

UNIVERSIDAD NACIONAL DE SAN AGUSTÍN DE AREQUIPA
FACULTAD DE INGENIERÍA DE PRODUCCIÓN Y SERVICIOS
ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS



MAG. RICHART SMITH ESCOBEDO QUISPE
MAG. CHRISTIAN ALAIN REVILLA ARROYO

GUÍA DE LABORATORIO

PROGRAMACIÓN WEB 2

III SEMESTRE
2019 - A

COMPETENCIAS

- Diseña responsablemente sistemas web, componentes o procesos para satisfacer necesidades dentro de restricciones realistas: económicas, medio ambientales, sociales, políticas, éticas, de salud, de seguridad, manufacturación y sostenibilidad.
- Construye responsablemente soluciones web siguiendo un proceso adecuado llevando a cabo las pruebas ajustadas a los recursos disponibles del cliente.
- Aplica de forma flexible técnicas, métodos, principios, normas, estándares y herramientas de la ingeniería web necesarias para la construcción de software e implementación de sistemas de información web.

Laboratorio
1

Instalación, Configuración, Demo con un Framework MVC

I

OBJETIVOS

- Instalar un framework Modelo-Vista-Controlador.
- Configurar un framework Modelo-Vista-Controlador.
- Crear un Hola Mundo con un framework Modelo-Vista-Controlador.

II

TEMAS A TRATAR

- Instalación de un framework MVC
- Instalación de Composer
- Creación de un nuevo proyecto

III

MARCO TEÓRICO

Instalación

CakePHP tiene algunos requisitos del sistema:

- Servidor HTTP. Por ejemplo: Apache. Se prefiere tener mod_rewrite, pero de ninguna manera es obligatorio. También puede usar nginx o Microsoft IIS si lo prefiere.
- PHP 5.6 como mínimo (compatible con 7.4).
- extensión PHP mbstring
- extensión PHP intl
- extensión PHP simplexml

- Extensión PHP PDO

En XAMPP, se incluye la extensión intl, pero debe descomentar `extension=php_intl.dll` en `php.ini` y reiniciar el servidor a través del Panel de control de XAMPP.

En WAMP, la extensión intl está "activada" por defecto pero no funciona. Para que funcione, debe ir a la carpeta php (por defecto) `C:\wamp\bin\php\php {versión}`, copiar todos los archivos que se parecen a `icu *.dll` y pegarlos en el directorio bin de apache `C:\wamp\bin\apache\apache {versión}\bin`. Luego reinicie todos los servicios y debería estar bien.

Si bien no se requiere un motor de base de datos, imaginamos que la mayoría de las aplicaciones lo utilizarán. CakePHP admite una variedad de motores de almacenamiento de bases de datos:

- MySQL (5.5.3 o superior)
- MariaDB (5.5 o mayor)
- PostgreSQL
- Microsoft SQL Server (2008 o superior)
- SQLite 3

Todos los controladores integrados requieren PDO. Debe asegurarse de tener instaladas las extensiones de PDO correctas.

Instalación de CakePHP

Antes de comenzar, debe asegurarse de que su versión de PHP esté actualizada:

```
php -v
```

Debe tener PHP 5.6 (CLI) o superior. La versión de PHP de su servidor web también debe ser 5.6 o superior, y debe ser la misma versión que usa su interfaz de línea de comandos (CLI).

Instalación de Composer

CakePHP utiliza Composer, una herramienta de gestión de dependencias, como método de instalación oficialmente admitido.

Instalación de Composer en Linux y macOS

- Ejecute el script de instalación como se describe en la documentación oficial de Composer y siga las instrucciones para instalar Composer.

<https://getcomposer.org/download/>

- Ejecute el siguiente comando para mover composer.phar a un directorio que esté en su ruta:

```
mv composer.phar /usr/local/bin/composer
```

Instalación de Composer en Windows

Para sistemas Windows, puede descargar el instalador de Windows de Composer.

<https://github.com/composer/windows-setup/releases/>

Puede encontrar más instrucciones para el instalador de Windows de Composer en el archivo README.

<https://github.com/composer/windows-setup>

Crea un proyecto CakePHP

Puede crear una nueva aplicación CakePHP usando el create-project comando del compositor :

```
composer create-project --prefer-dist cakephp/app:^3.9 my_app_name
```

Una vez que Composer termine de descargar el esqueleto de la aplicación y la biblioteca principal de CakePHP, debería tener una aplicación CakePHP en funcionamiento instalada a través de Composer. Asegúrese de mantener los archivos composer.json y composer.lock con el resto de su código fuente.

Ahora puede visitar la ruta donde instaló su aplicación CakePHP y ver la página de inicio predeterminada. Para cambiar el contenido de esta página, edite src / Template / Pages / home.ctp .

Aunque composer es el método de instalación recomendado, hay descargas preinstaladas disponibles en Github.

<https://github.com/cakephp/cakephp/tags>

Esas descargas contienen el esqueleto de la aplicación con todos los paquetes de proveedores instalados. También incluye el composer.phar para que tengas todo lo que necesitas para un uso posterior.

Producción

Una instalación de producción es una forma más flexible de configurar CakePHP. El uso de este método permite que todo un dominio actúe como una única aplicación CakePHP. Este ejemplo lo ayudará a instalar CakePHP en cualquier lugar de su sistema de archivos y lo hará disponible en <http://www.example.com> . Tenga en cuenta que esta instalación puede requerir los derechos para cambiar DocumentRoot en los servidores web Apache. Después de instalar su aplicación utilizando uno de los métodos anteriores en el directorio de su elección, asumiremos que eligió / cake_install, su configuración de producción se verá así en el sistema de archivos:

```
/cake_install/  
  bin/  
  config/  
  logs/  
  plugins/  
  src/  
  tests/  
  tmp/  
  vendor/  
  webroot/ (this directory is set as DocumentRoot)  
  .gitignore  
  .htaccess  
  .travis.yml  
  composer.json  
  index.php  
  phpunit.xml.dist  
  README.md
```

Los desarrolladores que usan Apache deben establecer la DocumentRoot directiva para el dominio en:

DocumentRoot /cake_install/webroot

Si su servidor web está configurado correctamente, ahora debería encontrar su aplicación CakePHP accesible en <http://www.example.com> .

IV

ACTIVIDADES

Instale la plataforma XAMP en su sistema operativo.

En sistemas MS Windows use el paquete XAMPP.

En sistemas GNU/Linux Debian ejecute el siguiente comando:

\$ sudo apt-get install apache2 php7.4 mysql-server

Instale los requerimientos del framework MVC declaradas en el Marco Teórico.

Realice los ejercicios resueltos.

Discuta en clase los ejercicios propuestos.

V

EJERCICIOS RESUELTOS

Programación BackEnd – En el Controlador

Cree un nuevo Controlador MessagesController:

```
public class MessagesController extends Controller {}
```

Cree un método nuevo dentro de la clase del Controlador llamado index():

```
public function index() {}
```

Cree un mensaje dentro del método como una cadena de texto:

```
$mensaje = 'Hola mundo';
```

Envíe el mensaje a la vista usando el método set():

```
$this->set('mensaje', $mensaje);
```

Programación en el Front-End – En la Vista

Cree un archivo 'index.ctp' relacionada al método del controlador.

