

The image shows a file tree on the left and a code editor on the right.

**File Tree:**

- assets
- commands
- config
  - console.php
  - db.php
  - params.php
  - test.php
  - test\_db.php
  - web.php
- controllers
- mail
- models
- runtime
- tests
- vendor
- views
- web
- .bowerrc
- .gitignore
- /\* codeception.yml
- /\* composer.json
- composer.lock
- LICENSE.md
- README.md
- requirements.php
- yii

**Code Editor (db.php):**

```
<?php  
1 1  
2 2  
3 3 return [  
4 4     'class' => 'yii\db\Connection',  
5 5     'dsn' => 'mysql:host=localhost;dbname=proyecto',  
6 6     'username' => 'root',  
7 7     'password' => '',  
8 8     'charset' => 'utf8',  
9 9 ];  
10 10
```

**Text Overlay:**

Conexión a la Base de Datos

Modificar

# Welcome to Gii a magical tool that can write code for you

Start the fun with the following code generators:

## Model Generator

This generator generates an ActiveRecord class for the specified database table.

[Start »](#)

## CRUD Generator

This generator generates a controller and views that implement CRUD (Create, Read, Update, Delete) operations for the specified data model.

[Start »](#)

## Controller Generator

This generator helps you to quickly generate a new controller class with one or several controller actions and their corresponding views.

[Start »](#)

## Form Generator

This generator generates a view script file that displays a form to collect input for the specified model class.

[Start »](#)

## Generadores

## Module Generator

This generator helps you to generate the skeleton code needed by a Yii module.

[Start »](#)

## Extension Generator

This generator helps you to generate the files needed by a Yii extension.

[Start »](#)

# Creando el modelo

[Home](#)[Help](#)[Application](#)

Model Generator

CRUD Generator

Controller Generator

Form Generator

Module Generator

Extension Generator

## Model Generator

This generator generates an ActiveRecord class for the specified database table.

**Table Name**

table

**Model Class**

Table

**Namespace**

app\models

**Base Class**

yii\db\ActiveRecord

**Database Connection ID**

dbc

 **Use Table Prefix****Generate Relations**

All relations

Se coloca una tabla  
de la Base de Datos

Se coloca el nombre del  
modelo en Mayúsculas

Preview

# Creando el modelo

Home

Help

Application

Model Generator

CRUD Generator

Controller Generator

Form Generator

Module Generator

Extension Generator

## Model Generator

This generator generates an ActiveRecord class for the specified database table.

**Table Name**

table

**Model Class**

Table

Preview

Generate

**Preview****Generate**

Se genera el Modelo  
asociado a la tabla

Click on the above **Generate** button to generate the files selected below:

 Create  Unchanged  Overwrite**Code File**models\ **Modelo**.php**Action**

create



# Creando el CRUD

[Home](#)[Help](#)[Application](#)

- [Model Generator](#)
- [CRUD Generator](#)
- [Controller Generator](#)
- [Form Generator](#)
- [Module Generator](#)
- [Extension Generator](#)

## CRUD Generator

This generator generates a controller and views that implement CRUD (Create, Read, Update, Delete) operations for the specified data model.

**Model Class**`app\models\ Modelo`**Nombre del modelo****Search Model Class**`app\models\ ModeloSearch`**Nombre del buscador****Controller Class**`app\controllers\ ModeloController`**Nombre del controlador****View Path****Preview****Base Controller Class**`yii\web\Controller`**Widget Used in Index Page**`GridView`



# Creando el CRUD

Home Help Application

Model Generator



CRUD Generator



Controller Generator



Form Generator



Module Generator



Extension Generator



## CRUD Generator

This generator generates a controller and views that implement CRUD (Create, Read, Update, Delete) operations for the specified data model.

### Model Class

app\models\ Modelo

### Search Model Class

app\models\ ModeloSearch

### Controller Class

app\controllers\ ModeloController

### View Path

### Base Controller Class

yii\web\Controller

### Widget Used in Index Page

GridView

Preview

Generate

Se generan  
los archivos

Model Generator



CRUD Generator



Controller Generator



Form Generator



Module Generator



Extension Generator



# CRUD Generator

# CRUD

This generator generates a controller and views that implement CRUD (Create, Read, Update, Delete) operations for the specified data model.

## Model Class

## Search Model Class

## Controller Class

## View Path




Click on the above **Generate** button to generate the files selected below

### Code File

 controllers\ **Modelo** Controller.php

 models\ **Modelo** Search.php

 views\ **modelo**\\_form.php

 views\ **modelo**\\_search.php

 views\ **modelo**\\_create.php

 views\ **modelo**\index.php

 views\ **modelo**\update.php

 views\ **modelo**\view.php

Se genera los  
archivos del CRUD

 Create  Unchanged  Overwrite

### Action

create

create

create

create

create

create

create

create

# TextInput

## Create Oferta

ID

Instituto

Plan Estudio

Modalidad

Vigencia

Autorizacion

Oficio

Con  
TextInput

Create

Regresar

# \_form.php

```
<?php  
use yii\helpers\Html;  
use yii\widgets\ActiveForm;  
use yii\helpers\ArrayHelper;  
  
/* @var $this yii\web\View */  
/* @var $model app\models\Oferta */  
/* @var $form yii\widgets\ActiveForm */  
?>
```

```
<div class="oferta-form">
```

```
    <?php $form = ActiveForm::begin(); ?>
```

```
    <?= $form->field($model, 'OFE_ID')->textInput() ?>
```

```
    <?= $form->field($model, 'OFE_INSTITUTO')->textInput() ?>
```

```
    <?= $form->field($model, 'OFE_PLAN_ESTUDIO')->textInput() ?>
```

```
    <?= $form->field($model, 'OFE_MODALIDAD')->textInput() ?>
```

```
    <?= $form->field($model, 'OFE_VIGENCIA')->textInput() ?>
```

```
    <?= $form->field($model, 'OFE_AUTORIZACION')->textInput(['maxlength' => true]) ?>
```

```
    <?= $form->field($model, 'OFE_OFICIO')->textInput(['maxlength' => true]) ?>
```

```
    <div class="form-group">
```

```
        <?= Html::submitButton($model->isNewRecord ? 'Create' : 'Update', ['class' => $model->isNewRecord ? 'btn btn-success' : 'btn btn-primary']) ?>  
        <?= Html::a('Regresar', Yii::$app->request->referrer, ['class' =>'btn btn-warning']) ?><br/><br/>
```

```
    </div>
```

```
    <?php ActiveForm::end(); ?>
```

