Hontza
Módulo Cliente
AVEQ

MANUAL DE DESARROLLO

Pamplona, 22 de Marzo 2006

Índice de contenido

El presente manual de desarrollo es un resumen de los principales elementos tecnológicos que componen el **módulo cliente** de la plataforma para la vigilancia competitiva desarrollada para AVEQ. Su objetivo es documentar la aplicación para su correcta comprensión técnica facilitando así su uso e integración en otras empresas.

1. Tecnología y lenguajes para el desarrollo del Módulo Cliente

Estos son los lenguajes de programación utilizados para su desarrollo:

a) Python como lenguaje de programación y GTK como entorno gráfico

El lenguaje de programación para el desarrollo ha sido Python, elegido por ser multiplataforma y por su gran versatilidad. Además, está muy bien acogido en la comunidad del Software Libre y su masivo uso en los últimos años lo convirtieron en la mejor opción para el desarrollo del Módulo Cliente.

A esto le sumamos las librerías gráficas GTK. Las cuales, de la misma manera que Python, son multiplataforma. Esto significa que uniendo ambos elementos (Python + GTK = PyGtk) solucionábamos la cuestión de "dibujar" ventanas para cualquier plataforma. De esta manera, el módulo cliente puede ser instaladado tanto en GNU/Linux, como Windows y otros sistemas operativos.

b) SoapPy como conector con los servicios web

SoapPy es la libería con la que gestionamos los servicios web preparados por Attest en el Módulo Servidor. Esta librería, al estar también desarrollada en Python, se integra de manera natural y efectiva con el resto de elementos que componen el Módulo Servidor.

c) SQLite como base de datos

SQLite es una pequeña librería que incluye un motor de base de datos que no precisa configuración. Es muy rápido, permite utilizar el lenguaje estandar SQL y destaca además por su versatilidad.

Aunque en la redacción del proyecto no se incluía un motor de base de datos para el Cliente, consideramos apropiado incluir una BBDD simple y ágil (como SQLite). La ventaja es que esta permite mayor eficiencia en la gestión de los datos enviados por el servidor, así como en todo lo relacionado con las búsquedas en el propio cliente.

d) Librería Py2exe para generar ejecutables en entornos Microsoft-Windows

Con esta librería, junto a otros elementos desarrollados por Investic, solucionamos uno de los principales problemas planteados al inicio del proyecto. Esto es, la ejecución nativa del módulo cliente en un entorno Windows. Esto significa que, utilizando esta libería el Módulo Cliente dispone de un ".exe" que una vez ejecutado arranca la aplicación.

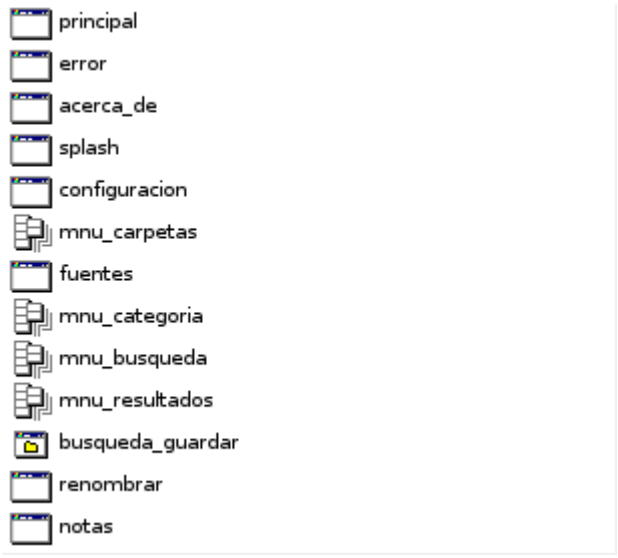
2. Descripción de los módulos que intervienen en el Módulo Cliente de Hontza

Estos son los módulos desarrollados por Investic para el Módulo Cliente de Hontza:

- **hontza.py**. Es el programa principal y está dedicado a la gestión de algunas señales (eventos), convirtiéndose así en el elemento aglutinador del resto de los módulos.
- **buscar.py**. Rutinas del marco (frame) buscar.
- **conexion.py**. Rutinas de conexión con el servidor (SOAP, WSDL...)
- **configuracion.py**. Gestión de la ventana de configuración del programa.
- **db.py**. Rutinas de acceso y manejo de la base de datos SQLite.
- **error.py**. Gestión de la ventana de errores.
- **fuentes.py**. Gestión de la ventana de selección de fuentes.
- **iconos.py.** Biblioteca gráfica que contiene iconos utilizados en la aplicación.
- **lst_busquedas.py**. Rutinas del marco de búsquedas.
- **lst_categoria.py.** Rutinas del marco de selección de categoría.
- **lst_resultados.py**. Rutinas del marco de resultados de las búsquedas.
- **splash.py.** Gestión de la ventana de arranque del programa (splash)
- **status.py.** Rutinas de manejo de la barra de estado de la ventana principal.
- **tre_busquedas.py.** Gestión y manejo del marco de carpetas de búsqueda

