

SWBForms

Guía del Programador

La plataforma **ágil** de **SWB**



Infotec
2019

Contenido

Introducción	3
Instalación y Puesta a punto de SWBForms	4
Empezar con SWBForms	5
Iniciar sesión en SWBForms	5
Elementos de SWBForms	5
Área de menú.....	6
Área de trabajo.....	6
La ficha Programación	7
Iniciar una nueva aplicación	8
DataSources	9
Crear un DataSource nuevo	11
Atributos extendidos.....	16
Creando el Menú de la aplicación	18
SC Grid Detail.....	20

Introducción

SWBForms es la plataforma de la suite **SemanticWebBuilder** para desarrollo ágil, basada en Java y Javascript que permite construir aplicaciones de negocio responsivas en muy corto tiempo con gran fortaleza de seguridad y desempeño. Con **SWBForms** es posible crear en minutos listas tabulares con funcionalidades complejas como agregar, editar y eliminar registros en línea; paginación automática además de filtrado, ordenación, intercambio de columnas, etc; crear formularios completos, agregarles validaciones de manera muy sencilla y rápida.

En este manual se explica el desarrollo de aplicaciones Web mediante **SWBForms**, un conjunto de herramientas de desarrollo que facilita la creación de aplicaciones Web creada por **INFOTEC** (Centro de Investigación de Investigación e Innovación en Tecnologías de la Información y Comunicación).

Con **SWBForms** se puede desarrollar aplicaciones Web mediante Javascript y JSP. Para ejemplificar el potencial para el desarrollo de aplicaciones de **SWBForms** en el transcurso de este manual se explica el desarrollo de una aplicación demo diseñada para el manejo de inventarios.

Instalación y Puesta a punto de SWBForms

El equipo de cómputo que correrá la plataforma SWBForms debe cubrir con los siguientes requerimientos:

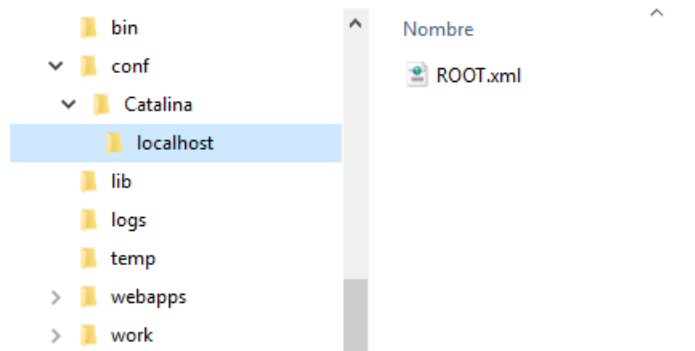
- **JDK 8**
- **Servicio MongoDB corriendo en el puerto 27017.**
- **Apache Tomcat 8 o superior**
- **Navegador web** (referentemente Chrome)

También es importante conocer el puerto configurado Tomcat, regularmente es el 8080.

Para instalar SWBForms en un ambiente local Windows siga los siguientes pasos:

1. Descargar el archivo de despliegue de SWBForms, *SWBForms-1.0-SNAPSHOT.war*, disponible en <https://github.com/SWBForms2>
2. Expandir *SWBForms-SNAPSHOT.war* en el sistema de archivos local
3. Crear un archivo de texto y nombrarlo ROOT.xml
4. ROOT.xml debe contener únicamente la siguiente información:

```
<Context path="/" docBase="C:\MiUbicacion\SWBForms-1.0-SNAPSHOT"/>
```
5. Este archivo ROOT.xml debe copiarse en el siguiente directorio del servidor de aplicaciones Tomcat: *ApacheTomcat_8/conf/Catalina/localhost/*



6. Iniciar Tomcat
7. Crear una cuenta para acceder al ambiente de programación. Escribir <http://localhost:8080/utils/createAdminUser.jsp> en la barra de direcciones del navegador web y pulsar la tecla Enter.
8. Agregar la nueva cuenta en la Base de datos. Escribir <http://localhost:8080/utils/importAdmin.jsp> en la barra de direcciones del navegador web y pulsar la tecla Enter.
9. Acceder a la dirección: <http://localhost:8080/admin>. Debe mostrarse una pantalla parecida a la mostrada en la Figura 1.

