

Tecnologías para Internet

Crear un CRUD con Yii2.

Yii 2 viene en dos plantillas la básica y la avanzada, la plantilla avanzada es muy fácil de usar, considerando si el proyecto a implementar requiere un modelo de usuario y autenticación. La mayoría de las aplicaciones web modernas necesitan esta funcionalidad y la plantilla avanzada tiene una solución.

Algunas de las características que se obtiene con la instalación de la plantilla avanzada es:

- Un esquema predefinido para la tabla de usuario.
- Formulario de inicio de sesión y registro de usuario.
- Recuperación de contraseña olvidada.
- Dominios separados del frontend y del backend.
- Generación automática del código de los modelos, controladores y vistas.
- Integración con Twitter Bootstrap.
- Widgets robustos para la presentación de información.
- Es altamente escalable y extensible y permite cargar funciones.

Entorno de trabajo

En la creación de un CRUD en Yii2 necesitamos el siguiente entorno de trabajo:

- XAMPP (<https://www.apachefriends.org/index.html>)
- phpMyAdmin
- Composer (<https://getcomposer.org/download/>) (<http://librosweb.es/tutorial/como-instalar-composer-globalmente/>)
- Sublime text (<https://www.sublimetext.com/>) o cualquier otro IDE.

Aplicación con CRUD en Yii2

1. En la carpeta de htdocs.
2. En la terminal, colocamos la siguiente instrucción en la terminal:

```
$ composer create-project --prefer-dist yiisoft/yii2-app-advanced  
yiicrud
```

Debe arrojar un resultado como el siguiente:

```
Installing yiisoft/yii2-app-advanced (2.0.9)
- Installing yiisoft/yii2-app-advanced (2.0.9)
  Loading from cache

Created project in yiicrud
Loading composer repositories with package information
Updating dependencies (including require-dev)
- Installing yiisoft/yii2-composer (2.0.4)
  Loading from cache

- Installing swiftmailer/swiftmailer (v5.4.3)
  Loading from cache

- Installing bower-asset/jquery (2.2.4)
  Loading from cache

- Installing bower-asset/yii2-pjax (v2.0.6)
  Loading from cache

- Installing bower-asset/punycode (v1.3.2)
  Loading from cache

- Installing bower-asset/jquery.inputmask (3.2.7)
  Loading from cache

- Installing cebe/markdown (1.1.1)
  Loading from cache

- Installing ezyang/htmlpurifier (v4.8.0)
  Loading from cache

- Installing yiisoft/yii2 (2.0.9)
  Loading from cache

- Installing yiisoft/yii2-swiftmailer (2.0.6)
  Loading from cache

- Installing yiisoft/yii2-codeception (2.0.5)
  Loading from cache

- Installing bower-asset/bootstrap (v3.3.7)
  Loading from cache

- Installing yiisoft/yii2-bootstrap (2.0.6)
  Loading from cache

- Installing yiisoft/yii2-debug (2.0.6)
  Loading from cache

- Installing bower-asset/typeahead.js (v0.11.1)
  Loading from cache

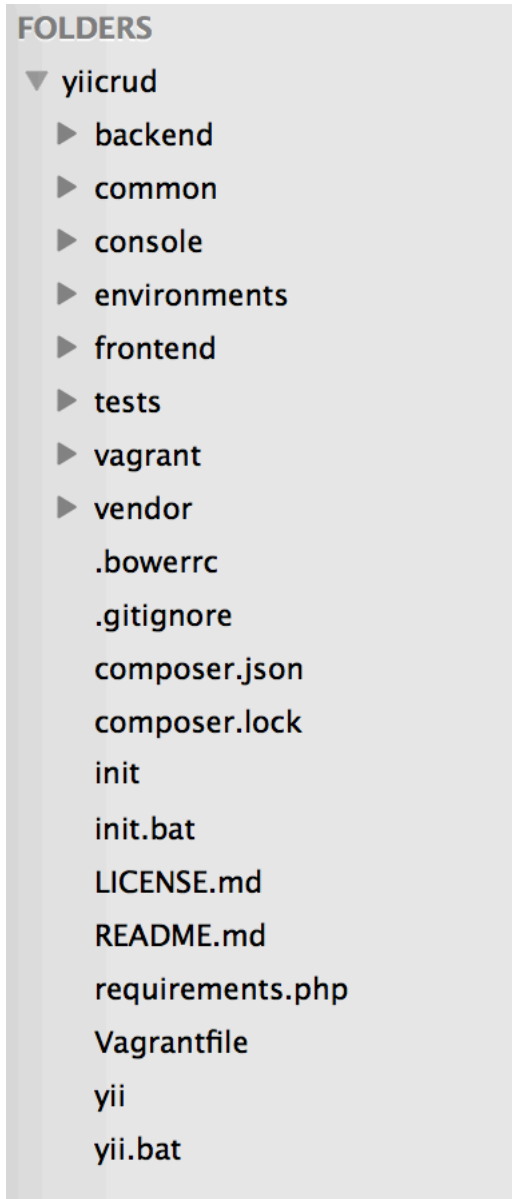
- Installing phpspec/php-diff (v1.1.0)
  Loading from cache

- Installing yiisoft/yii2-gii (2.0.5)
  Loading from cache

- Installing fzaninotto/faker (v1.6.0)
  Loading from cache

- Installing yiisoft/yii2-faker (2.0.3)
  Loading from cache

Writing lock file
Generating autoload files
```