# TextInput

Con  
TextInput

# DropDownList

## Create Oferta

ID

Instituto

Seleccione uno

Plan Estudio

Seleccione uno

Modalidad

Seleccione una

Seleccione una

Presencial

**Abierta**

Virtual

Mixta

Autorización

Oficio

Con  
DropDownList

Create

Regresar

&lt;?php

```
use yii\helpers\Html;
use yii\widgets\ActiveForm;
use yii\helpers\ArrayHelper;
use app\models\Instituto;
use app\models\Plan;
```

```
/* @var $this yii\web\View */
/* @var $model app\models\Oferta */
/* @var $form yii\widgets\ActiveForm */
```

```
$institutos = ArrayHelper::map(Instituto::find()->all(), 'INS_ID', 'INS_CORTO');
$planes = ArrayHelper::map(Plan::find()->all(), 'PLA_ID', 'PLA_NOMBRE');
$modalidades = ['Presencial' => 'Presencial', 'Abierta' => 'Abierta', 'Virtual' => 'Virtual', 'Mixta' => 'Mixta'];
$vigencias = ['Vigente' => 'Vigente', 'Liquidación' => 'Liquidación', 'Cancelada' => 'Cancelada'];
```

?&gt;

```
<div class="oferta-form">
```

```
    <?php $form = ActiveForm::begin(); ?>
```

```
    <?= $form->field($model, 'OFE_ID')->textInput() ?>
```

```
    <?= $form->field($model, 'OFE_INSTITUTO')->dropDownList($institutos, ['prompt' => 'Seleccione uno']) ?>
```

```
    <?= $form->field($model, 'OFE_PLAN_ESTUDIO')->dropDownList($planes, ['prompt' => 'Seleccione uno']) ?>
```

```
    <?= $form->field($model, 'OFE_MODALIDAD')->dropDownList($modalidades, ['prompt' => 'Seleccione una']) ?>
```

```
    <?= $form->field($model, 'OFE_VIGENCIA')->dropDownList($vigencias, ['prompt' => 'Seleccione una']) ?>
```

```
    <?= $form->field($model, 'OFE_AUTORIZACION')->textInput(['maxlength' => true]) ?>
```

```
    <?= $form->field($model, 'OFE_OFICIO')->textInput(['maxlength' => true]) ?>
```

# DropDownList

Llave  
Primaria

Nombre

&lt;?php

```
use yii\helpers\Html;
use yii\widgets\ActiveForm;
use yii\helpers\ArrayHelper;
use app\models\Instituto;
use app\models\Plan;
```

```
/* @var $this yii\web\View */
/* @var $model app\models\Oferta */
/* @var $form yii\widgets\ActiveForm */
```

```
$institutos = ArrayHelper::map(Instituto::find()->all(), 'INS_ID', 'INS_CORTO');
$planes = ArrayHelper::map(Plan::find()->all(), 'PLA_ID', 'PLA_NOMBRE');
$modalidades = ['Presencial' => 'Presencial', 'Abierta' => 'Abierta', 'Virtual' => 'Virtual', 'Mixta' => 'Mixta'];
$vigencias = ['Vigente' => 'Vigente', 'Liquidación' => 'Liquidación', 'Cancelada' => 'Cancelada'];
```

?&gt;

```
<div class="oferta-form">

    <?php $form = ActiveForm::begin(); ?>

    <?= $form->field($model, 'OFE_ID')->textInput() ?>
    <?= $form->field($model, 'OFE_INSTITUTO')->dropDownList($institutos, ['prompt' => 'Seleccione uno']) ?>
    <?= $form->field($model, 'OFE_PLAN_ESTUDIO')->dropDownList($planes, ['prompt' => 'Seleccione uno']) ?>
    <?= $form->field($model, 'OFE_MODALIDAD')->dropDownList($modalidades, ['prompt' => 'Seleccione una']) ?>
    <?= $form->field($model, 'OFE_VIGENCIA')->dropDownList($vigencias, ['prompt' => 'Seleccione una']) ?>
    <?= $form->field($model, 'OFE_AUTORIZACION')->textInput(['maxlength' => true]) ?>
    <?= $form->field($model, 'OFE_OFICIO')->textInput(['maxlength' => true]) ?>
```

