

Tipos de propiedades o Fields

```
type: string, int, date, float, double, long, text //primitivos
stype: grid, gridSelect, select, time, file //extendidos
```

//Tipos de links, objetos vinculados a formas

```
stype: subForm, tab
```

//DEPENDENCE SELECT

```
{name: "pais", title: "Pais", stype: "select", dataSource:"Países",
changed:"form.clearValue('estado');"},
{name: "estado", title: "Estado", stype: "select", dataSource:"Estados",
  getFilterCriteria : function () {
    var pais = this.form ? this.form.getValue("pais") :
this.grid.getEditedCell(this.rowNum, "pais");
    return {pais:pais};
  }
},
```

//Formulas en Grids

```
editorFormula: { text: "DateAdd(new Date(),'d',shipDays)"}
```

//Field Help; pone una interrogación al final del campo



```
fields : [
  {name: "recordtitle",
    title: "Título",
    helpText: "Capturar el Título del BIC",
    icons: [{
      src: "other/help.png",
      click: "isc.say(item.helpText)"
    }]
  }
]
```

//Keywords; Se van agregando strings y se guardan como un arreglo de strings

Palabras clave:

Revista x	Luna Córnea x	fotografía digital x	Gerardo Suter x	Pedro Meyer x	Fred Ritchin x
Claude Cahun x	Agustín Jiménez x	caballos x	Frida Kahlo x	Nahui Olin x	Bauhaus x
Werner Bischof x	Lola Álvarez Bravo x	Abbas x	Convocatoria Quinta Bienal de Fotografía 1993 x		
reseñas bibliográficas. x	Enter values				

keywords Palabras clave string 1		
Atributo	Valor	Tipo de Valor
editorType	MultiComboBoxItem	string
multiple	true	boolean
addUnknownValues	true	boolean
valueMap	{}	object
4 Registros		

//Section; uso de secciones con determinados campos que se expanden

Información Básica

tamaño de archivo

Título:

Descripción:

Palabras clave: Enter values

Institución: Centro de la Imagen

Lenguaje:

```
{ type:"section",defaultValue:"Información Básica",
  canCollapse:true,
  sectionExpanded:true,
  itemIds:[
    "recordtitle",
    "description",
    "keywords",
    "holder",
    "language"
  ]
},
{name:"recordtitle"},
{name:"description"},
{name:"keywords"},
{name:"holder"},
{name:"language"}
```

//Filtrar queries en DataProcessors por roles de usuario

```
function(request, dataSource, action){
  //your code
  var user=this.getUser();
  if(!user.roles.contains("admin"))
  {
    request.data["responsables"]=user._id;
  }
  return request;
}
```

//Filtrar queries complejo en DataProcessor validando roles y componentid

```
function(request, dataSource, action){
  if(request.componentId=="asignadas")
  {
    var or=[];
    if(user.roles.contains("admin"))
    {
      or.push({"responsable":{"$exists: true}});
    }else
    {
      or.push({"responsable":{"$ne:user._id"}});
      //request.data["responsable"]={$ne:user._id};
    }
    or.push({"id_beneficiario":{"$exists: true}});
    request.data["$or"]=or;
  }
  return request;
}
```

//Configurar columnas en la forma

Propiedades Extendidas de la Forma

Propiedad	Atributo	Tipo de Valor	Valor
	numCols	int	4
Correo electrónico, Password, N...	width	string	100%
	colWidths	object	[150,"",150,""]
Nombre Completo	rowSpan	int	2
Dirección	colSpan	int	3
Dirección	style	string	text
6 Registros			

Correo electrónico:	<input type="text" value="softjei@gmail.com"/>	Password:	<input type="password" value="*****"/>
Nombre Completo:	<input type="text" value="Javier Solis Gonzalez"/>	Roles:	<input type="text" value="Administrador, Programa..."/>
		Edad:	<input type="text"/>
Dirección:	<input type="text"/>		
Fecha de Creación:	13/02/2019	Usuario Creador:	softjei@gmail.com
Fecha de Modificación:	11/06/2019	Usuario Modificador:	softjei@gmail.com
		<input type="button" value="Regresar"/>	<input type="button" value="Guardar"/>

