



DATATEC API – DAPI
Configuración general

Sistema de Mercados Financieros

Update 3.023

Colombia

14-sep-2026
ICAP del Ecuador
support@datatec-ec.com

INDICE

1. Plataforma donde se puede ejecutar el DAPI.....	3
1.1. Plataformas de desarrollo.....	3
1.2. Plataformas de instalación.....	3
1.3. Requerimientos mínimos de hardware para uso del DAPI	3
2. Códigos de ejemplo del DAPI	3
3. Configuración del DAPI en el DMCliente	4
4. Configuración del cliente DAPI (dapi.cfg)	4
4.1. Temporal Configuración Archivo (dapi_tmp_previous_date_log_request.txt)	7
4.2. Lectura de archivo dapi.log generado en otro usuario	8
5. Instalación del DAPI en Windows	10
5.1. Instalación del control ActiveX (OCX).....	10
5.1.1. Revisión de la versión DAPI en el archivo OCX	11
5.2. Ejemplo de uso del DAPI en Windows - VbdxApiTest.....	11
5.2.1. Nota para equipos Windows 64-bits	11
No se requiere instalar el paquete MS Visual C++ 2019 Redistributable Package (x64). Ejemplo Visual Basic del uso del DAPI.	11
5.2.2. Ejemplo Visual Basic del uso del DAPI	11
5.3. Instalación y uso de la librería DAPI de Windows.....	12
5.4. Ejemplo de uso de la librería DAPI en Windows - Drvapitest.exe.....	13
5.4.1. Ejemplo C++ del uso de la librería DAPI.....	13
6. Instalación del DAPI en Linux.....	14
6.1. Configuración en Linux.....	15
6.2. Instalación y uso de la librería DAPI en Linux	15
6.3. Contenido del directorio de la aplicación de ejemplo y de la librería	15
6.4. Ejemplo de uso del DAPI en Linux	16

1. Plataforma donde se puede ejecutar el DAPI

1.1. Plataformas de desarrollo

El DAPI está disponible para las siguientes plataformas de desarrollo:

1. ActiveX Control de Windows (OCX): Compilado usando Microsoft Visual Studio 2019. Este control es compatible con aplicaciones desarrolladas en Microsoft Visual Studio, por ejemplo, Visual Basic, C#, C++, J#, etc.
2. Librería de Windows (LIB): Compilado usando Microsoft Visual Studio 2019. Esta librería es compatible con aplicaciones desarrolladas usando Microsoft Visual Studio usando lenguaje c++.
3. Librería estática de Linux (.a) y (.so)

NOTA: No se utiliza ssl para el dapi. Se recomienda habilitar una conexión privada, vpn, o dssl para garantizar la privacidad de la comunicación.

1.2. Plataformas de instalación

El DAPI como ActiveX control (OCX) corre en los siguientes sistemas operativos.

- Microsoft Windows 10 - 11 (64 bits)

El DAPI como librería, se puede usar en los siguientes sistemas operativos:

- Linux (64 bits)
- Windows (64 bits)

1.3. Requerimientos mínimos de hardware para uso del DAPI

La aplicación DAPI actúa como un Usuario Primario del sistema Datatec y tiene los mismos requerimientos de hardware. Se recomienda:

- Procesador: Dual core o superior.
- Memoria: 8 GB o superior.
- Disco duro: Al menos 500 GB disponibles.

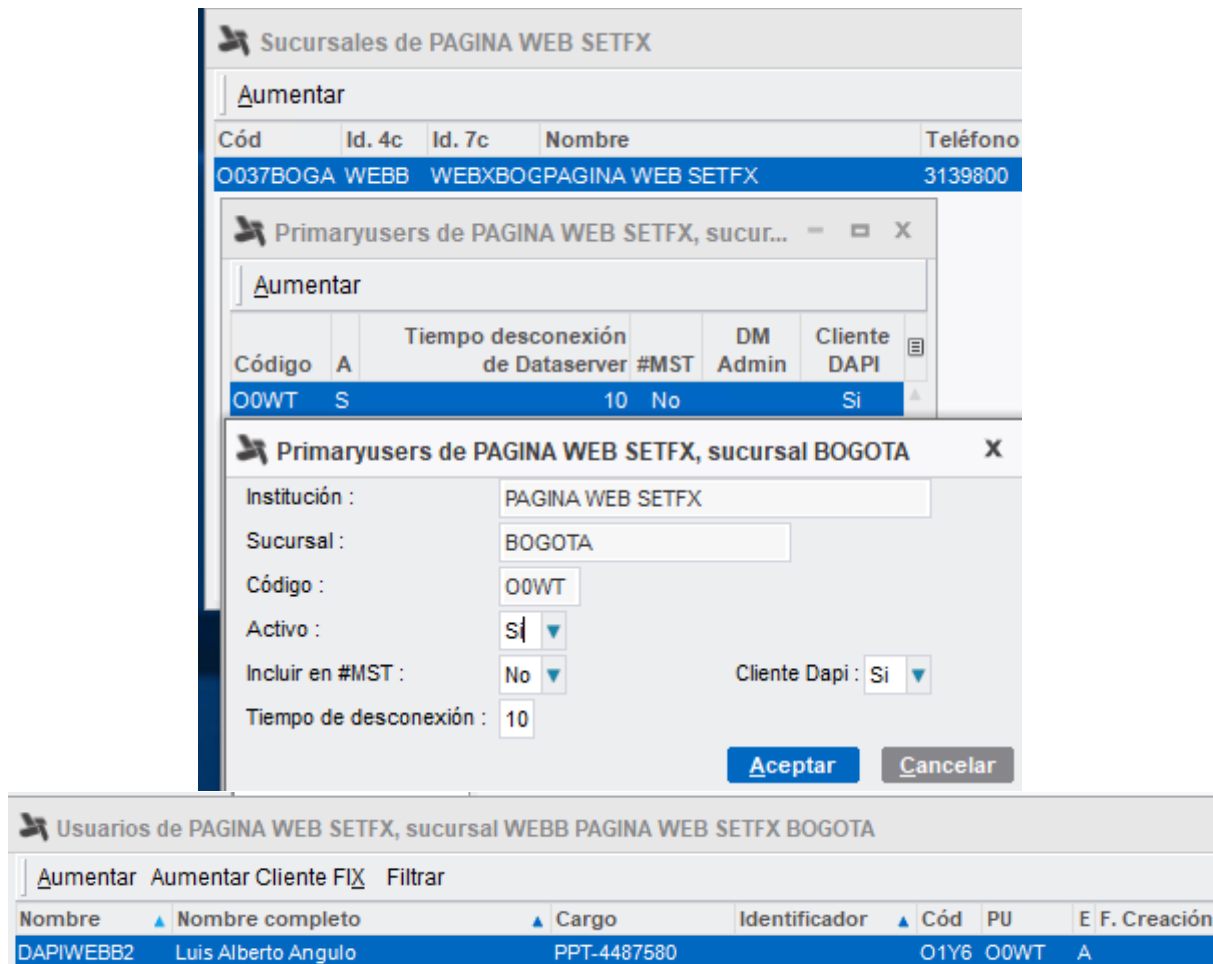
2. Códigos de ejemplo del DAPI

Con el kit de desarrollo se proveen dos códigos fuente que se dan como ejemplo del uso del DAPI:

