Guía de Inicio Rápido de Servoy

1. Conceptos Principales de Servoy

Generales

Java	Tecnología que se usa para el desarrollo de aplicaciones que convierten a la Web en un elemento más interesante y útil.
JDBC (Java Database Connectivity)	API que permite la ejecución de operaciones sobre bases de datos desde Java, independientemente del sistema operativo donde se ejecute o de la base de datos a la cual se accede, utilizando el lenguaje SQL del motor de base de datos que se utilice.
JAR (Java ARchive)	Tipo de archivo que permite ejecutar aplicaciones escritas en el lenguaje Java.
JavaScript	Lenguaje de programación interpretado. Se define como orientado a objetos y prototipos, imperativo, débilmente tipado y dinámico. Se utiliza principalmente en su forma del lado del cliente, implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas. Es el lenguaje utilizado por Servoy para escribir su lógica de negocios.
Evento	Suceso que ocurre en la ejecución del sistema.
Comando	Instrucción que se le ordena a un formulario.
Resources	Objeto donde Servoy guarda la información de las conexiones a bases de datos, hojas de estilo, seguridad e internacionalización para usarlo luego en la solución.
Application Server	Un servidor de aplicaciones gestiona la mayor parte (o la totalidad) de las funciones de lógica de negocio y de acceso a los datos de la aplicación. Los principales beneficios de la aplicación de la tecnología de servidores de aplicación son la centralización y la disminución de la complejidad en el desarrollo de aplicaciones. Servoy cuenta con su propia versión de servidor de aplicaciones el cual contiene la configuración de la implementación deseada.
Smart Client	El cliente nativo de Servoy que se instala y actualiza automáticamente en una máquina cliente.
Web Client	Cliente de Servoy que corre en un explorador de internet sin necesidad de instalación en la máquina cliente. Corre toda su lógica de negocios en el Servoy Application Server y se muestra en el explorador generando HTML y CSS. Soporte uso de AJAX.
Runtime Client	Aplicación nativa de una solución para una sola instalación. No requiere Servoy Application Server para ejecutarse. También contiene una base de datos PostgreSQL para los dato y puede ser ejecutado desde un disco duro, memoria USB o CD/DVD.
Headless Client	Cliente de Servoy sin interfaz gráfica. Se ejecuta en el Servoy Application Server.

Programación

Solución	Aplicación que puede ser ejecutada con cualquiera de los clientes de Servoy. Contiene formularios, lógica de negocios y definiciones en la base de datos.
Módulo	Una solución contenida en otra solución.
Scopes	Contexto de ejecución de JavaScript dentro de la solución. Un “scope” contiene variables y funciones. Servoy tiene 3 tipos: Global, Form y Base de Datos.
Form	Objeto que provee de una interfaz para el usuario y/o lógica de negocios. La interfaz se crea de Partes que contienen Elementos.
Elemento del Form	Control gráfico que se inserta en un Form. Los elementos incluyen: a. *Label* (Etiqueta) - Texto que aparece en un Form - generalmente estático aunque puede ser derivado en tiempo de ejecución. No puede ser editado por el usuario final. b. *Field* (Campo) - Elemento conectado a un Dataprovider. Puede ser editado por el usuario final y puede ser mostrado de varias maneras (normal, password, radios, check, entre otros). c. *Button* (Botón) - Etiqueta especial formateado para que se muestre como botón. Generalmente atado a un método. d. *Lines and shapes* (Líneas y Formas) - Objetos de dibujo para la estética de la interfase. e. *Portal* - Objeto que muestra datos de una relación de uno a muchos en Campos. f. *Tabpanel* - Objeto que permite mostrar varios Forms en pestañas dentro de otro. Los Forms agregados pueden estar relacionados por los datos o no. g. *Beans* - Objetos visuales de Java que extienden la funcionalidad.
Dataprovider	Propiedad de algunos objetos (tanto de la interfaz del usuario como en el código) que contiene datos. Puede ser una Variable Global, una Variable del Form o un campo de un registro de la base de datos.
Método	Función de JavaScript enganchada a un objeto. Es un grupo de código. Un método puede llamar a otros métodos, cambiar propiedades para elementos del Form, entre otra cosas.
Plugin	Extiende la funcionalidad de Servoy usando Java. Son accedidos desde el código. Hay una API disponible para escribir plugins para Servoy.