3. Dentro de la carpeta yii2-app-advanced ejecutar la siguiente instrucción:

\$php init

Se debe seleccionar la opción de Development para inicializar el proyecto en entorno de desarrollo.

```
Yii Application Initialization Tool v1.0

Which environment do you want the application to be initialized
in?

    [0] Development
    [1] Production

Your choice [0-1, or "q" to quit] 0

Initialize the application under 'Development' environment?
[yes|no] yes

Start initialization ...

generate backend/config/main-local.php
generate backend/config/params-local.php
generate backend/web/index-test.php
generate backend/web/index.php
generate common/config/main-local.php
generate common/config/params-local.php
generate console/config/main-local.php
generate console/config/params-local.php
generate frontend/config/main-local.php
generate frontend/config/params-local.php
generate frontend/web/index-test.php
generate frontend/web/index.php
generate tests/codeception/config/config-local.php
generate yii
generate cookie validation key in backend/config/main-
local.php
generate cookie validation key in frontend/config/main-
local.php
    chmod 0777 backend/runtime
    chmod 0777 backend/web/assets
    chmod 0777 frontend/runtime
    chmod 0777 frontend/web/assets
    chmod 0755 yii
    chmod 0755 tests/codeception/bin/yii

... initialization completed.
```

4. A continuación debemos crear una base de datos para el proyecto. Utilizando phpMyAdmin.
El nombre de la base de datos es **yiicrud** con cotejamiento **latin1_general_ci**

Bases de datos

 **Crear base de datos** 

5. Creamos un usuario con permisos para administrar la base de datos yiicrud.

En localhost creamos;

Usuario: **yiicrudadmin**

Contraseña: **pecas**

6. Modificando la conexión de la base de datos.

/ XAMPP/xamppfiles/htdocs/yiiCRUD/yii2-app-advanced/common/config/ main-local.php

```
<?php
return [
    'components' => [
        'db' => [
            'class' => 'yii\db\Connection',
            'dsn' => 'mysql:host=localhost;dbname=yiicrud',
            'username' => 'yiicrudadmin',
            'password' => 'pecas',
            'charset' => 'utf8',
        ],
        'mailer' => [
            'class' => 'yii\swiftmailer\Mailer',
            'viewPath' => '@common/mail',
            // send all mails to a file by default. You have to set
            // 'useFileTransport' to false and configure a transport
            // for the mailer to send real emails.
            'useFileTransport' => true,
        ],
    ],
];
```

7. Ejecutamos el siguiente comando en la terminal.

```
$. /yii migrate
```

```
Yii Migration Tool (based on Yii v2.0.9)

Creating migration history table "migration"...Done.
Total 1 new migration to be applied:
    m130524_201442_init

Apply the above migration? (yes|no) [no]:yes
*** applying m130524_201442_init
    > create table {%user%} ... done (time: 0.011s)
*** applied m130524_201442_init (time: 0.025s)

1 migration was applied.
Migrated up successfully.
```

8. Vista del proyecto yiicrud recién instalado

My Company

[Home](#)[About](#)[Contact](#)[Signup](#)[Login](#)

Congratulations!

