

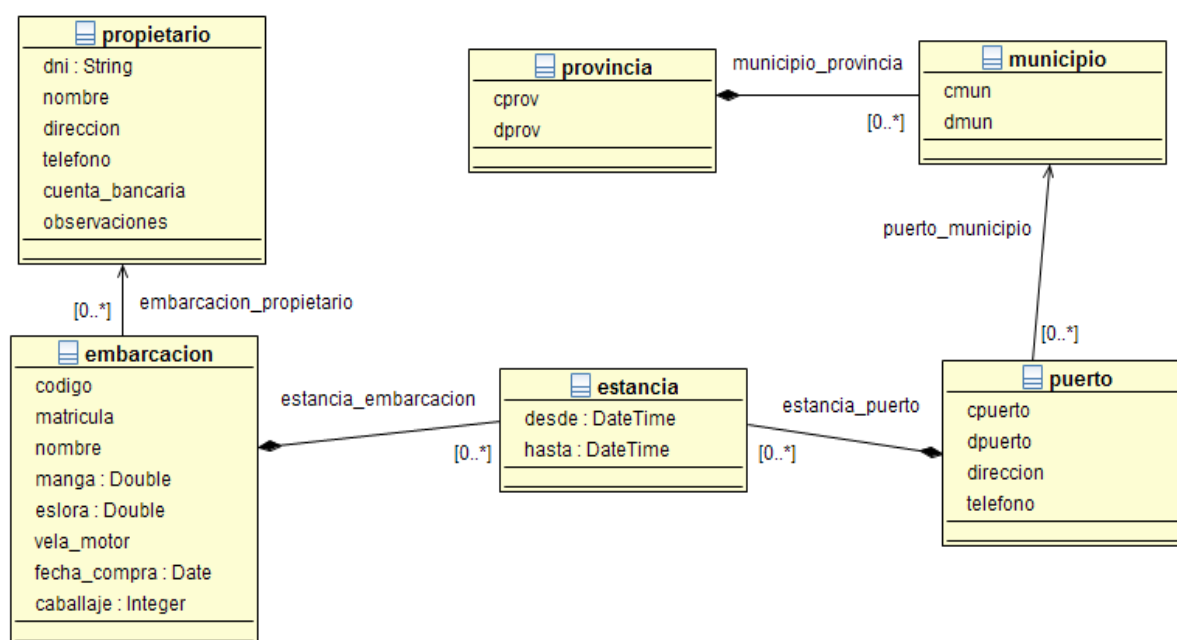
OBJETIVO

El objetivo de este documento es permitir al lector hacer una prueba inicial de las capacidades de la herramienta gvHIDRA. Para ello hemos seleccionado un enunciado sencillo con el que a partir de sus casos de uso (4 mantenimientos) se presentarán algunas de las características básicas del framework. Se trata de un enunciado “académico” ya que su propósito en exclusiva es la formación del lector por lo que puede que algunas de las restricciones no tengan sentido sacadas de contexto.

Tras terminar este tutorial, que no debe durar más de 1 hora, el usuario tendrá una pequeña aplicación funcional con la que iniciar su práctica con gvHIDRA.

ENUNCIADO: Embarcaciones Transeúntes

Se desea tener información de las embarcaciones que hacen escala en nuestros puertos. La información que tenemos de cada embarcación es un código, una matrícula, un nombre, un propietario, las dimensiones (manga y eslora), si tiene vela o motor, el caballaje, y la fecha del último cambio de titular. Los atributos código, matrícula y nombre son únicos. De cada estancia sabemos, además del puerto y la embarcación, la fecha y hora de entrada y de salida y donde está amarrado, con la restricción de que un barco sólo puede estar en un momento dado en un puerto. En el siguiente diagrama de clases se muestran más detalles:



 GENERALITAT VALENCIANA	CONSELLERIA D'INFRAESTRUCTURES I TRANSPORT		gvHIDRA	
	Proyecto: gvHIDRA		Proyecto: gvHIDRA	Versión: 1.0
			Fecha: 10/05/2011	Pág.: 2/25
TUTORIAL-EMPEZAR CON GVHIDRA				

ANÁLISIS/DISEÑO

La aplicación se va a descomponer en 4 ventanas y un listado.

1. Mantenimiento de Propietarios

Patrón de interfaz: Registro (FIL-EDI)

Requisitos:

- Se deben añadir máscaras en el registro (EDI) a teléfono y cuenta bancaria.
- El campo observaciones es un área de texto.

2. Mantenimiento de Provincias y Municipios

Patrón de interfaz: Maestro detalle Tabular-Tabular (FIL-LIS/LIS)

Requisitos:

- Las descripciones deben almacenarse en la BBDD en mayúsculas.

3. Mantenimiento de Embarcaciones

Patrón de interfaz: Tabular-Registro (FIL-LIS-EDI)

Requisitos:

- En el modo tabla no debe aparecer toda la información de la tabla. No interesa que aparezcan los siguientes campos manga, eslora, vela-motor y la fecha de compra.
- Comprobar que se hacen validaciones de tipos.
- El campo vela-motor debe ser de tipo radio.
- Creamos una ventana de selección para escoger un propietario.
- Validación que garantice que la matrícula y el nombre son únicos.
- Acción de interfaz. Cuando selecciona el campo vela-motor, muestra el campo caballaje sólo si el valor es motor.

4. Mantenimiento de Puertos y Estancias

Patrón de interfaz: Maestro detalle (FIL-EDI/LIS)

Requisitos:

- Creamos una lista desplegable dependiente para provincias-municipios en el maestro.
- Añadir una lista de Embarcaciones en el detalle.
- Añadir una validación que compruebe que el barco en esas fechas no está en otro puerto.

 GENERALITAT VALENCIANA	CONSELLERIA D'INFRAESTRUCTURES I TRANSPORT		gvHIDRA	
	Proyecto: gvHIDRA		Proyecto:	Versión:
			gvHIDRA	1.0
			Fecha:	Pág.:
		10/05/2011	3/25	
TUTORIAL-EMPEZAR CON GVHIDRA				

IMPLEMENTACIÓN

Para implementar la aplicación en este tutorial, iremos resolviendo cada una de las ventanas con sus respectivos requisitos, haciendo uso de la herramienta Genaro, para la generación de código de gvHIDRA.

Suponemos que se dispone de un entorno de trabajo gvHIDRA (en los ejemplos localhost) y de la plantilla base de aplicaciones. Para más información www.gvhidra.org. Ante cualquier duda o problema podéis dirigiros a nosotros en gvhidra@gva.es.

Creación de la Base de Datos

El primer paso para desarrollar la aplicación es tener disponible la base de datos. Para ello, hemos elaborado el siguiente script que crea el esquema, las tablas, los datos y los usuarios necesarios para trabajar. En este caso, hemos elegido como SGBD PostgreSQL; podemos utilizar Oracle o MySQL siempre que adaptemos el script (ver fichero dump.sql anexo)

Nota: se trata de una interpretación del enunciado, es susceptible de cambios por parte del lector.

Nota2: en el caso de corta pegar el contenido del script, tener cuidado con la codificación y con los retornos de carro que pueden provocar problemas inesperados.

