga	2 carga	3 carga	4 carga	5 carga
BBDD original	100.1649879	0.768797	0.468789	0.46549879	0.898794
BBDD completa	120.3154879	2.2144987	2.119871	2.1246879	2.454987

Tabla 3. Resultados obtenidos Tablet Toshiba, 1Ghz y 512MB RAM

	1 tabla/fichero	6 tablas/fichero	12 tablas/fichero	18 tablas/fichero	24 tablas/fichero
BBDD original	5.1457842	5.578497	6.124879	10.1248745	18.184784
BBDD completa	5.312487	6.032154	6.874512	13.215478	25.021487

Tabla 4. Comparación de resultados para distinta cantidad de ficheros XML.

	1 carga	2 carga	3 carga	4 carga	5 carga
BBDD original	1.549764	1.464871	1.687487	1.4315797	1.5474147
BBDD completa	5.2544341	5.2305145	4.9300146	5.2601863	5.2980170

Tabla 5. Resultados XML obtenidos con un Intel Core i7-2600, 3.40GHz, 8GB RAM

	1 carga	2 carga	3 carga	4 carga	5 carga
BBDD original	3.4853187	3.4487519	3.6154785	3.4418764	3.1577411
BBDD completa	8.3457874	8.1267487	8.1547874	8.1315748	8.4214487

Tabla 6. Resultados XML obtenidos con un Pentium4, 3GHz, 1GB RAM

	1 carga	2 carga	3 carga	4 carga	5 carga
BBDD original	5.1457974	5.464871	5.4578788	5.5157894	5.5447874
BBDD completa	8.2544341	8.2646811	8.112457	8.8745478	8.4648741

Tabla 7. Resultados XML obtenidos Tablet Toshiba, 1Ghz y 512MB RAM

Anexo H

H. Diagrama de GANTT de los tiempos empleados en el PFC.

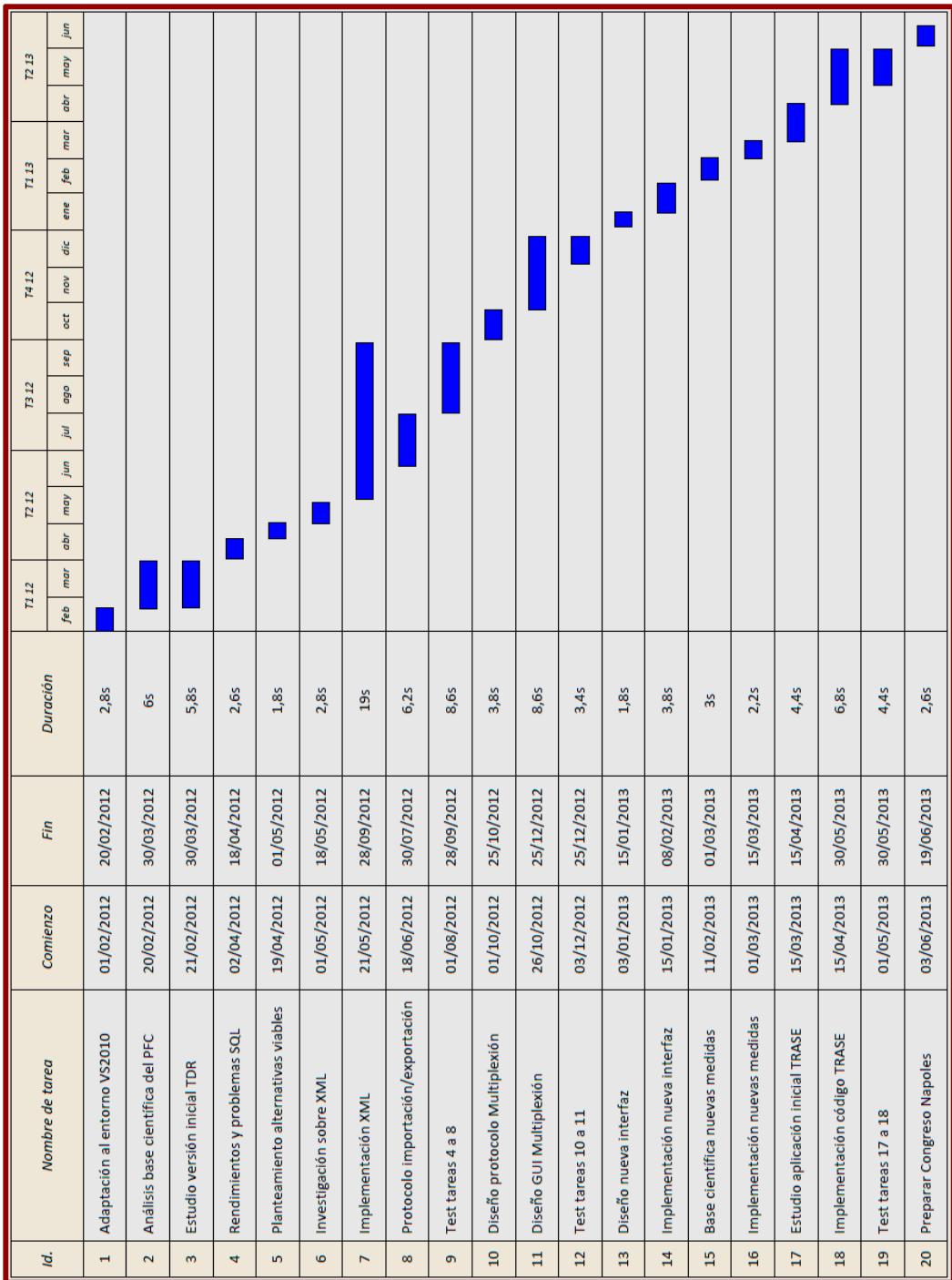


Figura 57. Gráfico de Gantt de las horas realizadas.