You have successfully created your Yii-powered application.

Get started with Yii

Heading

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur.

Yii Documentation »

Heading

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur.

Yii Forum »

Heading

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur.

Yii Extensions »

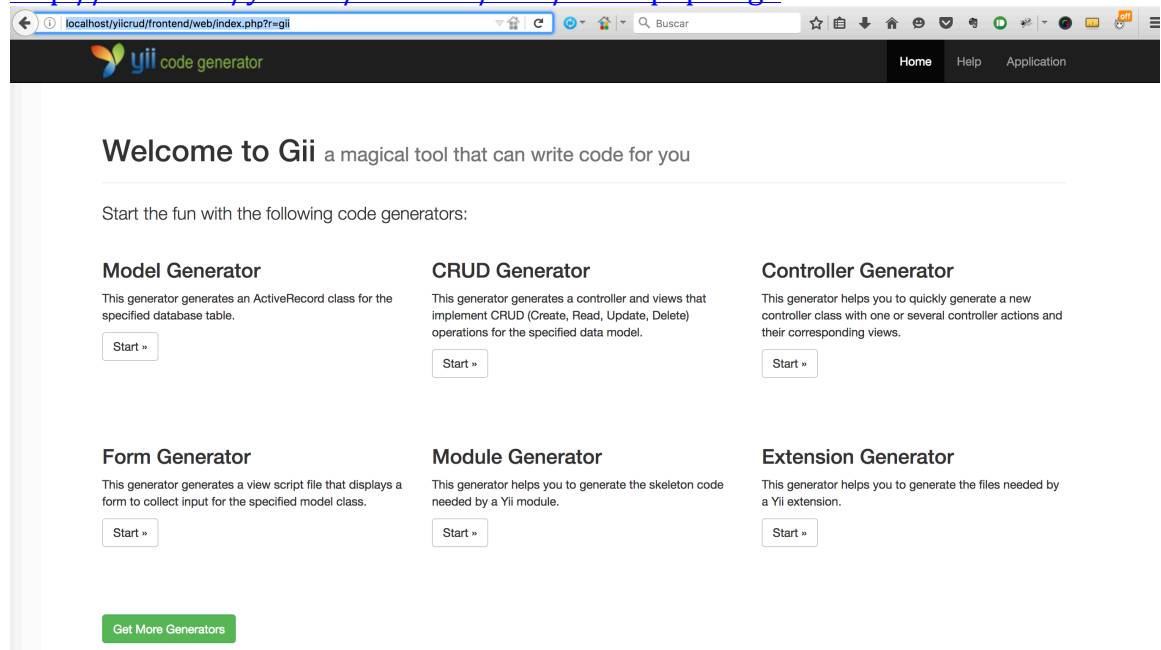
9. Creamos la tabla estudiante

```
CREATE TABLE `yiicrud`.`estudiantes` ( `id` BIGINT NOT NULL  
AUTO_INCREMENT , `nombre` VARCHAR(255) CHARACTER SET latin1  
COLLATE latin1_general_ci NOT NULL , `apellido` VARCHAR(255)  
CHARACTER SET latin1 COLLATE latin1_general_ci NOT NULL , `email`  
TEXT CHARACTER SET latin1 COLLATE latin1_general_ci NOT NULL ,  
`telefono` TEXT CHARACTER SET latin1 COLLATE latin1_general_ci  
NOT NULL , PRIMARY KEY (`id`)) ENGINE = InnoDB CHARACTER SET  
latin1 COLLATE latin1_general_ci;
```

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Extra	Acción
<input type="checkbox"/>	1	id	bigint(20)		No	Ninguna	AUTO_INCREMENT	 Cambiar  Eliminar  Primaria  Único  Índice  Más
<input type="checkbox"/>	2	nombre	varchar(255)	latin1_general_ci	No	Ninguna		 Cambiar  Eliminar  Primaria  Único  Índice  Más
<input type="checkbox"/>	3	apellido	varchar(255)	latin1_general_ci	No	Ninguna		 Cambiar  Eliminar  Primaria  Único  Índice  Espacial  Más
<input type="checkbox"/>	4	email	text	latin1_general_ci	No	Ninguna		 Cambiar  Eliminar  Primaria  Único  Índice  Espacial  Más
<input type="checkbox"/>	5	telefono	text	latin1_general_ci	No	Ninguna		 Cambiar  Eliminar  Primaria  Único  Índice  Espacial  Más