Datos

Base de Datos	Conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso.
SQL (Structured Query Language)	Lenguaje declarativo de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones en ellas.
PostgreSQL	Sistema de gestión de bases de datos objeto-relacional, distribuido bajo licencia BSD y con su código fuente disponible libremente. Es el sistema de gestión de bases de datos de código abierto más potente del mercado y el elegido por Servoy para acompañar su producto.
Tabla	Combinación de filas (registros) y columnas (campos) que definen el formato para guardar información en la base de datos. Los datos de la tabla pueden ser usados en Servoy por los Form de una solución.
Columna	Atributo de una tabla.
Foundset	Objeto de Servoy que representa el resultado de una consulta de una tabla de la base de datos. En Servoy, un FoundSet utiliza el caché para mostrar sus registros, pero no es necesario que “re-consulte” a la base de datos. Todos los Forms de una solución tienen un FoundSet que es el que muestran en pantalla. Todos los Forms de la misma tabla compartirán el FoundSet a menos que se indique lo contrario.
Record	Representación de una fila de la base de datos como un objeto en la solución. Los Records son contenidos por FoundSets.
Relación	Relación entre 2 tablas de la base de datos. Se puede dar por uno o más campos, aunque lo habitual es que sea de una clave primaria hacia un campo de otra tabla. También las hay desde variables globales a tablas y combinadas las 2 alternativas.
Dataset	Objeto de Servoy que guarda datos en formato de filas y columnas. Puede ser el resultado de una consulta SQL a la base de datos. A diferencia de los FoundSet, no tiene conexión directa con la base de datos. Es decir que, cualquier modificación en un Dataset, no será reflejado en la base de datos.
Calculation	Valor calculado que actúa como si fuera otra columna de una tabla en la base de datos. Si existe un campo en la tabla con el mismo nombre que una Calculation el valor se guarda en la base de datos. Si no existe un campo con el mismo nombre, ese valor se perderá y se volverá a calcular cuando se reinicie la aplicación. Son calculadas cuando son utilizadas (mostradas en un Form o nombradas en el código) o, automáticamente, cuando cambia algún valor referenciado en la Calculation.
Aggregation	Valor derivado de un cálculo sobre un campo de un conjunto de registros de una tabla. El cálculo puede ser sumar, contar, obtener máximo, obtener mínimo y obtener promedio.

Variable	Guarda un dato en tiempo de ejecución, pero no está atada a un campo de la base de datos. Pueden ser mostradas en pantallas. Se pueden definir en un Form o como Globales estando disponibles en cualquier contexto.

Eclipse

Workspace	Directorio donde se guarda todo el trabajo de Desarrollo en Eclipse. Contiene una carpeta por cada proyecto y metadata para Eclipse. Se pueden tener muchos workspace en una máquina pero sólo se permite trabajar con uno por vez.
Perspectiva	Vista del espacio de trabajo. Hay distintas perspectivas según el propósito. En Servoy las más utilizadas son "Servoy Design" para el desarrollo, "Debug" para debuggear una aplicación y "Team Synchronizing" para sincronizar con un repositorio SVN.

2. Instalación

El instalador es el mismo para el Servoy Developer como para el Servoy Application Server.

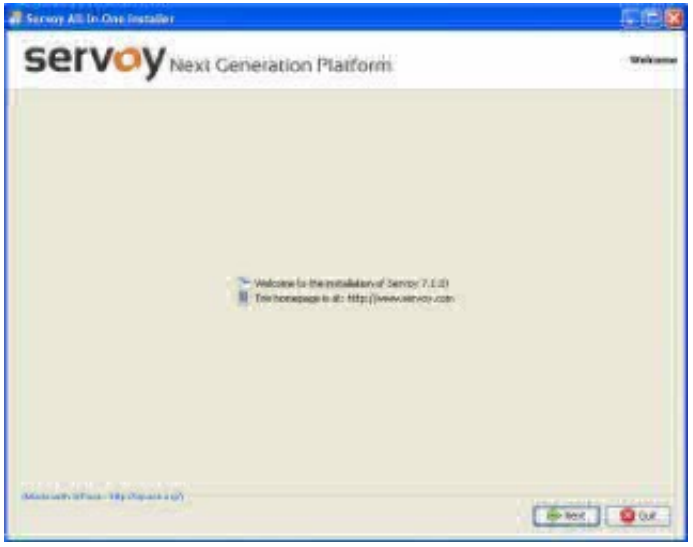
Primero debemos descargar el instalador de Servoy desde http://www.servoy.com/download. Allí, tendremos que registrarnos y luego, elegir la versión que se querrá instalar.

Las 2 versiones disponibles son:
- Multi-Plataforma desde un .jar
- Ejecutable de Windows (.exe)

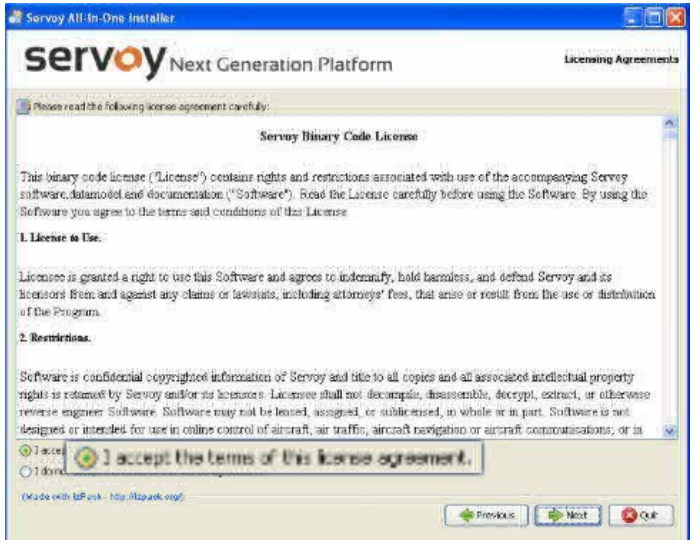
Una vez finalizada la descarga del instalador, debemos ejecutarlo. Para esto se lo puede haciendo doble clic sobre el archivo, haciendo clic derecho sobre el archivo y buscar el *Abrir Con* y elegir Java o desde la consola de comandos ejecutando la instrucción *java -jar servoy_installer.jar*.

