



# **DIVULGACIÓN ACADÉMICA**

Proyecto D

Desarrollo de Aplicaciones Web II

JENNIFER FUENTE MANZANARES  
ANDREA COLORADO ESTEBAN  
ENRIQUE CUESTA PALAFOX  
EVA SANABRIA SANTIAGO?  
ALFONSO ONDO OYONO  
JUAN RIEGO VILA

4º GIISI, Curso 2025-2026

# ÍNDICE

ÍNDICE.....	1
1. CONTROLADORES.....	5
Archivo: /codigo/controllers/UsuarioController.php.....	5
1. Control de Acceso y Comportamientos (behaviors).....	5
2. Gestión de Usuarios (CRUD) .....	5
3. Lógica de Actualización y Perfil.....	5
Archivo: /codigo/controllers/SiteController.php.....	5
1. Autenticación y Seguridad.....	6
2. Sistema de Auditoría (actionLogs) .....	6
3. Módulo de Preguntas y Respuestas (actionQa).....	6
4. Acciones Utilitarias .....	6
2. MODELOS .....	6
Archivo: /codigo/models/RegistroUsuarios.php .....	6
1. Validación de Datos (rules).....	7
2. Lógica de Registro (signup).....	7
Archivo: /codigo/models/UsuarioSearch.php.....	7
1. Escenarios y Reglas .....	7
2. Construcción de Consultas (search).....	7
Archivo: /codigo/models/UserLog.php.....	7
1. Automatización del Registro (add) .....	8
Archivo: /codigo/models/LoginForm.php.....	8
1. Validación en Línea (validatePassword) .....	8
2. Gestión de Sesión (login).....	8
Archivo: /codigo/models/Usuario.php.....	8
1. Implementación de IdentityInterface .....	8
2. Reglas de Negocio y Escenarios .....	8
3. Control de Roles (ACL) .....	9
4. Seguridad de Contraseñas .....	9
3. VISTAS.....	9
3.1. /codigo/views/usuario/ .....	9
Archivo: /codigo/views/index.php.....	9
Archivo: /codigo/views/usuario/_search.php .....	9
Archivo: /codigo/views/usuario/create.php y update.php .....	9
Archivo: /codigo/views/usuario/_form.php.....	10
Archivo: /codigo/views/usuario/view.php.....	10
Archivo: /codigo/views/usuario/perfil.php.....	10
Archivos: /codigo/views/usuario/Archivos Parciales (_tab_*.php).....	10

3.2. /codigo/views/site/ .....	11
Archivo: /codigo/views/site/login.php .....	11
Archivo: /codigo/views/site/register.php .....	11
BASE DE DATOS.....	12
Creación.....	12
Tabla Usuario .....	12
Tabla Institución .....	12
Tabla Materia.....	13
Tabla Personal.....	13
Tabla Documento .....	13
Tabla documento_comentario .....	13
Tabla documento_valoracion .....	14
Tabla pregunta.....	14
Tabla pregunta_guardada.....	14
Tabla pregunta_voto .....	14
Tabla respuesta .....	15
Tabla user_log .....	15
Tabla coleccion .....	15
Tabla coleccion_documento .....	15
Tabla usuario_coleccion .....	16
Tabla Servicio .....	16
Tabla ClienteEmpresa.....	16
Tabla Contratación .....	16
Tabla Evento.....	17
Tabla Incidencia .....	17
Resultado .....	17
Introducción de datos .....	18
Tabla usuario.....	18
Tabla institución .....	18
Tabla materia .....	19
Tabla personal.....	19
Tabla documento.....	20
Tabla servicio .....	20
Tabla user_log .....	20
Tabla pregunta.....	21
Tabla coleccion .....	21
Tabla contratacion.....	21
Tabla documento_comentario .....	21
Tabla documento_valoracion .....	22