10. Para crear el CRUD de la tabla estudiante, vamos a emplear el generador de CRUD que proporciona yii2, conocido como gii. Para ello ingresamos a la ruta:

<http://localhost/yiicrud/frontend/web/index.php?r=gii>



11. Creamos el modelo Estudiante.

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

estudiantes

Model Class

Estudiante

Namespace

frontend\models

Base Class

yii\db\ActiveRecord

Database Connection ID

db

☐ **Use Table Prefix**

Generate Relations

All relations

☐ **Generate Labels from DB Comments**

☒ **Generate ActiveQuery**

12. Realizamos el CRUD utilizando la funcionalidad gii.

yii code generator

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

frontend\models\Estudiante

Search Model Class

frontend\models\search\EstudianteSearch

Controller Class

frontend\controllers\EstudianteController

View Path

fronted\views

Base Controller Class

yii\web\Controller

Widget Used in Index Page

GridView

Enable I18N

Enable Pjax

Code Template

default (/Applications/XAMPP/xamppfiles/htdocs/yiicrud/vendor/yiisoft/yii2-gii/generators/crud/default)

Preview

Generate

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

Create

Unchanged

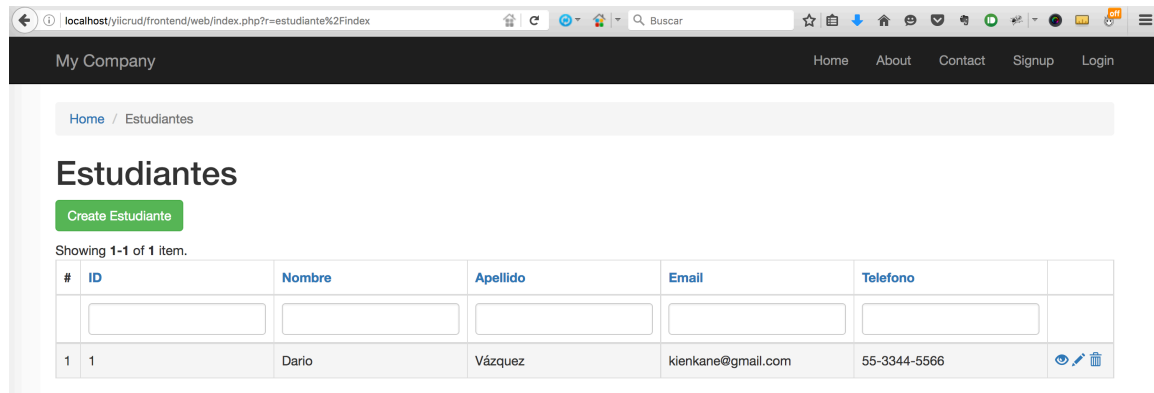
Overwrite

Code File	Action	
controllers/EstudianteController.php	unchanged	
models/search/EstudianteSearch.php	create	
fronted\views/_form.php	create	
fronted\views/_search.php	create	
fronted\views/create.php	create	
fronted\views/index.php	create	
fronted\views/update.php	create	
fronted\views/view.php	create	