Reciba la variable que tiene el mensaje del controlador:

```
<?php  
    echo '<p>' . $mensaje . '</p>';
```

Acceda al navegador web a la dirección análoga para comprobar:

<http://localhost/messages/index>

VI

EJERCICIOS PROPUESTOS

Desarrollar la solución para el siguiente requerimiento:

El Controlador debe enviar un mensaje de saludo de acuerdo a la hora del sistema.

- Buenos días.
- Buenas tardes.
- Buenas noches.

El mensaje debe llegar a la vista y debe mostrarse en el Navegador web.

VII

CUESTIONARIO

1. ¿Cómo se envía una variable del Controlador a una vista?
2. ¿Cuáles son las configuraciones regionales que se deben hacer en el servidor para utilizar una zona horaria determinada?

VIII

BIBLIOGRAFÍA

- [1] <https://book.cakephp.org/3/en/installation.html>

UNIVERSIDAD NACIONAL DE SAN AGUSTÍN DE AREQUIPA

FACULTAD DE INGENIERÍA DE PRODUCCIÓN Y SERVICIOS

ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS



MAG. RICHART SMITH ESCOBEDO QUISPE
MAG. CHRISTIAN ALAIN REVILLA ARROYO

GUÍA DE LABORATORIO

PROGRAMACIÓN WEB 2

III SEMESTRE
2019 - A

COMPETENCIAS

- Diseña responsablemente sistemas web, componentes o procesos para satisfacer necesidades dentro de restricciones realistas: económicas, medio ambientales, sociales, políticas, éticas, de salud, de seguridad, manufacturación y sostenibilidad.
- Construye responsablemente soluciones web siguiendo un proceso adecuado llevando a cabo las pruebas ajustadas a los recursos disponibles del cliente.
- Aplica de forma flexible técnicas, métodos, principios, normas, estándares y herramientas de la ingeniería web necesarias para la construcción de software e implementación de sistemas de información web.

Laboratorio**2****Modelo lógico – Modelo Físico****I****OBJETIVOS**

- Estudiar los diagramas UML/Entidad-Relación.
- Diseñar un Modelo lógico – Modelo físico.
- Obtener el diccionario de datos con una herramienta.

II**TEMAS A TRATAR**

- Diagramas UML/Entidad Relación
- Modelo Lógico
- Modelo Físico
- Diccionario de datos