Programación

Configuración del dns de conexión

Lo primero es configurar las conexiones para poder acceder a la Base de Datos que acabamos de crear. Para ello editamos el fichero *gvHidraConfig.inc.xml* ubicado en el raíz de nuestro proyecto.

Dentro de la zona *<DSNZone>* añadimos el siguiente código:

```
<DSNZone>
  <dbDSN id='g_dsn' sgb='pgsql'>
    <dbHost>localhost</dbHost>
    <dbPort>5432</dbPort>
    <dbDatabase>embarcacionesdb</dbDatabase>
    <dbUser>gvh_ae</dbUser>
    <dbPassword>gvh_ae</dbPassword>
  </dbDSN>

  <dbDSN id='gvh_dsn_log' sgb='postgres'>
    <dbHost>localhost</dbHost>
```

	CONSELLERIA D'INFRAESTRUCTURES I TRANSPORT	gvHIDRA	
	Proyecto: gvHIDRA	Proyecto: gvHIDRA	Versión: 1.0
		Fecha: 10/05/2011	Pág.: 4/25

TUTORIAL-EMPEZAR CON GVHIDRA

```

<dbPort>5432</dbPort>
<dbDatabase>embarcacionesdb</dbDatabase>
<dbUser>gvh_log</dbUser>
<dbPassword>gvh_log</dbPassword>
</dbDSN>
</DSNZone>

```

Nota: en el ejemplo se configura la conexión haciendo referencia a la ubicación en localhost del SGBD, si no es el caso, indicar el valor adecuado.

Ahora vamos a activar el debug del framework. Para ello, modificamos el Tag `logSettings` en el mismo archivo:

```
<logSettings status='LOG_ALL' dsnRef='gvh_dsn_log' />
```

Genaro: generador de código gvHidra

Para realizar los mantenimientos de la aplicación vamos a utilizar la herramienta Genaro, el proyecto está ubicado en la página web de gvHidra, en el punto de Descargas:

<http://www.gvhidra.org>

Una vez descargado lo descomprimos en el directorio “include” de la aplicación. Dentro del paquete hay un fichero “leeme.txt” con las instrucciones de uso.

Una vez hechas las instrucciones indicadas, accedemos a nuestra aplicación.

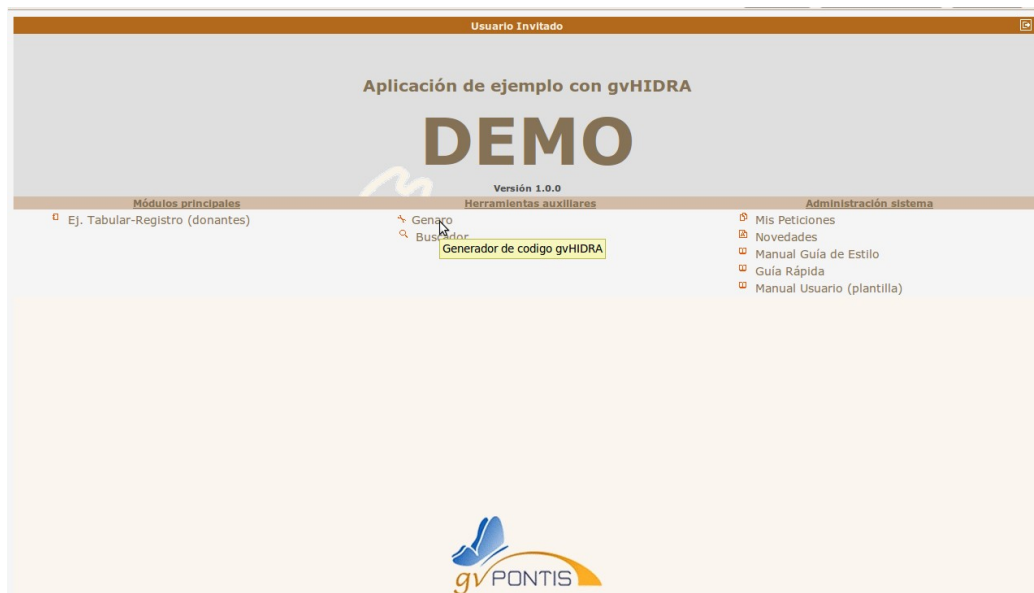
<http://localhost/Tutorial/>

Nota: esta referencia tiene en cuenta que el servidor web es localhost y el proyecto está ubicado en el directorio tutorial. Si no es así, ajuste los parámetros.

Nos aparecerá la página de validación, nos validamos con usuario “invitado” y password “1” y accederemos a la pantalla de entrada de la aplicación.

	CONSELLERIA D'INFRAESTRUCTURES I TRANSPORT	gvHIDRA	
	Proyecto:	gvHIDRA	Proyecto: gvHIDRA Versión: 1.0
			Fecha: 10/05/2011 Pág.: 5/25

TUTORIAL-EMPEZAR CON GVHIDRA



Ya podemos ver que en el menú Herramientas ya tenemos un acceso a Genaro.

MANTENIMIENTOS

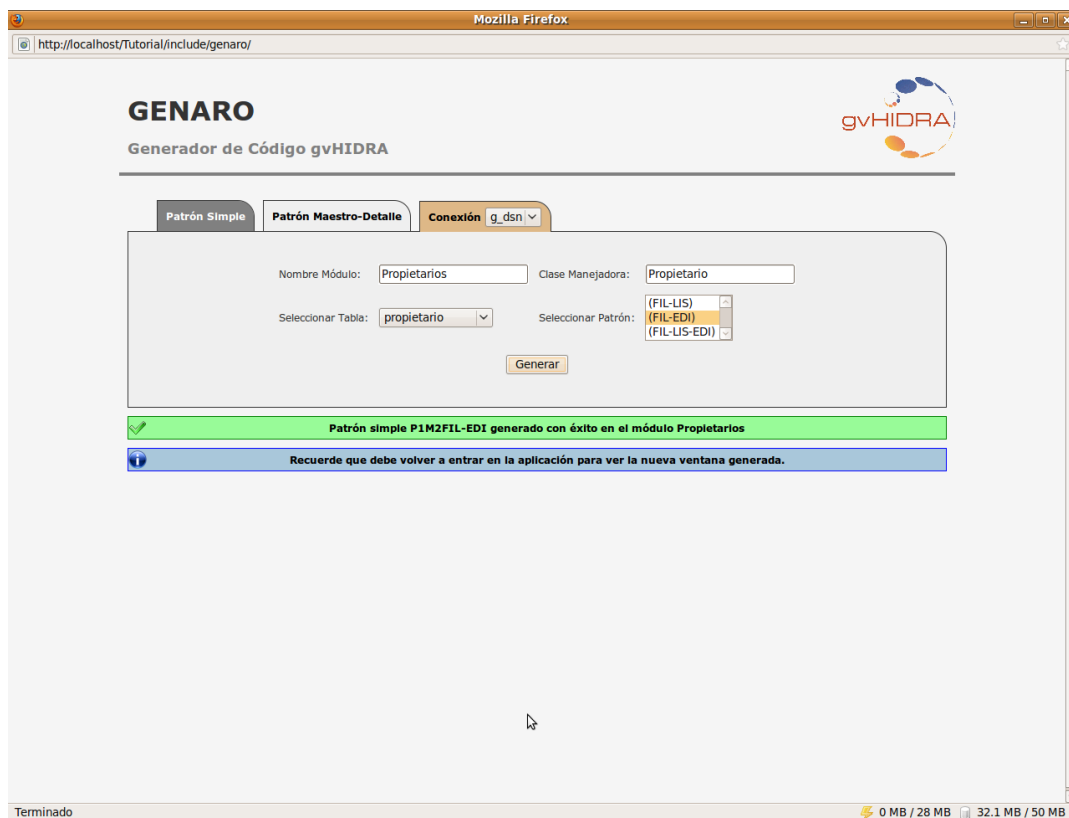
Lanzamos Genaro y lo primero que debemos hacer es elegir la conexión dsn que vamos a utilizar.



