

"ACTUALIZACIÓN "EL FARO"

El Faro

Iluminando a la comunidad con noticias que importan

Mantente informado con las últimas noticias locales, eventos comunitarios y todo lo que sucede en nuestra región.

↓ Ver Noticias

Maximiliano Galindo Díaz

Aplicaciones para internet

Semana 7
28 de septiembre

Objetivos del tutorial

Criterios:

- Determina requerimientos de usuario para modelado de base de datos
- Determina estructura de base de datos según análisis de requerimientos

Resultados de aprendizaje:

- Diseñar diagrama ER a partir de requisitos
- Implementar modelos CodeIgniter 4 con Active Record
- Crear relaciones 1:N entre tablas MySQL
- Desarrollar operaciones CRUD completas

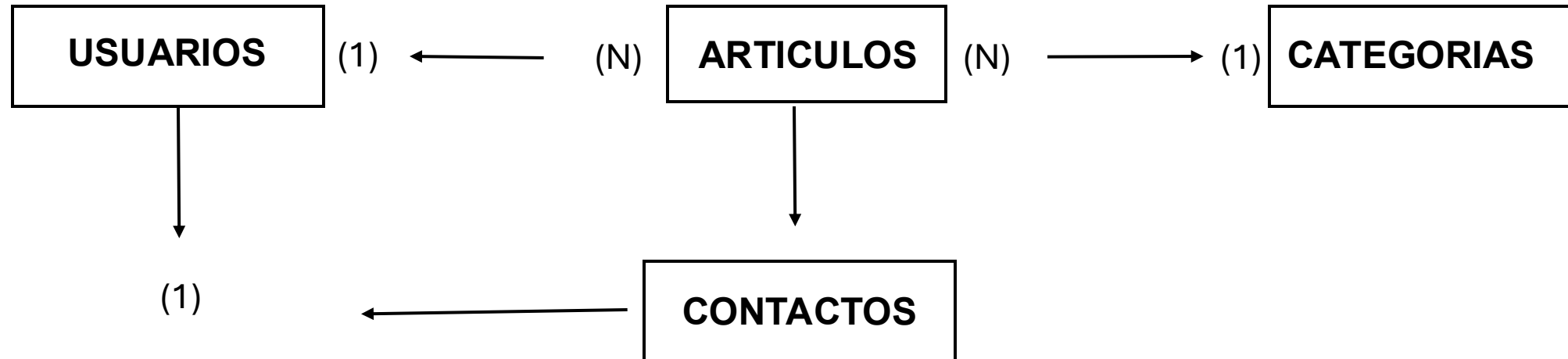
PASO 1 - Análisis de Requerimientos → Entidades

```
-- REQUERIMIENTO: Registrar nuevo artículo  
→ ENTIDAD: ArtículoModel (tabla 'articulos')  
  
-- REQUERIMIENTO: Presentar artículos por página  
→ ENTIDAD: ArtículoModel + CategoriaModel (JOIN)  
  
-- REQUERIMIENTO: Registrar datos de contacto  
→ ENTIDAD: ContactoModel (tabla 'contactos')  
  
-- REQUERIMIENTO: Mostrar usuarios registrados  
→ ENTIDAD: UsuarioModel (tabla 'usuarios')
```

Ejercicio práctico:

- Mostrar cómo cada requisito funcional se mapea a una entidad
 - Identificar atributos necesarios para cada entidad

PASO 2 - Diseño del Modelo Entidad-Relación (CON RELACIONES)



```
// Relación 1:N - Categoría → Artículos
```

```
class CategoriaModel {  
    public function articulos() {  
        return $this->  
>hasMany(ArticuloModel::class);  
    }  
}
```

Relaciones identificadas:

```
// Relación 1:N - Usuario → Artículos  
class UsuarioModel {  
    public function articulos() {  
        return $this->  
>hasMany(ArticuloModel::class);  
    }  
}
```

Diagrama de clases (parte 2)

```
class ArtículoModel extends Model {  
  - table: 'articulos'  
  - primaryKey: 'id'  
  - allowedFields: array  
  
  +  
  obtenerArticulosConCategoria($limite,  
    $categoriaId)  
    + obtenerArticuloCompleto($id)  
    +  
  obtenerArticulosRecientes($limite)  
    + buscarArticulos($termino)  
}
```

```
class ContactoModel extends Model {  
  - table: 'contactos'  
  - primaryKey: 'id'  
  - allowedFields: array  
  
  + obtenerContactosNoLeidos()  
  + marcarComoLeido($id)  
  + contarNoLeidos()  
}
```