III**MARCO TEÓRICO****Diagrama de Casos de uso**

Actores

Casos de uso

Relaciones entre un Actor y un Caso de uso

Relaciones entre Casos de uso

- Usa
- Extiende

Ejemplo: Diagrama UML para un Sistema de biblioteca

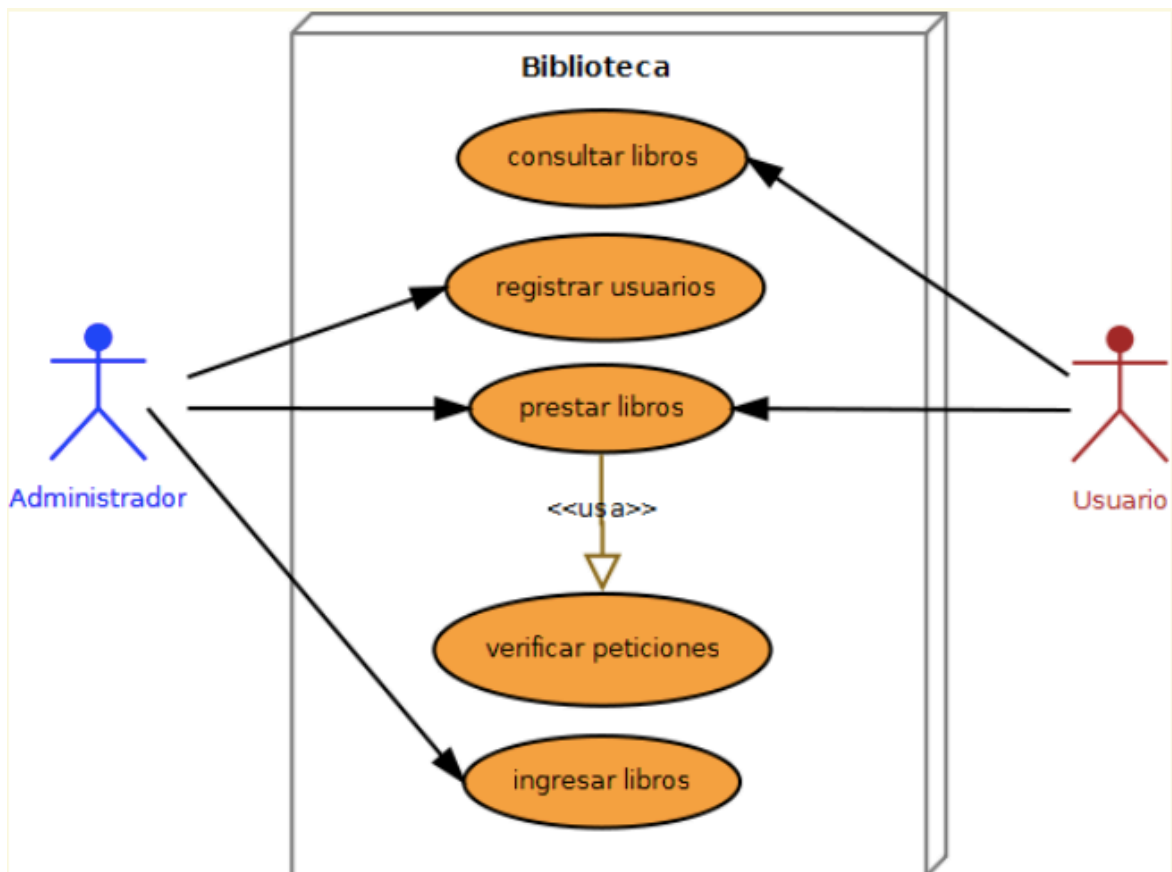


Diagrama Entidad Relación

Entidades

Relaciones

- Uno a uno
- Uno a muchos
- Muchos a Muchos

Cardinalidad

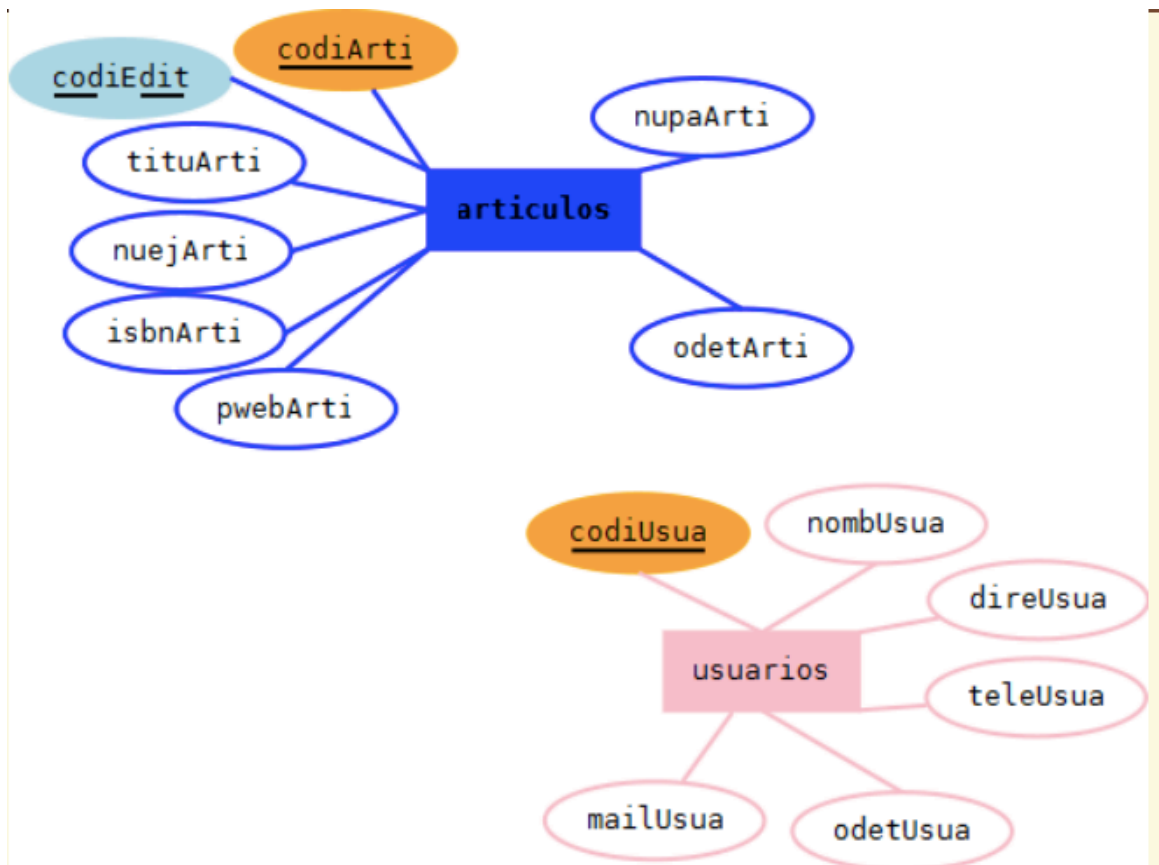
Clave Primaria

Clave secundaria

Ejemplo: Diagrama E-R para el sistema de biblioteca.

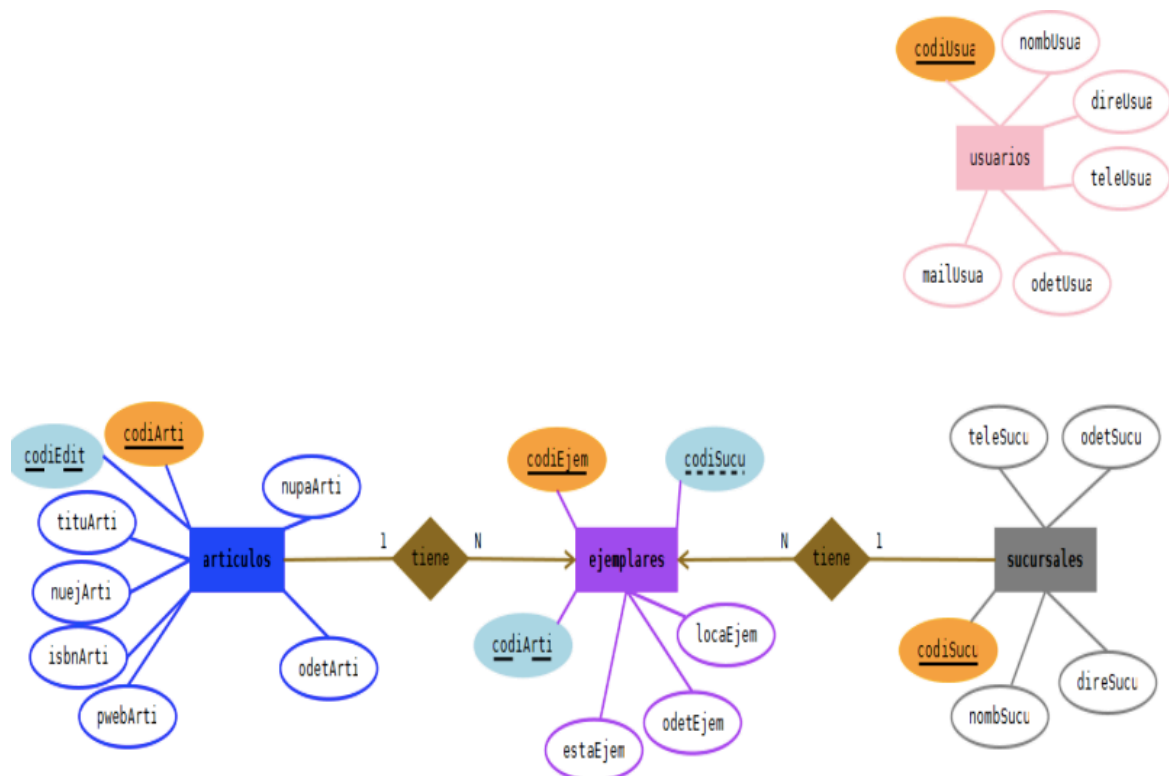
Una biblioteca almacena información de los libros, revistas, tesis, etc. (artículos) que toman en préstamo los alumnos, profesores, egresados, etc. (usuarios).

Por cada libro(artículo) se guarda un identificador único(codiArti), el título del artículo(tituArti) y el nombre(noedArti), dirección(diedArti) y teléfono de su editorial(teedArti).

**Opcional:**

La biblioteca dispone de varios locales a los que llama sucursales. La BD debe contener información a compartir por todas ellas.

Se dispone de una o varios ejemplares de cada libro(artículo), que se encuentran distribuidas por las sucursales.



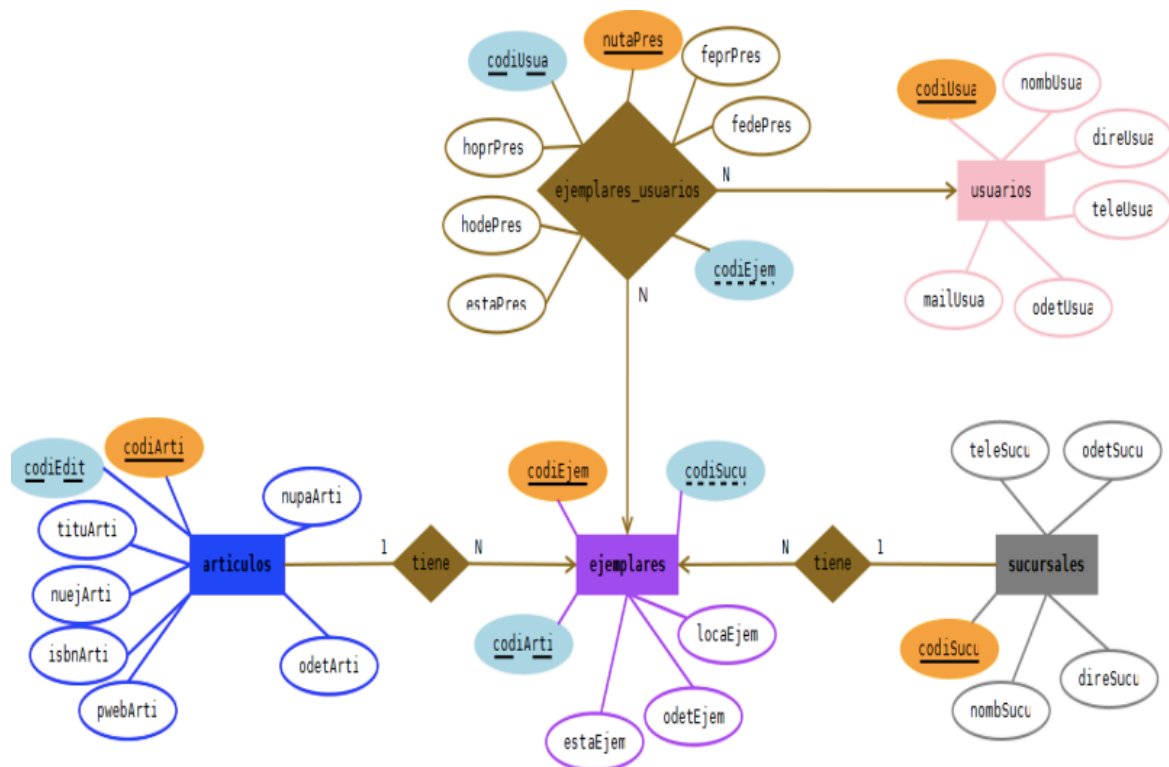
Opcional:

Les interesaría saber por cada libro(artículo) el número de ejemplares (nuejArti) asignadas a cada sucursal, y el identificador(codiSucu) y nombre(nombSucu) únicos de la sucursal junto a la dirección de la sucursal(direSucu).

Un ejemplar de un libro(artículo) se puede encontrar en un momento determinado físicamente en la sucursal a la que está asignado o bien puede estar prestado a un usuario.

De éste último sabemos cuál es el número de su tarjeta(nutaPres), (no hay dos lectores(usuarios) con el mismo número de tarjeta), sabemos su nombre(nombUsua), dirección(direUsua) y teléfono(teleUsua).

Se va guardando qué libros(artículos) tomo prestado un determinado usuario, de qué sucursal lo tomo y la fecha de préstamo(feprPres) y devolución(fedePres) de la última vez que se preste el artículo de la sucursal.



Se deben hacer consultas por autores y almacenar los autores de cada libro(articulo).

Puede ocurrir que hayan autores diferentes que se llamen igual.

Se debe distinguir a dos autores con el mismo nombre(nombAuto) por el libro(articulo) del que son autores.

Se deben hacer consultas por autores y almacenar los autores de cada libro(articulo).

Puede ocurrir que hayan autores diferentes que se llamen igual.

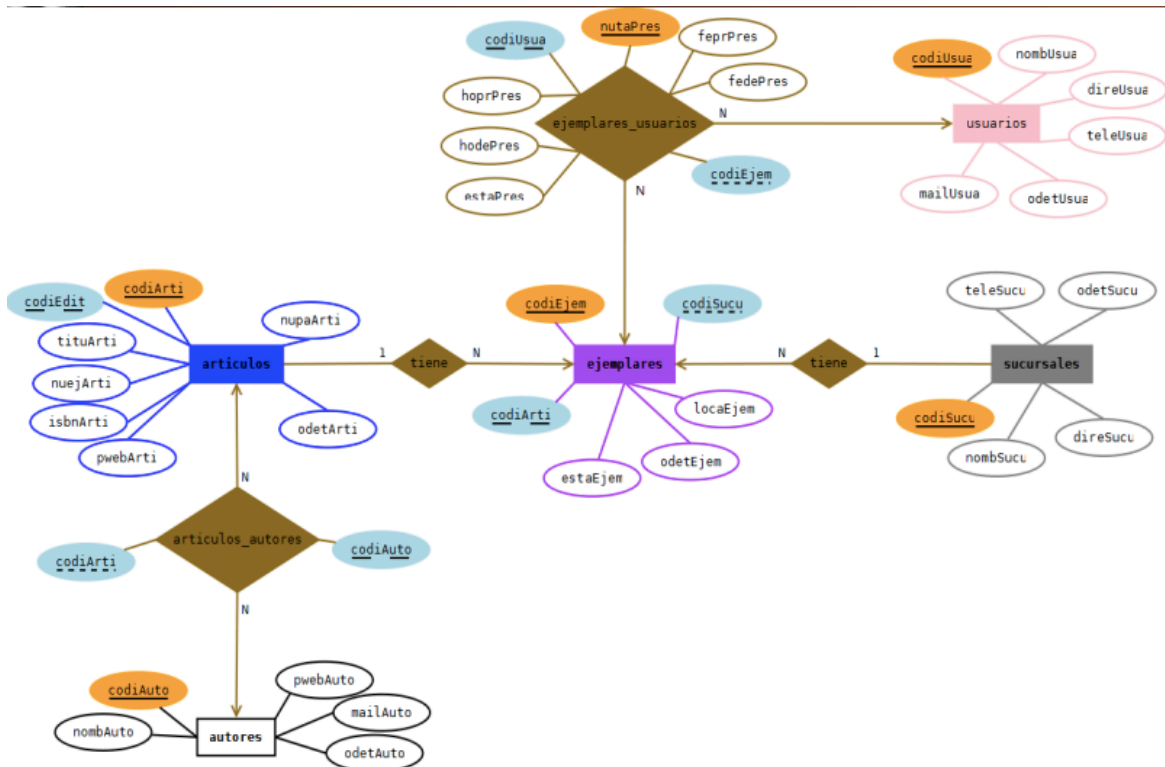
Se debe distinguir a dos autores con el mismo nombre(nombAuto) por el libro(articulo) del que son autores.

En la ficha de cada autor se tiene el código de autor(codiAuto) y el nombre(nombAuto).

De cada libro(articulo) se guarda el código(codiArti), título(tituArti), ISBN(isbnArti), editorial(editArti) y número de páginas(nupaArti).

Un autor puede escribir varios libros(artículos), y un libro(articulo) puede ser escrito por varios autores.

Un libro(articulo) está formado por ejemplares.



Cada ejemplar tiene su código(codiEjem) y su localización(lopeEjem).

Un libro(articulo) tiene muchos ejemplares y un ejemplar pertenece sólo a un libro(articulo).