Llave  
Primaria

Nombre

# DropDownList

# DropDownList

## DEPENDIENTES

### Create Oferta

ID

Instituto

Seleccione uno



Plan Estudio

Seleccione uno

Modalidad

Seleccione una

Vigencia

Seleccione una

Autorizacion

Oficio

Create

Regresar

# DropDownList DEPENDIENTES

**form.php**

```
use app\models\Plan;
/* @var $this yii\web\View */
/* @var $model app\models\Oferta */
/* @var $form yii\widgets\ActiveForm */
$institutos = ArrayHelper::map(Instituto::find()->all(), 'INS_ID', 'INS_CORTO');
$planes = ArrayHelper::map(Plan::find()->all(), 'PLA_ID', 'PLA_NOMBRE');
$modalidades = ['Presencial' => 'Presencial', 'Abierta' => 'Abierta', 'Virtual' => 'Virtual', 'Mixta' => 'Mixta'];
$vigencias = ['Vigente' => 'Vigente', 'Liquidación' => 'Liquidación', 'Cancelada' => 'Cancelada'];
?>
<div class="oferta-form">
<?php $form = ActiveForm::begin(); ?>
<?= $form->field($model, 'OFE_ID')->textInput() ?>
<?= $form->field($model, 'OFE_INSTITUTO')->dropDownList($institutos, [
    'prompt' => 'Seleccione uno',
    'onchange' =>
        $.post("/oferta/planes?instituto="+$(this).val(),
            function( data ) {
                $("select#oferta-ofe_plan_estudio").html( data );
            }
        );
    ]) ?>
<?= $form->field($model, 'OFE_PLAN_ESTUDIO')->dropDownList($planes, ['prompt' => 'Seleccione uno']) ?>
<?= $form->field($model, 'OFE_MODALIDAD')->dropDownList($modalidades, ['prompt' => 'Seleccione una']) ?>
```

**/controller/action?parámetro=valor**  
**"oferta/planes?instituto=" + ...**

**tipo#id del field que se actualizará:**  
**select#oferta-ofe\_plan\_estudio**

# OfertaController.php

```
* @param integer $id
* @return Oferta the loaded model
* @throws NotFoundHttpException if the model cannot be found
*/
protected function findModel($id)
{
    if (($model = Oferta::findOne($id)) !== null) {
        return $model;
    } else {
        throw new NotFoundHttpException('The requested page does not exist.');
    }
}
```

```
public function actionPlanes($instituto)
{
    $planes = \app\models\Plan::find()
        ->where(['PLA_INS' => $instituto])
        ->orderBy('PLA_ID ASC')
        ->all();
    echo '<option value="">Seleccione uno</option>';
    foreach($planes as $plan) {
        echo "<option value='{$plan->PLA_ID}'>{$plan->PLA_NOMBRE}</option>\n";
    }
}
```

## DropDownList DEPENDIENTES

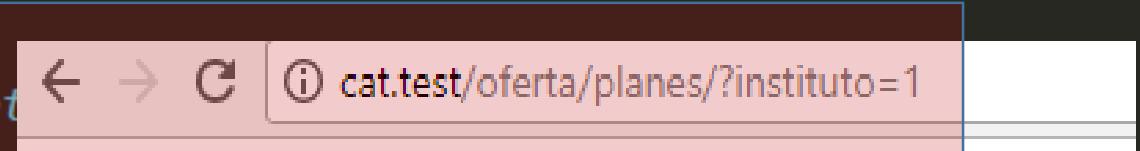
**actionPlanes(\$parámetro)**  
\$instituto es el parámetro

# OfertaController.php

```
* @param integer $id
* @return Oferta the loaded model
* @throws NotFoundHttpException if the model cannot be found
*/
protected function findModel($id)
{
    if (($model = Oferta::findOne($id)) !== null) {
        return $model;
    } else {
        throw new NotFoundHttpException('The requested page does not exist.');
    }
}

public function actionPlanes($instituto)
{
    $planes = \app\models\Plan::find()
        ->where(['PLA_INS' => $instituto])
        ->orderBy('PLA_ID ASC')
        ->all();
    echo '<option value="">Seleccione uno';
    foreach($planes as $plan) {
        echo "<option value='{$plan->PLA_ID}'>" . $plan->PLA_NOMBRE;
    }
}
```

## DropDownList DEPENDIENTES



Seleccione uno  
Arquitectura 2010  
Contador 2010  
Ambiental 2010  
Bioquímica 2010  
Civil 2010  
Eléctrica 2010  
Electromecánica 2010  
Electrónica 2010  
Acuicultura 2010  
Administración Ing 2010  
Agronomía 2010  
Gastronomía 2010

```
public function behaviors()
{
    return [
        'verbs' => [
            'class' => VerbFilter::className(),
            'actions' => [
                'delete' => ['POST'],
            ],
        ],
        'access' => [
            'class' => \yii\filters\AccessControl::className(),
            'only' => ['index', 'create', 'update', 'view', 'planes'],
            'rules' => [
                [
                    'allow' => true,
                    'roles' => ['@'],
                ],
            ],
        ],
    ];
}
```

# DropDownList DEPENDIENTES

Para leer más sobre **access** y **rules** consultar la liga:  
<http://www.yiiframework.com/doc-2.0/guide-security-authorization.html>

# DropDownList EDITABLES

Correo

Tipo

Seleccione Uno ▾

Estado

Ta ▾

Estado de México

Querétaro

Quintana Roo

Tabasco

Tamaulipas

# \_form.php

## DropDownList EDITABLES

```
<?= $form->field($model, 'INS_CORTO')->textInput(['maxlength' => true]) ?>

<?= $form->field($model, 'INS_CLAVE')->textInput(['maxlength' => true]) ?>

<?= $form->field($model, 'INS_CORREO')->textInput(['maxlength' => true]) ?>

<?= $form->field($model, 'INS_TIPO')->dropDownList($tipos, ['prompt' => 'Seleccione Uno']) ?>

<?= $form->field($model, 'INS_ESTADO')->textInput(['list' => 'INS_ESTADO']) ?>
<datalist id="INS_ESTADO"> ← →
    <?php foreach ($estados as $key => $value): ?>
        <option value="<?= $value ?>">
    <?php endforeach ?>
</datalist>

<?= $form->field($model, 'INS_CIUDAD')->textInput(['maxlength' => true]) ?>
```

## Administración de Catálogos

Planes

**Boton 1**

Texto del Boton. Está algo largo para que se justifique automáticamente.

Oferta  
Educativa

**Boton 2**

Texto del Boton. Con la imagen centrada de color verde.