1. **VbdxApiTest**. Aplicación de ejemplo en Visual Basic usando el ActiveX control del DAPI para Windows 32/64 bits.
2. **Drvapitest**. Aplicación tipo consola de ejemplo en C++ usando la librería del DAPI para Linux y Windows (32/64 bits).

3. Configuración del DAPI en el DMCliente

El administrador del sistema debe configurar el usuario tipo DAPI usando la aplicación DMCliente. Para ejecutar esta tarea, debe seleccionar en la ventana de Fichero de Maestros la institución y sucursal a la cual pertenecerá este nuevo usuario DAPI para crear un nuevo Primaryuser (PU). El usuario DAPI tiene una propiedad especial llamada Cliente Dapi que debe ser configurada con Si.



The screenshot shows the DMCliente application interface. It consists of three overlapping windows:

- Sucursales de PAGINA WEB SETFX:** A table with columns: Cód, Id. 4c, Id. 7c, Nombre, Teléfono. It contains one row: 0037BOGA WEBB, WEBXBOGPAGINA WEB SETFX, 3139800.
- Primaryusers de PAGINA WEB SETFX, sucursal...:** A table with columns: Código, A, Tiempo desconexión de Dataserver, #MST, DM Admin, Cliente DAPI. It contains one row: 00WT, S, 10, No, Si.
- Primaryusers de PAGINA WEB SETFX, sucursal BOGOTA:** A form with fields:
 - Institución: PAGINA WEB SETFX
 - Sucursal: BOGOTA
 - Código: 00WT
 - Activo: Si (dropdown)
 - Incluir en #MST: No (dropdown)
 - Tiempo de desconexión: 10
 - Cliente Dapi: Si (dropdown)

Buttons 'Aceptar' and 'Cancelar' are visible at the bottom of the third window.

Below the windows, there is a table titled 'Usuarios de PAGINA WEB SETFX, sucursal WEBB PAGINA WEB SETFX BOGOTA' with columns: Nombre, Nombre completo, Cargo, Identificador, Cód, PU, E F. Creación. It contains one row: DAPIWEBB2, Luis Alberto Angulo, PPT-4487580, O1Y6, 00WT, A.

Una vez que el PU ha sido creado, proceder a agregar el nuevo usuario DAPI. No hay necesidad de asignar Accesos de mercado a este usuario. La única restricción es que este usuario debe utilizar el PU especial que se creó anteriormente.

4. Configuración del cliente DAPI (dapi.cfg)

El administrador del sistema debe crear en el DMCliente al DAPI como un usuario independiente. Además, debe proveer al usuario de esta interfaz el archivo DAPI.cfg, que contiene los siguientes datos: nombre de usuario, contraseña, código de sucursal, código de usuario, entre otros datos. A continuación, se muestra un archivo DAPI.cfg de ejemplo. El carácter '#' se utiliza para marcar el inicio de un comentario.

```

dapi.cfg @ D:\CO\DAPI\3022\Bolsa_DAPIWEBB_vb\vbdxapitest\bin64\
File Edit Search AutoText View Help
-----
# Datatec Api (DAPI) sample configuration file.
# -----

oper_file_version = "DAPI WEB DAPIWEBB"
data_server       = $03B
ds_ip_address     = 192.168.135.12
ds_tcp_port       = 2016
# PrimaryUser code is the local_code value.
local_code        = 0247
user_code         = 037N
branch_code       = WEBBBOGA
user_name         = DAPIWEBB
password          = datatecl

# If another dataserver is defined to connect the DAPI application
# it is necessary to specify the name, ip and port.
#data_server      = $02N
#ds_ip_address    = 192.168.121.103
#ds_tcp_port      = 2008

# If there were more dataservers in the environment, their names
# must be specified even if the DAPI application is not connected to them.
#data_server      = $03N
#data_server      = $04N

# transaction_feed token is used to get the "Orden_Transaccion" messages (111, 121).
# These messages are got in high/low messages log and it is the most common
# feed used.
transaction_feed = 1

# language options. SPANISH, ENGLISH
#language         = SPANISH

# Delay in milliseconds when DAPI sends messages to Dataserver to prevent
# an involuntary feed that causes problems in the environment.
Send_Msg_Delay_Millisec = 500

# The next directives can be used to read the data from a testing file,
# indicating a range of date/time for the messages to be read, and also the
# the name of the output log file for this test reading.
#testing_file     = dapi_bbv.log
#testing_start    = 20090710091500
#testing_end      = 20090710092000
#testing_output_file = test_dapi_output.log

```

A continuación, se muestra la lista de declaraciones permitidas en el archivo DAPI.cfg:

Declarativa	Ejemplo	Descripción
oper_file_version	Transaction Feed API	Versión
local_code	G0OC	Código de PrimaryUser asignado al DAPI por el Administrador del Sistema.
user_code	G0IM	Código de usuario asignado por el Administrador del Sistema. Este código es diferente al código de PrimaryUser.
branch_code	G0JKUIOA	Código de sucursal de usuario.
data_server	\$02Q	Código del equipo DataServer. Nota: Si otros Dataservers son definidos para que el DAPI se conecte, se debe especificar su nombre, IP y puerto (ds_ip_address y ds_tcp_port).