Tabla pregunta_guardada.....	22
Tabla pregunta_voto .....	22
Tabla respuesta .....	22
Tabla colección_documento .....	23
Tabla usuario_coleccion .....	23
Tabla Cliente-Empresa .....	23
Tabla Contratación .....	23
Tabla Evento .....	24
Tabla Incidencia .....	24
Implementar base de datos .....	24
BACKEND.....	25
Colecciones .....	25
1. Características Principales .....	25
2. Modelo de Datos y Relaciones.....	26
3. Seguridad y Control de Acceso .....	26
4. Flujos de Trabajo .....	27
models/Documento.php .....	27
Reglas de validación (rules).....	27
controllers/MaterialesController.php .....	28
FRONTEND .....	31
CSS .....	31
Construcción de la Portada (site/index.php).....	31
VISTA MATERIALES .....	31
Documento.php (Modelo Principal).....	31
Materia.php, Personal.php, Institucion.php (Modelos Auxiliares).....	31
Capa Lógica (Controlador) .....	32
Método actionIndex().....	32
Desarrollo del Frontend (Vista y Diseño).....	32
Barra Lateral de Filtros (<aside>) .....	32
Tarjetas de Resultados (Grid) .....	32
VISTA COLECCIONES.....	33
1. Nomenclatura de Componentes .....	33
2. Obtención y Muestreo de Datos.....	33
views/materiales/view.php .....	33
views/materiales/index.php .....	34
views/materiales/subir.php .....	36
views/layouts/main.php .....	37
Iniciar sesión .....	37
Q&A .....	38

ARQUITECTURA DEL PROYECTO .....	46
BIBLIOGRAFÍA .....	51

# 1. CONTROLADORES

## **Archivo: /codigo/controllers/UsuarioController.php**

Este archivo implementa la clase `UsuarioController`, la cual extiende de `yii\web\Controller`. Su función principal es la gestión del ciclo de vida de los datos (CRUD) correspondientes al modelo `Usuario`, así como la administración de permisos específicos sobre dichos registros.

### **1. Control de Acceso y Comportamientos (behaviors)**

El método `behaviors` define los filtros de acción, destacando el componente `AccessControl`. Se establece una jerarquía de autorización basada en roles:

- ❖ **Regla de Acceso Básico:** Permite a cualquier usuario autenticado (@) acceder a las acciones `perfil` y `update`. Esto faculta a los usuarios para gestionar su propia información.
- ❖ **Regla de Administrador:** Restringe las acciones críticas (`index`, `view`, `create`, `delete`) exclusivamente a usuarios cuyo atributo `rol` sea estrictamente igual a `'admin'`.
- ❖ **VerbFilter:** Asegura que la acción `delete` solo pueda ser ejecutada mediante peticiones HTTP con el método `POST`, previniendo eliminaciones accidentales o malintencionadas vía `GET`.

### **2. Gestión de Usuarios (CRUD)**

- ❖ **Listado (actionIndex):** Utiliza el modelo `UsuarioSearch` para filtrar y proveer datos a la vista mediante un `ActiveDataProvider`, facilitando la paginación y ordenamiento en la interfaz.
- ❖ **Creación (actionCreate):** Instancia un nuevo objeto `Usuario`. Es relevante destacar la lógica condicional que detecta si existe una contraseña en texto plano (`password_plain`) para invocar el método `setPassword`, garantizando que la credencial se almacene encriptada antes de persistir el modelo.
- ❖ **Visualización (actionView):** Renderiza el detalle de un usuario específico tras localizarlo mediante `findModel`.

### **3. Lógica de Actualización y Perfil**

- ❖ **Acción actionPerfil:** Diseñada para recuperar y mostrar la entidad del usuario actualmente autenticado (`Yii::$app->user->id`), ofreciendo una vista de "Mi Cuenta".
- ❖ **Acción actionUpdate:** Contiene una lógica de negocio compleja y robusta:
  - **Escenarios:** Se activa el escenario `update`, lo cual fuerza validaciones específicas (como la confirmación de la contraseña actual).
  - **Validación de Permisos:** Se verifica explícitamente si el solicitante es administrador o el propietario del perfil. De lo contrario, se lanza una excepción `ForbiddenHttpException`.
  - **Protección de Roles:** Se impide que un usuario no administrador modifique su propio rol, restaurando el valor original (`getOldAttribute`) antes del guardado.
  - **Persistencia de Sesión:** Tras una actualización exitosa por parte del propio usuario, se invoca `Yii::$app->user->login` para refrescar la sesión. Esto previene que el cambio de hash de seguridad cierre la sesión del usuario involuntariamente.