Profesores

**Boton 3**

Texto del Boton. Con la imagen justificada a la izquierda.

Carreras  
Boton 4

Texto del Boton. Con la imagen ocupando toda el área.

Carreras profesores

**Boton 5**

Texto del Boton. Con dos imágenes colocadas en una posición específica.

**Boton 6**

Texto del Boton. No especificado. Queda en color gris y sin imagen.

**Boton 6**

Texto del Boton. Se cambia el color a este botón, y queda sin imagen.

Botones del  
Menú

```
.botones {
    align: center;
    width: 200px;
    height: 200px;
    padding: 100px 10px 10px 100px;
    border: 1px solid #ddd;
    cursor: pointer;
    display: inline-block;
    background-size: 50% 50%;
    background-repeat: no-repeat;
    background-position: center top;
    border-radius: 10px;
}
.botones:nth-child(1) {background-color: #A02227; background-image: url(../img/planes.png);}
.botones:nth-child(2) {background-color: #348838; background-image: url(../img/oferta.png);}
.botones:nth-child(3) {background-color: #4CA0F5; background-image: url(../img/profesores.png);
    background-position: 10% 5%;}
.botones:nth-child(4) {background-color: #58A9AC; background-image: url(../img/carreras.png); background-size: cover;}
.botones:nth-child(5) {background-color: #D27628; background-image: url(../img/carreras.png), url(../img/otro.png);
    background-size: 50% 50%, 50% 50%; background-position: 10% 10%, 80% 10%;}
.botones h1 {
    text-align: center;
    color: white;
    font-family: Tahoma;
    font-size: 30px;
}
.botones p {
    text-align: justify;
    text-justify: inter-word;
}
.botones:hover {
    -moz-transform: scale(1.1);
    -webkit-transform: scale(1.1);
    transform: scale(1.1);
}
```

# Botones

## menu.css

```
menu.css
FOLDERS
  cat
  assets
  commands
  config
  mail
  runtime
  tests
  vendor
  views
  web
    assets
    css
      /* menu.css */
      /* site.css */
      .botones {
        align: center;
        width: 200px;
        height: 200px;
        padding: 100px 10px 10px 10px;
        border: 1px solid #ddd;
        cursor: pointer;
        display: inline-block;
        background-size: 50% 50%;
        background-repeat: no-repeat;
        background-position: center top;
        border-radius: 10%;
      }
      .botones:nth-child(1) {background-color: #A02227; background-image: url(..../img/planes.png);}
      .botones:nth-child(2) {background-color: #348B38; background-image: url(..../img/oferta.png);}
      .botones:nth-child(3) {background-color: #4CA0F5; background-image: url(..../img/profesores.png);
        background-position: 10% 5%;}
      .botones:nth-child(4) {background-color: #58A9AC; background-image: url(..../img/carreras.png);
        background-size: cover;}
      .botones:nth-child(5) {background-color: #D2762B;
        background-image: url(..../img/carreras.png), url(..../img/otro.png);
        background-size: 50% 50%, 50% 50%; background-position: 10% 10%, 80% 10%;}
      .botones h1 {
        text-align: center;
        color: white;
        font-family: Tahoma;
        font-size: 30px;
      }
      .botones p {
        text-align: justify;
        text-justify: inter-word;
      }
      .botones:hover {
        -moz-transform: scale(1.1);
        -webkit-transform: scale(1.1);
        transform: scale(1.1);
      }
```

Para leer más sobre **css**, consultar la liga  
<https://www.w3schools.com/css/>

Para elegir el botón con la clase: **planes**, modificar  
**.botones[class='botones planes'] { ... }**

Para leer más sobre los **selectores y plantillas**, consultar las ligas  
[https://www.w3schools.com/cssref/css\\_selectors.asp](https://www.w3schools.com/cssref/css_selectors.asp)  
[https://www.w3schools.com/css/css\\_rwd\\_templates.asp](https://www.w3schools.com/css/css_rwd_templates.asp)

# Botones

FOLDERS

- ▶ cat
- ▶ assets
  - AppAsset.php
- ▶ commands
- ▶ components
- ▶ config
- ▶ controllers
- ▶ mail
- ▶ models
- ▶ runtime
- ▶ tests
- ▶ vendor
- ▶ views
- ▶ web
- ▶ widgets
- .bowerrc
- .gitignore
- /\* codeception.yml
- /\* composer.json
- composer.lock
- <> LICENSE.md
- <> README.md
- requirements.php
- yii
- /\* yii.bat

```
AppAsset.php
1 <?php
2 /**
3 * @link http://www.y
4 * @copyright Copyright (c) 2012 Qiang Xue
5 * @license http://www.y
6 */
7
8 namespace app\assets;
9
10 use yii\web\AssetBundle;
11
12 /**
13 * Main application asset bundle.
14 *
15 * @author Qiang Xue <qiang.xue@gmail.com>
16 * @since 2.0
17 */
18 class AppAsset extends AssetBundle
19 {
20     public $basePath = '@webroot';
21     public $baseUrl = '@web';
22     public $css = [
23         'css/site.css',
24         'css/menu.css'],
25     ];
26     public $js = [
27         'js/funciones.js',
28     ];
29     public $depends = [
30         'yii\web\YiiAsset',
31         'yii\bootstrap\BootstrapAsset',
32     ];
33 }
```