		Nota: Se deben definir tantos DataServer como cuantos están corriendo en el Sistema de Datatec, para que el DAPI reciba los mensajes originados en todos los Dataserver.
ds_ip_address	192.168.112.22	Dirección ip del equipo DataServer previamente definido con la directiva data_server.
ds_tcp_port	2008	Puerto tcp del equipo DataServer previamente definido con la directiva data_server. data_server está escuchando. Cuando se conecta una aplicación vía Internet, el puerto es usualmente el 443.
transaction_feed	1	0 – No recibir mensajes de transacciones realizadas en el sistema 1 - Recibir los mensajes de las transacciones realizadas en el sistema. Nota: Los parámetros guibos_feed y transaction_feed no deben ser especificados simultáneamente. Solo uno puede ser definido con valor 1.
testing_file	dapis.log	Indica el nombre del archivo de log que se utilizará para retroalimentar el DAPI. Esta opción permite probar la funcionalidad del DAPI mediante un conjunto de mensajes intercambiados en de una conexión real. De esta forma el DAPI puede recibir mensajes sin tener que estar conectado a un ambiente de pruebas. Mediante esta opción es posible además depurar posibles errores encontrados en una ejecución anterior de la que se tenga el archivo de log.
testing_start	20091215091500	Especifica la fecha (AAAAMMDD) y hora (hhmmss) de inicio desde la cual el DAPI empieza a leer el archivo testing_file.
testing_end	20091215091500	Especifica la fecha (AAAAMMDD) y hora (hhmmss) en que el DAPI termina la ejecución de la prueba.
testing_output_file	Test_dapi_output.log	Especifica el nombre del archivo de salida de la prueba.
language	spanish	Indica el idioma que se emplea en los tipos de los mensajes que se pasan a la aplicación. Si esta directiva no es definida se utilizarán los nombres antiguos. Se define en minúscula. Si se define language=spanish, se utilizará el tipo de mensaje en español. <u>Ejemplo:</u> mensaje_trans_moneda Si se define language=english, se utilizará el tipo de mensaje en inglés. <u>Ejemplo:</u> currency_trade_message

NOTA: Todos los DataServers disponibles en el sistema a los cuales el DAPI se conectará deben incluirse en el archivo *DAPI.cfg*.

Cada DataServer debe ser identificado por un conjunto de tres parámetros:

- data_server
- ds_ip_address
- ds_tcp_port

Con todos los DataSevers configurados, se permitirá que la aplicación DAPI pueda conectarse al sistema con otro DataServer en caso de que pierda conexión con el primero. De esta manera, reduce el tiempo fuera de servicio. De la misma manera, si hay DataServers adicionales en el sistema sus nombres deben ser configurados aun cuando el DAPI no se conecte a ellos.

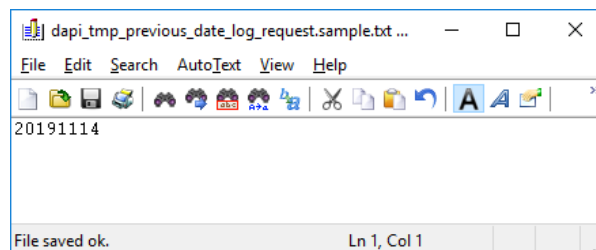
En las aplicaciones de ejemplo distribuidas para plataformas soportadas se incluye un archivo de ejemplo *dapi.cfg.sample*.

4.1. Temporal Configuración Archivo (dapi_tmp_previous_date_log_request.txt)

De forma extraordinaria se puede utilizar el archivo temporal *dapi_tmp_previous_date_log_request.txt*, para indicar al dapi que solicite la retransmisión de todos los mensajes de un día definido en el pasado. Tomar en cuenta que en el DataServer debe existir el directorio con los datos de la fecha a solicitar.

Este archivo debe crearse en el mismo directorio de *dapi.cfg*. Para evitar futuras retransmisiones, el archivo se elimina después de ser leído.

La fecha requerida debe definirse en los primeros caracteres de la primera línea y en formato YYYYMMDD, si hay algún error al leer o eliminar el archivo, ó con el reconocimiento de la fecha, se devuelve error y no se continúa con el proceso normal del DAPI.



Los posibles errores son:

Error	Code	Description
DAPI_ERR_INVALID_DATE	27	Invalidate date in the tmp config file
DAPI_ERR_CANNOT_DELETE_TMP_CONFIG	28	Can not delete tmp config file
DAPI_ERR_READING_TMP_CONFIG	25	Reading temporal config file
DAPI_ERR_EMPTY_TMP_CONFIG	26	Empty temporal config file

4.2. Lectura de archivo dapi.log generado en otro usuario

Por motivo de pruebas o validaciones específicas, se puede leer el contenido de un archivo dapi.log generado en otro usuario.

En el archivo dapi.cfg en la última sección se tiene la directiva #testing_file

```

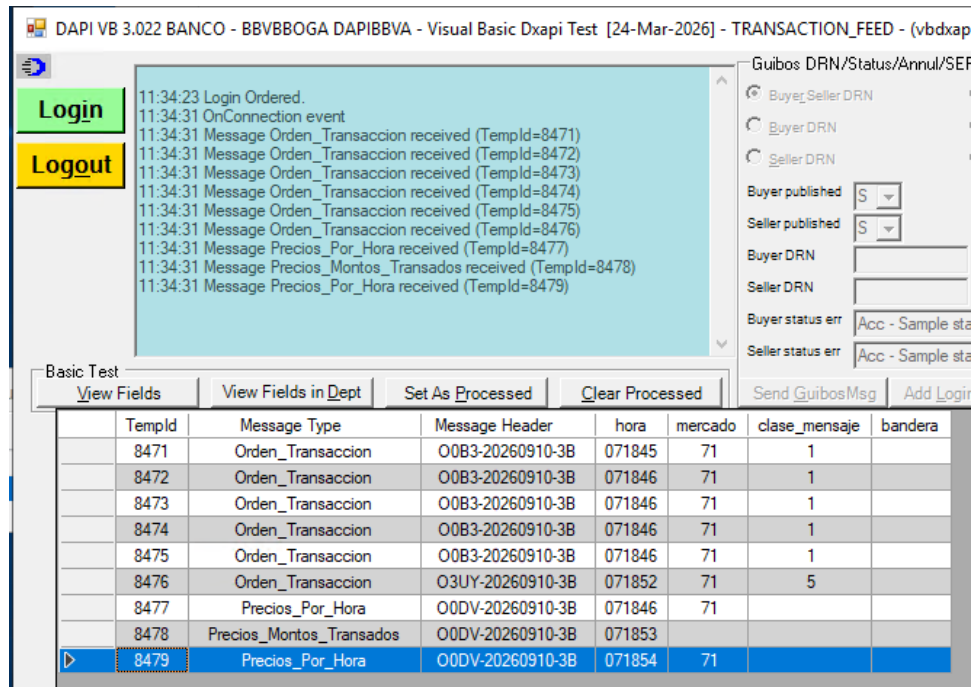
dapi.cfg @ D:\CO\DAPI\3022\Bolsa_DAPIWEBB_vb\vbdxapitest\bin64\
File Edit Search AutoText View Help
# The next directives can be used to read the data from a testing file,
# indicating a range of date/time for the messages to be read, and also the
# the name of the output log file for this test reading.
#testing_file = dapi_ori.log
#testing_start = 20090710091500
#testing_end = 20090710092000
#testing_output_file = test_dapi_output.log