## **Archivo: /codigo/controllers/SiteController.php**

La clase `SiteController` gestiona las rutas estáticas generales de la aplicación, el proceso de autenticación y características transversales como el registro de actividades (logs).

## 1. Autenticación y Seguridad

- ❖ **Login (actionLogin):** Gestiona el inicio de sesión mediante el modelo [LoginForm](#). Se ha integrado una funcionalidad de auditoría que invoca `\app\models\UserLog::add('Login Success')` tras una autenticación exitosa, permitiendo la trazabilidad de accesos.
- ❖ **Logout (actionLogout):** Cierra la sesión del usuario y redirige al inicio. Al igual que en el login, se registra el evento en el log de auditoría antes de destruir la sesión.
- ❖ **Registro (actionRegister):** Implementa la creación de nuevos usuarios externos mediante el modelo [RegistroUsuarios](#). Si el proceso `signup()` es exitoso, se notifica al usuario mediante un mensaje "flash" y se redirige al flujo principal.

## 2. Sistema de Auditoría (actionLogs)

Esta acción proporciona una interfaz para visualizar el historial de eventos del sistema:

- ❖ **Control de Acceso:** Verifica si el usuario es "invitado" (`isGuest`) para redirigirlo al login, protegiendo la información sensible de los registros.
- ❖ **Provisión de Datos:** Instancia un `ActiveDataProvider` basado en el modelo [UserLog](#), ordenando los resultados de forma descendente por fecha de creación (`created_at`), lo cual es estándar para visualizar logs cronológicos.

## 3. Módulo de Preguntas y Respuestas (actionQa)

Esta función orquesta la visualización de una sección de soporte o foro:

- ❖ **Carga Ansiosa (Eager Loading):** Utiliza `with(['usuario', 'respuestas.usuario'])` en la consulta al modelo [Pregunta](#). Esta técnica optimiza el rendimiento de la base de datos al reducir el problema de consultas N+1, cargando las relaciones necesarias en una sola operación.
- ❖ **Estadísticas:** Calcula métricas en tiempo real (total de preguntas, pendientes, respondidas y resueltas) para ofrecer un resumen de estado en la vista.

## 4. Acciones Utilitarias

El controlador define acciones independientes en el método `actions()`:

- ❖ **Captcha:** Configurado para pruebas automatizadas (fijado a `'testme'` en entorno de test) y producción, protegiendo formularios contra bots.
- ❖ **Error:** Centraliza el manejo de excepciones HTTP, delegando la renderización a la clase [yii\web\ErrorAction](#).

# 2. MODELOS

## Archivo: [/codigo/models/RegistroUsuarios.php](#)

Este archivo define la clase [RegistroUsuarios](#), la cual extiende de [yii\base\Model](#). Su propósito es actuar como una capa de abstracción específica para el proceso de registro, desacoplando la validación de entrada de datos de la persistencia final en la base de datos.

## 1. Validación de Datos (rules)

El modelo implementa un conjunto estricto de reglas de validación para garantizar la integridad de los datos antes de su procesamiento:

- ❖ **Unicidad:** Se utilizan validadores `unique` apuntando a la clase `\app\models\Usuario` para asegurar que tanto el `username` como el `email` no existan previamente en el sistema.
- ❖ **Sanitización:** Se aplica el filtro `trim` para eliminar espacios en blanco accidentales.
- ❖ **Formato:** Se valida estrictamente el formato de correo electrónico y la longitud mínima de la contraseña.

## 2. Lógica de Registro (signup)

El método `signup` encapsula la lógica transaccional de creación de usuario:

- ❖ **Validación Previa:** Se invoca `validate()` para asegurar que los datos cumplen todas las reglas definidas.
- ❖ **Mapeo de Datos:** Se transfieren los datos del formulario al modelo `Usuario` (`ActiveRecord`).
- ❖ **Persistencia:** Se guarda el nuevo registro. Es importante notar que, según los comentarios en el código, actualmente la contraseña se almacena en texto plano, existiendo una nota técnica para implementar `generatePasswordHash` en futuras iteraciones para cumplir con estándares de seguridad criptográfica.