**'css/archivo.css'**

**'js/archivo.js'**

# Botones

The image shows a code editor interface with a sidebar titled "FOLDERS". The sidebar lists various project files and folders. A green arrow points from the word "index.php" in the sidebar to the file tab in the editor. The main area of the editor displays the "index.php" file content.

```
index.php
1 <?php
2 use yii\helpers\Html;
3 /* @var $this yii\web\View */
4 $this->title = 'Administración de Catálogos';
5 ?>
6 <div class="botones">
7     <a href="#">
8         <h1>Botón 1</h1><p><b>Texto del Botón.</b> Está algo largo para que se justifique automáticamente.</p>
9     </a>
10 </div>
11 <div class="botones">
12     <a href="#">
13         <h1>Botón 2</h1><p><b>Texto del Botón.</b> Con la imagen centrada de color verde.</p>
14     </a>
15 </div>
16 <div class="botones">
17     <a href="#">
18         <h1>Botón 3</h1><p><b>Texto del Botón.</b> Con la imagen justificada a la izquierda.</p>
19     </a>
20 </div>
21 <div class="botones">
22     <a href="">
23         <h1>Botón 4</h1><p><b>Texto del Botón.</b> Con la imagen ocupando toda el área.</p>
24     </a>
25 </div>
26 <div class="botones">
27     <a href="">
28         <h1>Botón 5</h1><p><b>Texto del Botón.</b> Con dos imágenes colocadas en una posición específica.</p>
29     </a>
30 </div>
31 <div class="botones">
32     <a href="">
33         <h1>Botón 6</h1><p><b>Texto del Botón.</b> No especificado. Queda en color gris y sin imagen.</p>
34     </a>
35 </div>
36 <div class="botones" style="background-color: #9A35A3 ;">
37     <a href="">
38         <h1>Botón 6</h1><p><b>Texto del Botón.</b> Se cambia el color a este botón, y queda sin imagen.</p>
39     </a>
40 </div>
```

# Java Script

```
function planes(instituto) {
    bootbox.dialog({
        size: 'small',
        message: "<p><img src='/img/loading.gif' width='50' height='50'>Espere un momento...</p>",
    });

    if(instituto !== '') {
        $.post('/oferta/planes', { _csrf:csrfToken, datos:{ instituto:instituto } })
            .done(function(data) {
                $("select#oferta-ofe_plan_estudio").html( data );
            })
            .fail(function(f) {
                console.log(f.responseText);
            });
    } else {
        bootbox.alert({
            size: 'small',
            message: "Se necesita seleccionar un instituto.",
        });
    }
}
```

Código desde  
DropDownList  
DEPENDIENTES

archivo.js

FOLDERS

- cat
- assets
- commands
- config
- controllers
- mail
- models
- runtime
- tests

- vendor
- web
  - assets
  - css
  - img
  - js
    - funciones.js

- favicon.ico
- index-test.php
- index.php
- robots.txt

- .bowerrc
- .gitignore

- codeception.yml
- composer.json
- composer.lock
- LICENSE.md
- README.md
- requirements.php
- yii
- yii.bat

Se pueden colocar las funciones de **JQuery** en un **archivo.js**

Para leer más sobre **JQuery**, consultar la liga  
<https://jquery.com/>

# DropDownList DEPENDIENTES

**form.php**

```
1 <?php
2
3 use yii\helpers\Html;
4 use yii\widgets\ActiveForm;
5 use yii\helpers\ArrayHelper;
6 use app\models\Instituto;
7 use app\models\Plan;
8
9 /* @var $this yii\web\View */
10 /* @var $model app\models\Oferta */
11 /* @var $form yii\widgets\ActiveForm */
12
13 $institutos = ArrayHelper::map(Instituto::find()->all(), 'INS_ID', 'INS_CORTO');
14 $planes = ArrayHelper::map(Plan::find()->all(), 'PLA_ID', 'PLA_NOMBRE');
15 $modalidades = ['Presencial' => 'Presencial', 'Abierta' => 'Abierta', 'Virtual' => 'Virtual', 'Mixta' => 'Mixta'];
16 $vigencias = ['Vigente' => 'Vigente', 'Liquidación' => 'Liquidación', 'Cancelada' => 'Cancelada'];
17
18 ?>
19
20 <div class="oferta-form">
21
22     <?php $form = ActiveForm::begin(); ?>
23
24     <?= $form->field($model, 'OFE_ID')->textInput() ?>
25
26     <?= $form->field($model, 'OFE_INSTITUTO')->dropDownList($institutos, [
27         'prompt' => 'Seleccione uno',
28         'onchange' => 'planes($(this).val())']) ?>
29
30     <?= $form->field($model, 'OFE_PLAN_ESTUDIO')->dropDownList($planes, ['prompt' => 'Seleccione uno']) ?>
31
32     <?= $form->field($model, 'OFE_MODALIDAD')->dropDownList($modalidades, ['prompt' => 'Seleccione una']) ?>
```

**Se modifica el DropDownList**

**Se invoca la función del archivo.js 'planes(valor)'**



ADMINISTRADOR ▾

CERRAR SESIÓN (JEFÉ)

# Usuario Jefe

## Administración de Catálogos

Institutos

Institutos

Carreras

Carreras

Planes

Planes de Estudio

Oferta  
Educativa

Oferta Educativa

Profesores

Profesores

Departamentos

Departamentos

Relación  
Ins-Dep

Institutos-Deptos

Usuarios

Roles



ADMINISTRADOR ▾

CERRAR SESIÓN (OTRO)