```

Aquí se debe retirar el comentario en la directiva testing_file =, y luego se debe colocar el nombre del archivo que se desea leer. NUNCA debe llamarse dapi.log, se recomienda que se ponga un nombre que identifique donde originalmente se creó el archivo.

Ejemplo:

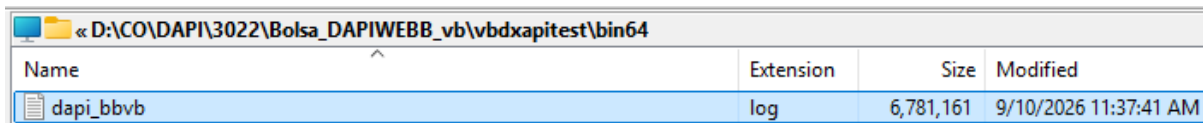
DAPI BBVB generó el siguiente dapi.log



TempId	Message Type	Message Header	hora	mercado	clase_mensaje	bandera
8471	Orden_Transaccion	O0B3-20260910-3B	071845	71	1	
8472	Orden_Transaccion	O0B3-20260910-3B	071846	71	1	
8473	Orden_Transaccion	O0B3-20260910-3B	071846	71	1	
8474	Orden_Transaccion	O0B3-20260910-3B	071846	71	1	
8475	Orden_Transaccion	O0B3-20260910-3B	071846	71	1	
8476	Orden_Transaccion	O3UY-20260910-3B	071852	71	5	
8477	Precios_Por_Hora	O0DV-20260910-3B	071846	71		
8478	Precios_Montos_Transados	O0DV-20260910-3B	071853			
8479	Precios_Por_Hora	O0DV-20260910-3B	071854	71		

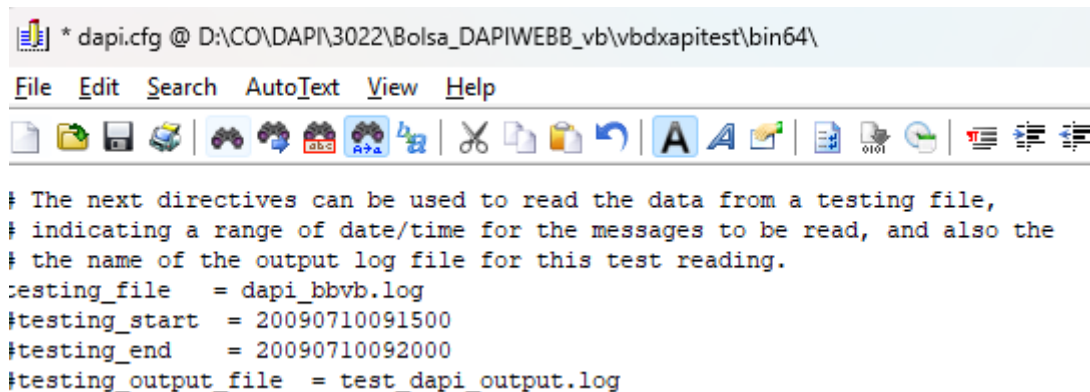
« D:\CO\DAPI\3022\Banco_BBV_vb\vbdxapitest\bin64				
Name	Extensi...	Size	Attrib...	Modified
AxInterop.dxapiLib.dll	dll	11,264	---A-...	3/24/2026 3:54:33 PM
dapi	log	6,780,968	---A-...	9/10/2026 11:34:31 AM

Desde DAPI WEB se quiere leer el contenido del dapi.log de BBVB, para esto se debe renombrar el log, por ejemplo, como dapi_bbvb.log; luego copiarlo a la misma ruta donde se ejecuta el DAPI WEB.



Name	Extension	Size	Modified
dapi_bbvb	log	6,781,161	9/10/2026 11:37:41 AM

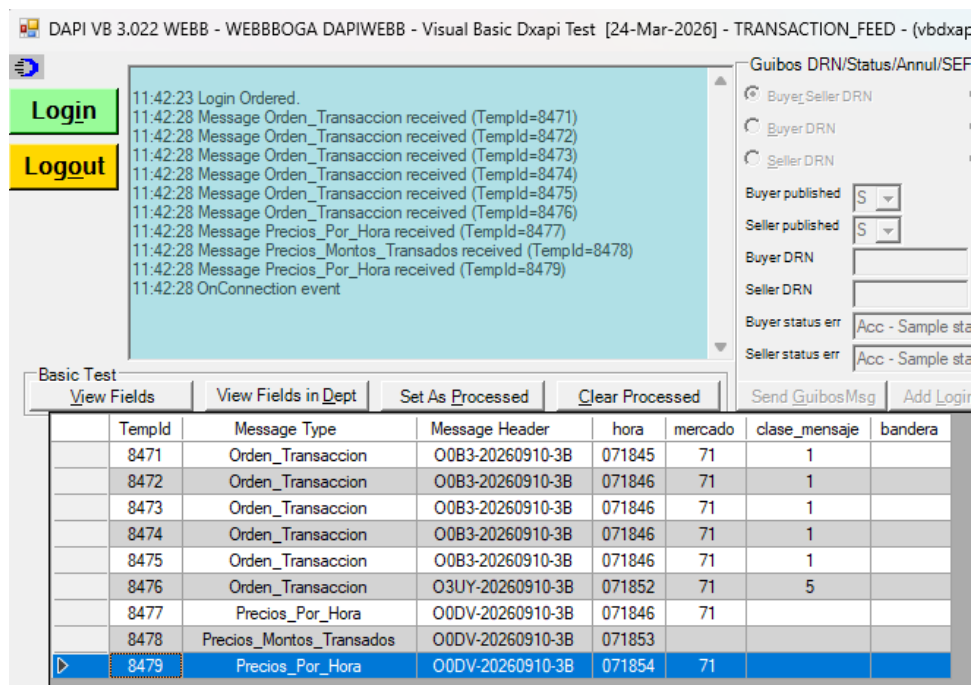
Editar el archivo dapi.cfg y en la directiva testing_file, colocar el nombre del archivo que se va a leer; en esta caso dapi_bbvb.log.



```
* dapi.cfg @ D:\CO\DAPI\3022\Bolsa_DAPIWEBB_vb\vbdxapitest\bin64\

# The next directives can be used to read the data from a testing file,
# indicating a range of date/time for the messages to be read, and also the
# the name of the output log file for this test reading.
testing_file      = dapi_bbvb.log
#testing_start    = 20090710091500
#testing_end      = 20090710092000
#testing_output_file = test_dapi_output.log
```