1.- Mantenimiento de propietarios

Parámetros del Genaro:

- Patrón : *FIL-EDI*. → *patrón simple*
- Nombre Módulo: Propietarios
- Clase Manejadora: *Propietario*.
- Tabla: *propietario* .



Ya podemos acceder a la ventana, desde la opción del menú de la pantalla principal. Podemos comprobar en la estructura de directorios que se han generado los diferentes ficheros que manejarán la ventana en cada uno de los directorios:

- actions
- include
- views
- plantillas

	CONSELLERIA D'INFRAESTRUCTURES I TRANSPORT	gvHIDRA	
	Proyecto: gvHIDRA	Proyecto: gvHIDRA	Versión: 1.0
		Fecha: 10/05/2011	Pág.: 7/25

TUTORIAL-EMPEZAR CON GVHIDRA

La herramienta nos crea todos los componentes de un mantenimiento básico de gvHIDRA. Partiendo de lo que ha generado tenemos que implementar los requisitos que marca el enunciado del análisis.

Requisitos:

1. Máscaras

<<Aplicar una máscara sobre los valores en los campos de texto teléfono y cuenta>>

Para aplicar una máscara a un campo debemos hacerlo a partir de su tipo en la Clase Manejadora. Para ello editamos el fichero generado *actions/Propietario.php*.

Buscamos la definición de tipo de datos de los campos teléfono. Antes de la asignación del tipo (*addFiledType*), aplicamos el método **setInputMask** a la variable *\$string* asignándoles la máscara *(+##)-#####* (cada símbolo # representa un dígito en esa posición). Esta máscara tiene que afectar tanto al campo definido en el panel de búsqueda como en el panel registro (*fil_telefono*, *edi_telefono*).

Lo mismo hacemos para el tipo de los campos de la cuenta cuenta bancaria con la siguiente máscara *#### #### ## #####*:

El código resultante es:

```
$string = new gvHidraString(false, 15);
$string->setInputMask('(+##)-#####');
$this->addFieldType('fil_telefono',$string);
$this->addFieldType('edi_telefono',$string);

$string = new gvHidraString(false, 23);
$string->setInputMask('#### #### ## #####');
$this->addFieldType('fil_cuenta_bancaria',$string);
$this->addFieldType('edi_cuenta_bancaria',$string);
```


2. Áreas de texto

<<el campo *observaciones* sea un área de texto en vez de un campo de texto simple>>

Para conseguirlo editamos el fichero *tpl* generado. Reemplazamos en el panel *'fil'* el siguiente código:

```
<td>{CWCampoTexto      textoAsociado="Observaciones"      nombre="fil_observaciones"
size="0"                maxlength="0"                      editable="true"
value=$defaultData_Propietario.fil_observaciones
dataType=$dataType_Propietario.fil_observaciones}</td>
por
```

```
<td>{CWAreaTexto      textoAsociado="Observaciones"      nombre="fil_observaciones"
rows="5"              cols="100"                        editable="true"
value=$defaultData_Propietario.fil_observaciones
dataType=$dataType_Propietario.fil_observaciones}</td>
```

	CONSELLERIA D'INFRAESTRUCTURES I TRANSPORT	gvHIDRA	
	Proyecto: gvHIDRA	Proyecto: gvHIDRA	Versión: 1.0
		Fecha: 10/05/2011	Pág.: 8/25

TUTORIAL-EMPEZAR CON GVHIDRA

Hacemos lo mismo el panel 'edi':

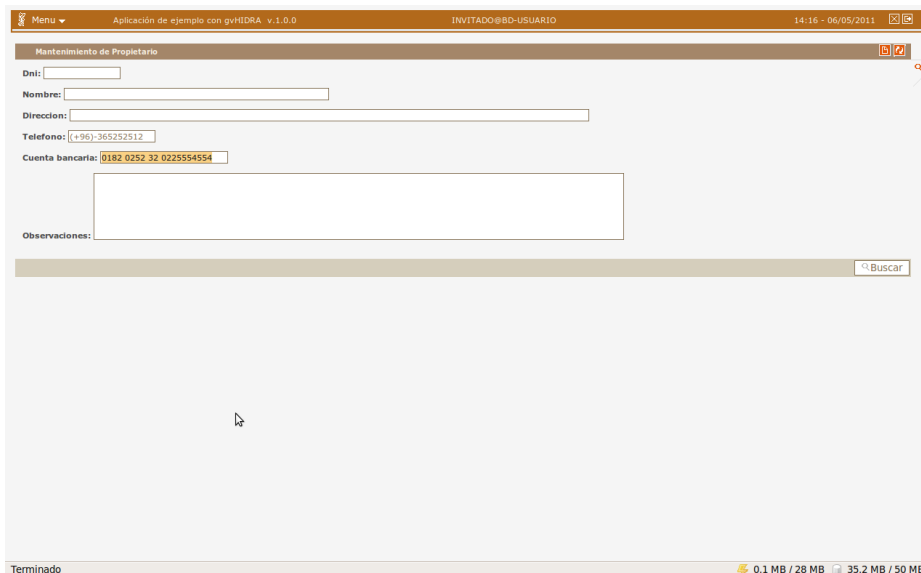
```
<td>{CWCampoTexto      textoAsociado="Observaciones"      nombre="edi_observaciones"
size=""                  maxlength=""                        editable="true"
value=$defaultData_Propietario.edi_observaciones
dataType=$dataType_Propietario.edi_observaciones}</td>
por
```

```
<td>{CWAreaTexto      textoAsociado="Observaciones"      nombre="edi_observaciones"
rows="5"              cols="100"                        editable="true"
value=$defaultData_Propietario.edi_observaciones
dataType=$dataType_Propietario.edi_observaciones}</td>
```

Por ultimo, fijamos el tamaño máximo, 500, del campo observaciones en la clase manejadora (*actions/Propietario.php*):

```
$string = new gvHidraString(false, 500);
$this->addFieldType('fil_observaciones',$string);
$this->addFieldType('edi_observaciones',$string);
```

La pantalla finalmente nos queda como la siguiente figura. Podemos comprobar que, por ejemplo, en el campo teléfono solo se pueden introducir dígitos, y que se estructuran según la máscara que hemos aplicado.