## Archivo: /codigo/models/UsuarioSearch.php

La clase `UsuarioSearch` extiende directamente del modelo `Usuario`. Su función es implementar el patrón de búsqueda y filtrado utilizado comúnmente en las vistas de administración (Grid Views).

### 1. Escenarios y Reglas

- ❖ **Bypass de Escenarios:** Sobrescribe el método `scenarios()` para utilizar la implementación base de `Model`, permitiendo una asignación masiva de atributos segura durante la búsqueda sin las restricciones de validación del modelo de edición.
- ❖ **Reglas de Búsqueda:** Define reglas `safe` para los campos de texto, permitiendo que cualquier entrada del usuario sea procesada como criterio de filtro.

### 2. Construcción de Consultas (search)

El método `search` retorna un objeto `ActiveDataProvider`, fundamental para la paginación y ordenamiento de datos en Yii2:

- ❖ **Filtrado Dinámico:** Utiliza métodos como `andFilterWhere` con el operador `like`. Esto permite realizar búsquedas parciales (ej. buscar "Juan" encuentra "Juan Pérez") en campos como `username`, `nombre` y `email`.
- ❖ **Validación de Filtros:** Si los parámetros de búsqueda no son válidos, el método está diseñado para retornar el proveedor de datos sin filtros o vacío, protegiendo la consulta SQL subyacente.

## Archivo: /codigo/models/UserLog.php

Este modelo representa la entidad de auditoría del sistema, extendiendo de `ActiveRecord` y mapeando a la tabla `user_log`. Su objetivo es proporcionar trazabilidad sobre las acciones de los usuarios.



## 1. Automatización del Registro (add)

El método estático `add($action)` actúa como un "helper" o servicio de infraestructura:

- ❖ **Detección de Contexto:** Determina automáticamente si el usuario está autenticado o es un invitado (`guest`). Si está logueado, captura su ID y un identificador legible (email); si no, registra la acción bajo el identificador 'Guest'.
- ❖ **Captura de Metadatos:** Almacena información crítica para la seguridad forense, como la dirección IP (`userIP`) y el agente de usuario (`userAgent`) del navegador.
- ❖ **Timestamp:** Genera la marca de tiempo exacta de la acción.

## Archivo: /codigo/models/LoginForm.php

`LoginForm` es un modelo de formulario (Model) que no representa una tabla en la base de datos, sino el proceso de autenticación.

### 1. Validación en Línea (validatePassword)

Implementa un validador personalizado (inline validator) que se ejecuta solo si las validaciones de formato básicas pasan. Este método:

- ❖ Recupera la instancia del usuario mediante `getUser()`.
- ❖ Delega la verificación de credenciales al método `validatePassword` del modelo `Usuario`.
- ❖ Añade errores al modelo si la autenticación falla, proporcionando retroalimentación al usuario.

## 2. Gestión de Sesión (login)

El método `login` es el encargado de interactuar con el componente de aplicación `Yii::$app->user`. Gestiona la duración de la sesión basándose en la propiedad `rememberMe`, estableciendo una cookie de larga duración (30 días) si el usuario así lo solicita.

## Archivo: /codigo/models/Usuario.php

Este es el modelo central del sistema. Extiende de `ActiveRecord` e implementa `IdentityInterface`, lo cual es obligatorio para que Yii2 reconozca esta clase como la identidad del usuario logueado.

### 1. Implementación de IdentityInterface

Define los métodos requeridos por el núcleo de seguridad del framework (`findIdentity`, `getId`, `validateAuthKey`, etc.), permitiendo que el sistema gestione la sesión del usuario de manera transparente.

## 2. Reglas de Negocio y Escenarios

- ❖ **Escenario 'update':** Define una regla condicional donde `current_password` es obligatorio solo durante la edición del perfil.
- ❖ **Validación Personalizada:** El método `validateCurrentPassword` asegura que, para realizar cambios sensibles, el usuario deba probar su identidad introduciendo su contraseña actual.