PASO 3: Implementación de ArtículoModel.php

```
<?php
namespace App\Models;
use CodeIgniter\Model;

class ArtículoModel extends Model
{
    // PASO 3.1 - Configuración básica
    // del modelo
    protected $table =
'articulos';
    protected $primaryKey = 'id';
    protected $useAutoIncrement = true;
```

```
// PASO 3.2 - Campos permitidos para
// inserción segura
    protected $allowedFields = [
        'titulo', 'contenido',
        'resumen', 'imagen',
        'categoria_id', 'usuario_id',
        'fecha_publicacion', 'activo'
    ];
```

Implementación de ArtículoModel.php (parte 2)

```
// PASO 3.3 - Validaciones automáticas
protected $validationRules = [
    'titulo' =>
    'required|min_length[5]|max_length[200]
',
    'contenido' =>
    'required|min_length[10]',
    'categoria_id' =>
    'required|integer'
];
```

Explicación de cada sección:

- ¿Por qué usar \$allowedFields? (seguridad)
- ¿Cómo funcionan las \$validationRules? (validación automática)
- ¿Qué es \$primaryKey? (clave primaria)


```
// PASO 4.1 - Método para obtener artículos con categorías
public function obtenerArticulosConCategoria($limite = 10, $categoriaId = null)
{
    $builder = $this->db->table('articulos a');
    $builder->select('a.*, c.nombre as categoria_nombre, u.nombre as autor');
    $builder->join('categorias c', 'a.categoria_id = c.id', 'left');
    $builder->join('usuarios u', 'a.usuario_id = u.id', 'left');
    $builder->where('a.activo', 1);

    if ($categoriaId) {
        $builder->where('a.categoria_id', $categoriaId);
    }

    $builder->orderBy('a.fecha_publicacion', 'DESC');
    $builder->limit($limite);

    return $builder->get()->getResultArray();
}
```

PASO 4 - Métodos Personalizados de ArtículoModel (OPERACIONES AVANZADAS)

Explicación de cada sección:

- ¿Por qué usar \$allowedFields? (seguridad)
- ¿Cómo funcionan las \$validationRules? (validación automática)
- ¿Qué es \$primaryKey? (clave primaria)

```
// PASO 4.2 - Método de búsqueda
public function buscarArticulos($termino)
{
    return $this->like('titulo', $termino)
        ->orLike('contenido', $termino)
        ->orLike('resumen', $termino)
        ->where('activo', 1)
        ->findAll();
}
```

PASO 5 - Implementación de Relaciones entre Modelos

Implementación de relaciones 1:N:

```
-- PASO 5.1 - SQL para crear tabla con claves foráneas
CREATE TABLE articulos (
  id INT PRIMARY KEY AUTO_INCREMENT,
  titulo VARCHAR(255) NOT NULL,
  contenido TEXT NOT NULL,
  categoria_id INT NOT NULL,
  usuario_id INT NOT NULL,
  fecha_publicacion DATETIME DEFAULT CURRENT_TIMESTAMP,
  activo TINYINT(1) DEFAULT 1,
  FOREIGN KEY (categoria_id) REFERENCES categorias(id) ON DELETE RESTRICT,
  FOREIGN KEY (usuario_id) REFERENCES usuarios(id) ON DELETE CASCADE
```

```
-- PASO 5.2 - Crear índices para optimización
CREATE INDEX idx_articulos_categoria ON articulos(categoria_id);
CREATE INDEX idx_articulos_usuario ON articulos(usuario_id);
CREATE INDEX idx_articulos_fecha ON articulos(fecha_publicacion);
```

Explicación de CONSTRAINTS:

- ON DELETE RESTRICT vs ON DELETE CASCADE
- Importancia de los índices para rendimiento
- Cómo verificar relaciones en phpMyAdmin

PASO 6 - Uso del Modelo en Controladores (INTEGRACIÓN MVC)

Ejemplo práctico en Home controller:

```
<?php
namespace App\Controllers;

use App\Models\ArticuloModel;
use App\Models\CategoriaModel;

class Home extends BaseController
{
    // PASO 6.1 - Instanciar modelos en constructor
    protected $articuloModel;
    protected $categoriaModel;

    public function __construct()
    {
        $this->articuloModel = new ArticuloModel();
        $this->categoriaModel = new
CategoriaModel();
    }
}
```

```
// PASO 6.2 - Usar modelo para obtener datos
```

```
public function index()
```

```
{
```

```
    $data = [
```

```
        'articulos' => $this->articuloModel->obtenerArticulosConCategoria(10),
```

```
        'categorias' => $this->categoriaModel->obtenerCategoriasActivas()
```

```
    ];
```

```
    return view('home/index', $data);
```

```
}
```