Antes de ejecutar el DAPI WEB, se recomienda eliminar el archivo dapi.log. Cuando se ejecuta el DAPI WEB, va a leer la directiva y buscar el archivo dapi_bbvb.log. Cuando lo lea, mostrará en la ventana del DAPI los mismo datos que tenía originalmente BBVB



DAPI VB 3.022 WEBB - WEBBBOGA DAPIWEBB - Visual Basic Dxapi Test [24-Mar-2026] - TRANSACTION_FEED - (vbdxap

Guibos DRN/Status/Annul/SEF

Buyer/Seller DRN

Buyer DRN

Seller DRN

Buyer published S

Seller published S

Buyer DRN

Seller DRN

Buyer status err Acc - Sample sta

Seller status err Acc - Sample sta

Send GuibosMsg Add Login

Basic Test

View Fields View Fields in Dept Set As Processed Clear Processed

TempId	Message Type	Message Header	hora	mercado	clase_mensaje	bandera
8471	Orden_Transaccion	O0B3-20260910-3B	071845	71	1	
8472	Orden_Transaccion	O0B3-20260910-3B	071846	71	1	
8473	Orden_Transaccion	O0B3-20260910-3B	071846	71	1	
8474	Orden_Transaccion	O0B3-20260910-3B	071846	71	1	
8475	Orden_Transaccion	O0B3-20260910-3B	071846	71	1	
8476	Orden_Transaccion	O3UY-20260910-3B	071852	71	5	
8477	Precios_Por_Hora	O0DV-20260910-3B	071846	71		
8478	Precios_Montos_Transados	O0DV-20260910-3B	071853			
8479	Precios_Por_Hora	O0DV-20260910-3B	071854	71		

5. Instalación del DAPI en Windows

La distribución del DAPI para Windows contiene los siguientes archivos:

- **windows_64\ActiveX\AxdxapiLib.dll.**
- **windows_64\ActiveX\dxapi.ocx.**
- **windows_64\ActiveX\Interop.dxapiLib.dll.**
- **windows_64\lib\dcapi.lib.**
- **windows_64\lib\dxapi.lib.**
- **sample_windows\vbdxapitest** (app ejemplo en VB usando el control OCX para 32 y 64 bits).
- **sample_windows\vbdxapitest\bin64** (están los dll, ejecutable y archivos de configuración de ejemplo para 64 bits).

El archivo dxapi.ocx, que se incorpora como un objeto COM también incluye todas sus dependencias. Existen múltiples librerías que, dependiendo de la aplicación c++ que se genera, serán requeridas por Visual Studio 2019, por ejemplo:

- USER32.dll
- GDI32.dll
- MSIMG32.dll
- WINSPOOL.DRV
- ADVAPI32.dll
- SHELL32.dll
- SHLWAPI.dll
- UxTheme.dll
- ole32.dll
- OLEAUT32.dll
- gdiplus.dll
- OLEACC.dll
- IMM32.dll
- WINMM.dll
- WS2_32.dll

5.1. Instalación del control ActiveX (OCX)

El DAPI ActiveX control se distribuye con los siguientes archivos:

1. El archivo **dxapi.ocx**
2. Un archivo de configuración **dapi.cfg**
3. Un archivo temporal de configuración **dapi_tmp_previous_date_log_request.txt**

Si ya se ha instalado una versión anterior del DAPI en el equipo, antes de realizar la instalación de la nueva versión, RETIRE la instalación antigua ubicándose en el directorio donde se encuentra el archivo anterior **dxapi.ocx** y ejecute la siguiente línea desde **\windows\sysWOW64**.

Regsvr32 /u dxapi.ocx

Copie el nuevo **dxapi.ocx** al disco duro del equipo de destino y registre el ActiveX control usando la siguiente línea de comando:

Regsvr32 dxapi.ocx

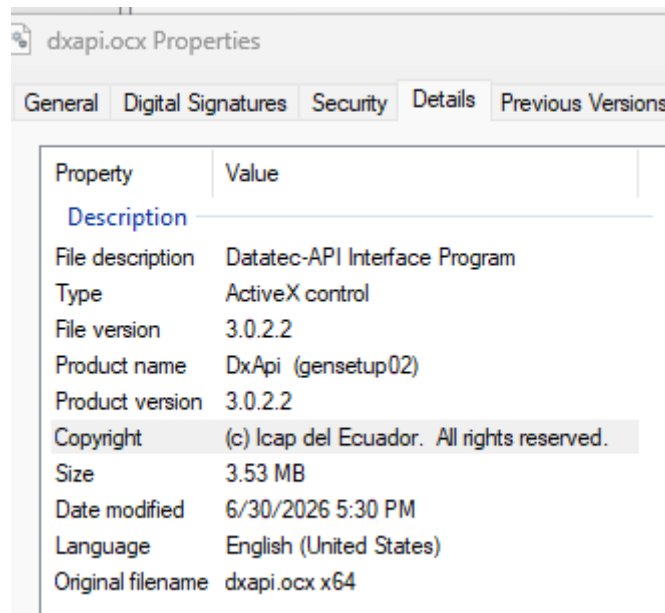
Esto instala el ActiveX control en el equipo y puede ser usado por la aplicación del usuario generada con MS Visual Studio.

A continuación, configure los parámetros del archivo **dapi.cfg** de acuerdo al ambiente y Sistema de Datatec donde se corre la aplicación del usuario.