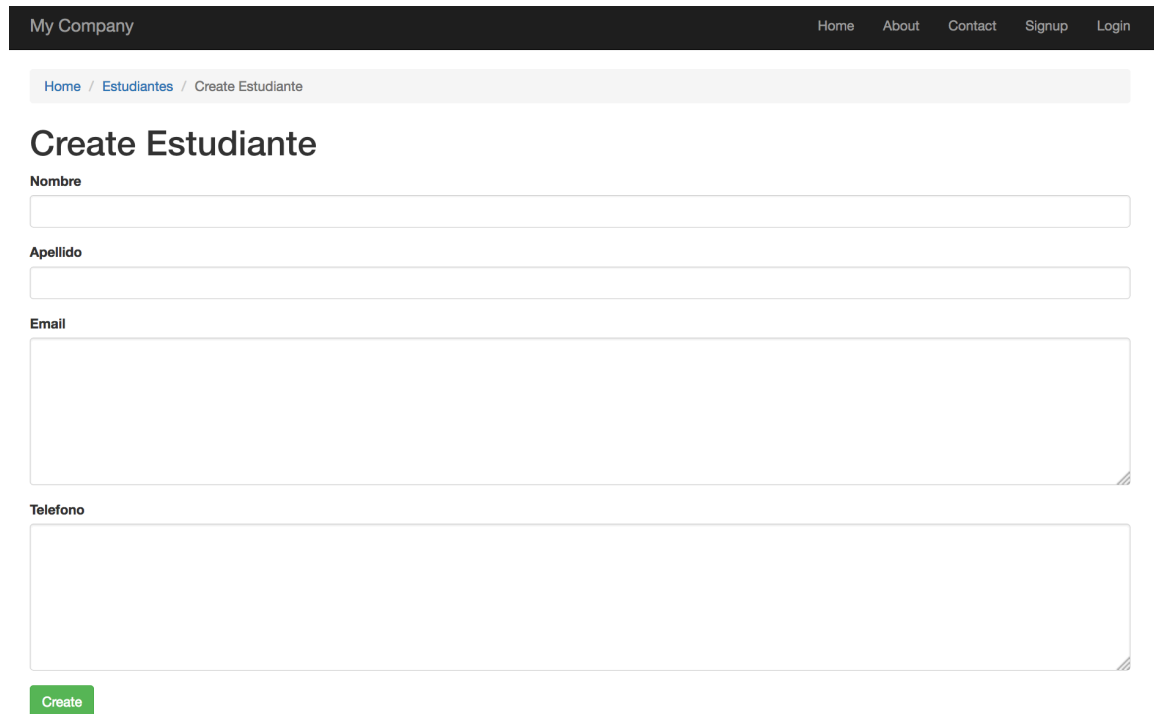
The code has been generated successfully.

Generating code using template "/Applications/XAMPP/xamppfiles/htdocs/yiicrud/vendor/yiisoft/yii2-gii/generators/crud/default"
skipped controllers/EstudianteController.php
generated models/search/EstudianteSearch.php
generated frontend\views_form.php
generated frontend\views_search.php
generated frontend\views/create.php
generated frontend\views/index.php
generated frontend\views/update.php
generated frontend\views/view.php
done!

13. La vista de estudiante index.php



14. La vista estudiante create.php



15. La vista estudiante update.php

My Company

HomeAboutContactSignupLogin

Home / Estudiantes / 1 / Update

Update Estudiante: 1

Nombre

Apellido

Email

Telefono

Update

16. Vista estudiante view.php

My Company

HomeAboutContactSignupLogin

Home / Estudiantes / 1

1

UpdateDelete

ID	1
Nombre	Dario
Apellido	Vázquez
Email	kienkane@gmail.com
Telefono	55-3344-5566

17. Agregando un campo de búsqueda general en la vista index del modelo Estudiante. Código basado y tomado de <http://jquery-manual.blogspot.mx/2015/02/yii-framework-2-crud-activerrecord-read.html>

Para ello nos vamos al archivo views/estudiante/index.php:

```
<!-- -->
<?php $f = ActiveForm::begin([
    "method" => "get",
    "action" => Url::toRoute("estudiante/index"),
    "enableClientValidation" => true,
]);
?>

<div class="form-group">
    <?= $f->field($form, "q")->input("search") ?>
</div>

<?= Html::submitButton("Buscar", ["class" => "btn btn-primary"])
?>

<?php $f->end() ?>

<h3><?= $search ?></h3>
<!-- -->
```

En este punto es importante dar anotar que el archivo sufrió cambios debido a que no usaremos la clase EstudianteSearch, dado que vamos a realizar una búsqueda general. Checar el documento entero.

```
<ul class="collection">
    <?php $controller = Yii::$app->controller; ?>
    <?php foreach($model as $value):?>
        <li class="collection-item avatar">
            <i class="material-icons circle">account_box</i>
            <span class="title"><?= $value['nombre'] ; ?></span>
            <p>
                <?= "Estudiante: ". $value['nombre'] . " ". $value['apellido'] ?><br>
                <?= "Email: ".$value['email'] . " Teléfono: " . $value['telefono'] ; ?>
            </p>
            <?php $paramsURL = array_merge(["{$controller->id}/view",
                ['id'=>$value['id']] ); ?>
            <a href="<?= Yii::$app->urlManager->createUrl($paramsURL);?>"
                class="secondary-content"><i class="material-icons">visibility</i></a>
        </li>
    <?php endforeach; ?>
</ul>
```