# Usuario Otro

## Administración de Catálogos

Institutos

Institutos

Carreras

Carreras

Profesores

Otro1

Profesores

Otro2

Profesores

Profesores

Departamentos

Departamentos

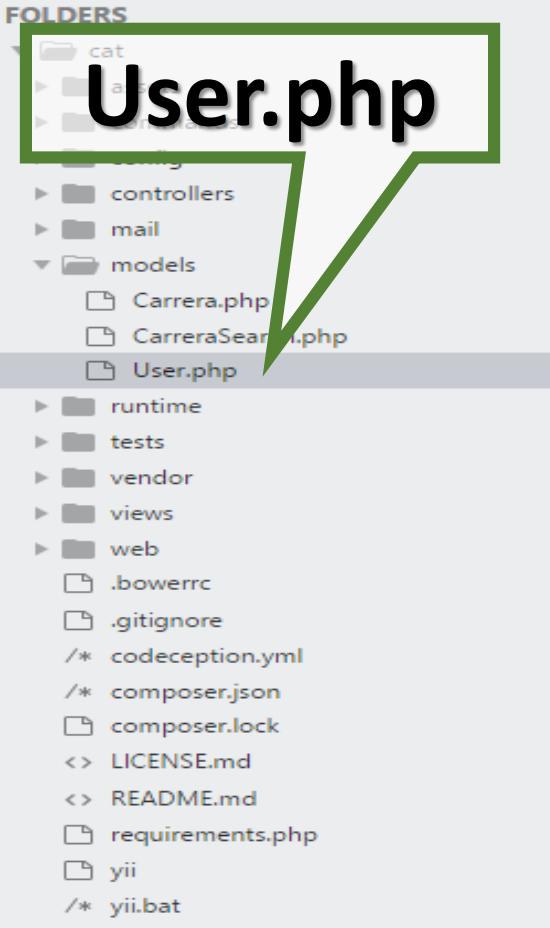
Relación  
Ins-Dep

Institutos-Deptos

Usuarios

Roles

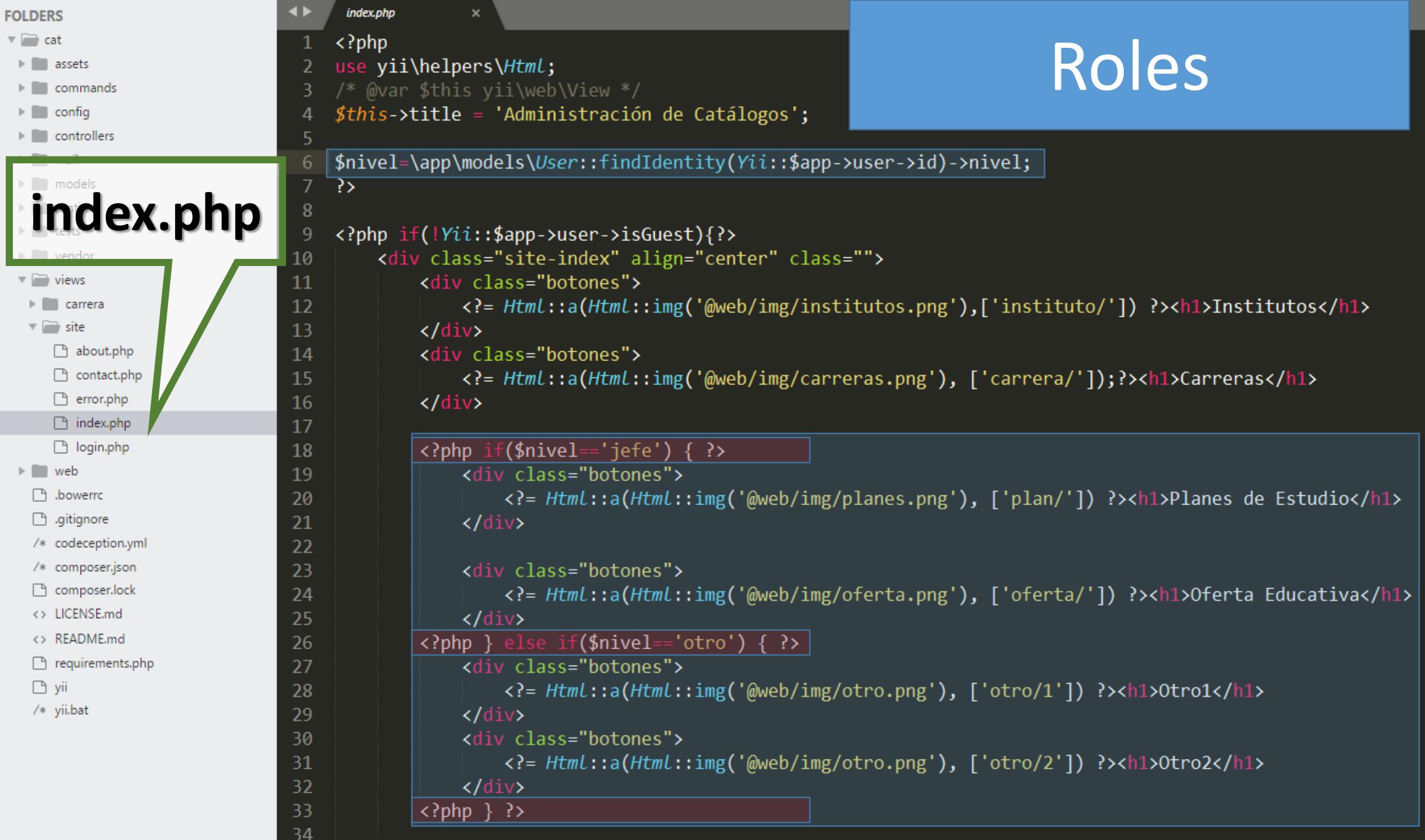
# Roles



```
User.php
```

```
1 <?php
2
3 namespace app\models;
4
5 class User extends \yii\base\Object
6 {
7     public $id;
8     public $username;
9     public $password;
10    public $authKey;
11    public $accessToken;
12    public $nivel;
13
14    private static $users = [
15        '100' => [
16            'id' => '100',
17            'username' => 'admin',
18            'password' => 'admin',
19            'authKey' => 'test100key',
20            'accessToken' => '100-token',
21            'nivel' => 'admin',
22        ],
23        '101' => [
24            'id' => '101',
25            'username' => 'jefe',
26            'password' => 'jefe',
27            'authKey' => 'test101key',
28            'accessToken' => '101-token',
29            'nivel' => 'jefe',
30        ],
31        '102' => [
32            'id' => '102',
33            'username' => 'otro',
34            'password' => 'otro',
35            'authKey' => 'test102key',
36            'accessToken' => '102-token',
37            'nivel' => 'otro',
38        ],
39    ];
40}
```