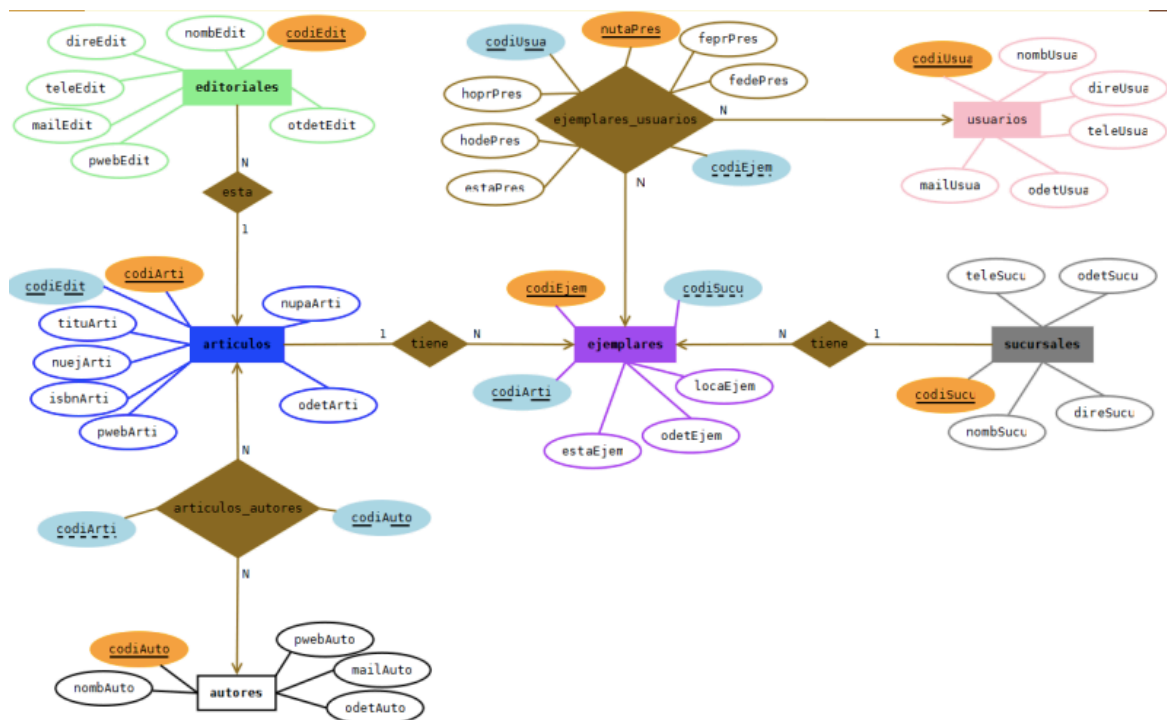
Los usuarios de la biblioteca disponen de fichas en la biblioteca y sacan ejemplares de ella.

De cada usuario se guarda el código, nombre(codiUsua), dirección(direUsua) y teléfono(teleUsua).

Los ejemplares son prestados a los usuarios.

Un usuario puede tomar prestados varios ejemplares, y un ejemplar puede ser prestado a varios usuarios.

De cada préstamo interesa guardar la fecha de préstamo(feprPres) y la fecha de devolución(fedePres).



Modelo físico - Creación de la base de datos

Ahora que tenemos CakePHP instalado, configuremos la base de datos para nuestra aplicación CMS . Si aún no lo ha hecho, cree una base de datos vacía para usar en este tutorial, con un nombre de su elección, por ejemplo cake_cms. Puede ejecutar el siguiente SQL para crear las tablas necesarias:

USE cake_cms;

```
CREATE TABLE users (
  id INT AUTO_INCREMENT PRIMARY KEY,
  email VARCHAR(255) NOT NULL,
  password VARCHAR(255) NOT NULL,
  created DATETIME,
  modified DATETIME
);
```

```
CREATE TABLE articles (
  id INT AUTO_INCREMENT PRIMARY KEY,
  user_id INT NOT NULL,
  title VARCHAR(255) NOT NULL,
  slug VARCHAR(191) NOT NULL,
  body TEXT,
  published BOOLEAN DEFAULT FALSE,
  created DATETIME,
```

```
modified DATETIME,  
UNIQUE KEY (slug),  
FOREIGN KEY user_key (user_id) REFERENCES users(id)  
) CHARSET=utf8mb4;  
  
CREATE TABLE tags (  
  id INT AUTO_INCREMENT PRIMARY KEY,  
  title VARCHAR(191),  
  created DATETIME,  
  modified DATETIME,  
  UNIQUE KEY (title)  
) CHARSET=utf8mb4;  
  
CREATE TABLE articles_tags (  
  article_id INT NOT NULL,  
  tag_id INT NOT NULL,  
  PRIMARY KEY (article_id, tag_id),  
  FOREIGN KEY tag_key(tag_id) REFERENCES tags(id),  
  FOREIGN KEY article_key(article_id) REFERENCES articles(id)  
);  
  
INSERT INTO users (email, password, created, modified)  
VALUES  
('cakephp@example.com', 'secret', NOW(), NOW());  
  
INSERT INTO articles (user_id, title, slug, body, published, created, modified)  
VALUES  
(1, 'First Post', 'first-post', 'This is the first post.', 1, now(), now());
```

Es posible que haya notado que la `articles_tag` tabla utiliza una clave primaria compuesta. CakePHP admite claves primarias compuestas en casi todas partes, lo que le permite tener esquemas más simples que no requieren idcolumnas adicionales .

Los nombres de tabla y columna que usamos no fueron arbitrarios. Al usar las convenciones de nomenclatura de CakePHP , podemos aprovechar CakePHP de manera más efectiva y evitar la necesidad de configurar el marco. Si bien CakePHP es lo suficientemente flexible para adaptarse a casi cualquier esquema de base de datos, adherirse a las convenciones le ahorrará tiempo, ya que puede aprovechar los valores predeterminados basados en convenciones que ofrece CakePHP.

Configuración de la base de datos

A continuación, digamos a CakePHP dónde está nuestra base de datos y cómo conectarse a ella. Reemplace los valores en la `Datasources.default` matriz en su archivo `config / app_local.php` con los que se aplican a su configuración. Una matriz de configuración completa de muestra podría tener el siguiente aspecto:


```
<?php
return [
    // More configuration above.
    'Datasources' => [
        'default' => [
            'className' => 'Cake\Database\Connection',
            'driver' => 'Cake\Database\Driver\Mysql',
            'persistent' => false,
            'host' => 'localhost',
            'username' => 'cakephp',
            'password' => 'AngelF00dC4k3~',
            'database' => 'cake_cms',
            'encoding' => 'utf8mb4',
            'timezone' => 'UTC',
            'cacheMetadata' => true,
        ],
    ],
    // More configuration below.
];
```

Una vez que haya guardado su archivo config / app_local.php , debería ver que la sección 'CakePHP puede conectarse a la base de datos' tiene un gorro de cocinero verde.

Una copia del archivo de configuración predeterminado de CakePHP se encuentra en config / app.default.php .

Diccionario de datos

clientes

Columna	Tipo	Nulo	Predeterminado	Enlaces a	
codigo (<i>Primaria</i>)	varchar(6)	No			
nombre	varchar(50)	No			
apellidoPaterno	varchar(50)	No			
apellidoMaterno	varchar(50)	No			
telefono	varchar(15)	No			

Índices

Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
PRIMARY	BTREE	Sí	No	codigo	9	A	No	