Para iniciar sesión podemos hacer uso de la cuenta que creamos en los pasos 7 y 8

Usuario: admin@infotec.mx

Contraseña: swbforms

Empezar con SWBForms

Iniciar sesión en SWBForms

Para iniciar el trabajo con SWBForms, debemos iniciar sesión:

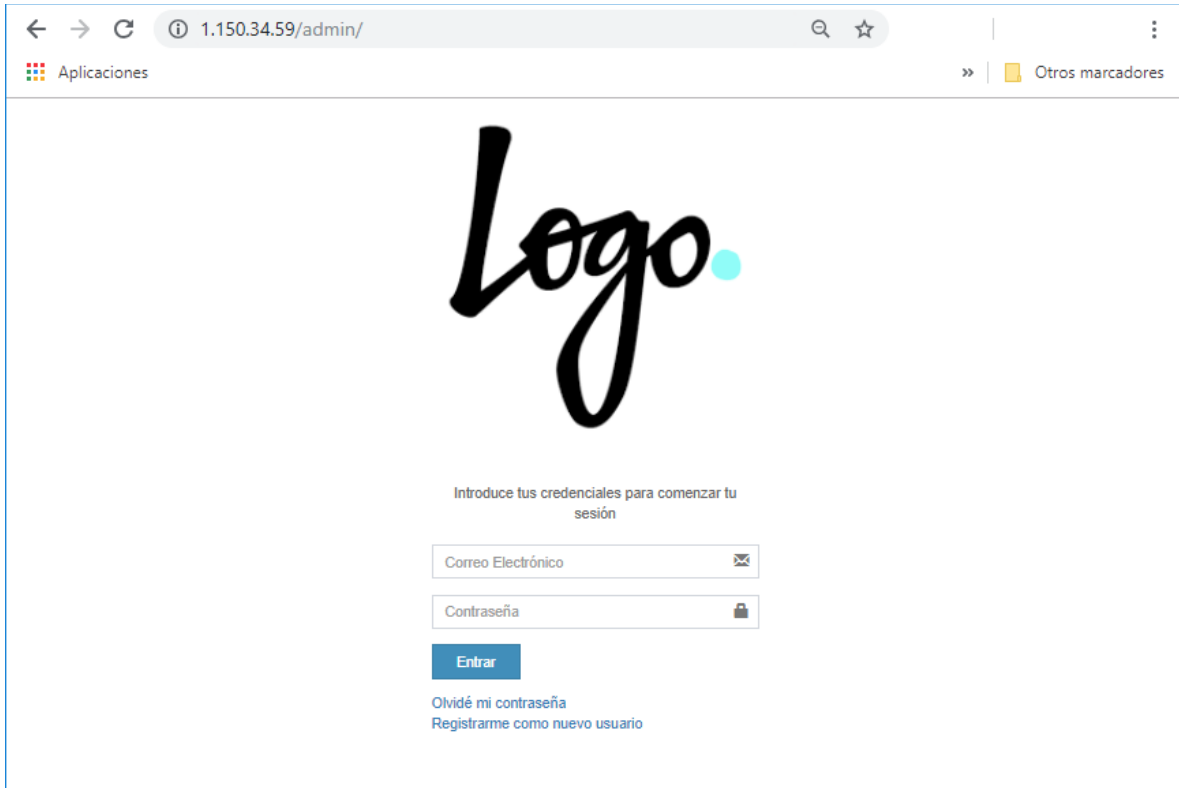


Figura 1. Iniciar sesión con SWBForms

Para lograr esto, accedemos a nuestra instancia de desarrollo. En la figura de arriba, podemos observar una instancia de **SWBForms** lista para iniciar sesión.

Una vez iniciada nuestra sesión, veremos una pantalla similar a la mostrada en la Figura 2.