Paso a paso del instalador

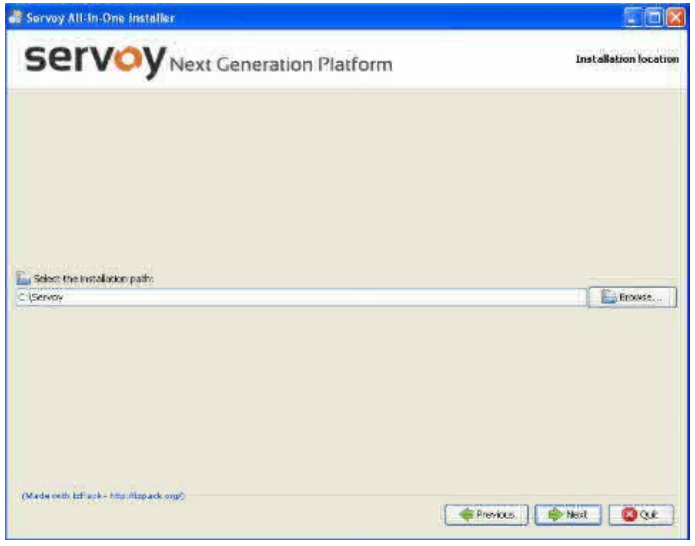
1 - Pantalla de Bienvenida. Tocamos “Next”.



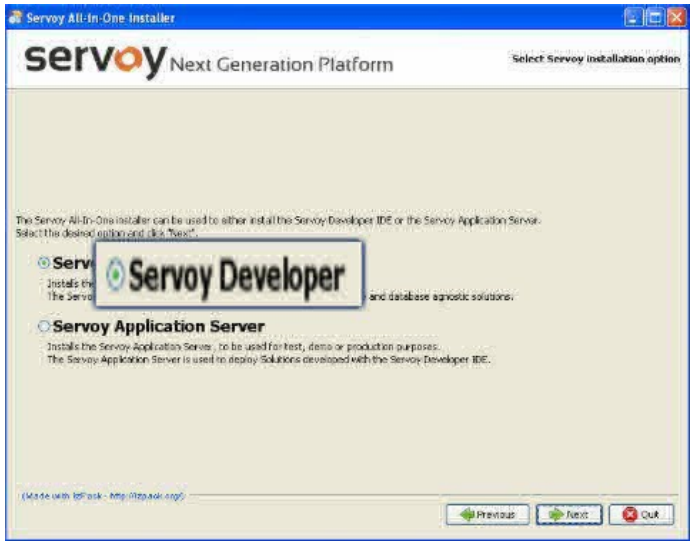
2 - Contrato de Licencia. Aceptamos (marcando como en la imágen) y tocamos “Next”.



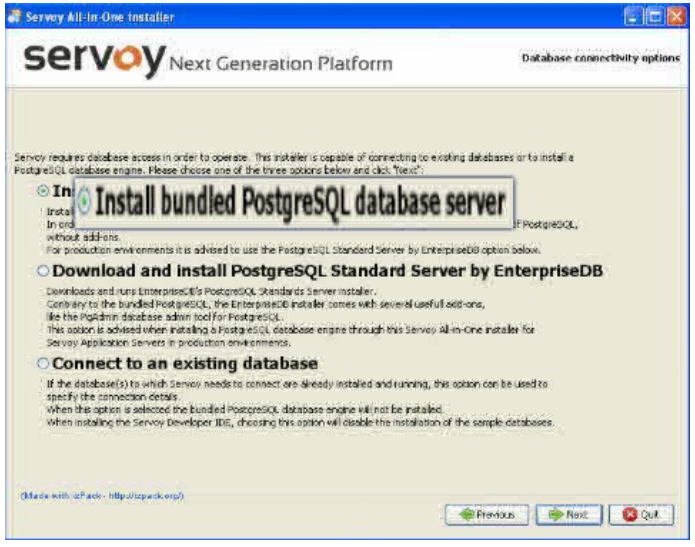
3 - Indicamos la ruta donde se lo va a instalar. El directorio de instalación debe tener permisos de Lectura/Escritura para los usuarios que vayan a utilizar el Servoy. Tocamos “Next”.



4 - Elegimos qué producto de Servoy se va a instalar. En este caso, marcamos “Servoy Developer” y luego “Next”.