18. Creamos la clase encargada de realizar o armar la búsqueda, para ello guardaremos el siguiente código con el nombre models/FormSearch.php

```
<?php

namespace frontend\models;
use Yii;
use yii\base\Model;

class FormSearch extends Model{

    public $q;

    public function rules(){
        return [
            ["q", "match", "pattern" => "/^[0-9@a-záéíóúñ\s.]+$/i", "message" =>
                "Sólo se aceptan letras y números"]
        ];
    }

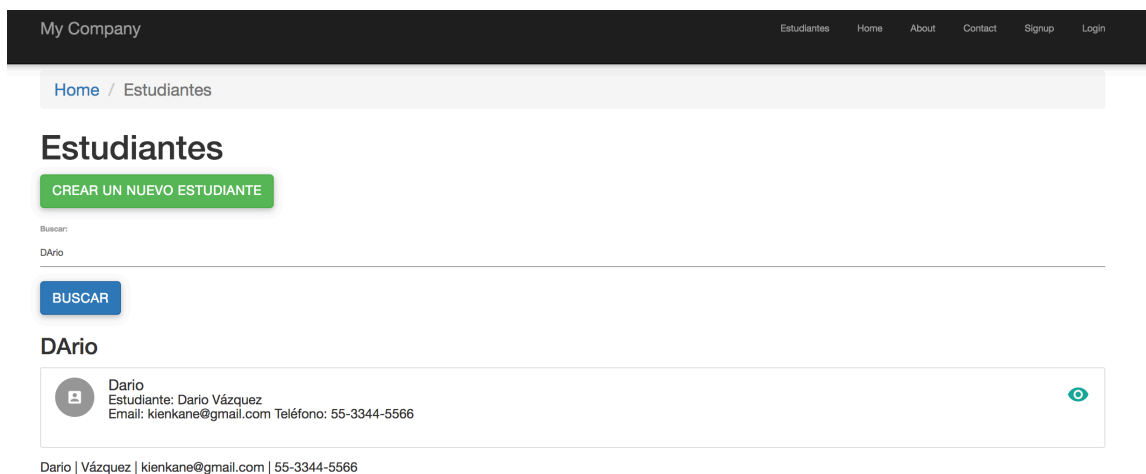
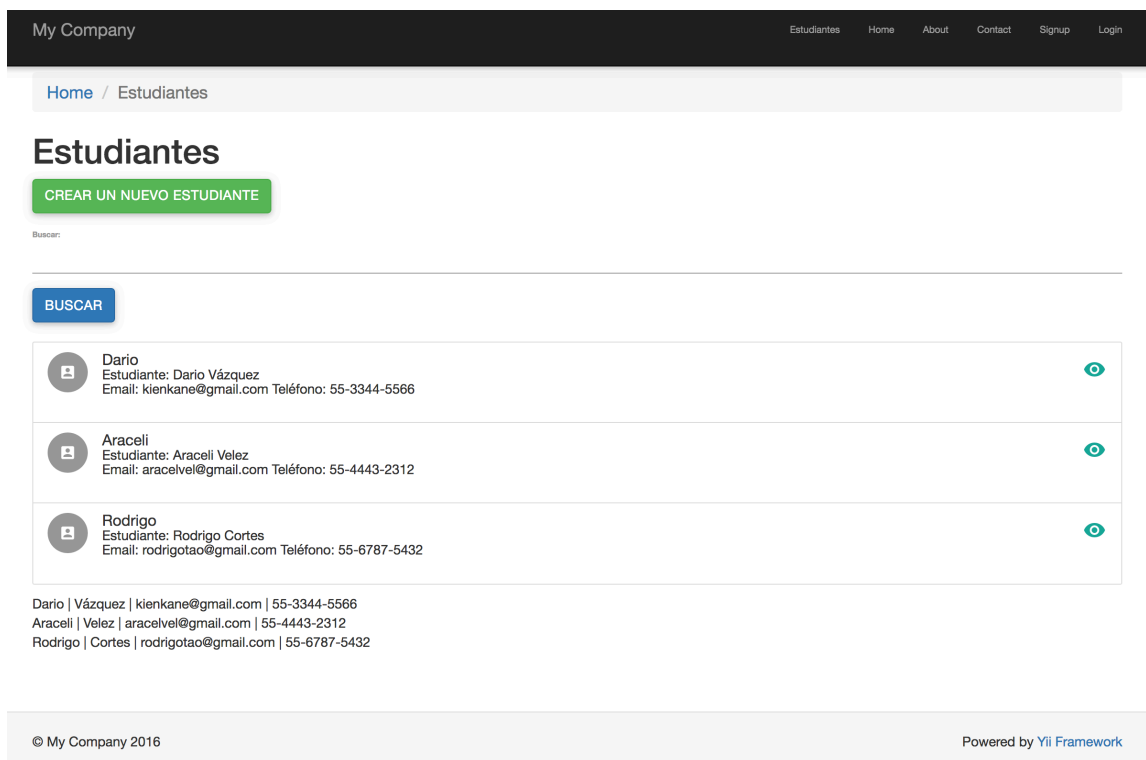
    public function attributeLabels(){
        return [
            'q' => "Buscar:",];
    }
}
```