# Roles



The screenshot shows a code editor with a file named `index.php` open. The file's content is displayed in a dark-themed code editor. The code handles user roles and displays different content based on the user's level (`nivel`). The `index.php` file is highlighted with a green border.

```
<?php
use yii\helpers\Html;
/* @var $this yii\web\View */
$this->title = 'Administración de Catálogos';

$nivel=\app\models\User::findIdentity(Yii::$app->user->id)->nivel;

?>

<?php if(!Yii::$app->user->isGuest){?>
    <div class="site-index" align="center" class="">
        <div class="botones">
            <?= Html::a(Html::img('@web/img/institutos.png'), ['instituto/']) ?><h1>Institutos</h1>
        </div>
        <div class="botones">
            <?= Html::a(Html::img('@web/img/carreras.png'), ['carrera/']);?><h1>Carreras</h1>
        </div>

        <?php if($nivel=='jefe') { ?>
            <div class="botones">
                <?= Html::a(Html::img(' @web/img/planes.png'), ['plan/']) ?><h1>Planes de Estudio</h1>
            </div>

            <div class="botones">
                <?= Html::a(Html::img(' @web/img/oferta.png'), ['oferta/']) ?><h1>Oferta Educativa</h1>
            </div>
        <?php } else if($nivel=='otro') { ?>
            <div class="botones">
                <?= Html::a(Html::img(' @web/img/otro.png'), ['otro/1']) ?><h1>Otro1</h1>
            </div>
            <div class="botones">
                <?= Html::a(Html::img(' @web/img/otro.png'), ['otro/2']) ?><h1>Otro2</h1>
            </div>
        <?php } ?>
```

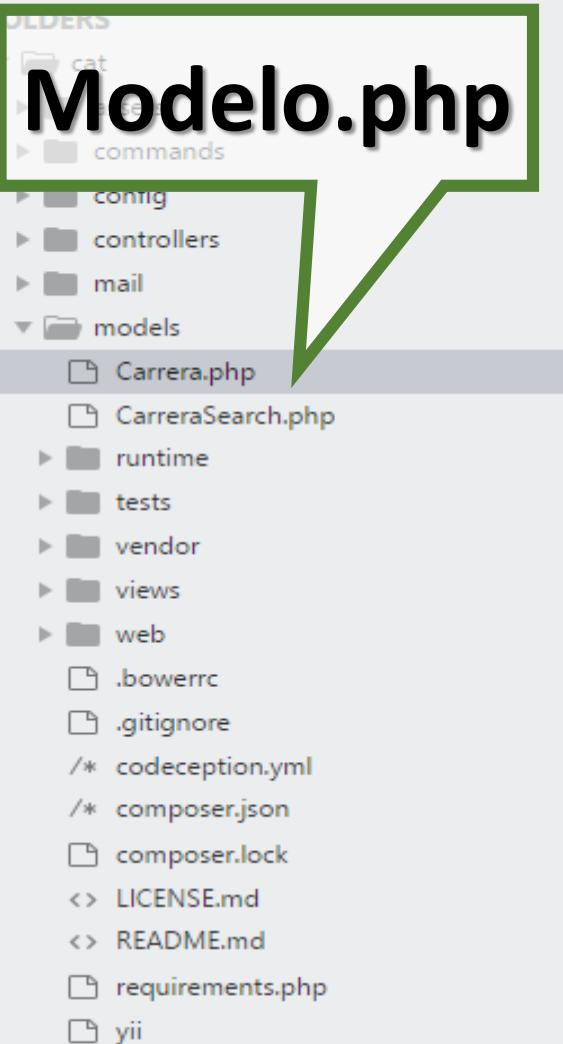
# Busqueda y Ordenamiento con campos personalizados que no están en la base de datos

Crear Carrera

Mostrando 1-3 de 3 elementos.

#	ID	Nombre	Otro campo agregado	
			tec	
1	1	Arquitectura	ARQUITECTURA	 
2	46	Técnico Superior en Buceo Industrial	TÉCNICO	 
3	47	Técnico Superior en Minería	TÉCNICO	 

# Modelo.php



```
Carrera.php
34     public function rules()
35     {
36         return [
37             [['CAR_NOMBRE']], 'required',
38             [['CAR_NOMBRE']], 'string',
39         ];
40     }
41
42     /**
43      * @inheritdoc
44     */
45     public function attributeLabels()
46     {
47         return [
48             'CAR_ID' => 'ID',
49             'CAR_NOMBRE' => 'Nombre',
50             'otro' => 'Otro campo agregado',
51         ];
52     }
53
54     //Aquí va el campo calculado
55     public function getOtro()
56     {
57         //En este ejemplo solo se toma la primera palabra
58         //del campo CAR_NOMBRE y se convierte a mayúsculas
59         $separado = explode(" ", $this->CAR_NOMBRE); //Se separa el nombre
60         $dato = $separado[0]; //Se toma la primera palabra
61         $dato = mb_strtoupper($dato, 'UTF-8'); //Se convierte a mayúsculas
62         return $dato;
63     }
64 }
```