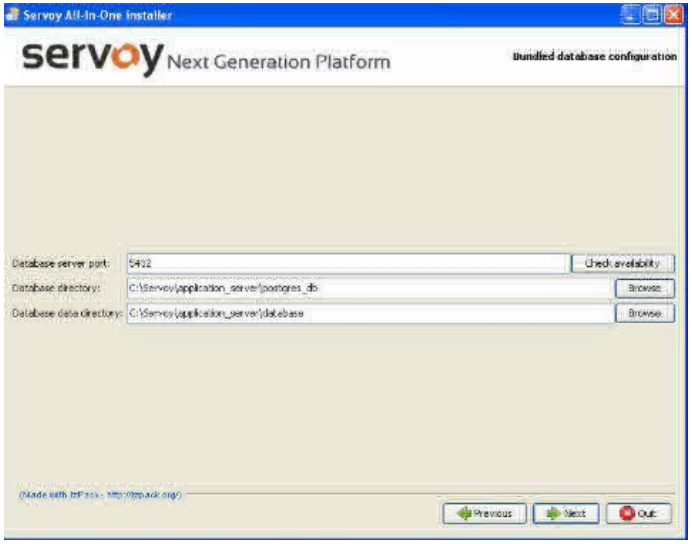


5 - Conexión con la Base de Datos. Aquí tenemos 3 opciones:
A – Instalar PostgreSQL junto con Servoy.
B – Instalar PostgreSQL como una aplicación independiente.
C – Conectarse a una Base de Datos ya existente.

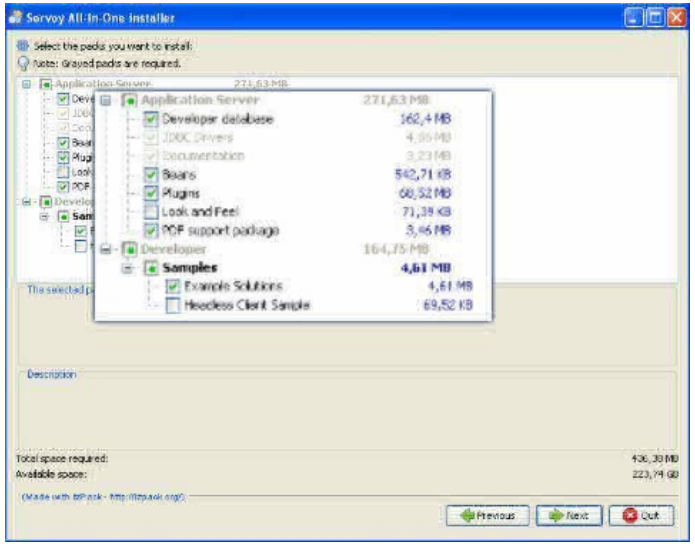


Elegimos la opción “A” como indica la imágen y luego tocamos “Next”.

6 - Configuración de la Base de Datos. Indicamos el puerto y los directorios donde se va a instalar el PostgreSQL. Tocamos “Next”.



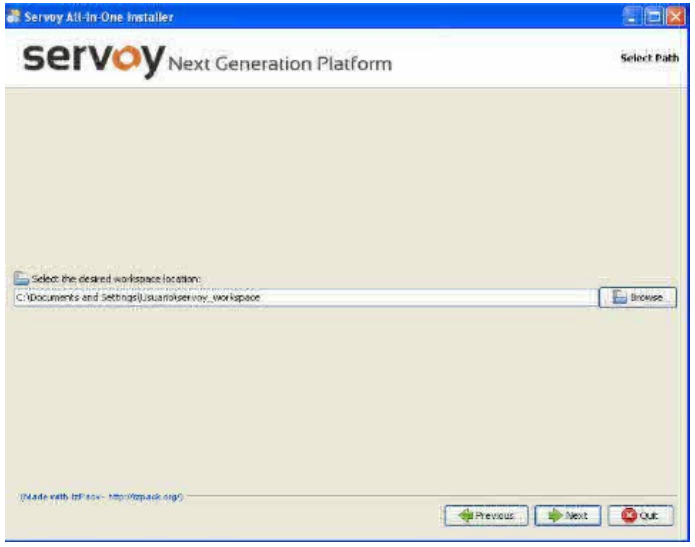
7 - Opciones de Instalación. Marcamos como indica la imágen y tocamos “Next”.



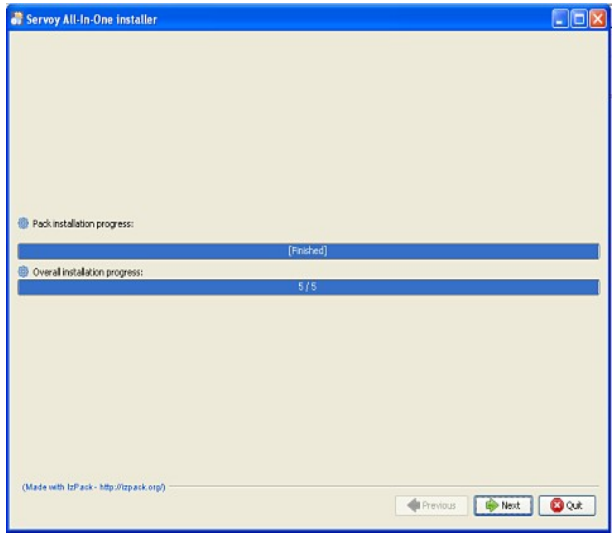
Las opciones son las siguientes:

Paquete	Obligatorio	Contenido
Application Server	Sí	Servoy Application Server
Developer	No	Servoy Developer
JDBC Driver Files	Sí	JDBC drivers sugeridos por Servoy
Database	No	PostgreSQL
Runtime Builder	No	Plugin para el Servoy Developer para crear Clientes Runtime
Application Server Service	No	Instalar el Servoy Application Server como servicio
Example Files	No	Soluciones de Ejemplo
Headless Client Sample	No	Ejemplo para un Servoy Headless Client
Documentation Files	Sí	Documentación para el Cliente de Servoy
Beans	No	Beans sugeridos por Servoy
Plugins	No	Plugins sugeridos por Servoy
Look and Feel Files	No	Archivos de diseño para el Cliente Smart de Servoy
PDF Support Package	No	Plugin para soportar trabajar con PDF's dentro de Servoy

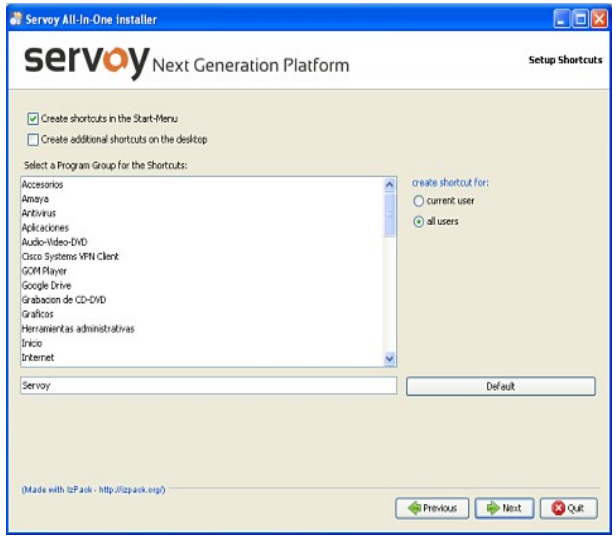
8 - Indicamos la ruta donde se va a alojar el Workspace de Servoy. Tocamos “Next”.



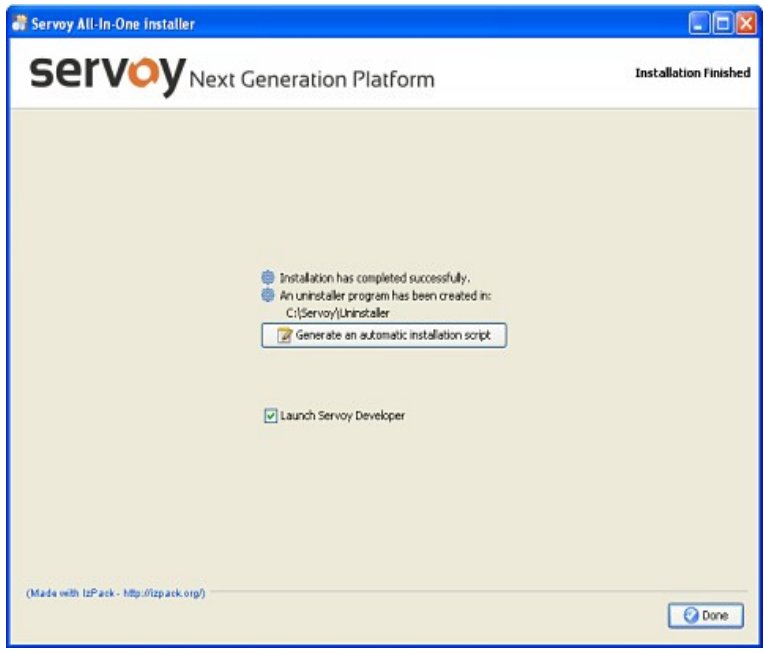
9 - Esperamos a que termine la instalación. Tocamos “Next”.



10 - Indicamos si queremos un acceso directo. Tocamos “Next”.



11 - La última pantalla nos da la opción de crear un script de instalación para cuando necesitemos hacerlo de forma no atendida. No lo hacemos y tocamos “Finish”. El Servoy Developer está listo para usarlo.



3. Perspectiva de Diseño

Solution Explorer	Árbol jerárquico con todos los obejtos disponibles para el desarrollo en Servoy. Todo lo que se puede hacer en Servoy lo va a encontrar en el Solution Explorer.
Resources	Se encuentran las conexiones a Bases de Datos, hojas de estilo, seguridad e internacionalización.
Solution	La solución activa. Dentro de ella podremos ver los Scopes, Forms, Relaciones, Valuelist, Media (imágenes) y los módulos que la componen.
JS Lib	Funciones de JavaScript disponibles. Las hay para todos los tipos de datos (Texto, Número, Vectores, etc.) como también para el flujo de una función o para tratar archivos XML.
Application	Funciones para la aplicación por ejemplo crear nuevas ventanas, salir de la aplicación o redireccionar a una URL, entre otras.
Solution Model	Funciones para crear formularios y elementos de forma dinámica en tiempo de ejecución.
Database Manager	Funciones para trabajar con la Base de Datos. Tratamiento de transacciones, consultas SQL a la Base de datos, manejo del guardado automático de Servoy como las principales.
Utils	Funciones útiles para distintas cosas. Formatear números o fechas, encriptar una cadena son algunas de ellas.
History	Funciones para manejar el historial del usuario. Limitado en Web Client ya que el historial lo maneja el browser.
Security	Funciones para manejar la seguridad de la aplicación. Login, logout, obtener el usuario conectado entre otras.
i18n	Funciones para la internacionalización. Agregar nuevas claves, obtener una clave en cierto idioma y muchas funciones más.
JSUnit	Funciones para test unitarios de funciones.
Plugins	Agrupa todos los plugins instalados. Cada uno será una opción donde se pueden observar sus funciones asociadas.
Propiedades	Las propiedades del objeto seleccionado aparece a la derecha de la pantalla por defecto. Dependiendo del objeto seleccionado se muestra un grupo de propiedades correspondientes para configurar.
Form Designer	Editor visual de los Forms. Aparece en el centro de la pantalla y se maneja con Drag n Drop para agregar y desplazar los elementos dentro de un Form. Cuenta con herramientas de agrupación y de alineamiento.