Imprimir

IV

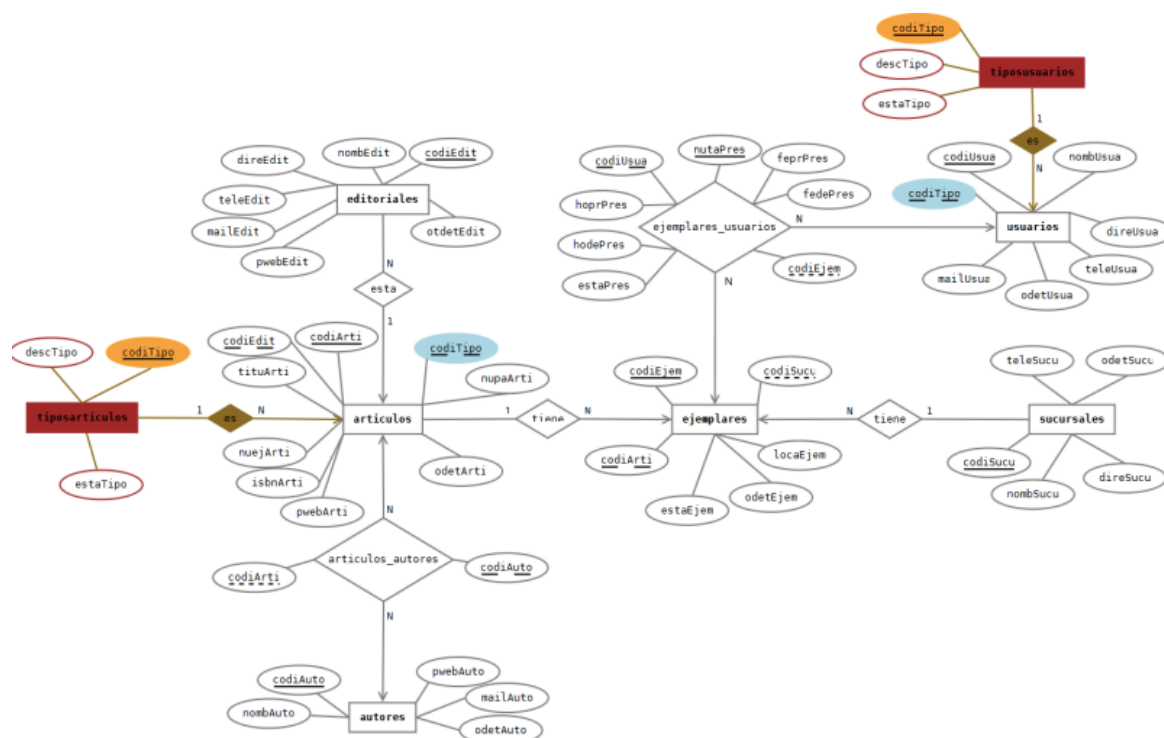
ACTIVIDADES

- Cree un modelo lógico para el contexto donde implementará su aplicación web.
- Cree el modelo físico en una Base de datos SQL o No-SQL.
- Especifique el Diccionario de Datos.

V

EJERCICIOS RESUELTOS

El profesor explica en clase la creación de un modelo lógico para un sistema.



Crear la base de datos.

Crear la entidades tablas.

Crear el diccionario de datos.

VI

EJERCICIOS PROPUESTOS

- Cree un modelo lógico para el contexto donde implementará su aplicación web.
- Cree el modelo físico en una Base de datos SQL o No-SQL.
- Especifique el Diccionario de Datos.

VII

CUESTIONARIO

1. ¿Qué herramientas pueden generar diagramas UML/E-R?
2. ¿Qué tipo de Base de datos utilizará SQL, No-SQL?
3. ¿Qué estándares utilizará para los nombres de los atributos?

VIII

BIBLIOGRAFÍA

- [1] <https://book.cakephp.org/3/en/tutorials-and-examples/cms/database.html>

UNIVERSIDAD NACIONAL DE SAN AGUSTÍN DE AREQUIPA

FACULTAD DE INGENIERÍA DE PRODUCCIÓN Y SERVICIOS

ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS



MAG. RICHART SMITH ESCOBEDO QUISPE
MAG. CHRISTIAN ALAIN REVILLA ARROYO

GUÍA DE LABORATORIO

PROGRAMACIÓN WEB 2

III SEMESTRE
2019 - A

COMPETENCIAS

- Diseña responsablemente sistemas web, componentes o procesos para satisfacer necesidades dentro de restricciones realistas: económicas, medio ambientales, sociales, políticas, éticas, de salud, de seguridad, manufacturación y sostenibilidad.
- Construye responsablemente soluciones web siguiendo un proceso adecuado llevando a cabo las pruebas ajustadas a los recursos disponibles del cliente.
- Aplica de forma flexible técnicas, métodos, principios, normas, estándares y herramientas de la ingeniería web necesarias para la construcción de software e implementación de sistemas de información web.

Laboratorio**3****CRUD – Create – Read – Update
- Delete****I****OBJETIVOS**

- Crear un CRUD (Create – Read – Update – Delete) en un framework MVC.

II**TEMAS A TRATAR**

- Crear un registro
- Leer registros
- Actualización de registros
- Eliminación de registros

III**MARCO TEÓRICO****CREATE - Obteniendo el primer resultado**

El first() método le permite obtener solo la primera fila de una consulta. Si la consulta no se ha ejecutado, se aplicará una cláusula: LIMIT 1

```
// In a controller or table method.  
$query = $articles->find('all', [  
    'order' => ['Articles.created' => 'DESC']  
]);  
$row = $query->first();
```

Este enfoque reemplaza find('first') en versiones anteriores de CakePHP. Es posible que también desee utilizar el get() método si está cargando entidades por clave principal.

El first() método volverá null si no se encuentran resultados.

Obtener un recuento de resultados

Una vez que haya creado un objeto de consulta, puede usar el `count()` método para obtener un recuento de resultados de esa consulta:

```
// In a controller or table method.  
$query = $articles->find('all', [  
    'conditions' => ['Articles.title LIKE' => '%Ovens%']  
]);  
$number = $query->count();
```

Consulte Devolución del recuento total de registros para conocer el uso adicional del `count()` método

Insertar datos

La forma más fácil de insertar datos en la base de datos es creando una nueva entidad y pasándola al `save()` método en la `Table` clase:

```
use Cake\ORM\TableRegistry;  
  
// Prior to 3.6 use TableRegistry::get('Articles')  
$articlesTable = TableRegistry::getTableLocator()->get('Articles');  
$article = $articlesTable->newEntity();  
  
$article->title = 'A New Article';  
$article->body = 'This is the body of the article';  
  
if ($articlesTable->save($article)) {  
    // The $article entity contains the id now  
    $id = $article->id;  
}
```

El `save()` método devuelve la entidad guardada correctamente o false en caso de error.

Actualización de datos

Actualizar sus datos es igualmente fácil, y el `save()` método también se utiliza para ese propósito:

```
use Cake\ORM\TableRegistry;  
  
// Prior to 3.6 use TableRegistry::get('Articles')  
$articlesTable = TableRegistry::getTableLocator()->get('Articles');  
$article = $articlesTable->get(12); // Return article with id 12  
  
$article->title = 'CakePHP is THE best PHP framework!';  
$articlesTable->save($article);
```

CakePHP sabrá si realizar una inserción o una actualización en función del valor de retorno del `isNew()` método. Entidades que se recuperaron con `get()` o `find()` que siempre volverán false cuando `isNew()` se les llame.

Ahorrar con asociaciones

De forma predeterminada, el `save()` método también guardará un nivel de asociaciones:

```
// Prior to 3.6 use TableRegistry::get('Articles')
$articlesTable = TableRegistry::getTableLocator()->get('Articles');
$author = $articlesTable->Authors->findByUserName('mark')->first();

$article = $articlesTable->newEntity();
$article->title = 'An article by mark';
$article->author = $author;

if ($articlesTable->save($article)) {
    // The foreign key value was set automatically.
    echo $article->author_id;
}
```

El `save()` método también puede crear nuevos registros para asociaciones:

```
$firstComment = $articlesTable->Comments->newEntity();
$firstComment->body = 'The CakePHP features are outstanding';

$secondComment = $articlesTable->Comments->newEntity();
$secondComment->body = 'CakePHP performance is terrific!';

$tag1 = $articlesTable->Tags->findByName('cakephp')->first();
$tag2 = $articlesTable->Tags->newEntity();
$tag2->name = 'awesome';

$article = $articlesTable->get(12);
$article->comments = [$firstComment, $secondComment];
$article->tags = [$tag1, $tag2];

$articlesTable->save($article);
```

DELETE - Eliminar datos

```
class Cake\ORM\Table
```

```
Cake\ORM\Table::delete( Entidad $ entidad , $ opciones = [] )
```

Una vez que haya cargado una entidad, puede eliminarla llamando al método de eliminación de la tabla de origen:

```
// In a controller.
$entity = $this->Articles->get(2);
```

```
$result = $this->Articles->delete($entity);
```

Al eliminar entidades suceden algunas cosas:

Se aplicarán las reglas de eliminación . Si las reglas fallan, se evitará la eliminación.