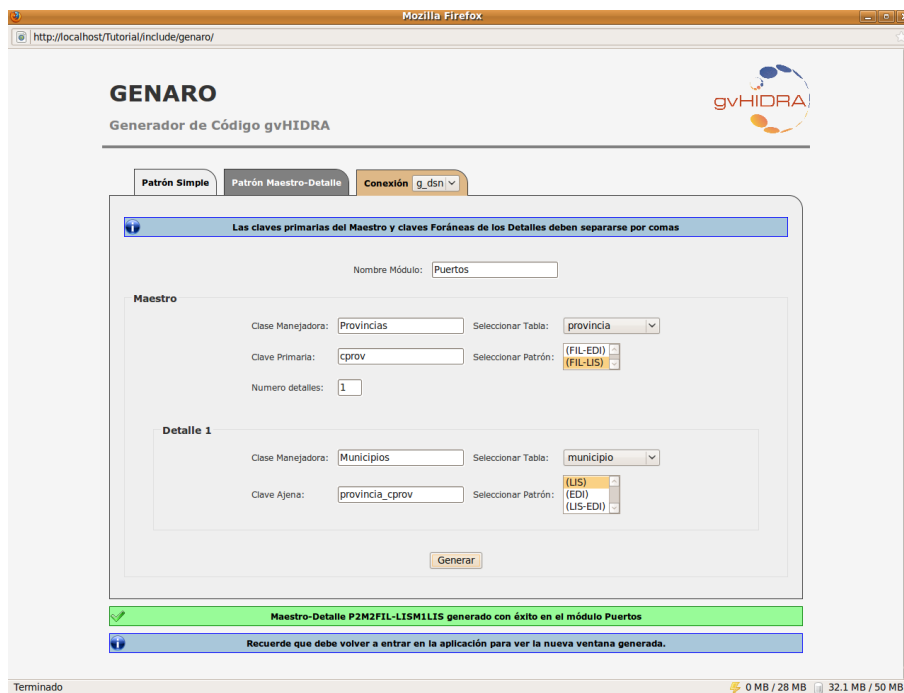
2.- Pantalla de Provincias y Municipios

- Patrón : Maestro(FIL-LIS) / Detalle(LIS) .
- Nombre Módulo: Puertos

	CONSELLERIA D'INFRAESTRUCTURES I TRANSPORT	gvHIDRA	
	Proyecto: gvHIDRA	Proyecto: gvHIDRA	Versión: 1.0
		Fecha: 10/05/2011	Pág.: 9/25

TUTORIAL-EMPEZAR CON GVHIDRA

- Clases Manejadoras: Maestro: Provincia, Detalle: Municipio .
- Tablas: provincia, municipio.



Requisitos

1.- preInsertar y preModificar

<<garantizar que se almacena en mayúsculas las descripciones de las provincias y de los municipios>>

– Para conseguir esto, vamos a implementar los métodos *preInsertar* y *preModificar*, tanto en el fichero *actions/Provincia.php* como en *actions/Municipio.php*.

En *actions/Provincia.php*:

```
public function preInsertar($objDatos)
{
    $m_datos = $objDatos->getAllTuplas();
    foreach($m_datos as $indice => $tupla)
    {
        $m_datos[$indice]["lis_dprov"] = strtoupper($tupla["lis_dprov"]);
    }
    $objDatos->setAllTuplas($m_datos);

    return 0;
}
```

	CONSELLERIA D'INFRAESTRUCTURES I TRANSPORT	gvHIDRA	
	Proyecto: gvHIDRA	Proyecto: gvHIDRA	Versión: 1.0
		Fecha: 10/05/2011	Pág.: 10/25

TUTORIAL-EMPEZAR CON GVHIDRA

```

public function preModificar($objDatos)
{
    $m_datos = $objDatos->getAllTuplas();
    foreach($m_datos as $indice => $tupla)
    {
        $m_datos[$indice]["lis_dprov"] = strtoupper($tupla["lis_dprov"]);
    }
    $objDatos->setAllTuplas($m_datos);

    return 0;
}

```

En *actions/Municipio.php*:

```

public function preInsertar($objDatos)
{
    $m_datos = $objDatos->getAllTuplas();
    foreach($m_datos as $indice => $tupla)
    {
        $m_datos[$indice]["lis_dmun"] = strtoupper($tupla["lis_dmun"]);
    }
    $objDatos->setAllTuplas($m_datos);

    return 0;
}

public function preModificar($objDatos)
{
    $m_datos = $objDatos->getAllTuplas();
    foreach($m_datos as $indice => $tupla)
    {
        $m_datos[$indice]["lis_dmun"] = strtoupper($tupla["lis_dmun"]);
    }
    $objDatos->setAllTuplas($m_datos);

    return 0;
}

```

	CONSELLERIA D'INFRASTRUCTURES I TRANSPORT		gvHIDRA	
	Proyecto: <div style="text-align: center;">gvHIDRA</div>		Proyecto:	gvHIDRA
			Fecha:	10/05/2011
			Versión:	1.0
			Pág.:	11/25

TUTORIAL-EMPEZAR CON GVHIDRA

La pantalla resultante es:

Menu

Aplicación de ejemplo con gvHIDRA v.1.0.0

INVITADO@BO-USUARIO

14:33 - 06/05/2011

Provincias

☐

Cprov

☒

03

ALICANTE

☐

12

CASTELLÓN

☐

46

VALENCIA

Dprov

(Nº reg. 3) Pag. 01 de 01

Municipios

☐

Cmun

☐

03700

Dénia

☐

03739

Jávea

Dmun

Provincia cprov

03

03

(Nº reg. 2) Pag. 01 de 01

Terminado

0.1 MB / 28 MB35.3 MB / 50 MB

3.- Pantalla de Embarcaciones

- Patrón : (FIL-LIS-EDI) → patrón simple
- Nombre Módulo: Puertos
- Clase Manejadora: Embarcacion.
- Tabla: embarcacion



Requisitos

1.- Simplificar el patrón tabular

<<En el modo tabular que no aparezcan los campos *manga*, *eslora*, *vela_motor* y *fecha_compra*>>

–Modificamos la variable `$str_select` del fichero `actions/Embarcacion.php`:

```
$str_select = "SELECT codigo as \"lis_codigo\", matricula as \"lis_matricula\", nombre as \"lis_nombre\", caballaje as \"lis_caballaje\", propietario_dni as \"lis_propietario_dni\" FROM embarcacion";
```

–De los *matchings* (llamadas `$this->addMatching("lis_*,)`), en la sección del modo listar en tabla (panel *lis*), quitamos las líneas correspondientes a los atributos eliminados. La sección quedará de este modo:

```
$this->addMatching("lis_codigo","codigo","embarcacion");
```

	CONSELLERIA D'INFRAESTRUCTURES I TRANSPORT	gvHIDRA	
	Proyecto:	gvHIDRA	Proyecto: gvHIDRA
			Versión: 1.0
		Fecha: 10/05/2011	Pág.: 13/25

TUTORIAL-EMPEZAR CON GVHIDRA

```
$this->addMatching("lis_matricula","matricula","embarcacion");

$this->addMatching("lis_nombre","nombre","embarcacion");

$this->addMatching("lis_caballaje","caballaje","embarcacion");

$this->addMatching("lis_propietario_dni","propietario_dni","embarcacion");
```