Script Editor	Permite crear y modificar funciones y variables del contexto (Form, global o tabla) correspondiente. Aparece en el mismo lugar que el Form Designer.
Problems View	Muestra los errores y llamados de atención de la solución. Se muestra en la parte de abajo de la pantalla. Para ver mayor información sobre lo informado, se puede hacer doble clic encima. Con clic derecho y la opción "Quick Fix" se tendrán sugerencias sobre cómo resolver el problema en algunos casos.

NOTA

En Eclipse al hacer clic derecho sobre cualquier elemento se mostrarán todas las opciones disponibles para tal.

4. Conexiones a Bases de Datos

Servoy puede conectarse a cualquier base de datos que tenga un driver JDBC disponible. Si Ud no posee ninguna base de datos, Servoy viene con PostgreSQL disponible para instalar.

Crear una Base de Datos PostgreSQL

1 - Hacer clic derecho sobre una base de datos y elegir la opción *Create PostgreSQL Database*
2 - Poner un nombre a la base de datos
3 - Esperar a que nos indique que el proceso esta OK.

Conectarse a una base de datos existente

1 - Copiar el Driver JDBC (con el Servoy Developer cerrado) en la carpeta *$SERVOY_DIR$/application_server/drivers*
2 - Abrir el Servoy Developer, hacer clic derecho sobre *Database Servers* y elegir la opción *New Server*.
3 - Aparecerá un submenu preguntando qué base de datos es, elegir la mejor opción. Aunque aparezca en la lista puede ser que no exista el Driver para conectarse.
4 - Llenar los datos correspondientes. El usuario debe tener permisos de administrador.
5 - Grabar

5. Crear una Solución

Crear Nueva Solución

1 - Elija una de las 3 opciones:

a. *File* > *New* > *Servoy Solution* del menú del Servoy Developer.
b. Haga clic sobre el botón *New* de la barra de herramientas. En la ventana emergente, elija el nodo de *Servoy* y luego *Servoy Solution*.
c. Haga clic derecho sobre *All Solutions* en el Solution Explorer o en la Solución Activa. Seleccione la opción *Create new solution* del menu.

Aparecerá una ventana donde debe eligir un nombre para la solución.

2 - Seleccione el tipo de solución:

Normal – Solución estándar que puede ser accedida por cualquier cliente o usada como módulo.
Module – Solución que va a ser parte de otra solución.
Web Client Only – Solución sólo accedida por Web Client.
Smart Client Only – Solución sólo accedida por Smart Client.
Login – Solución que permite a usuarios conectarse a otra solución.
Authenticator – Solución que autentica usuarios con el servidor.
Pre-import hook module – Solución que se utiliza antes de importar otra en el servidor.
Post-import hook module – Solución que se utiliza luego de importar otra en el servidor.
Mobile - Solución sólo accedida por Mobile Client.
Mobile Shared Module - Solución módulo de soluciones de tipo Mobile.

3 - Seleccione un proyecto de recursos.

4 - Haga clic en *Finish* y la nueva solución será creada y activada.

RECOMENDACIÓN

Sólo tenga un proyecto de recursos por workspace para evitar confusiones.

Importar Solución Existente

NOTA

Puede encontrar soluciones de ejemplo en *$SERVOY_DIR$/application_server/solutions/examples* para importar.

1 - Elija una de las 2 opciones:
   a. *File* > *Import*. En la ventana emergente, seleccione *Servoy* > *Import Solution* y haga clic en *Next*.
   b. Haga clic derecho sobre *All solutions* en el Solution Explorer y seleccione *Import a solution*.

2 - Seleccione el archivo .*servoy* que desea importar.

3 - Mantenga los default de las opciones para importar.

4 - Haga clic en *Finish*.

5 - Hay posibilidades que aparezcan algunas preguntas sobre importar datos, hojas de estilo y otros elementos contenidos en el archivo. Haga clic en *Yes*.

6 - Una ventana con el resultado le aparecerá. Si el proceso terminó exitosamente, la solución importada será activada.

6. Crear un Form

1 - Elija una de las 3 opciones:

a. Clic derecho sobre la solución activa en el Solution Explorer y seleccione *Create New Form*.
b. Clic derecho sobre Forms en el Solution Explorer y seleccione seleccione *Create New Form*.
c. Clic sobre el botón *New Form* en la barra de herramientas del Servoy Developer.