### 3. Control de Roles (ACL)

El modelo encapsula la lógica de autorización mediante constantes (`ROL_ADMIN`, `ROL_GESTOR`, etc.) y métodos de ayuda como `puedeAccederBackend()`. Esto centraliza las políticas de acceso en el modelo, evitando dispersar lógica condicional compleja en las vistas o controladores.

### 4. Seguridad de Contraseñas

Actualmente, los métodos `validatePassword` y `setPassword` operan con comparación directa (texto plano). El código incluye bloques comentados preparados para la transición a `Yii::$app->security` (hashing con Bcrypt/Argon2), lo cual indica una arquitectura preparada para escalar en seguridad, aunque actualmente se encuentra en modo de desarrollo o prototipado.

## 3. VISTAS

### 3.1. /codigo/views/usuario/

#### **Archivo: /codigo/views/index.php**

Esta vista representa la interfaz principal de administración de usuarios. Su objetivo es listar los registros existentes y ofrecer puntos de entrada para las operaciones CRUD.

- ❖ **Componente GridView:** Se utiliza el widget `yii\grid\GridView` para generar una tabla dinámica de datos.
- ❖ **Columna Personalizada (Roles):** Destaca la implementación de una columna con lógica anónima (`closure`) para el atributo rol. En lugar de mostrar el valor crudo de la base de datos, se define un filtro (`filter`) mediante un arreglo asociativo y se formatea el valor de salida (`ucfirst`), mejorando la legibilidad semántica de la interfaz.
- ❖ **Navegación:** Integra la barra de herramientas estándar (`ActionColumn`) que genera automáticamente los enlaces a las vistas `view`, `update` y `delete`.

#### **Archivo: /codigo/views/usuario/\_search.php**

Este archivo contiene el formulario de filtrado avanzado que opera en conjunto con la vista index.

- ❖ **Formulario GET:** A diferencia de los formularios de creación/edición, este utiliza el método `HTTP GET`. Esto permite que los parámetros de búsqueda se reflejen en la URL, facilitando el compartimiento de resultados de búsqueda específicos.
- ❖ **Controles de Interfaz:** Provee campos de texto para coincidencias parciales y una lista desplegable (`dropDownList`) para el filtrado exacto por rol, garantizando la consistencia con los datos definidos en el modelo.

#### **Archivo: /codigo/views/usuario/create.php y update.php**

Estas dos vistas actúan como contenedores contextuales (wrappers). Aunque comparten la lógica de formulario, sus responsabilidades difieren en la preparación del entorno visual:

- ❖ **Abstracción de Lógica:** Ambas vistas delegan la renderización del formulario propiamente dicho al archivo parcial `_form.php`. Esto cumple con el principio DRY (Don't Repeat Yourself).
- ❖ **Gestión de Contexto:** Se encargan de establecer el título de la página (`$this->title`) y la estructura de navegación (`breadcrumbs`), proporcionando al usuario una ubicación clara dentro de la jerarquía de la aplicación.

## **Archivo: /codigo/views/usuario/\_form.php**

Este es el archivo más complejo de la capa de vista, ya que contiene la lógica de entrada de datos y reglas de visualización condicional.

- ❖ **Seguridad Visual (RBAC):** Implementa una verificación explícita (`$isAdmin`) para renderizar el selector de "Roles". Esto previene que un usuario estándar pueda elevar sus propios privilegios manipulando el formulario, ya que el campo ni siquiera se genera en el HTML si no se poseen los permisos adecuados.
- ❖ **Manejo de Contraseñas:**
  - **Distinción de Escenarios:** Diferencia visualmente entre la creación (donde la contraseña es obligatoria) y la actualización (donde es opcional).
  - **Confirmación de Seguridad:** En el modo de edición (`!$isNew`), despliega una sección de alerta visual (estilo warning) exigiendo la contraseña actual para confirmar cambios sensibles, reforzando la seguridad operativa.

## **Archivo: /codigo/views/usuario/view.php**

Esta vista presenta el detalle completo de una entidad Usuario específica.

- ❖ **Componente DetailView:** Utiliza este widget para la presentación estructurada de atributos individuales.
- ❖ **Protección de Información:** Se observa la exclusión deliberada de los campos relacionados con la contraseña (`password`, `auth_key`, etc.) en la lista de atributos mostrados, cumpliendo con las buenas prácticas de seguridad de la información.