# Campos personalizados

Se agrega la etiqueta del Campo

Se agrega la función del Campo

# Campos personalizados

The screenshot shows a code editor with the file `CarreraSearch.php` open. A green arrow points from the title "ModeloSearch.php" in the sidebar to the file tab in the editor. The code editor has a dark theme with syntax highlighting. A callout box highlights the line `public $otro;` with the text "Se agrega el campo personalizado". Another callout box highlights the line `['otro'], 'safe']` in the `rules()` method with the text "Se agrega la regla safe para el campo personalizado".

```
<?php  
namespace app\models;  
use yii;  
use yii\base\Model;  
use yii\data\ActiveDataProvider;  
use app\models\Carrera;  
  
/**  
 * CarreraSearch represents the model behind the search form about `app\models\Carrera`.  
 */  
class CarreraSearch extends Carrera  
{  
    public $otro;  
  
    /**  
     * @inheritdoc  
     */  
    public function rules()  
    {  
        return [  
            ['CAR_ID'], 'integer',  
            ['CAR_NOMBRE'], 'safe',  
            ['otro'], 'safe'  
        ];  
    }  
}
```

Se agrega el campo personalizado

Se agrega la regla **safe** para el campo personalizado

# Campos personalizados

## ModeloSearch.php

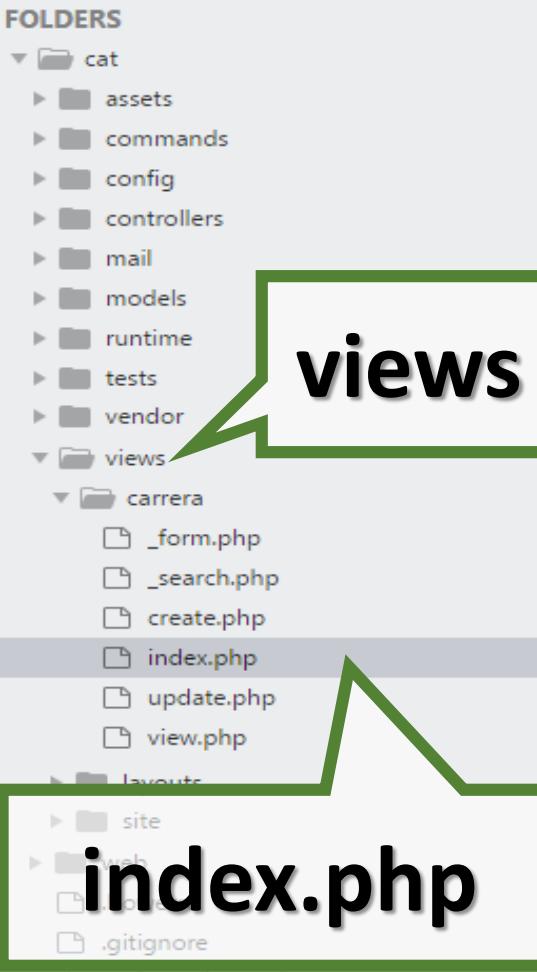
FOLDERS  
cat  
assets  
commands  
config  
main  
models  
Carrera.php  
CarreraSearch.php  
runtime  
tests  
vendor  
views  
web  
.bowerrc  
.gitignore  
/\* codeception.yml  
/\* composer.json  
composer.lock  
<> LICENSE.md  
<> README.md  
requirements.php  
yii  
/\* yii.bat

```
CarreraSearch.php x
46     public function search($params)
47     {
48         $query = Carrera::find();
49         // add conditions that should always apply here
50         // ...
51         $dataProvider = new ActiveDataProvider([
52             'query' => $query,
53         ]);
54
55
56
57         $dataProvider->setSort([
58             'attributes' => [
59                 'CAR_ID',
60                 'CAR_NOMBRE',
61                 'otro' => [
62                     'asc' => ['CAR_NOMBRE' => SORT_ASC],
63                     'desc' => ['CAR_NOMBRE' => SORT_DESC],
64                     'default' => SORT_ASC
65                 ],
66             ],
67         ]);
68
69         $this->load($params);
70
71         if (!$this->validate()) {
72             // uncomment the following line if you do not want to return any records when validation fails
73             // $query->where('0=1');
74             return $dataProvider;
75         }
76
77         // grid filtering conditions
78         $query->andFilterWhere([
79             'CAR_ID' => $this->CAR_ID,
80         ]);
81
82         $query->andFilterWhere(['like', 'CAR_NOMBRE', $this->CAR_NOMBRE])
83             ->andFilterWhere(['like', "SUBSTRING(CAR_NOMBRE,1,(LOCATE(' ',CAR_NOMBRE)))", $this->otro]);
84
85         return $dataProvider;
86     }
87 }
```

Se agrega el ordenamiento para el campo personalizado

Se agrega el filtro de búsqueda para el campo personalizado

# Campos personalizados



```
index.php x
1 <?php
2
3 use yii\helpers\Html;
4 use yii\grid\GridView;
5
6 /* @var $this yii\web\View */
7 /* @var $searchModel app\models\CarreteraSearch */
8 /* @var $dataProvider yii\data\ActiveDataProvider */
9
10 $this->title = 'Carreras';
11 $this->params['breadcrumbs'][] = $this->title;
12 ?>
13 <div class="carrera-index">
14     <p>
15         <?= Html::a('Crear Carrera', ['create'], ['class' => 'btn btn-success']) ?>
16     </p>
17     <?= GridView::widget([
18         'dataProvider' => $dataProvider,
19         'filterModel' => $searchModel,
20         'columns' => [
21             ['class' => 'yii\grid\SerialColumn',
22              'contentOptions' => ['style' => 'color:#aaa;width:80px;text-align:center;'],
23             ],
24
25             ['attribute' => 'CAR_ID',
26              'contentOptions' => ['style' => 'width:80px;text-align:right;'],
27             ],
28             ['CAR_NOMBRE'],
29             ['otro'],
30             ],
31             ],
32             ],
33             ]); ?>
34     </div>
```

A green callout points to the line 'CAR\_NOMBRE' in the code, with the text 'Se agrega el campo personalizado'.

Para leer más sobre campos personalizados, consultar la liga:  
<http://www.yiiframework.com/wiki/621/filter-sort-by-calculated-related-fields-in-gridview-yii-2-0/>