2 - Seleccione la tabla de la base de datos con la cual va a trabajar el nuevo Form. Si hace clic en el botón ... se abrirá un árbol con todas las bases y tablas disponibles para que elija. Si no quiere que el form trabaje con una tabla seleccion *-none-*.

3 - El nombre del Form se llenará automáticamente con el nombre de la tabla elegida. Igualmente, puede editarlo.

RECOMENDACIÓN

Recomendamos seguir una convención para que sea fácil ubicar los forms posteriormente. Por ejemplo "clientes_dtl" para un Form de clientes que muestre de un registro por vez y "clientes_tbl" para uno que muestre una tabla de clientes.

4 - Puede elegir la hoja de estilo que va a utilizar el Form.

5 - Verifique que se va a crear en la solución que desea. Puede cambiar a un módulo si lo desea.

6 - Haga clic en *Next*.

7 - Seleccione los datos a mostrar en el Form. Pueden ser tanto columnas de la tabla seleccionada, como variables, calculations, aggregations o campos relacionados. Sosteniendo la tecla *Shift* puede seleccionar un rango y con la tecla *Control* puede multiseleccionar.

8 - Elija las opciones sobre cómo se agregarán los campos en el Form.
Place With Labels : Colocará una etiqueta para cada dato seleccionado.
Place Horizontal : Colocará los datos uno al lado del otro (si no seleccionado, los colocará uno debajo del otro).
Fill Name Property : Llenará automáticamente la propiedad "Name" de cada elemento.
Fill Text Property : Llenará automáticamente la propiedad "Text" de cada elemento.
Place As Labels : Colocará los datos como etiquetas y no como campos de texto.

NOTA

Los Forms normalmente son creados como vista de un registro por vez, a menos que se active la opción *Place Horizontal* que lo hará como forma de tabla. Puede modificarse luego de ser creado.

9 - Haga clic en *Finish*. Se abrirá el Form creado en el Form Designer.

Editar un Form

Debe seleccionar el Form deseado en el Solution Explorer y luego elegir una de las 2 opciones:

a. Presione Ctrl-Shift-A
b. Clic derecho sobre el nombre y seleccionar *Open in Form Designer*

RECOMENDACIÓN

Para poder abrir un Form haciendo doble clic sobre su nombre en el Solution Explorer hace falta cambiar una preferencia. Para eso debe ir a *Window* > *Preferences* > *Servoy* > *Solution Explorer* > *Form Node Double Click Operation* y seleccionar la opción *Open Form Designer*.

7. Probar una aplicación

Con Servoy es posible probar y debuggear sus aplicaciones directamente desde el entorno de desarrollo. Esto incluye tanto clientes Smart como Web gracias a que el Servoy Developer levanta una instancia del Application Server de Servoy que permite 2 conexiones - 1 para Web y otra para Smart. Los cambios que se apliquen en la aplicacion se verán reflejados en los clientes sin necesidad de inciarlos nuevamente (sólo en entorno de desarrollo).

NOTA

Las aplicaciones Web suelen ser muy difíciles de debuggear en un browser ya que su lógica de negocios se divide entre el cliente y el servidor asincrónicamente. En Servoy esto no ocurre ya que toda la lógica de negocios corre en el servidor.

Smart Client

Elija una de las 2 opciones:

a. Actions > Start Smart Client
b. Botón de Start Smart Client en la barra de herramientas

Web Client

Elija una de las 2 opciones:

a. Actions > Start Web Client
b. Botón de Start Web Client en la barra de herramientas

NOTA

Puede cambiar el browser donde se prueban sus aplicaciones yendo a Window > Preferences > General > Web Browser y agregar su browser preferido agregándolo a la lista.

8. Implementando la Lógica de Negocios

A pesar de que la plataforma de Servoy está basada en Java por completo, el desarrollador no debe saber Java. Toda la lógica de negocios se escribe en JavaScript, lo cual lo hace mucho más productivo que escribir en Java puro. Servoy también le provee funciones para hacerlo más eficiente.

Creación de Variables

Elija una de las 2 opciones:

a. En el Solution Explorer, dirijirse al contexto deseado (Ejemplo: *Active Solution* > *Globals* > *variables*). Hacer clic derecho sobre *variables* y seleccionar la opción *Create Variable*. Poner un nombre, elegir un tipo de datos y, opcionalmente, un valor por defecto. La variable será creada en el archivo correspondiente (*globals.js* en el ejemplo), el cual se abrirá en el Script Editor.
b. En el Solution Explorer, dirijirse al contexto deseado (Ejemplo: *Active Solution* > *Globals* > *variables*). Seleccionar *variables* y cliquear sobre el botón *Create Variable* de la barra de herramientas inferior del Solution Explorer. Poner un nombre, elegir un tipo de datos y, opcionalmente, un valor por defecto. La variable será creada en el archivo correspondiente (*globals.js* en el ejemplo), el cual se abrirá en el Script Editor.
c. En el archivo correspondiente, fuera de todas las funciones, agregar