Elementos de SWBForms

Vamos a ver los elementos básicos de SWBForms. Al iniciar SWBForms aparece una pantalla inicial como esta:

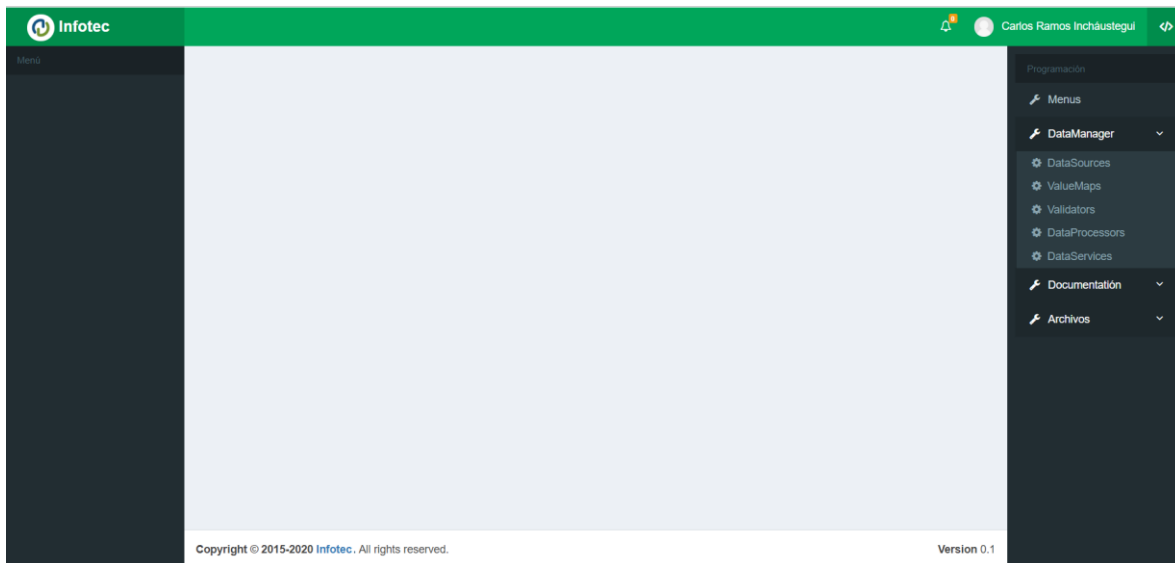


Figura 2. Vista de SWBForms

Empezaremos con los componentes fundamentales, de esta forma conoceremos los nombres de los diferentes elementos y será más fácil entender el resto del manual. La pantalla anterior puede no coincidir exactamente con la que ves en tu computadora, ya que eso depende de la distribución del war.

Área de menú

Contiene la estructura de opciones que se agregan desde la paleta Programación. Cuando agregamos opciones de menú se visualizan en esta área, una vez refrescada la pantalla.



Figura 3. Partes de SWBForms.

Área de trabajo

Esta área muestra la funcionalidad de la aplicación accedida mediante las opciones del menú. Esta área y la de menú, representa el punto focal para un usuario final. Provee la interfaz para realizar la operación que enmarca el propósito de la aplicación implementada.

La ficha Programación

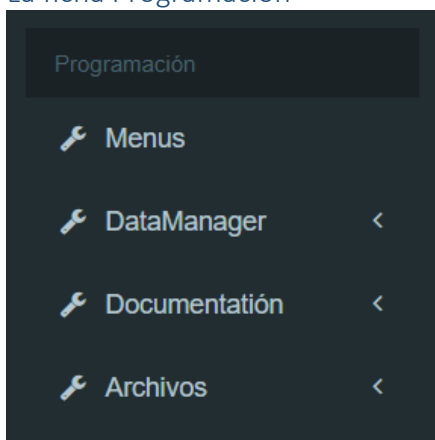


Figura 4. Ficha Programación


Para acceder a la ficha Programación debes contar con el role Programador (Programmer). SWBForms, despliega la ficha Programación, haciendo clic en el botón  que se encuentra en la parte superior derecha de la pantalla. Contiene elementos como: **Menús**, **DataManager**, **Documentación** y **Archivos**. Para cerrar la ficha Programación y volver al área de trabajo hacemos clic nuevamente en el ícono anterior.



Figura 5. Abrir y cerrar la Ficha Programación.

- **Menús.** Nos permite definir el Menú de la Aplicación. Vea la Figura 5.
- **DataManager.** Nos permite definir los datos y el comportamiento de la Aplicación.
 - **DataSourcees.** Define datos.
 - **ValueMaps.** Define datos.
 - **Validadores.** Define comportamiento.

- **Data Processors.** Define comportamiento.
- **Data Services.** Define comportamiento.
- **Documentación.** Documentación del diseño generada automáticamente.
- **Archivos.** Nos permite acceder a los archivos de Sistema.