19. En el siguiente punto modificamos el controlador
controllers/EstudianteController.php, en este archivo tendremos que modificar
la acción index dado que realizaremos la búsqueda y filtrado de información sin
utilizar la clase EstudianteSearch.php

```
[...]
use frontend\models\FormSearch;
use yii\helpers\Html;
[...]

public function actionIndex(){
    $table = new Estudiante;
    $model = $table->find()->all();
    $form = new FormSearch;
    $search = null;
    if($form->load(Yii::$app->request->get())){
        if ($form->validate()){
            $search = Html::encode($form->q);
            $query = "SELECT * FROM estudiantes WHERE nombre LIKE '%$search%' OR ";
            $query .= "apellido LIKE '%$search%' OR email LIKE '%$search%' OR telefono
                LIKE '%$search%'";
            $model = $table->findBySql($query)->all();
        }else{
            $form->getErrors();
        }
    }
    return $this->render('index', [
        "model" => $model,
        "form" => $form,
        "search" => $search,]);
}
```

20. Con estos cambios tendremos algo como en las imágenes siguientes:



21. Agregar estilos de css a las vistas de estudiantes nos colocamos en el archivo `assets/AppAsset.php` y agregamos la siguiente línea.

```
public $css = [  
    'css/site.css',  
    'css/style.css',  
];
```

22. Como podemos ver necesitamos crear un archivo web/css/style.css donde daremos algunos estilos al campo de búsqueda de la vista estudiantes/index

```
nav ul a {
    font-size: 1.5rem;
}
.form-group .labelSearchGen {
    font-size: 1.5rem;
    float:left;
    margin:10px 15px 0 10px;
}

.form-group .searchGen{
    font-size: 1.3rem;
    width: 150px;
    float:left;
}

.btnSearch{
    margin:0px 15px;
    background-color: #5cb85c;
    border-color: #4cae4c;
    transition: all 0.2s ease-out 0s;
    color: #fff;
}
```

23. Con estos cambios es necesario actualizar el formulario de búsqueda del archivo estudiante/index.php agregando las clases al label, input y botón del formulario.

```
<div class="form-group">
    <?=$f->field($form, "q")->input("search",['class' => 'searchGen'])-
>label('Buscar',['class'=>'labelSearchGen']); ?>
</div>

<?= Html::submitButton("<i class='material-icons'>search</i>", ["class" =>
"btnSearch btn "]) ?>
```

El resultado se presenta en la imagen siguiente:

Estudiantes

CREAR UN NUEVO ESTUDIANTE

Buscar



Modificar la vista Create y Update de Estudiante

El objetivo es agregar y modificar más de un número telefónico en la vista Create y en la vista Update de estudiantes.