Explicación de CONSTRAINTS:

- ON DELETE RESTRICT vs ON DELETE CASCADE
- Importancia de los índices para rendimiento
- Cómo verificar relaciones en phpMyAdmin

```
// PASO 6.3 - Método para búsqueda
public function buscar()
{
    $termino = $this->request->getGet('q');
    $data['resultados'] = $this->articuloModel->buscarArticulos($termino);
    $data['termino'] = $termino;

    return view('home/buscar', $data);
}
?>
```

PASO 7 - Vista que Utiliza Datos del Modelo (FRONTEND)

Código de vista home/index.php:

```
<!-- PASO 7.1 - Recibir datos del controlador -->
<div class="container">
  <h1>Últimas Noticias</h1>

  <!-- PASO 7.2 - Iterar sobre artículos -->
  <?php foreach ($articulos as $articulo): ?>
    <article class="card mb-4">
      <div class="card-body">
        <h2><?= esc($articulo['titulo']) ?></h2>
```

```

<!-- PASO 7.3 - Mostrar datos relacionados -->
<div class="article-meta">
    <span class="badge bg-primary">
        <?= esc($articulo['categoria_nombre']) ?>
    </span>
    <span class="text-muted">
        Por <?= esc($articulo['autor']) ?>
        el <?= date('d/m/Y', strtotime($articulo['fecha_publicacion'])) ?>
    </span>
</div>

<p><?= esc($articulo['resumen']) ?></p>
<a href="/articulo/<?= $articulo['id'] ?>" class="btn btn-primary">
    Leer más
</a>
</div>
</article>
<?php endforeach; ?>
</div>

```

Buenas prácticas mostradas:

- Uso de esc() para prevenir XSS
- Formateo de fechas con date()
- Clases de Bootstrap para responsividad
- URLs amigables

PASO 8 - Implementación Completa del CRUD (OPERACIONES ADMIN)

Ejemplo de AdminArticulosController:

```
// PASO 8.1 - Crear nuevo artículo
public function crear()
{
    if ($this->request->getMethod() === 'POST') {
        $datos = [
            'titulo' => $this->request->getPost('titulo'),
            'contenido' => $this->request->getPost('contenido'),
            'categoria_id' => $this->request->getPost('categoria_id'),
            'usuario_id' => session()->get('usuario_id')
        ];

        // El modelo valida automáticamente
        if ($this->articuloModel->save($datos)) {
            return redirect()->to('/admin/articulos')->with('success', 'Artículo creado');
        }
    }
}
```

Validaciones automáticas del modelo en acción

```
    return view('admin/articulos/crear');  
}  
  
// PASO 8.2 - Editar artículo existente  
public function editar($id)  
{  
    $articulo = $this->articuloModel->find($id);  
    return view('admin/articulos/editar', ['articulo' =>  
$articulo]);  
Validaciones auto
```

Pruebas y Validación (CHECKLIST TÉCNICO)

```
// TEST 1: ¿Se crean artículos correctamente?
$model = new ArtículoModel();
$id = $model->insert([
    'titulo' => 'Test Article',
    'contenido' => 'Content here',
    'categoria_id' => 1,
    'usuario_id' => 1
]);
echo $id ? "✓ Artículo creado (ID: $id)" : "X Error al crear";

// TEST 2: ¿Las relaciones funcionan?
$articulos = $model->obtenerArticulosConCategoria(5);
echo count($articulos) . " artículos con categorías cargados";
```

Checklist de cumplimiento:

- 4 modelos implementados (Artículo, Categoría, Usuario, Contacto)
- Relaciones 1:N funcionando
- Validaciones automáticas activas
- CRUD completo operativo
- Diseño responsivo mantenido
- Arquitectura MVC intacta

Descargas y Recursos (MATERIAL PRÁCTICO)

Archivos incluidos en el tutorial:

/models/

- `ArticuloModel.php` CRUD + búsqueda + relaciones
- `CategoriaModel.php` Gestión categorías + slugs
- `UsuarioModel.php` Autenticación + seguridad
- `ContactoModel.php` Formularios + notificaciones