Iniciar una nueva aplicación

Para iniciar, lo primero que necesitamos entender que es exactamente un DataSource. SWBForms es una plataforma orientada a los datos, así que los DataSources representan la piedra angular del desarrollo de cualquier aplicación web basada en SWBForms.

Para la implementación de **menú de usuario, formularios (Formas) y listados tabulares (Grids)**, lo primero es contar con un DataSource.

Un DataSource representa un concepto o entidad, como los sustantivos en un dominio. Cada DataSource es un modelo que describe una cantidad de propiedades y comportamientos. A partir de un DataSource, se puede definir un conjunto de cualquier tamaño de instancias.

Pongamos un ejemplo para ilustrar lo antes dicho, supongamos un DataSource llamado *Producto*. Cada vez que se crea un nuevo *producto* a partir del DataSource *Producto* estamos creando una instancia de ese DataSource. Un producto tiene nombre de producto, clave de producto, número de lote, costo, precio público, peso, tamaño, color, etc. Todos estos datos son las propiedades del producto y constituyen el estado de ese producto en particular.

Dato interesante

DataObject

SWBForms emplea, internamente, una Clase llamada DataObject para acceder a cualquier instancia de un DataSource y poder manipularlo.

Un DataObject puede almacenar datos en varios formatos, esto permite que DataObject sea compatible con una amplia variedad de destinos que si solo se pudiera representar un solo formato de dato. Los destinos aquí hacen referencia a las entidades de cualquier dominio.

DataObject proporciona varios constructores sobrecargados que facilitan el llenado de una instancia de DataSource.

Dentro de los comportamientos que puede tener un producto, siguiendo con el ejemplo, sería que no permita que un usuario establezca su precio público fuera de lo que señala cierta política de la empresa.

DataSourcees

DataSourcees Home > Programación > DataSourcees

Identificador	Descripción	Backend	Frontend	Roles de Co...	Roles de Cre...	Roles de Edi...	Roles de Eli...		
User	Usuarios	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	--	--	--	--		
Log	Log	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	--	--	--	--		

2 Registros

Agregar nuevo DataSource

Muestra la vista de impresión

Apariencia de la vista

Exportar los datos

Figura 6. La Ficha Programación permite gestionar los DataSourcees de las aplicaciones implementadas con SWBForms.

Todos los DataSourcees hacen uso de, principalmente, las siguientes propiedades:

- **name.** El nombre del atributo permite identificarlo dentro del DataSource.
- **title.** El título que será visible al usuario final, por lo general se expresa en lenguaje que permita al usuario final reconocerlo como un dato de su operación.
- **type.** Tipo de dato. Cada una de las propiedades de un DataSource debe incluir el atributo type. La Sección 3.4 del Manual SWBForms incluye una lista completa de tipos de datos disponibles


A continuación, se presenta la Figura 4 con una representación del DataSource Person.

```
eng.dataSources["Person"] = {
  scis: "Person",
  modelid: "SWBF2",
  dataStore: "mongodb",
  displayField: "fullname",
  fields: [
    {name: "firstName", title: "Nombre", type: "string", length: "64", required: true},
    {name: "lastName", title: "Primer Apellido", type: "string", length: "64", required: true},
    {name: "secLastName", title: "Segundo Apellido", type: "string", length: "64", required: false},
    {name: "gender", title: "Género", type: "select", valueMap: "gender", required: true, defaultValue: "male"},
    {name: "birthday", title: "Cumpleaños", type: "date", required: false}
  ]
};
```

Figura 7. DataSource **Person**.

Para crear un **DataSource** nuevo o editar uno ya existente:

1. Abre la ficha Programación.

Usa el mouse para llegar al botón .

2. Abre la lista desplegable DataManager.









Da clic sobre el la lista desplegable **DataManager**.

3. Seleccionar la opción DataSources.





Cuando hagas clic sobre esta opción, se abrirá la pantalla DataSources.

La siguiente Figura, muestra cómo se ve esta pantalla.