Finalmente, hay que incluir en la aplicación el código necesario para interactuar con el DAPI. Usualmente se requiere incluir el ActiveX control dentro de una ventana (por ejemplo: dialog application), crear una variable de este control y redefinir los eventos necesarios (principalmente el evento *OnMessageReceived*, esto será discutido más adelante). Si se requiere enviar mensajes al sistema de Datatec se deben llamar a los métodos que la clase del DAPI proporciona para este propósito con la misma variable antes creada.

5.1.1. Revisión de la versión DAPI en el archivo OCX

Es posible revisar la versión del DAPI en el archivo OCX. Para esto, **clic-derecho** en el archivo, opción **Propiedades**, escoger la **viñeta Details** en Windows.



5.2. Ejemplo de uso del DAPI en Windows - VbDxApiTest

5.2.1. Nota para equipos Windows 64-bits

No se requiere instalar el paquete MS Visual C++ 2019 Redistributable Package (x64). Ejemplo Visual Basic del uso del DAPI.

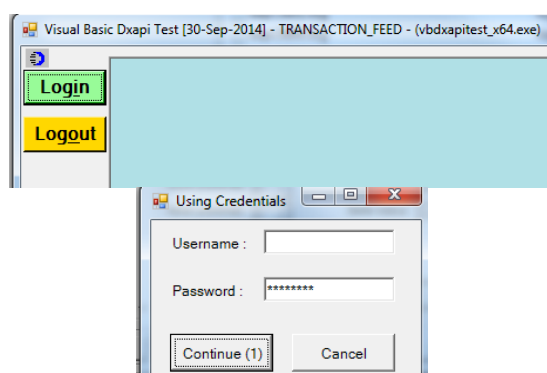
5.2.2. Ejemplo Visual Basic del uso del DAPI

Para ilustrar el uso del DAPI y sus funciones se suministra un fácil y sencillo ejemplo programado en VB. Se encuentra en la carpeta **sample_windowsvbDxapitest** de la distribución del DAPI. Para usarlo copie el contenido del directorio en el equipo donde desea correr el ejemplo y siga los siguientes pasos:

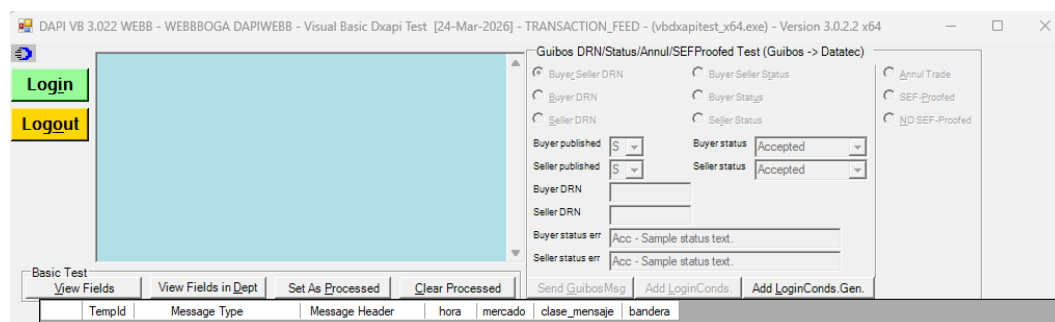
1. Configure el usuario DAPI usando el DataManager Cliente.

2. Actualice el archivo de configuración dapi.cfg con la información del usuario DAPI y los DataServers a los que puede conectarse.
3. Asegúrese que el archivo *dapi.cfg* se encuentre en el mismo directorio donde se ejecutará la aplicación de ejemplo *vbdxapitest*.
4. Asegúrese que el archivo *dapi_tmp_previous_date_log_request.txt* se encuentre en el mismo directorio donde se ejecutará la aplicación de ejemplo *vbdxapitest*.
5. El directorio desde donde se ejecutará la aplicación de ejemplo debe estar ubicado en un disco local, ya que ésta **no se ejecuta correctamente en unidades de red**.
6. El control ActiveX *dxapi.ocx* debe estar correctamente registrado en el equipo (véase la sección Instalación del DAPI en Windows).
7. Los ejecutables *vbdxapitest.exe* y *vbdxapitest_64.exe*, se encuentran bajo *vbdxapitest\bin32* y *vbdxapitest\bin64* para 32 y 64 bits respectivamente

Ejecute el archivo *vbdxapitest.exe*, debe presionar el botón **Login** y se puede ignorar los datos de usuario y clave que se solicita.



La siguiente imagen muestra un ejemplo de la ventana luego de un ingreso exitoso al sistema.



En este ejemplo el control ActiveX (*dxapi.ocx*) se incluye en la plantilla (*Form1*). El nombre que tiene este control en la aplicación de ejemplo es: *Axdxapi1*.

5.3. Instalación y uso de la librería DAPI de Windows

En caso de usar la librería del DAPI para Windows, copiar el directorio *sample_windows\drvapitest* al proyecto deseado. En nuevas versiones actualizar las librerías correspondientes al sistema operativo (Windows) y compilador (32/64 bits) desde los directorios **windows*Lib** a los subdirectorios lib correspondientes en *drvapitest*

Finalmente ajustar el proceso de mensajes emulando *drvapitest.cpp* de acuerdo con el requerimiento particular. Revisar y codificar los métodos definidos en el archivo *drvapi.cpp*, especialmente aquellos

métodos que atienden a los eventos que el DAPI genera durante su ejecución. De ser necesario se puede crear una nueva clase que tenga como base la clase CdcApi.

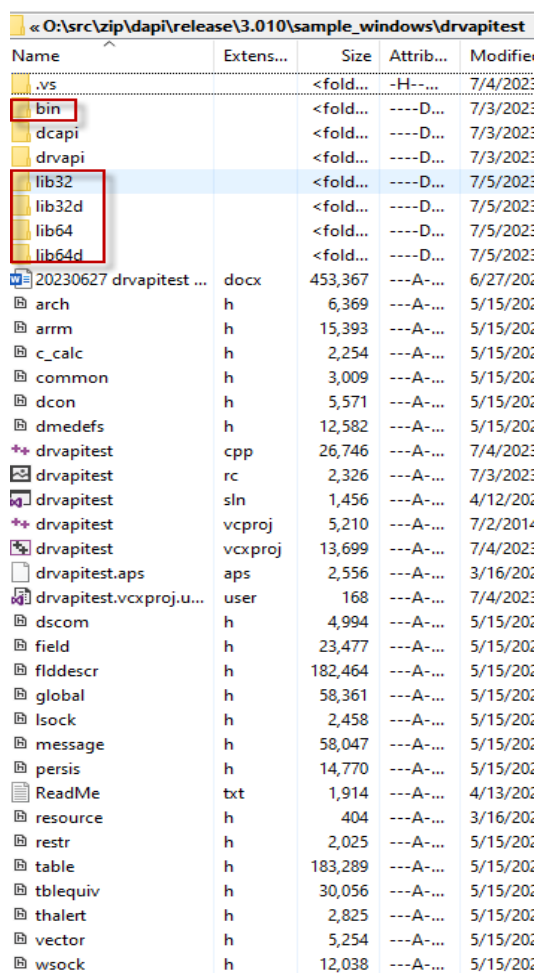
5.4. Ejemplo de uso de la librería DAPI en Windows - Drvapitest.exe