– Finalmente, borramos del panel *lis* los campos *lis_manga*, *lis_eslora*, *lis_vela_motor* y *lis_fecha_compra*. Con ello la *tpl* quedará:

```
{CWCampoTexto textoAsociado="Codigo" nombre="lis_codigo" size="10"
editable="true" oculto="false" value=$defaultData_Embarcacion.lis_codigo
dataType=$dataType_Embarcacion.lis_codigo}
{CWCampoTexto textoAsociado="Matricula" nombre="lis_matricula" size="12"
editable="true" oculto="false" value=$defaultData_Embarcacion.lis_matricula
dataType=$dataType_Embarcacion.lis_matricula}
{CWCampoTexto textoAsociado="Nombre" nombre="lis_nombre" size="50"
editable="true" oculto="false" value=$defaultData_Embarcacion.lis_nombre
dataType=$dataType_Embarcacion.lis_nombre}
{CWCampoTexto textoAsociado="Caballaje" nombre="lis_caballaje" size="4"
editable="true" oculto="false" value=$defaultData_Embarcacion.lis_caballaje
dataType=$dataType_Embarcacion.lis_caballaje}
{CWCampoTexto textoAsociado="Propietario dni" nombre="lis_propietario_dni"
size="13" editable="true" oculto="false"
value=$defaultData_Embarcacion.lis_propietario_dni
dataType=$dataType_Embarcacion.lis_propietario_dni}
```

2.- Lista tipo radioButton

<<convertir el campo de texto campo vela/motor en un radioButton>>

En gvHidra, un radioButton es realmente una lista *gvHidraList* (*CWLista* en la *tpl*) marcándole el tipo con el método *setRadio()*. Para definirla hacemos los siguientes pasos:

– Primero, como vamos a redefinir el tipo de este campo, pasa de ser un campo de texto a una lista, previamente debemos eliminar la definición de tipo que se nos ha generado por defecto en el constructor de la clase manejadora (archivo *actions/Embarcacion.php*). Eliminar las siguientes líneas del constructor:

```
$this->addFieldType('fil_vela_motor',$string);

$this->addFieldType('edi_vela_motor',$string);
```

– Ahora ya podemos definir la lista en el constructor. Hay que añadirlas tanto para el panel *fil* como para el panel *edi*.

```
$listaVelaMotor_fil = new gvHidraList('fil_vela_motor');
$listVelaMotor_fil->setRadio(true);
$listVelaMotor_fil->addOption('v','Vela');
```

 GENERALITAT VALENCIANA	CONSELLERIA D'INFRAESTRUCTURES I TRANSPORT		gvHIDRA	
	Proyecto: gvHIDRA		Proyecto:	Versión:
			gvHIDRA	1.0
			Fecha:	Pág.:
			10/05/2011	14/25
TUTORIAL-EMPEZAR CON GVHIDRA				

```

$listaVelaMotor_fil->addOption('m','Motor');
$listaVelaMotor_fil->setSelected('m');
$this->addList($listaVelaMotor_fil);

$listaVelaMotor_edt = new gvHidraList('edt_vela_motor');
$listaVelaMotor_edt->setRadio(true);
$listaVelaMotor_edt->addOption('v','Vela');
$listaVelaMotor_edt->addOption('m','Motor');
$listaVelaMotor_edt->setSelected('m');
$this->addList($listaVelaMotor_edt);

```

Nota: la llamada a setSelected fija un valor por defecto que se utilizará cuando la lista se represente sin un respaldo de datos. Típicamente esto ocurre cuando estamos en el modo de búsqueda o en el modo de inserción.

– En “plantillas/Embarcacion/p_Embarcacion.tpl” reemplazamos en el panel fil:

```

<td>{CWCampoTexto    textoAsociado="Vela    motor"    nombre="fil_vela_motor"
size="1"                                maxlength="1"                                editable="true"
value=$defaultData_Embarcacion.fil_vela_motor
dataType=$dataType_Embarcacion.fil_vela_motor}</td>

```

por

```

<td>{CWLista    nombre="fil_vela_motor"    textoAsociado="Vela    /    motor"
editable="true"                                datos=$defaultData_Embarcacion.fil_vela_motor
dataType=$dataType_Embarcacion.fil_vela_motor}</td>

```

–En el panel edt reemplazamos:

```

<td>{CWCampoTexto    textoAsociado="Vela    motor"    nombre="edt_vela_motor"
size="1"                                maxlength="1"                                editable="true"
value=$defaultData_Embarcacion.edt_vela_motor
dataType=$dataType_Embarcacion.edt_vela_motor}</td>

```

por

```

<td>{CWLista    nombre="edt_vela_motor"    textoAsociado="Vela    /    motor"
editable="true"                                datos=$defaultData_Embarcacion.edt_vela_motor
dataType=$dataType_Embarcacion.edt_vela_motor}</td>

```

3.- Visibilidad de campos

<<hacer visible el campo de texto caballaje cuando se selecciona la opción motor, y oculto si se selecciona la opción vela >>

–Incluimos dos acciones de interfaz (una para fil, y otra para edt) en el constructor de actions/Embarcacion.php.:

```

$this->addAccionInterfaz('fil_vela_motor','visibilidadCaballaje');

```

	CONSELLERIA D'INFRAESTRUCTURES I TRANSPORT	gvHIDRA	
	Proyecto: gvHIDRA	Proyecto: gvHIDRA	Versión: 1.0
		Fecha: 10/05/2011	Pág.: 15/25

TUTORIAL-EMPEZAR CON GVHIDRA

```
$this->addAccionInterfaz('edi_vela_motor','visibilidadCaballaje');
```

– Implementamos el método *visibilidadCaballaje* que, como vemos en el código anterior, es el encargado de manejar el evento de interfaz. Concretamente tiene que cambiar el item seleccionado de la lista vela / motor :

```
public function visibilidadCaballaje($objDatos)
{
    //si estamos en el panel fil:
    if($objDatos->getModoActivo()=='fil')
    {
        //obtenemos el valor del campo fil_vela_motor
        //y si es motor, ponemos a visible el campo fil_caballaje
        $seleccionado = $objDatos->getValue('fil_vela_motor');
        if($seleccionado=='m')
            $objDatos->setVisible('fil_caballaje',true);
        else
            $objDatos->setVisible('fil_caballaje',false);
    }

    //si estamos en el panel edi:
    else if ($objDatos->getModoActivo()=='edi')
    {
        //obtenemos el valor del campo edi_vela_motor
        //y si es motor, ponemos a visible el campo edi_caballaje
        $seleccionado = $objDatos->getValue('edi_vela_motor');
        if($seleccionado=='m')
            $objDatos->setVisible('edi_caballaje',true);
        else
            $objDatos->setVisible('edi_caballaje',false);
    }

    return 0;
}
```