DataSources Home > Programación > DataSources

Identificador	Descripción	Backend	Frontend	Roles de Co...	Roles de Cre...	Roles de Edi...	Roles de Elim...	
User	Usuarios	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	--	--	--	--	 
Log	Log	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	--	--	--	--	 
Producto	Productos de v...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Admin, Progra...	Admin, Progra...	Admin, Progra...	Admin, Program...	 
TipoProducto	Tipo de Producto	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Admin, Progra...	Admin, Progra...	Admin, Progra...	Admin, Program...	 

4 Registros




Agregar nuevo DataSource 
 Modificar la definición del DataSource 
 Eliminar el DataSource 

Figura 8. DataManager nos permite administrar nuestros DataSources.

Crear un DataSource nuevo

Vamos a explicar los campos que debemos capturar para crear un nuevo DataSource.

- **Identificador** nos permite asignarle un nombre al DataSource a nivel de Sistema, es decir, la manera en que identificará en la estructura de clases estáticas. Es importante evitar el uso de acentos, tildes y espacios en blanco. También es conveniente mencionar que **SWBForms** recomienda mantener los mismos valores para los campos Identificador y Nombre de Colección, eso no implica que Nombre de Colección no pueda cambiar de valor.

The screenshot shows a web form for creating a new DataSource. The form is titled "DataSources" and has a breadcrumb trail: Home > Programación > DataSources > Alta. A tab labeled "Agregar Menu" is active. The form contains several input fields: "Identificador" (empty), "Descripción" (empty), "Nombre de Colección" (empty), "Nombre del Modelo" (containing "_modelid"), "Data Store" (containing "_dataStore"), "Campo de Despliegue" (empty), and four dropdown menus for "Roles de Consulta", "Roles de Creación", "Roles de Edición", and "Roles de Eliminación". There are also two checkboxes for "Backend" and "Frontend", both of which are unchecked. At the bottom right, there are two blue buttons: "Regresar" and "Guardar".

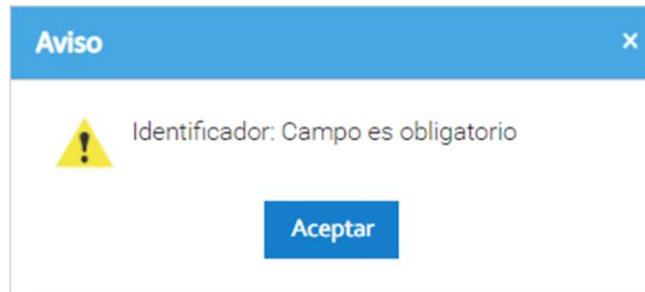
Figura 9. Crear un DataSource nuevo.

- **Descripción**, en este campo podemos agregar un texto que facilite el entendimiento del propósito del DataSource.
- **Nombre de Colección** es el nombre a nivel de Base de datos del DataSource. Por lo cual es conveniente evitar el uso de acentos, tildes y espacios en blanco.

- **Campo de Despliegue**, nos permite designar la propiedad que se mostrará por default cuando instanciamos un Producto en particular. Nuevamente recomendamos enfáticamente evitar el uso de acentos, tildes y espacios en blanco.
- **Backend, Frontend**, por el momento vamos a marcar estas casillas, más adelante explicaremos a detalle su funcionamiento.
- Los **Roles** para **Consulta, Creación, Edición y Eliminación**, también lo explicaremos en el Capítulo Seguridad. Momentáneamente vamos a asignarle permisos a todos los roles. Para lo cual, marcamos todas las casillas de la lista desplegable de cada tipo de permiso.
- **Nombre del Modelo y Data Store** son valores de solo lectura, estos campos exponen que sus valores son variables globales de sistema.

Nota

Eventualmente encontrarás que el título o etiqueta de ciertos campos aparecen en negritas, esto significa que ese campo es obligatorio. En el caso de omitir dichos campos, **SWBForms** nos lo notificará por medio de un cuadro de diálogo con un mensaje parecido al siguiente:



A continuación, mostramos cómo se vería la creación del DataSource Producto:

DataSources Home > Programación > DataSources > Alta

Agregar Menu

Identificador : Producto

Descripción : Productos de venta

Nombre de Colección : Producto

Nombre del Modelo : _modelid

Data Store : _dataStore

Campo de Despliegue : nombreProducto

☒ Backend

☒ Frontend

Roles de Consulta : Admin, Programmer, Super User, User

Roles de Creación : Admin, Programmer, Super User, User

Roles de Edición : Admin, Programmer, Super User, User





Roles de Eliminación : Admin, Programmer, Super User, User

Regresar Guardar

Figura 10. Crear un nuevo DataSource **Producto**.