5.4.1. Ejemplo C++ del uso de la librería DAPI

Un programa drvapitest que normalmente se usa en Linux, también se lo ha generado para Windows, el programa ejemplo es de tipo consola. El archivo ejecutable se llama drvapitest.exe.

Se distribuye en la carpeta `\sample_windows\drvapitest\` de del DAPI. Bajo este directorio se incluyen las librerías para compilar y ejecutar las versiones *debug* y *release* en 32 o 64 bits. (lib32d, lib64d, lib32 y lib64). También se incluye el directorio `\sample_windows\drvapitest\bin` que contiene los archivos de configuración.

El contenido del directorio drvapitest es:



Name	Extens...	Size	Attrib...	Modifie...
.vs		<fold...	-H--...	7/4/202...
bin		<fold...	----	7/3/202...
dcapi		<fold...	----	7/3/202...
drvapi		<fold...	----	7/3/202...
lib32		<fold...	----	7/5/202...
lib32d		<fold...	----	7/5/202...
lib64		<fold...	----	7/5/202...
lib64d		<fold...	----	7/5/202...
20230627 drvapitest ...	docx	453,367	---A-...	6/27/20...
arch	h	6,369	---A-...	5/15/20...
arm	h	15,393	---A-...	5/15/20...
c_calc	h	2,254	---A-...	5/15/20...
common	h	3,009	---A-...	5/15/20...
dcon	h	5,571	---A-...	5/15/20...
dmedefs	h	12,582	---A-...	5/15/20...
drvapitest	cpp	26,746	---A-...	7/4/202...
drvapitest	rc	2,326	---A-...	7/3/202...
drvapitest	sln	1,456	---A-...	4/12/20...
drvapitest	vcproj	5,210	---A-...	7/2/201...
drvapitest	vcxproj	13,699	---A-...	7/4/202...
drvapitest.aps	aps	2,556	---A-...	3/16/20...
drvapitest.vcxproj.u...	user	168	---A-...	7/4/202...
dscom	h	4,994	---A-...	5/15/20...
field	h	23,477	---A-...	5/15/20...
flddescr	h	182,464	---A-...	5/15/20...
global	h	58,361	---A-...	5/15/20...
lsock	h	2,458	---A-...	5/15/20...
message	h	58,047	---A-...	5/15/20...
persis	h	14,770	---A-...	5/15/20...
ReadMe	txt	1,914	---A-...	4/13/20...
resource	h	404	---A-...	3/16/20...
restr	h	2,025	---A-...	5/15/20...
table	h	183,289	---A-...	5/15/20...
tblequiv	h	30,056	---A-...	5/15/20...
thalert	h	2,825	---A-...	5/15/20...
vector	h	5,254	---A-...	5/15/20...
wsock	h	12,038	---A-...	5/15/20...

En los directorios lib32 y lib32d, están las librerías dcapi.lib y crypt.lib *release* y *debug* respectivamente, compiladas para 32 bits.

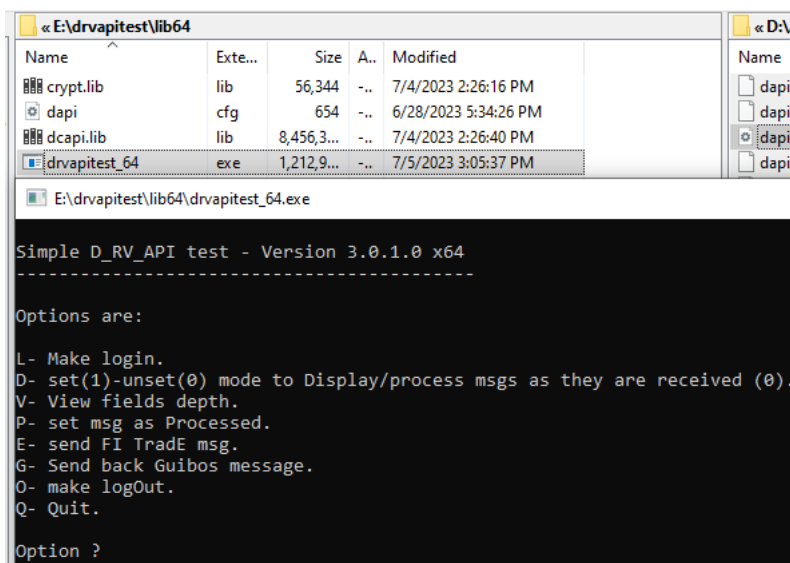
En los directorios lib64 y lib64d, están las librerías dcapi.lib y crypt.lib *release* y *debug* respectivamente, compiladas para 64 bits.

Con nuevas versiones, las librerías `dcapi.lib` y `crypt.lib` deben copiarse en `lib*`, según la plataforma requerida. Si desea regenerar el ejecutable abrir el archivo solución (.sln) usando Visual Studio (2019) y realice el proceso para la generación del ejecutable en configuración y plataforma requeridos. Si usa nuevas versiones de Visual Studio (por ejemplo 2022), al abrir el archivo .sln se preguntará si desea migrar la plataforma a la del VS, seleccione la que corresponda a fin de que pueda compilar el proyecto.

Para una prueba inicial copie el contenido del directorio `drvapitest` en el equipo donde desea correr el ejemplo y siga estos pasos:

1. Configure el usuario DAPI usando el DataManager Cliente.
2. Actualice el archivo de configuración `dapi.cfg` con la información del usuario DAPI y los DataServers a los que puede conectarse.
3. Asegúrese que el archivo `dapi.cfg` se encuentre en el mismo directorio donde se ejecutará la aplicación de ejemplo `drvapitest/lib32`, `drvapitest/lib64`, para ejecutar versiones release ó `drvapitest/lib32d`, `drvapitest/lib64d`, para versiones *debug*.
4. Asegúrese que el archivo `dapi_tmp_previous_date_log_request.txt` se encuentre en el mismo directorio donde se ejecutará la aplicación de ejemplo `drvapitest/lib*`.
5. Ejecute el programa.