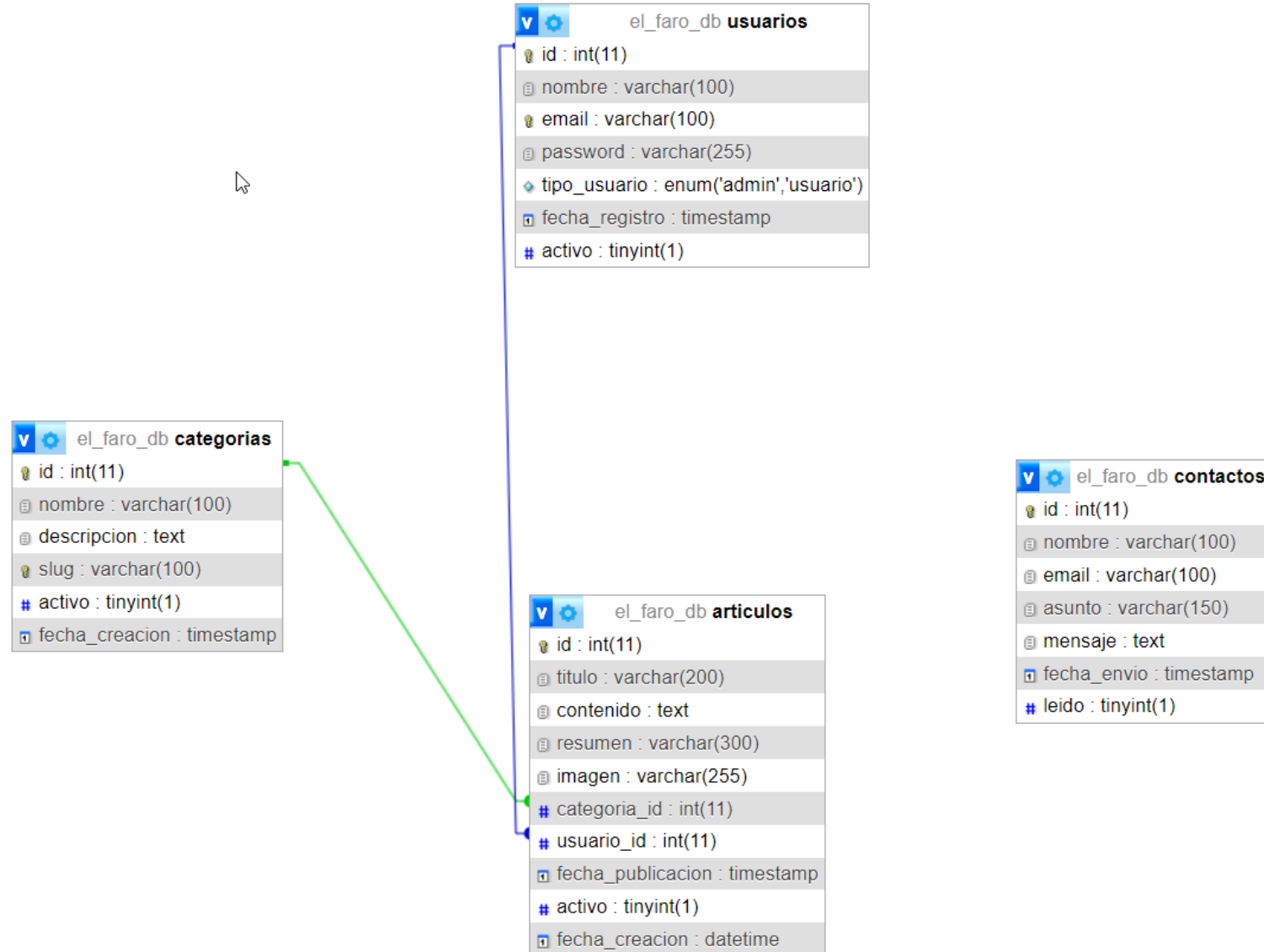
/sql/

- `database.sql` Estructura completa con relaciones
- `datos_ejemplo.sql` Datos de prueba


/controllers/


- `Home.php` Frontend con uso de modelos
- `AdminArticulos.php` CRUD administrativo


Anexos (1)




Anexos (2)

 Inicio Secciones ▼ Contacto

Buscar noticias... 

 Administrador ▼

 Información del Artículo

Título del Artículo *

Ingresa un título atractivo...