El botón **Regresar** nos permite ver la lista de DataSources en nuestro sistema.

DataSources Home > Programación > DataSources

Identificador	Descripción	Backend	Frontend	Roles de Co...	Roles de Cre...	Roles de Edi...	Roles de Eli...	
User	Usuarios	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	--	--	--	--	 
Log	Log	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	--	--	--	--	 




2 Registros   

Figura 11. Lista de DataSources. Aquí podemos crear modificar o eliminar DataSources.

Una vez completado el llenado del formulario, hacemos clic en el botón **Guardar**. **SWBForms** crea el nuevo DataSource, y muestra un cuadro de diálogo informando que los datos fueron registrados correctamente.

Ahora la pantalla del DataSource Producto muestra tres pestañas más, aparte de la pestaña **Información**:


- **Propiedades.**
Formulario para definir las propiedades y comportamientos del DataSource.
- **Datos.**
Muestra un listado tabular de los valores de todas las instancias del DataSource almacenadas en Base de datos.
- **Importar.**
Nos provee un archivo CSV con el layout (estructura) de nuestro DataSource para cargar nuevas instancias del DataSource.



Figura 12. La pestaña **Propiedades** nos permite definir las propiedades de un DataSource.

La siguiente pestaña, **Propiedades**, nos permite definir las propiedades del DataSource junto con todos sus atributos. Recordemos que, en el paso anterior, asignamos *nombreProducto* como Campo de Despliegue, ahora es momento de definir esta propiedad. La descripción de cada campo nos ayudará a entenderlos. Si aún siguen sin entenderse, puede consultar el Manual de SWBForms.

- **Identificador.** El nombre con que se identifica el atributo dentro del DataSource.
- **Título.** El título que será visible al usuario final, por lo general se expresa en lenguaje que permita al usuario final reconocerlo como un dato de su operación.
- **Tipo.** Tipo de dato. Cada una de las propiedades de un DataSource debe incluir el atributo type. La Sección 3.4 del Manual SWBForms incluye una lista completa de tipos de datos disponibles.
- **Requerido.** Especifica si esta propiedad (campo) es obligatorio en un formulario.
- **Orden.** Especifica el orden en el que se desplegará la propiedad (campo) en un formulario.

Hacemos clic en el botón  para agregar una nueva propiedad. Cuando terminemos de agregar los atributos principales, damos Enter para guardar los cambios. Esta pantalla no incluye un botón Guardar, en lugar de eso para persistir los cambios realizados, simplemente damos Enter y SWBForms guarda automáticamente todos los cambios realizados.

Identificador	Título	Tipo	Requerido	Orden
nombreProducto	Nombre del Producto	string	<input checked="" type="checkbox"/>	10

Figura 13. Definición de la propiedad nombreProducto.

Una vez ingresado el Orden, damos Enter y automáticamente, se guardan los cambios.

Identificador	Título	Tipo	Requerido	Orden
nombreProducto	Nombre del Producto	string	<input checked="" type="checkbox"/>	10

Figura 14. No existe un botón guardar, una vez registrada una propiedad, SWBForms los registra automáticamente.

Atributos extendidos

El cuadro de diálogo de Atributos Extendidos nos permite gestionar de entre una amplia lista de atributos extendidos.

Identificador	Título	Tipo	Requerido	Orden
nombreProducto	Nombre del Producto	string	<input checked="" type="checkbox"/>	10

Permite abrir/cerrar el cuadro de diálogo de atributos extendidos

Figura 15. Definición de atributos extendidos para la propiedad nombreProducto.

Los atributos extendidos menos comprendidos son **stype** y **dataSource**. A continuación, revisaremos para qué sirven estos atributos extendidos.

Estos atributos se emplean para referenciar instancias de otro DataSource desde el DataSource que estamos definiendo. El resultado final es que **SWBForms** construye una lista desplegable para

DataSources Home > Programación > DataSources > Detalle

Información Propiedades Datos Importar

Identificador	Título	Tipo	Requerido	Orden
nombreProducto	Nombre del Producto	string	<input checked="" type="checkbox"/>	10
tipoProducto	Tipo de Producto	string	<input type="checkbox"/>	—

Atributo	Tipo de Valor	Valor
stype	string	select
dataSource	string	TipoProducto

2 Registros

2 Registros

Figura 16. Definición de dos atributos extendidos, stype y dataSource.

seleccionar uno o varias instancias del DataSource referenciado.