Se `Model.beforeDeleted` desencadena el evento. Si se detiene este evento, se cancelará la eliminación y se devolverá el resultado del evento.

La entidad será eliminada.

Se eliminarán todas las asociaciones dependientes. Si las asociaciones se eliminan como entidades, se enviarán eventos adicionales.

Se eliminarán todos los registros de la tabla de unión de las asociaciones `BelongsToMany`.

El `Model.afterDelete` evento se activará.

De forma predeterminada, todas las eliminaciones ocurren dentro de una transacción. Puede deshabilitar la transacción con la opción atómica:

```
$result = $this->Articles->delete($entity, ['atomic' => false]);
```

Eliminaciones en cascada

Al eliminar entidades, también se pueden eliminar los datos asociados. Si sus asociaciones `HasOne` y `HasMany` están configuradas como `dependent`, las operaciones de eliminación también se realizarán en cascada hacia esas entidades. De forma predeterminada, las entidades de las tablas asociadas se eliminan mediante `Cake\ORM\Table::deleteAll()`. Puede elegir que el ORM cargue las entidades relacionadas y eliminarlas individualmente configurando la `cascadeCallbacks` opción en `true`. Un ejemplo de asociación de `HasMany` con estas dos opciones habilitadas sería:

```
// In a Table's initialize method.  
$this->hasMany('Comments', [  
    'dependent' => true,  
    'cascadeCallbacks' => true,  
]);
```

Establecer `cascadeCallbacks` en `true`, da como resultado eliminaciones considerablemente más lentas en comparación con las eliminaciones masivas. La opción `cascadeCallbacks` solo debe habilitarse cuando su aplicación tiene un trabajo importante manejado por detectores de eventos.

Eliminaciones masivas

```
Cake\ORM\Table::deleteAll( $ condiciones )
```

Puede haber ocasiones en las que eliminar filas una por una no sea eficaz o útil. En estos casos, es más eficaz utilizar una eliminación masiva para eliminar muchas filas a la vez:


```
// Delete all the spam
function destroySpam()
{
    return $this->deleteAll(['is_spam' => true]);
}
```

Una eliminación masiva se considerará exitosa si se eliminan 1 o más filas. La función devuelve el número de registros eliminados como un número entero.

deleteAll no activará los eventos beforeDelete / afterDelete. Si necesita que se activen las devoluciones de llamada, primero cargue las entidades find() y elimínelas en un bucle.

IV

ACTIVIDADES

- Crear un registro
- Leer registros
- Actualización de registros
- Eliminación de registros

V

EJERCICIOS RESUELTOS

creación del controlador de artículos

Con nuestro modelo creado, necesitamos un controlador para nuestros artículos. Los controladores en CakePHP manejan las solicitudes HTTP y ejecutan la lógica empresarial contenida en los métodos del modelo para preparar la respuesta. Colocaremos este nuevo controlador en un archivo llamado ArticlesController.php dentro del directorio src / Controller . Así es como debería verse el controlador básico:

```
<?php
// src/Controller/ArticlesController.php

namespace App\Controller;

class ArticlesController extends AppController
{
}
```

Ahora, agreguemos una acción a nuestro controlador. Las acciones son métodos de controlador que tienen rutas conectadas. Por ejemplo, cuando un usuario solicita www.example.com/articles/index (que también es lo mismo que

www.example.com/articles), CakePHP llamará al `index` método de su `ArticlesController`. Este método debe consultar la capa del modelo y preparar una respuesta al representar una plantilla en la vista. El código para esa acción se vería así:

```
<?php
// src/Controller/ArticlesController.php

namespace App\Controller;

class ArticlesController extends AppController
{
    public function index()
    {
        $this->loadComponent('Paginator');
        $articles = $this->Paginator->paginate($this->Articles->find());
        $this->set(compact('articles'));
    }
}
```

Al definir la función `index()` en nuestro `ArticlesController`, los usuarios ahora pueden acceder a la lógica allí solicitando `www.example.com/articles/index`. De manera similar, si definiéramos una función llamada `foobar()`, los usuarios podrían acceder a ella en `www.example.com/articles/foobar`. Puede tener la tentación de nombrar sus controladores y acciones de una manera que le permita obtener URL específicas. Resista esa tentación. En su lugar, siga las convenciones de CakePHP para crear nombres de acción legibles y significativos. Luego, puede usar Enrutamiento para conectar las URL que desea con las acciones que ha creado.

Nuestra acción de controlador es muy simple. Obtiene un conjunto paginado de artículos de la base de datos, utilizando el modelo de artículos que se carga automáticamente mediante convenciones de nomenclatura. Luego se usa `set()` para pasar los artículos a la Plantilla (que crearemos pronto). CakePHP renderizará automáticamente la plantilla después de que se complete la acción de nuestro controlador.

Agregar acción de eliminación

A continuación, abramos una forma para que los usuarios eliminen artículos. Comience con una `delete()` acción en `ArticlesController`:

```
// src/Controller/ArticlesController.php

public function delete($slug)
{
    $this->request->allowMethod(['post', 'delete']);

    $article = $this->Articles->findBySlug($slug)->firstOrFail();
    if ($this->Articles->delete($article)) {
        $this->Flash->success(__('The {0} article has been deleted.', $article->title));
    }
}
```

```
        return $this->redirect(['action' => 'index']);  
    }  
}
```

Esta lógica elimina el artículo especificado por \$slugy se utiliza \$this->Flash->success() para mostrar al usuario un mensaje de confirmación después de redirigirlo a /articles. Si el usuario intenta eliminar un artículo mediante una solicitud GET, allowMethod() lanzará una excepción. Las excepciones no detectadas son capturadas por el controlador de excepciones de CakePHP y se muestra una bonita página de error. Hay muchas excepciones integradas que se pueden usar para indicar los diversos errores HTTP que su aplicación podría necesitar generar.

Permitir que se elimine contenido mediante solicitudes GET es muy peligroso, ya que los rastreadores web podrían eliminar accidentalmente todo su contenido. Es por eso que usamos allowMethod() en nuestro controlador.

VI

EJERCICIOS PROPUESTOS

- Cree un método para crear un registro en la base de datos usando el framework MVC.
- Cree un método para editar un registro en la base de datos usando el framework MVC.
- Cree un método para eliminar un registro en la base de datos usando el framework MVC.

VII

CUESTIONARIO

1. ¿Cómo se pueden guardar varios registros?
2. ¿Cómo se pueden eliminar varios registros?
3. ¿Cómo resolvería eliminar registros previa evaluación?
4. Investigue herramientas para autogenerar CRUD.