– Por último, en la *tpl* añadimos al final de los campos *fil_vela_motor* y *edi_vela_motor* la directiva *actualizaA=<nombre del campo que se actualizara>* para que la acción de interfaz se lance cuando se produzca el evento:

```
<td>{CWLista  nombre="fil_vela_motor"  textoAsociado="Vela  /  motor"
editable="true"                      datos=$defaultData_Embarcacion.fil_vela_motor
dataType=$dataType_Embarcacion.fil_vela_motor
actualizaA="fil_caballaje"}</td>
```

```
<td>{CWLista  nombre="edi_vela_motor"  textoAsociado="Vela  /  motor"
editable="true"                      datos=$defaultData_Embarcacion.edi_vela_motor
```

	CONSELLERIA D'INFRAESTRUCTURES I TRANSPORT	gvHIDRA	
	Proyecto: gvHIDRA	Proyecto: gvHIDRA	Versión: 1.0
		Fecha: 10/05/2011	Pág.: 16/25

TUTORIAL-EMPEZAR CON GVHIDRA

```

dataType=$dataType_Embarcacion.edi_vela_motor
actualizaA="edi_caballaje"></td>

```

4.- Comprobación en preInsertar

<<verificar que los campos de texto matricula y nombre son únicos>>

-Implementamos el siguiente código en el método *preInsertar* que ya aparece predefinido en *actions/Embarcacion.php*:

```

public function preInsertar($objDatos)
{
    //obtenemos el valor actual en el campo edi_nombre
    $nombre=$objDatos->getValue('edi_nombre');
    //consulta para obtener el número de registros con el mismo nombre
    $consulta = "select count(nombre) as \"num\"from embarcacion where
nombre='\". $nombre. \"'";
    $res = $this->consultar($consulta);

    if($res[0]['num']>0)
    {
        $this->showMessage('APL-01',array($nombre));
        return -1;
    }

    //hacemos lo mismo para el campo edi_matricula
    $matricula=$objDatos->getValue('edi_matricula');

    $consulta = "select count(matricula) as \"num\"from embarcacion where
matricula='\". $matricula. \"'";
    $res = $this->consultar($consulta);
    if($res[0]['num']>0)
    {
        $this->showMessage('APL-02',array($matricula));
        return -1;
    }

    return 0;
}

```

- Finalmente, tenemos que definir los mensajes *APL-01* y *APL-02*. Para ello, en el fichero *mensajes.php*, en el raíz del proyecto, encontramos un array (*g_mensajesParticulares*) que contendrá todos los mensajes que se necesiten para la aplicación.

```

'APL-01'=>array('descCorta'=>'Embarcación existente.', 'descLarga'=>'No se pudo
realizar la inserción porque la embarcación %0% ya está dada de

```



```
alta.', 'tipo'=>'ERROR'),
'APL-02'=>array('descCorta'=>'Embarcación existente .', 'descLarga'=>'No se pudo
realizar la inserción porque la embarcación con matrícula %0% ya está dada de
alta.', 'tipo'=>'ERROR')
```

5.- Ventana de selección

<<añadir una ventana de selección de propietarios al lado del campo de texto dni del propietario>>

– Primero tenemos que definir la fuente de datos, para ello vamos al fichero *actions/principal/AppMainWindow.php*, en el método *AppMainWindow()* hay que añadir la definición de la ventana de selección. La definición se hace mediante el método *setSelectionWindow_DBSource*:

```
$conf->setSelectionWindow_DBSource('PROPIETARIOS', 'select dni as "dni", nombre
as "nombre" from propietario');
```

– Tenemos que definir las ventanas de selección en la clase manejadora correspondiente (*actions/Embarcacion.php*). Esta definición se declara en el constructor, hay que añadir las dos, una para el panel fil y otra para el edi, con el método *gvHidraSelectionWindow()*. Como primer parámetro hay que indicar el nombre del campo en la *tpl*, campo asociado a la ventana y cuyo contenido se actualizará con el valor elegido. Hay que añadir el *match* necesario para relacionar ese campo con la columna correspondiente en la tabla *Embarcacion* de la BD:

```
$propietarios_fil = new
gvHidraSelectionWindow('fil_propietario_dni', 'PROPIETARIOS');
$propietarios_fil->addMatching('fil_propietario_dni', 'dni');
$this->addSelectionWindow($propietarios_fil);

$propietarios_edi = new
gvHidraSelectionWindow('edi_propietario_dni', 'PROPIETARIOS');
$propietarios_edi->addMatching('edi_propietario_dni', 'dni');
$this->addSelectionWindow($propietarios_edi);
```

– En *p_Embarcacion.tpl* añadimos al lado de los campos de dni del propietario (y dentro de los mismos tags <td></td>) las definiciones de los botones *tooltip* que abrirán las ventanas de selección:

```
<td>

{CWCampoTexto textoAsociado="DNI Propietario"
nombre="fil_propietario_dni" size="13" editable="true"
value=$defaultData_Embarcacion.fil_propietario_dni
dataType=$dataType_Embarcacion.fil_propietario_dni}
```

	CONSELLERIA D'INFRAESTRUCTURES I TRANSPORT	gvHIDRA	
	Proyecto: gvHIDRA	Proyecto: gvHIDRA	Versión: 1.0
		Fecha: 10/05/2011	Pág.: 18/25

TUTORIAL-EMPEZAR CON GVHIDRA

```
{CWBotonTooltip imagen="13" titulo="Busqueda de Propietarios" funcion="abrirVS"
actuaSobre="fil_propietario_dni"
formActua="fil" panelActua="fil" claseManejadora="Embarcacion"}
</td>
```

```
<td>

{CWCampoTexto textoAsociado="DNI Propietario" nombre="edi_propietario_dni"
size="13" editable="true" value=$defaultData_Embarcacion.edi_propietario_dni
dataType=$dataType_Embarcacion.edi_propietario_dni}

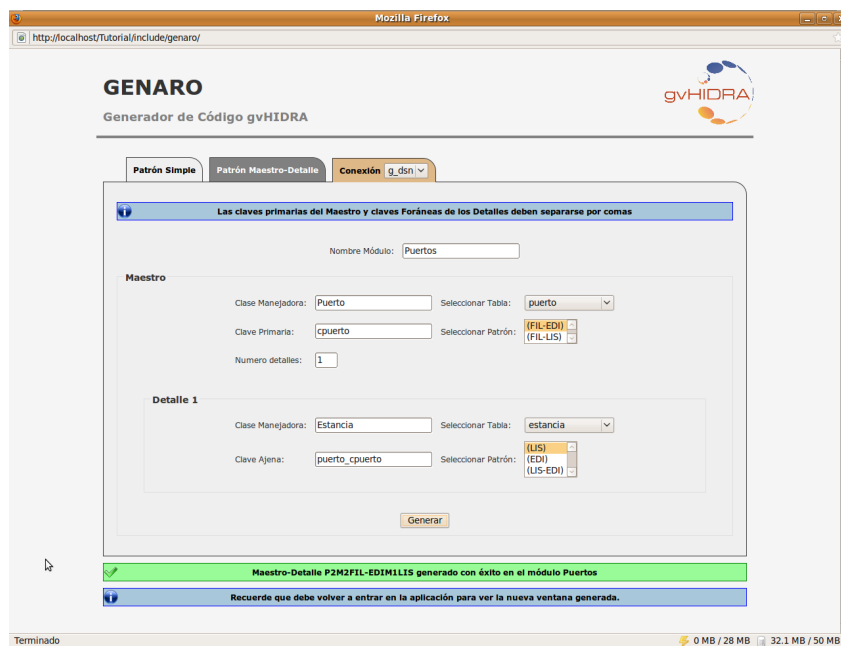
{CWBotonTooltip imagen="13" titulo="Busqueda de Propietarios" funcion="abrirVS"
actuaSobre="edi_propietario_dni"
formActua="edi" panelActua="FichaEdicion" claseManejadora="Embarcacion"}
</td>
```