Mínimo 5 caracteres, máximo 200

Categoría *

Seleccionar categoría...

▼

Imagen del Artículo

Elegir archivo

No se eligió ningún archivo

Formatos: JPG, PNG, GIF. Máximo 2MB


Resumen/Entradilla

Breve resumen del artículo que aparecerá en las listas...

Máximo 300 caracteres. Este texto aparecerá en la página principal.


Contenido del Artículo *


Escribe aquí el contenido completo del artículo...

 Vista Previa

El título aparecerá aquí...

El resumen aparecerá aquí...

 30/09/2025 02:39


 Consejos de Redacción

✓ **Título:** Claro y atractivo

✓ **Resumen:** Resume la noticia en 2-3 líneas

✓ **Contenido:** Información completa y verificada

✓ **Imagen:** Relacionada con el contenido

 Categorías Disponibles

Noticias Locales

Deportes

Cultura

Economía

Tecnología

Selecciona la categoría que mejor describa tu artículo para facilitar la navegación de los lectores.

Anexos (3)

Editar Artículo

Modifica el contenido del artículo: Inauguración del nuevo centro comunitario

[← Volver a Artículos](#)

[Ver Publicado](#)

Editar Información del Artículo

 Creado: 24/09/2025 17:33

Título del Artículo *

Inauguración del nuevo centro comunitario

Mínimo 5 caracteres, máximo 200

Categoría *

Noticias Locales



Imagen del Artículo

Elegir archivo

No se eligió ningún archivo

Formatos: JPG, PNG, GIF. Máximo 2MB. Dejar vacío para mantener imagen actual.

Resumen/Entradilla

Nuevo centro comunitario beneficiará a 500 familias de la zona norte

Máximo 300 caracteres. Este texto aparecerá en la página principal.

Contenido del Artículo *

El alcalde inauguró oficialmente el nuevo centro comunitario que servirá a más de 500 familias de la zona norte de la ciudad. Las instalaciones cuentan con biblioteca, sala de computación y espacios recreativos.

Estadísticas del Artículo

Estado actual: **Publicado**

Fecha de creación:

24/09/2025 17:33

Categoría actual:

Noticias Locales

ID del artículo: #6

Vista Previa



Sin imagen

Inauguración del nuevo centro comunitario

Nuevo centro comunitario beneficiará a 500 familias de la zona norte

 24/09/2025 17:33

Anexos (4)

El Faro

[Inicio](#) [Secciones](#) [Contacto](#)

Buscar noticias...

Administrador

Total Usuarios

Usuarios Activos




Administradores

Usuarios Regulares

Listado de Usuarios Registrados

Buscar usuario...

Filtrar

ID	Nombre	Email	Tipo	Fecha Registro	Estado	Acciones
1	 Administrador	admin@elfaro.cl	Administrador	24/09/2025 17:33	Activo	 

Mostrando 1 usuario(s) registrado(s)

Opciones de Exportación

Exporta la lista de usuarios para análisis externo o backup.

Exportar CSV

Exportar Excel

Exportar PDF

El Faro

Enlaces Rápidos

Inicio

Secciones

Noticias Locales

Síguenos



Anexos (5)

Para

redaccion@elfaro.cl <redaccion@elfaro.cl> X

CC CCO

Agregar un asunto

☐ Acepto la [política de privacidad](#) y el tratamiento de mis datos

 Enviar mensaje



Noticias y Reportajes

¿Tienes una noticia que contar?

redaccion@elfaro.cl



Publicidad

Promociona tu negocio con nosotros

publicidad@elfaro.cl




Soporte Técnico

Problemas con el sitio web


soporte@elfaro.cl

Síguenos

 Facebook

 Twitter

 Instagram

 YouTube

BIBLIOGRAFÍA

BIBLIOGRAFÍA

- - CodeIgniter 4 User Guide - Models (2024)
- https://codeigniter.com/user_guide/models/
-
- - Material didáctico Semana 7 - Bases de Datos MVC
 - AIEP Educación Online (2025)
 -
- - MySQL Documentation - Foreign Keys
 - <https://dev.mysql.com/doc/>
 -
- - Carbon - Beautiful code snippets
 - <https://carbon.now.sh>

URL DE ACCESO

- [ACCESO AL SITIO WEB]
-

ACCESO AL SITIO WEB:

- <http://localhost/elfaro/public>

- Credenciales de prueba:
Usuario: admin@elfaro.cl
Contraseña: [admin123]