//Edicion de grid condicional

```
canEditCell : function (rowNum, colNum)
{
    var record = this.getRecord(rowNum);
    var fieldName = this.getFieldName(colNum);
    var rol=record?record.rol:this.getEditValues(rowNum).rol;

    console.log("canEditCell", record, fieldName, rol);

    if(rol)
    {
        if (fieldName == "estado_id" && rol == "admin")
        {
            return false;
        }else if (fieldName == "jurisdiccion_id" && (rol == "admin" || rol ==
"admin_edo"))
        {
            return false;
        }else if (fieldName == "unidad_id" && (rol == "admin" || rol == "admin_edo" ||
rol == "admin_jurisdiccion"))
        {
            return false;
        }
    }
    return this.Super("canEditCell", arguments);
}
```

//Evento al teminar edicion de celda

```
editorExit : function(editCompletionEvent, record, newValue, rowNum, colNum)
{
    console.log("editorExit", rowNum, colNum);
    this.refreshRow(rowNum);
}
```

//FormsProcessor

```
//init
function(request, dataSource, action, trxParams){
    //your code
    request.data.fullname="Jei Test";

    request.data._errors={
        "email":"Error email 2",
        "fullname": "Error name 2"
    };
    request.data._hints={
        "email":"Jei 2",
        "fullname": "Jei Solis 2"
    };
    return request;
}

//Change
function(request, dataSource, action, trxParams){
    //your code
    if(request.data.fullname!="Juan")
    {
        request.data._enabled={sexo:false};
        request.data._visible={updated:false};
        request.data._msg="El nombre debe de ser Juan";
    }else
    {
        request.data._enabled={sexo:true};
        request.data._visible={updated:true};
    }
    return request;
}
```

Otra base de datos (mongo) extra a la de defecto

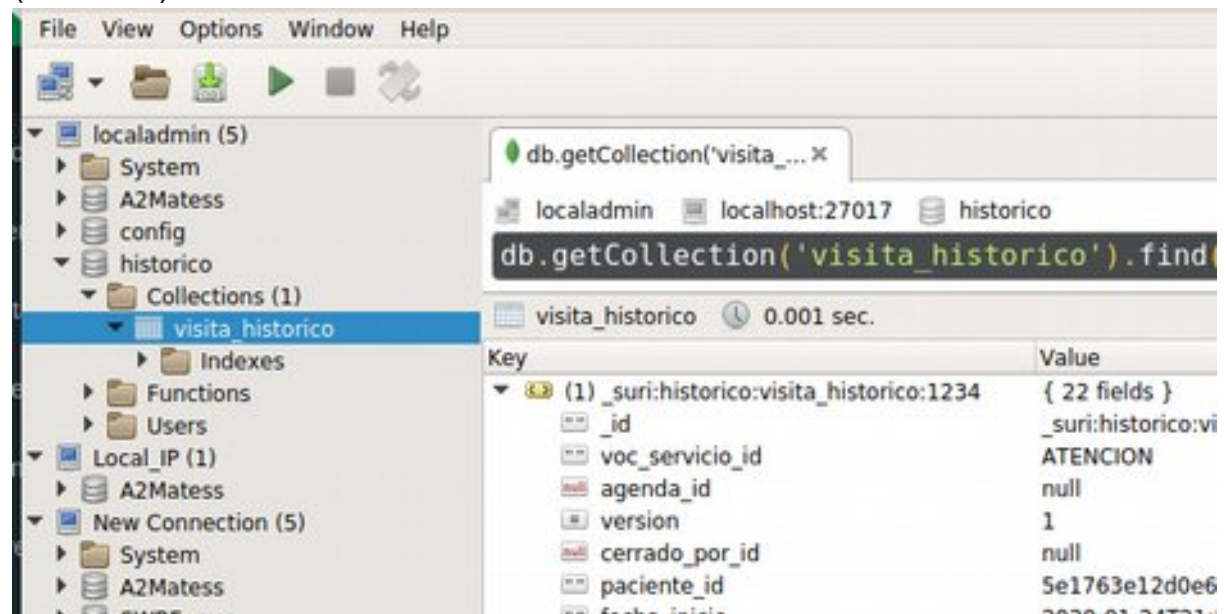
Configurar el DataStore (conexión) en el:

WEB-INF/global_base.js

(mongodb_remoto)



Crear o tener la Base de Datos
(historico)



Al dar de alta el DataSource no olvidar configurar:

Modelo <- Base de Datos
DataStore <- Conexión