Con todo esto, la ventana de selección de propietarios resultante es:



4.- Pantalla de Puertos y Estancias

- Patrón: P2M2 (FIL-EDI)M1(LIS)
- Nombre Módulo: Puertos
- Clases Manejadoras: Maestro: Puerto - Detalle: Estancia
- Tablas: puerto, estancia



Requisitos

1.- Listas dependientes

<<crear una lista desplegable dependiente de provincias y municipios en el panel Maestro (panel puertos)>>

- Editamos el fichero *plantillas/p_Puerto.tpl*, y añadimos al panel maestro dos campos de tipo lista (una para provincia y otra para municipios). Tenemos que:

- Eliminar el campo de texto *fil_municipio_cmun*
- En la definición de la lista de provincias añadimos el atributo "*actualizaA*" de manera que la lista de provincia actualice la de municipios cuando cambia el ítem seleccionado en '*fil_provincia_cprov*'.

```
<tr>
<td>{CWLista      nombre="fil_provincia_cprov"      textoAsociado="Provincia"
editable="true"      datos=$defaultData_Puerto.fil_provincia_cprov
dataType=$dataType_Puerto.fil_provincia_cprov
actualizaA="fil_municipio_cmun"}</td>
</tr>
```

	CONSELLERIA D'INFRASTRUCTURES I TRANSPORT	gvHIDRA	
	Proyecto: gvHIDRA	Proyecto: gvHIDRA	Versión: 1.0
		Fecha: 10/05/2011	Pág.: 21/25

TUTORIAL-EMPEZAR CON GVHIDRA

```
<tr>
<td>{CWLista      nombre="fil_municipio_cmun"      textoAsociado="Municipio"
editable="true"      datos=$defaultData_Puerto.fil_municipio_cmun
dataType=$dataType_Puerto.fil_municipio_cmun}</td>
</tr>
```

De manera similar procedemos a añadir las dos listas al panel edi del maestro:

```
<tr>
<td>{CWLista      nombre="edi_provincia_cprov"      textoAsociado="Provincia"
editable="true"      datos=$defaultData_Puerto.edi_cod_prov
dataType=$dataType_Puerto.edi_
actualizaA="edi_municipio_cmun"}</td>
</tr>

<tr>
<td>{CWLista      nombre="edi_municipio_cmun"      textoAsociado="Municipio"
editable="true"      datos=$defaultData_Puerto.edi_municipio_cmun
dataType=$dataType_Puerto.edi_municipio_cmun}</td>
</tr>
```

- Pasamos a la clase manejadora, editamos el fichero *actions/Puerto.php*. Primero tenemos que eliminar la definición que nos ha generado tanto para el campo *fil_municipio_cmun* como para el *edi_municipio_cmun*, eliminamos las siguientes líneas:

```
$string = new gvHidraString(false, 5);
$this->addFieldType('fil_municipio_cmun',$string);
$this->addFieldType('edi_municipio_cmun',$string);
```

- Ahora añadiremos las definiciones de las dos listas (tanto en el panel '*fil*' como en el '*edi*') dentro del constructor de la clase, y especificando también las dependencias entre las listas:

```
$listaProvincias_fil = new gvHidraList('fil_provincia_cprov','PROVINCIAS');
$listaProvincias_fil->addOption("", "");
$this->addList($listaProvincias_fil);

$listaMunicipios_fil = new gvHidraList('fil_municipio_cmun','MUNICIPIOS');
$listaMunicipios_fil->addOption("", "");
$listaMunicipios_fil->setDependence(array('fil_provincia_cprov'),array('municipio.provincia_cprov'))
;
$this->addList($listaMunicipios_fil);

$listaProvincias_edi = new gvHidraList('edi_provincia_cprov','PROVINCIAS');
$listaProvincias_edi->addOption("", "");
$this->addList($listaProvincias_edi);

$listaMunicipios_edi = new gvHidraList('edi_municipio_cmun','MUNICIPIOS');
$listaMunicipios_edi->addOption("", "");
$listaMunicipios_edi->
```

	CONSELLERIA D'INFRAESTRUCTURES I TRANSPORT	gvHIDRA	
	Proyecto: gvHIDRA	Proyecto: gvHIDRA	Versión: 1.0
		Fecha: 10/05/2011	Pág.: 22/25

TUTORIAL-EMPEZAR CON GVHIDRA

```
>setDependence(array('edi_provincia_cprov'),array('municipio.provincia_cprov'))
;
$this->addList($listaMunicipios_edi);
```

– Hay que crear las definiciones de PROVINCIAS y MUNICIPIOS para que el framework cargue estas listas. Para ello añadimos en el fichero “*principal/AppMainWindow.php*” las definiciones de las consultas SQL (PROVINCIAS, MUNICIPIOS). Recordad que los alias deben ser *valor* y *descripcion*.

```
$conf->setList_DBSource('PROVINCIAS','select cprov as "valor", dprov as
"descripcion" from provincia');
$conf->setList_DBSource('MUNICIPIOS','select cmun as "valor", dmun as
"descripcion" from municipio');
```

–Por último debemos modificar la select de edición (*\$str_select*). Si comprobamos el diseño de la tabla, no tenemos referencia de la provincia del puerto sino es a partir del municipio. Por ello, para disponer de este dato, debemos incorporarlo a la consulta. Para ello, modificamos la select añadiendo la referencia a la provincia a través de la tabla municipio. Añadimos la where necesaria para que el join funcione correctamente (método *setWhereForSearchQuery*).

```
$str_select = "SELECT cpuerto as \"cpuerto\", dpuerto as \"edi_dpuerto\",
direccion as \"edi_direccion\", telefono as \"edi_telefono\", municipio_cmun as
\"edi_municipio_cmun\", provincia_cprov as \"edi_provincia_cprov\" FROM puerto,
municipio";
$this->setSelectForSearchQuery($str_select);

$str_where = "municipio.cmun = puerto.municipio_cmun";
$this->setWhereForSearchQuery($str_where);
```

2.- Lista desplegable

<<añadir en el detalle (Estancias) una lista para seleccionar una embarcación>>

–Primero editamos el fichero “*plantillas/p_Puerto.tpl*”, y modificamos el campo de texto *lis_embarcacion* para que sea una lista. Borramos el campo de texto.

```
{CWCampoTexto textoAsociado="Embarcacion codigo" nombre="lis_embarcacion_codigo"
..
```

y lo sustituimos por:

```
{CWLista textoAsociado="Embarcacion codigo" nombre="lis_embarcacion_codigo"
datos=$defaultData_Estancia.lis_embarcacion_codigo editable="true"
dataType=$dataType_Estancia.lis_embarcacion_codigo}
```