CÓDIGO
```
var nombreVar = "miValor";
```

Creación de Métodos

Elija una de las 2 opciones:

a. En el Solution Explorer, dirijirse al contexto deseado (Ejemplo: *Active Solution* > *Globals*). Hacer clic derecho sobre *Globals* y seleccionar la opción *Create Method*. Poner un nombre. El método será creado en el archivo correspondiente (*globals.js* en el ejemplo), el cual se abrirá en el Script Editor.
b. En el Solution Explorer, dirijirse al contexto deseado (Ejemplo: *Active Solution* > *Globals*). Seleccionar *Globals* y cliquear sobre el botón *Create Method* de la barra de herramientas inferior del Solution Explorer. Poner un nombre. El método será creado en el archivo correspondiente (*globals.js* en el ejemplo), el cual se abrirá en el Script Editor.
c. En el archivo correspondiente, fuera de otras funciones, agregar

CÓDIGO
```
function miMetodo() {

}
```

Invocación de Métodos y Variables

Para invocar a una función o variable, primero se debe saber cuál es el contexto en el que se está parado. Para ésto, es útil mirar el nombre del archivo abierto. Suele ser el nombre del Form, tabla o contexto seguido de ".js".

Ejemplos:

1 - Globales desde otro contexto:

```
CÓDIGO
    globals.funcion(parámetros)
    globals.variable = 1
```

2 - Forms desde otro contexto:

```
CÓDIGO
    forms.nombreForm.funcion()
    forms.nombreForm.controller.show()
    forms.nombreForm.variable = 1
```

3 - Contexto "cont" desde otro contexto:

```
CÓDIGO
    scopes.cont.funcion()
    scopes.cont.variable = "Estoy en otro contexto"
```

4 - Globales o Forms en el mismo contexto:

```
CÓDIGO
    funcion()
    variable = "Estoy en el mismo contexto"
```

Invocación de Funciones de Servoy

Las funciones de Servoy se pueden invocar desde el Solution Explorer, sin necesidad de escribir ni una letra en el Script Editor. Con el archivo correspondiente abierto, se puede invocar la llamada de una función o un ejemplo de cómo se puede usar esa función.

Invocación de Llamada

Elija una de las 2 opciones:

a. En el Solution Explorer, dirijirse a la sección donde se encuentra la función deseada (Ejemplo DatabaseManager). En la lista de métodos de la parte de abajo del Solution Explorer, hacer clic derecho sobre la función deseada y seleccionar la opción *Move Code*. El código será copiado en el archivo correspondiente.
b. En el Solution Explorer, dirijirse a la sección donde se encuentra la función deseada (Ejemplo DatabaseManager). En la lista de métodos de la parte de abajo del Solution Explorer, seleccionar la función deseada y cliquear el botón *Move Code* de la barra de herramientas inferior del Solution Explorer. El código será copiado en el archivo correspondiente.

Invocación de Ejemplo

Elija una de las 2 opciones:

a. En el Solution Explorer, dirijirse a la sección donde se encuentra la función deseada (Ejemplo DatabaseManager). En la lista de métodos de la parte de abajo del Solution Explorer, hacer clic derecho sobre la función deseada y seleccionar la opción *Move Sample*. El código de ejemplo de la función será copiado en el archivo correspondiente.
b. En el Solution Explorer, dirijirse a la sección donde se encuentra la función deseada (Ejemplo DatabaseManager). En la lista de métodos de la parte de abajo del Solution Explorer, seleccionar la función deseada y cliquear el botón *Move Sample* de la barra de herramientas inferior del Solution Explorer. El código de ejemplo de la función será copiado en el archivo correspondiente.

> RECOMENDACIÓN
>
> Servoy también usa el auto-completado para su código. Para invocarlo, debe tocar *Ctrl+Space* y le aparecerán las sugerencias sobre cómo puede seguir la instrucción.

9. Estilizando la Solución

Servoy soporta el uso de CSS (hojas de estilo en cascada de HTML) para estilizar sus aplicaciones. Una hoja de estilo es un archivo de texto puro con determinado formato que se guarda en la parte de Resources del Workspace. Puede haber tantos archivos como se quiera y se puede utilizar más de uno por solución, aunque sólo uno por Form. Incluso puede cambiar el estilo programáticamente. Funcionan tanto para Web Client como para Smart Client.

Un estilo está dividido en clases. Una clase es un conjunto de propiedades de un elemento particular. El estilo se aplica a un Form y la clase se aplica tanto a un Form como a los elementos dentro del mismo. Los elementos que pueden ser usados son: form, label, button, field, check, radio, combobox, tabpanel y portal.

Ejemplo:
Label con letra blanca y fondo negro como default y una clase *título* con la letra más grande y en negrita

CÓDIGO

```
label {
color: #ffffff;
background-color: #000000;
font: 10pt Verdana;
 }
label.titulo {
color: #ffffff;
 background-color: #000000;
 font: bold 14pt Verdana;
  }
```

¿Cómo usarlo en una solución?

Para aplicar un estilo a un Form, se debe elegir el estilo deseado en la propiedad *StyleName* del Form. Luego, se puede elegir una clase determinada o simplemente que tome la clase default en la propiedad *StyleClass*.

Para los elementos del Form, sólo se podrán aplicar las clases que estén dentro del estilo elegido para el Form que integran. Para aplicar una clase determinada, se la debe elegir en la propiedad *StyleClass* (sólo aparecerán las clases para el elemento seleccionado).