3. Definición y composición de la Interface de usuario (Glade)

La construcción de la estructura de la interface de usuario ha sido desarrollada utilizando Glade-2. En la siguiente imagen se reproduce dicha estructura:



Veamos con detalle los principales elementos que constituyen esta estructura:

- **PRINCIPAL**. Ventana donde se agrupan la mayoria de los elmentos del programa, los más importantes son:

  mnt_carpeta: Árbol para la gestión de carpetas.

  mnt_categorias: Lista para la visualización y gestión de las categorías.

  mnt_busqueda: Lista donde se gestionan las búsquedas.

  mnt_resultados: Lista que recoge los distintos resultados.

- **ERROR**. Esta ventana es la encargada de recoger los distintos errores, falta de conexión, problemas con las fuentes etc.
- **ACERCA_DE**. Ventana de información relativa al programa.
- **SPLASH**. Esta venta es la encargada de la entrada en el programa, informa del proceso en el que nos encontramos en cada momento, y una vez acabado da paso a la principal.
- **CONFIGURACIÓN**. Es la primera ventana en aparece y es la encargada de suministrar los datos necesarios para establecer la comunicacion con el módulo servidor.
- **FUENTE**. Esta pantalla permite la gestión de las fuentes, son dos listas que mediante check-box permiten selecionar las distintas fuentes de cada categoría.
- **BÚSQUEDA-GUARDAR**. Esta ventana es la encargada de guardar las distintas opciones que Hontza ofrece, por ejemplo los resultados obtenidos por una búsqueda.

- **RENOMBRAR**. Es la ventana encargada de renombrar las carpetas del arbol mnt_carpetas.
- **NOTAS.** Esta ventana es la encargada de poner las notas a los distintos resultados.

4. Descripción de la base de datos y modelo de entidad relación

Para la gestión de datos el cliente para la vigilacia competitiva Hontza hace uso de las librerías SQLite. Estas son las tablas que componen la base de datos:

categorias			
campo	tipo	default	otros
id	INTEGER		UNIQUE PRIMARY KEY
nombre	VARCHAR(100)	""	
seleccion	BOOLEAN	0	
activa	BOOLEAN	0	

fuentes			
campo	tipo	default	otros
id	INTEGER		UNIQUE PRIMARY KEY
nombre	VARCHAR(100)	""	
op_fe	BOOLEAN	0	
op_and	BOOLEAN	0	
op_or	BOOLEAN	0	
id_categoria	INTEGER	0	
seleccion	BOOLEAN	0	
activa	BOOLEAN	0	

busquedas			
campo	tipo	default	otros
id	INTEGER		UNIQUE PRIMARY KEY
texto	VARCHAR(100)	""	
op_fe	BOOLEAN	0	
op_and	BOOLEAN	0	
op_or	BOOLEAN	0	
id_categoria	INTEGER	0	
num_resultados	INTEGER	0	
fecha	DATE	""	
id_carpeta	INTEGER	0	
fuentes	VARCHAR(255)		

carpetas			
campo	tipo	default	otros
id	INTEGER		UNIQUE PRIMARY KEY
nombre	VARCHAR(50)	""	
pertenece	INTEGER	0	
abierta	BOOLEAN	0	

configuracion			
campo	tipo	default	otros
id_cliente	VARCHAR(50)	“”	
servidor	VARCHAR(255)	“”	

resultados			
campo	tipo	default	otros
id	INTEGER		UNIQUE PRIMARY KEY
titulo	VARCHAR(255)	“”	
texto	VARCHAR(255)	“”	
enlace	VARCHAR(255)	“”	
id_fuente	INTEGER	0	
idioma	VARCHAR(2)	“”	
id_busqueda	INTEGER	0	
icono	INTEGER	0	
checksum	VARCHAR(32)	“”	
fecha	DATE	“”	
nota	VARCHAR(255)	“”	

La relación entre las distintas tablas se puede ver por medio de los campos nombrados con id_"texto", donde "texto" indica la tabla con la que esta relacionada. Por ejemplo, la Tabla Fuentes está relacionada con Resultados. Dicha relación se puede comprobar porque esta última contiene el campo "id_fuente" refiriéndose así a la Tabla Fuentes.