24. Para ello tenemos que ir a la vista `views/estudiante/create.php` y modificar las siguientes líneas:

```
<?php
use yii\helpers\Html;
use yii\widgets\ActiveForm;
$this->title = 'Create Estudiante';
$this->params['breadcrumbs'][] = ['label' => 'Estudiantes', 'url' => ['index']];
?>
<?php $this->registerJsFile(Yii::$app->request->baseUrl . '/js/addTelefonos.js', ['depends' =>
[yii\web\jQueryAsset::className()]]); ?>

<div class="container-fluid">
<div class="empresa-create">
<h3><?= Html::encode($this->title) ?></h3>
<?php
    $form = ActiveForm::begin([
        'id' => 'estudiante-create-form',
        'options' => ['class' => 'form-horizontal'],
    ]);?>
<h5>Datos de Cuenta</h5>
<?= $form->field($model, 'nombre')->textInput(['maxlength' => 255]) ?>
<?= $form->field($model, 'apellido')->textInput(['maxlength' => 255]) ?>
<?= $form->field($model, 'email')->textInput(['maxlength' => 200]) ?>
<div>
    Teléfono
    <a id="agregarTelefono" class="btn-floating btn-small waves-effect waves-light "><i
class="material-icons">add</i></a>

</div>
<div id="contenedorTels">
    <div class="added">
        <input type="text" name="mitexto[]" id="campo_1" placeholder="Teléfono 1"/><a href="#"
class="eliminar">&times;</a>
    </div>
</div>
<?= $form->field($model, 'telefono')->hiddenInput(['id' => 'inputTel'])->label(false); ?>
<div class="form-group">
    <div class="col-lg-offset-1 col-lg-11">
        <?= Html::submitButton('Create', ['class' => 'btn btn-primary', 'id'=>
'crearEstudiante']) ?>
    </div>
</div>
<?php ActiveForm::end(); ?>
</div>
</div>
```


25. Modificamos el archivo `models/Estudiante.php` con el siguiente código en el método `rules()`.

```
/**
 * @inheritdoc
 */
public function rules(){
    return [
        [['nombre', 'apellido'], 'required'],
        [['nombre', 'apellido', 'email', 'telefono'], 'string', 'max'
=> 255],
    ];
}
```

26. Agregamos el código en javascript para generar los teléfonos de forma dinámica

```
$(document).ready(function() {
    var MaxInputs      = 8; //Número Maximo de Campos
    var contenedor      = $("#contenedorTels"); //ID del contenedor
    var AddButton       = $("#agregarTelefono"); //ID del Botón Agregar
    var EnviarButton    = $("#crearEstudiante");

    //var x = número de campos existentes en el contenedor
    var x = $("#contenedorTels div").length + 1;
    var FieldCount = x-1; //para el seguimiento de los campos
    $(AddButton).click(function (e) {
        if(x <= MaxInputs) //max input box allowed{
            FieldCount++;
            $(contenedor).append('<div class="added" ><input type="text"
name="mitexto[]" id="campo_' + FieldCount +' " placeholder="Teléfono ' +
FieldCount +'"/><a href="#" class="eliminar">&times;</a></div>');
            x++; //text box increment
        }
        return false;
    });
    $("body").on("click", ".eliminar", function(e){ //click en eliminar
campo
        if( x > 1 ) {
            $(this).parent('div').remove(); //eliminar el campo
            x--;
        }
        return false;
    });
    $(EnviarButton).on("click", function (e) {
        var cad = "";
        $("#contenedorTels .added input").each(function (index){
            if( $(this).val() !== "" ) {
                cad += $(this).val() + "|";
            }
        });
        $( "#inputTel" ).val(cad);
    });
});
```

27. El resultado se puede ver en la siguiente imagen

Home / Estudiantes


Create Estudiante

Datos de Cuenta

Nombre

Apellido

Email

Teléfono 

Teléfono 1

x

CREATE

Home / Estudiantes / 8 / Actualizar

Actualizar Estudiante: 8

Datos personales

Nombre

Talla

Apellido

vázquez

Email

talia@gmail.com

AGREGAR TELÉFONO

345 35435 345435

x

345 345 665 567

x

UPDATE

28. Se pueden descargar el proyecto desde el repositorio en Github
<https://github.com/kienkane/yiicrud2.git>