VIII

BIBLIOGRAFÍA

- [1] <https://book.cakephp.org/3/en/orm/retrieving-data-and-resultsets.html>
- [2] <https://book.cakephp.org/3/en/orm/saving-data.html>
- [3] <https://book.cakephp.org/3/en/orm/deleting-data.html>

UNIVERSIDAD NACIONAL DE SAN AGUSTÍN DE AREQUIPA

FACULTAD DE INGENIERÍA DE PRODUCCIÓN Y SERVICIOS

ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS



MAG. RICHART SMITH ESCOBEDO QUISPE
MAG. CHRISTIAN ALAIN REVILLA ARROYO

GUÍA DE LABORATORIO

PROGRAMACIÓN WEB 2

III SEMESTRE
2019 - A

COMPETENCIAS

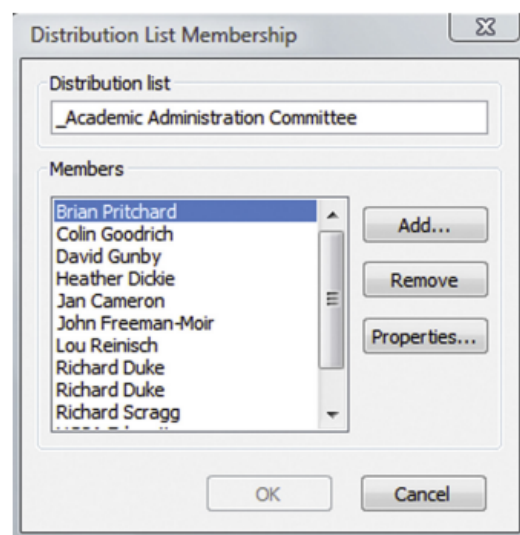
- Diseña responsablemente sistemas web, componentes o procesos para satisfacer necesidades dentro de restricciones realistas: económicas, medio ambientales, sociales, políticas, éticas, de salud, de seguridad, manufacturación y sostenibilidad.
- Construye responsablemente soluciones web siguiendo un proceso adecuado llevando a cabo las pruebas ajustadas a los recursos disponibles del cliente.
- Aplica de forma flexible técnicas, métodos, principios, normas, estándares y herramientas de la ingeniería web necesarias para la construcción de software e implementación de sistemas de información web.

Laboratorio**4****AJAX****I****OBJETIVOS**

- Crear operaciones asincronas utilizando un framework MVC.

II**TEMAS A TRATAR**

- Crear una vista accesible para realizar operaciones CRUD.
- Agregue operaciones asíncronas a esa vista.

III**MARCO TEÓRICO****Interfaz gráfica de usuarios para hacer CRUD****FIGURE 2.2**

In Outlook's Distribution List Membership dialog box, list buttons are in a group box, separate from the control buttons.

Vinculación a archivos Javascript

`Cake\View\Helper\HtmlHelper::script($ url mixta , $ opciones mixtas)`

Incluya un archivo de script, contenido localmente o como una URL remota.

De forma predeterminada, las etiquetas de secuencia de comandos se agregan al documento en línea. Si anula esto estableciendo `$options['block']` en true, las etiquetas de secuencia de comandos se agregarán al script bloque que puede imprimir en otra parte del documento. Si desea anular qué nombre de bloque se usa, puede hacerlo configurando `$options['block']`.

`$options['once']` controla si desea o no incluir este script una vez por solicitud o más de una vez. Este valor predeterminado es true.

Puede usar `$options` para establecer propiedades adicionales a la etiqueta de secuencia de comandos generada. Si se utiliza una matriz de etiquetas de secuencia de comandos, los atributos se aplicarán a todas las etiquetas de secuencia de comandos generadas.

Este método de inclusión de archivos JavaScript asume que el archivo JavaScript especificado reside dentro del directorio webroot / js :

```
echo $this->Html->script('scripts');
```

Saldrá:

```
<script src="/js/scripts.js"></script>
```

También puede vincular archivos con rutas absolutas para vincular archivos que no están en webroot / js :

```
echo $this->Html->script('/otherdir/script_file');
```

También puede vincular a una URL remota:

```
echo $this->Html->script('http://code.jquery.com/jquery.min.js');
```

Saldrá:

```
<script src="http://code.jquery.com/jquery.min.js"></script>
```

El primer parámetro puede ser una matriz para incluir varios archivos.

```
echo $this->Html->script(['jquery', 'wysiwyg', 'scripts']);
```

Saldrá:

```
<script src="/js/jquery.js"></script>
```

```
<script src="/js/wysiwyg.js"></script>
```

```
<script src="/js/scripts.js"></script>
```

AJAX

Este objeto será el contexto de todas las devoluciones de llamada relacionadas con Ajax. De forma predeterminada, el contexto es un objeto que representa la configuración de Ajax utilizada en la llamada (\$.ajaxSettings combinada con la configuración pasada \$.ajax).

Por ejemplo, especificar un elemento DOM como contexto hará que el contexto para la completadevolución de llamada de una solicitud, así:

```
$.ajax({  
  url: "test.html",  
  context: document.body  
}).done(function() {  
  $( this ).addClass( "done" );  
});
```

IV

ACTIVIDADES

- Crear una sola interfaz para crear un CRUD.
- Usar AJAX para implementar las funcionalidades del CRUD.

V

EJERCICIOS RESUELTOS

Vista

```
<!DOCTYPE html>  
<html>  
  <head>  
    <title>Ajax - Example</title>  
    <script src="jquery-3.5.1.min.js"></script>  
  <script>  
    function ajaxFunction(){  
      $.ajax({  
url: "ajax.php",  
beforeSend: function( xhr ) {  
  xhr.overrideMimeType( "text/plain; charset=utf-8;" );  
}  
})  
.done(function( data ) {  
  if ( console && console.log ) {  
    console.log( "result:", data );  
    $('#ajaxResult').html(data);  
  }  
})  
}
```



```
}  
});  
    </script>  
    </head>  
    <body>  
        <button type="button" onclick="ajaxFunction()">Ajax function</button>  
        <div id="ajaxResult"></div>  
    </body>  
</html>
```

Backend

```
<?php
```

```
$team = array(  
    1=>'José',  
    2=>'María',  
    3=>'Jesus',  
    4=>'Estefany',  
    5=>'Carlos',  
    6=>'Teresa',  
    7=>'Luis',  
    8=>'Juan',  
    9=>'Sofia',  
    10=>'Marco'  
);  
  
$number = rand(1,10);  
  
echo $team[$number];  
  
?>
```

JQuery

<https://github.com/rescobedoq/ajax-php/blob/main/jquery-3.5.1.min.js>

VI

EJERCICIOS PROPUESTOS

- Crear una sola interfaz para crear un CRUD.
- Usar AJAX para implementar las funcionalidades del CRUD.

VII

CUESTIONARIO

1. ¿Cómo recibir un array desde el controlador y convertirlo en un JSON?
2. ¿Cómo recibir código HTML desde el controlador usando AJAX?
3. ¿Cómo recibir XML/JSON desde el controlador usando AJAX?

VIII

BIBLIOGRAFÍA

- [1] [Designing with the Mind in Mind Simple Guide to Understanding. User Interface Design Guidelines. Second Edition. Jeff Johnson](#)
- [2] <https://github.com/rescobedoq/ajax-php>
- [3] <https://github.com/rescobedoq/iw/tree/master/cakephp/3.8/v0.7/cakecms>