Nota

Para seleccionar múltiples instancias de un DataSource referenciado en otro, agregamos el atributo extendido **multiple** y establecemos su valor a *true*. El **Manual de SWBForms** explica a detalle la mayoría de atributos extendidos.

Creando el Menú de la aplicación

En este capítulo, veremos cómo construir una jerarquía de menú funcional para la aplicación hipotética que venimos trabajando.

Como mencionamos anteriormente, la interfaz para acceder a la funcionalidad de una aplicación implementada en **SWBForms**. Para ello, haremos uso de la opción Menús de la ficha Programación. Esta funcionalidad está implementada por **recursos** automáticos o implementados directamente por nosotros.

Los recursos automáticos están íntimamente relacionados con los DataSources definidos para nuestra aplicación. En el caso concreto de nuestra aplicación hipotética, serían los DataSources Producto y TipoProducto.

Existen varios tipos de opciones para configurar cada elemento del menú:

- **Head.**
Define el elemento del menú como un ancla para una estructura compleja de elementos de menú
- **Group Menu.**
Nos permite crear un nuevo grupo para organizar convenientemente la jerarquía de opciones.
- **URL Content.**
Nos re direcciona a un recurso abandonando la pantalla inicial. Este recurso puede ser, por ejemplo, una JSP interna o una página web externa.
- **Ajax Content.**
Define un recurso como una pestaña del elemento de menú padre. A su vez, el elemento padre debe ser de tipo SC Grid Detail o SC Form. Siendo el tipo SC Grid Detail el más popular por su simplicidad.
- **IFrame Content.**
Es análogo a Ajax Content con la diferencia de que la invocación no es asíncrona.
- **SC Grid.**
Define una lista tabular con el contenido del DataSource.
- **SC Grid Detail.**
Define una lista tabular con el contenido del DataSource, agregando el comportamiento de drill down sobre cada fila para acceder a información detallada de la instancia del DataSource.
- **SC Form.**
Define un formulario con el DataSource especificado.

Supongamos que deseamos agregar el catálogo de Productos al menú principal. Veamos a continuación cómo se ve este menú:

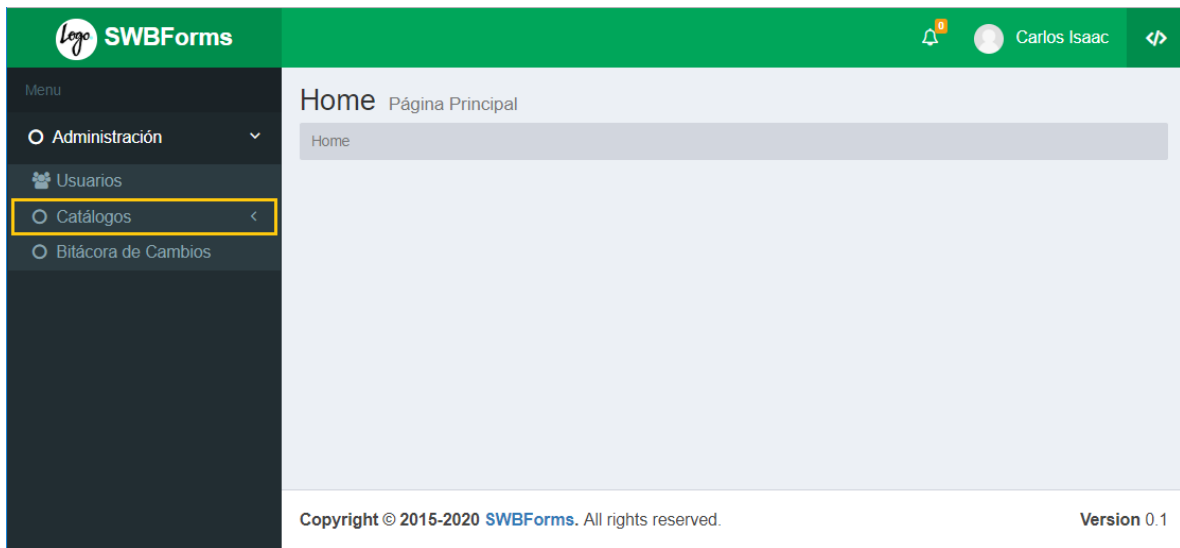


Figura 17. Queremos agregar el elemento de menú Productos bajo Catálogos.