-Eliminamos el tipo que tenemos definido por defecto para el campo 'lis_embarcacion_codigo'.

```
$string = new gvHidraString(false,10);
$this->addFieldType('lis_embarcacion_codigo',$string);
```

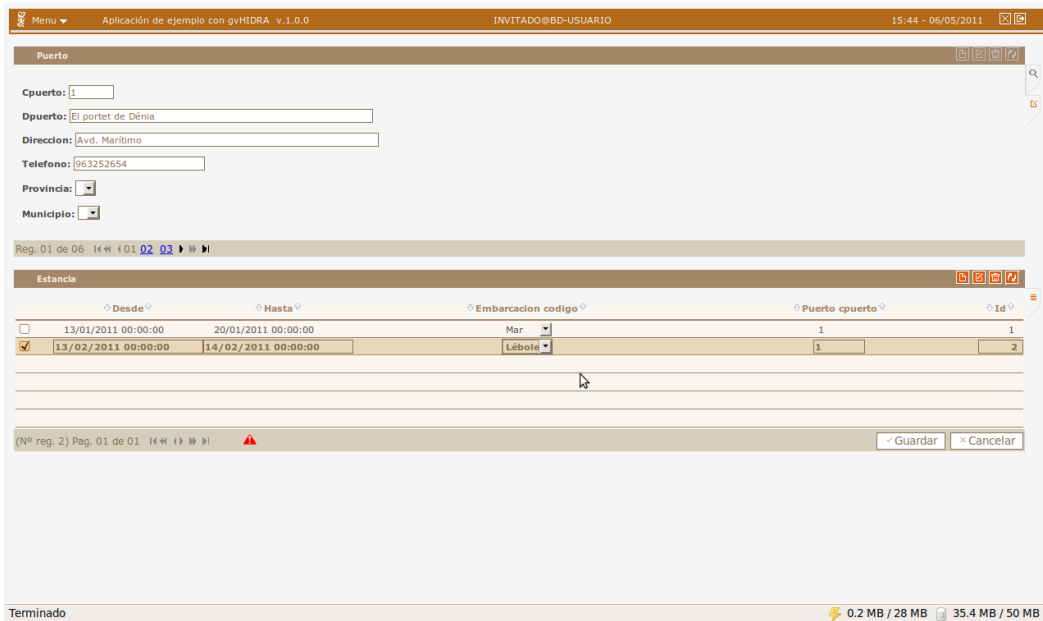
-Y ahora definimos la lista de embarcaciones en el constructor de la clase manejadora en el fichero "actions/Estancia.php"

```
$listaEmbarcaciones = new
gvHidraList('lis_embarcacion_codigo','EMBARCACIONES');
$this->addList($listaEmbarcaciones);
```

-Y por último declaramos la consulta *SQL* en el fichero "principal/AppMainWindow.php"

```
$conf->setList_DBSource('EMBARCACIONES','select codigo as "valor", nombre as
"descripcion" from embarcacion');
```

La pantalla de puertos y estancias resultante sera:



3.- Comprobación en preInsertar

<<comprobar que durante el periodo de una nueva instancia, no está el barco ubicado en otro puerto.>>

- Editamos la clase manejadora del detalle (*Estancia.php*) e incluimos un método que calcule si la embarcación está ubicada en otro puerto durante las fechas indicadas, *existeEstancia()*.

```
public function existeEstancia($puerto, $embarcacion, $desde, $hasta)
```

```
{
    $consulta = "SELECT count(*) as \"ocupado\" FROM estancias WHERE
embarcacion_codigo = '\".$embarcacion.\"' and puerto_cpuerto <> '\".$puerto.\"' AND
(desde BETWEEN '\".$desde.\"' AND '\".$hasta.\"') OR (hasta BETWEEN '\".$desde.\"'
AND '\".$hasta.\"')";
    $res = $this->consultar($consulta);
    if ($res[0]['ocupado']>0)
        return 0;
    else
        return -1;
}
```

- Antes de una inserción o modificación tendremos que comprobar esta restricción, para ello en los métodos *preInsertar* y *preModificar* que ya aparecen predefinidos, insertamos el siguiente código.

```
public function preInsertar($objDatos)
{
    $mDatos = $objDatos->getAllTuplas();
    foreach($mDatos as $indice=>$tupla)
    {
        $puerto = $tupla['lis_puerto_cpuerto'];
        $embarcacion = $tupla['lis_embarcacion_codigo'];
        $desde = $tupla['lis_desde'];
        $hasta = $tupla['lis_hasta'];
        // Preparamos la fecha para adaptarla al tipo de la BD
        $desde = $this->getConnection()->prepararFecha($desde);
        $hasta = $this->getConnection()->prepararFecha($hasta);
        $ocupado = $this->existeEstancia($puerto,$embarcacion,$desde,
$hasta);
        if ($ocupado == 0)
        {
            $this->showMessage('APL-03',array($embarcacion));
            return -1;
        }
    }
    return 0;
}

public function preModificar($objDatos)
{
    $mDatos = $objDatos->getAllTuplas();
    foreach($mDatos as $indice=>$tupla)
    {
        $puerto = $tupla['lis_puerto_cpuerto'];
        $embarcacion = $tupla['lis_embarcacion_codigo'];
        $desde = $tupla['lis_desde'];
        $hasta = $tupla['lis_hasta'];
        // Preparamos la fecha para adaptarla al tipo de la BD
        $desde = $this->getConnection()->prepararFecha($desde);
        $hasta = $this->getConnection()->prepararFecha($hasta);
        $ocupado = $this->existeEstancia($puerto,$embarcacion,$desde,
$hasta);
    }
}
```


	CONSELLERIA D'INFRAESTRUCTURES I TRANSPORT	gvHIDRA	
	Proyecto:	gvHIDRA	Versión: 1.0
		Fecha: 10/05/2011	Pág.: 25/25

TUTORIAL-EMPEZAR CON GVHIDRA

```

        if ($ocupado == 0)
        {
            $this->showMessage('APL-04',array($embarcacion));
            return -1;
        }
    }
    return 0;
}

```

-Como hemos añadido que se mostrará un mensaje al usuario alertándole del problema tenemos que crearlos ('APL-03' y 'APL-04') en el fichero *mensajes.php*.

```

'APL-03'=>array('descCorta'=>'Error al insertar','descLarga'=>'No se puede
realizar la inserción porque la embarcación %0% para esas fechas se encuentra
en otro puerto','tipo'=>'ERROR'),
'APL-04'=>array('descCorta'=>'Error al modificar','descLarga'=>'No se puede
realizar la inserción porque la embarcación %0% para esas fechas se encuentra
en otro puerto','tipo'=>'ERROR'),

```

Nota: Para todas las pantallas, en los archivos *.tpl (dentro del directorio plantillas), se ha modificado el atributo *textoAsociado* de los campos de texto, introduciendo un texto mas presentable (el generador por defecto le asigna los nombres de las columnas en la BD).

Historial de versiones

Versión	Fecha	Autor	Comentarios
1.0	10-05-2011	Toni Félix	