Para ejecutar el `drvapitest.exe` o `drvapitest_x64.exe`, es suficiente con presionar doble clic del mouse, la ventana resultado es:



Las opciones listadas son las mismas que el programa `drvapitest` Linux.

6. Instalación del DAPI en Linux

La distribución para Linux contiene los siguientes archivos:

- `linux_64\lib64\libdapi.a`
- `linux_64\lib64\libdapi.so`
- `libstdc++.so.6`
- `sample_linux\linux7_64\lnxdrvapitest.zip` (librería estática .a y código de ejemplo desarrollado en C++).

- sample_linux\linux7_64\lnxdrvapitest_so.zip (librería dinámica .so y código de ejemplo desarrollado en C++).

6.1. Configuración en Linux

El DAPI para el sistema operativo Linux ha sido generado y probado en un ambiente con las siguientes características:

Para 64 bits en equipo Linux de 64 bits:

- Red Hat Enterprise Linux Server release 8 y 9 para 64 bits.
- **gcc (GCC) 8.5.0 en Linux 8**
- **gcc (GCC) 10.5.0 en Linux 9**

6.2. Instalación y uso de la librería DAPI en Linux

Luego de seleccionar la plataforma para el Dapi y desempaquetar el archivo `Inxdrvapitest.zip`; se obtiene el directorio **Inxdrvapitest** y bajo éste, una lista de otros directorios y archivos.

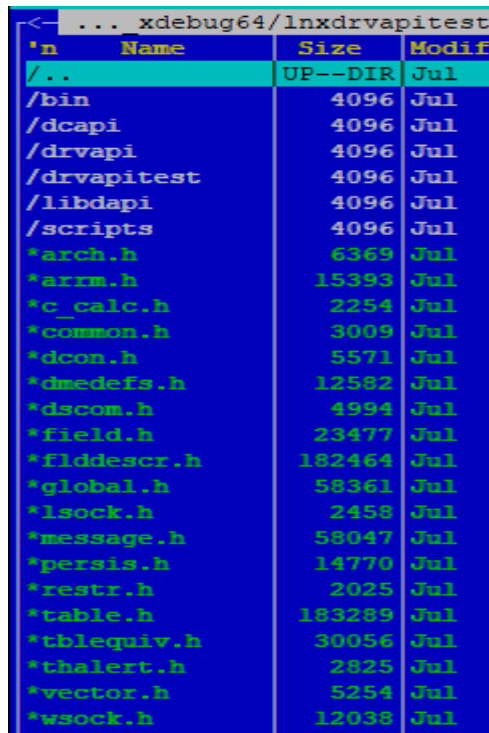
La librería del DAPI **libdapi.a** se distribuye compilada usando el compilador gcc de la versión indicada previamente. Por lo tanto, para usar los métodos del DAPI se debe enlazar la aplicación que se esté desarrollando con esta librería que se encuentra en: **Inxdrvapitest/libdapi/lib/libdapi.a**

El archivo header de la librería DAPI es `dcapi.h` y se encuentra en: **Inxdrvapitest/dcapi/dcapi.h**

Los mismos pasos descritos anteriormente se aplica para el archivo **Inxdrvapitest_so**, en este caso se utiliza la librería dinámica **libdapi.so**

6.3. Contenido del directorio de la aplicación de ejemplo y de la librería

Luego de seleccionar la plataforma para el Dapi y desempaquetar el archivo `Inxdrvapitest.zip` o `Inxdrvapitest_so.zip`; se tiene la siguiente estructura de directorios y archivos:



n	Name	Size	Modif
	UP--DIR		Jul
	/bin	4096	Jul
	/dcapi	4096	Jul
	/drvapi	4096	Jul
	/drvapitest	4096	Jul
	/libdapi	4096	Jul
	/scripts	4096	Jul
*	arch.h	6369	Jul
*	arm.h	15393	Jul
*	c_calc.h	2254	Jul
*	common.h	3009	Jul
*	dcon.h	5571	Jul
*	dmedefs.h	12582	Jul
*	dscom.h	4994	Jul
*	field.h	23477	Jul
*	flddescr.h	182464	Jul
*	global.h	58361	Jul
*	lsock.h	2458	Jul
*	message.h	58047	Jul
*	persis.h	14770	Jul
*	restr.h	2025	Jul
*	table.h	183289	Jul
*	tblequiv.h	30056	Jul
*	thalert.h	2825	Jul
*	vector.h	5254	Jul
*	wsock.h	12038	Jul

Figura 24: Estructura del DAPI en Linux

6.4. Ejemplo de uso del DAPI en Linux

La distribución del DAPI para Linux (de 32/64 bits) cuenta con un ejemplo de la utilización de la librería **libdapi.a/libdapi.so** mediante un ejecutable programado en C++ (**drvapitest**). Para que funcione correctamente esta aplicación debe asegurarse de cumplir el siguiente procedimiento:

1. Configure el usuario DAPI usando el DataManager Cliente.
2. Actualice el archivo de configuración **dapi.cfg** con la información del usuario DAPI y los DataServers a los que puede conectarse.
3. Asegúrese que el archivo **dapi.cfg** se encuentre en el mismo directorio donde se ejecutará la aplicación de ejemplo **drvapitest**.

El ejecutable de ejemplo del Dapi se encuentra bajo el directorio **lnxdrvapitest/drvapitest/xrelease32**, **lnxdrvapitest/drvapitest/xrelease64**, **lnxdrvapitest_so/drvapitest/xrelease32**, o **lnxdrvapitest_so/drvapitest/xrelease64**. Deberá entonces ubicarse en el directorio y ejecutar el archivo correspondiente de esta manera:

[myuser@ xrelease32]\$./drvapitest

```
Simple D_RV_API test
Options are:

L- Make login.
D- set(1)-unset(0) mode to Display/process msgs as they are received (0).
V- View fields depth.
P- set msg as Processed.
E- send FI Trade msg.
G- Send back Guibos message.
O- make logOut.
Q- Quit.

Option ?
Status changed 21, Connected
Connection established.
```