SC Grid Detail

El tipo de Menú SC Grid Detail, es uno de los más utilizados por su funcionalidad. Resulta muy conveniente cuando deseamos mostrar una lista inicial de instancia de algún DataSource y acceder al detalle de uno en particular bajo la operación Drill Down.

A continuación, mostramos cómo quedaría definido el elemento de menú **Producto**, dentro de la jerarquía **Catálogos**.

Vamos a explicar los campos que debemos capturar para crear un nuevo DataSource.

The screenshot shows a web application interface for managing menus. The title is 'Menus' and there is a breadcrumb trail: 'Home > Programación > Menus > Alta'. A button 'Agregar Menu' is visible. The form contains the following fields:

- Identificador:
- Nombre:
- Padre:
- Sub Nombre:
- Orden:
- Extra URLParams:
- Tipo:
- Icon:
- Clase del Icono:
- Roles de Acceso:
- Estatus:
- DataSource:
- Características del Grid:
- Propiedades del Grid:

Below these fields is a table for 'Propiedades del Grid':

Propiedad	Atributo	Tipo de Va...	Valor
No hay objetos a mostrar.			

At the bottom of the table are three icons: a green plus, a blue pencil, and a red minus. Below the table is a text area for 'JScript Adicional al Grid'.

Identificador. El nombre con que se identifica el atributo dentro del DataSource.

Nombre. Título con que se muestra el recurso para Producto.

Padre. Elemento padre de este elemento del Menú.

Sub Nombre. Subtítulo del recurso para Producto.

Orden. Posición del elemento Menú dentro de la jerarquía Padre.

Extra URLParams. Parámetros adicionales enviados al recurso para Producto.

Tipo. Tipo de elemento de menú.

Ícono. Ícono del elemento menú.

Clase del ícono. Clase CSS del ícono.

Roles de Acceso.

Figura 18. Crear nuevo elemento de menu. Parte 1.

Propiedades de la Forma : Tipo de Producto, Nombre del Producto

Propiedad	Atributo	Tipo de Va...	Valor
No hay objetos a mostrar.			

Propiedades Extendidas de la Forma :

JScript Adicionales a la Forma :

Roles de Creación :

Roles de Edición :

Roles de Eliminación :

Regresar Guardar

Figura 19. Crear nuevo elemento de menú. Parte 2.

Data Source. DataSource de referencia del elemento de menú. En este caso concreto, Producto.

Propiedades del Grid. Define las propiedades del DataSource visibles en el Grid.

Propiedades Extendidas del Grid. Atributos extendidos para mejorar la vista de las propiedades anteriores.

JScript Adicional al Grid.

Comportamiento

adicional del Grid.

Propiedades de la Forma. Define las propiedades del DataSource visibles en el formulario.

Propiedades Extendidas de la Forma. Atributos extendidos para mejorar la vista de las propiedades del formulario. Estos atributos comprenden vista y comportamiento.

JScript Adicionales a la Forma. Comportamiento adicional del formulario.

Roles de Creación. Asigna el o los roles que tendrán permiso de crear nuevas instancias del DataSource

Roles de Edición. Asigna el o los roles que tendrán permiso de editar las instancias del DataSource.

Roles de Eliminación. Asigna el o los roles que tendrán permiso de eliminar instancias del DataSource.

Una vez terminada la definición del elemento de menú, hacemos clic en el botón **Guardar**. Para verificar los cambios recientemente realizados, será necesario refrescar, para lo cual podemos recargar la página con el mouse o pulsar la tecla F5.

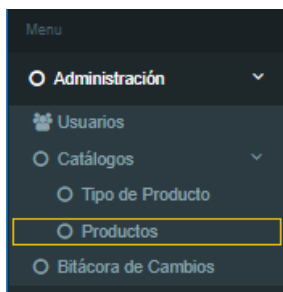


Figura 20. Menú Productos listo!