El archivo *inicialización.sql* es el encargado de generar la estructura de la base de datos. Al hacer uso de SQLite, en este archivo se pueden ver sentencias SQL (las de toda la vida), lo cual hace mas fácil modificar y redefinir la base de datos.

Hay que tener en cuenta que el archivo *inicialización.sql*, una vez creada la estructura de la base de datos, inserta unos datos obligatorios que son los siguientes:

```
INSERT INTO carpetas (nombre, abierta, pertenece) VALUES ('Mis búsquedas', 1, 0)
INSERT INTO configuracion (id_cliente, servidor) VALUES ('', '')
```

La inserción en la Tabla Carpetas se realiza para tener por defecto una carpeta "Mis búsquedas", la cual no podrá ser borrada. Y en lo referente a la configuración, se insertan los campos vacíos para en caso de fallar la conexión poder entrar en la aplicación.

Todos los datos se guardan en el archivo *hontza.db*.

5. Crear ejecutable para windows

Pygtk es un lenguaje de programación multiplataforma, por lo tanto, si se tienen instalados Python y GTK en cualquier plataforma, no debería haber ningún problema en la ejecución de la aplicación. Nuestro propósito al elegir este conjunto de tecnologías era, y es, mostrar cómo es posible desarrollar una aplicación para usuarios no avanzados donde tanto la instalación como la ejecución no suponga ningún problema en cualquier plataforma.

Para generar un ejecutable para la plataforma privativa Windows XP hemos hecho uso de py2exe, el cual a su vez precisa de una serie de programas para poder generer el "punto exe". Esta es la relación de programas necesarios:

- python-2.4.1.msi
- gtk-win32-2.8.8-rc2.exe
- pygtk-2.8.2-1.win32-py2.4.exe
- py2exe-0.5.2.win32-py2.4.exe
- diffutils-2.8.7-1.exe
- patch-2.5.9-3.exe
- pysqlite-2.0.5.win32-py2.4.exe
- pyXML-0.8.4.win32-py2.4.exe
- SOAPpy-0.12.0.zip
- glade-2.6.0-rc1.exe

Una vez tengamos todos estos programas instalados, es preciso situarnos en el directorio donde tengamos el proyecto Hontza y donde están los diferente archivos python y ejecutar esta orden en la línea de comandos:

```
setup.py py2exe -O1 --packages encodings
```

El archivo *setup.py* contiene la descripción de los distintos elementos que necesita py2exe para poder generar el punto.exe. Y la orden –packages encodings sirve para que soporten correctamente las diferentes codificaciones.

Este es el contenido del archivo *setup.py*:

```
from distutils.core import setup
import py2exe
import glob

opts = {
"py2exe": {
```

```
"includes": ["pango", "atk", "gobject", "gtk"],
"dll_excludes": ["iconv.dll", "intl.dll",
"libatk-1.0-0.dll", "libgdk_pixbuf-2.0-0.dll",
"libgdk-win32-2.0-0.dll", "libglib-2.0-0.dll",
"libgmodule-2.0-0.dll", "libgobject-2.0-0.dll",
"libgthread-2.0-0.dll", "libgtk-win32-2.0-0.dll",
"libpango-1.0-0.dll", "libpangowin32-1.0-0.dll"]
}
}

setup(
name = "Hontza",
version = "0.5",
windows = [{
"script": "hontza.py"
}],
options=opts,
data_files=[("pixmaps", glob.glob("pixmaps/*.png")),
("glade", glob.glob("glade/*.*"))
],
)
```

Con esto ya tenemos el ejecutable, pero es necesario GTK para poder hacerlos funcionar. Por ello, tendremos que copiar la carpeta "dist" generada por py2exe y copiarla dentro de una copia de GTK/bin que habíamos instalado anteriormente, y quedaría de siguiente manera: GTK/bin/hontza.exe.

Para facilitar la instalación de Hontza hemos hecho uso de estas dos aplicaciones:

Nsis: Este es el compilador para hacer instaladores (http://nsis.sourceforge.net/)

Hmne: Este es el asistente para hacer uso del compilador Nsis (http://hmne.sourceforge.net)

INVESTIC
Soluciones tecnológicas de uso y Software Libre
www.investic.net
investic.interzonas.info
Plaza del Castillo 43 Bis 2ºC
Tlf. 948 22 15 63
info@investic.net