## **Archivo: /codigo/views/usuario/perfil.php**

Esta vista funciona como un "Dashboard" o cuadro de mando personal para el usuario autenticado. Su arquitectura es modular, agregando información de diversos modelos (`Documento`, `Coleccion`, `Pregunta`, `UserLog`).

- ❖ **Instanciación de Proveedores de Datos:** El archivo configura múltiples instancias de `ActiveDataProvider` directamente. Esto permite alimentar diferentes secciones de la interfaz (Materiales, Colecciones, Q&A) de manera independiente y paginada.
- ❖ **Registro de Auditoría:** Incluye una sección manual (tabla HTML) para visualizar los últimos accesos del usuario (`UserLog`), proporcionando transparencia sobre la seguridad de la cuenta.

## **Archivos: /codigo/views/usuario/Archivos Parciales (\_tab\_\*.php)**

Los archivos `_tab_materiales.php`, `_tab_colecciones.php` y `_tab_qa.php` representan una estrategia de fragmentación de vistas.

- ❖ **\_tab\_materiales.php:** Renderiza un `GridView` simplificado para los documentos subidos por el usuario, enfocándose en el título y el tipo de acceso.
- ❖ **\_tab\_colecciones.php:** Implementa lógica específica de negocio en la interfaz. Destaca el botón personalizado en la `ActionColumn` que permite "Dejar de seguir" una colección, ejecutando una petición `POST` hacia el controlador correspondiente.
- ❖ **\_tab\_qa.php:** Gestiona la visualización de la participación del usuario en foros. Utiliza lógica condicional dentro de las columnas para verificar la integridad referencial (ej. verificar si la pregunta padre aún existe antes de renderizar el enlace), evitando errores en tiempo de ejecución si el contenido relacionado ha sido eliminado.

## **3.2. /codigo/views/site/**

### **Archivo: /codigo/views/site/login.php**

Esta vista es la encargada de renderizar la interfaz de autenticación. Se caracteriza por un diseño moderno y centrado, utilizando las clases de utilidad del framework CSS **Bootstrap 5**, lo que garantiza una visualización responsiva y limpia.

#### **1. Arquitectura de la Interfaz (UI)**

- ❖ **Contenedor Centrado:** El formulario está encapsulado dentro de una estructura de rejilla (**row justify-content-center**), limitando el ancho visual en pantallas grandes (**col-md-5**) y utilizando un componente de tipo "tarjeta" (**card shadow**) para elevar visualmente el contenido sobre el fondo.
- ❖ **Gestión de Errores Visuales:** Antes de iniciar el formulario, se implementa una comprobación condicional (**if (\$model->hasErrors())**) que renderiza un sumario de errores (**Html::errorSummary**). Esto proporciona retroalimentación inmediata y agrupada al usuario en caso de credenciales inválidas, mejorando la usabilidad.

#### **2. Implementación del Formulario (ActiveForm)**

- ❖ **Configuración de Campos:**
  - **Foco Automático:** El campo **username** incluye la propiedad autofocus, agilizando la interacción del usuario al cargar la página.
  - **Placeholders:** se utilizan marcadores de posición en los inputs para limpiar la interfaz, eliminando la necesidad de etiquetas (**label(false)**) redundantes visualmente, logrando un diseño minimalista.
- ❖ **Persistencia de Sesión:** Incluye el control checkbox para el atributo **rememberMe**, vinculado directamente a la lógica del modelo **LoginForm** para gestionar la cookie de autenticación de larga duración.

### **Archivo: /codigo/views/site/register.php**

El archivo **register.php** implementa la interfaz de registro de nuevos usuarios. A diferencia del login, utiliza un diseño más estándar alineado a la izquierda, típico de formularios de toma de datos extensos.

#### **1. Vinculación de Datos (Data Binding)**

La vista instancia un **ActiveForm** que se vincula al modelo **RegistroUsuarios** (analizado previamente). Cada campo generado (**textInput**, **passwordInput**) corresponde a una propiedad del modelo, asegurando que las reglas de validación definidas en el backend (como **required** o **email**) se reflejen en la validación del lado del cliente si está habilitada.

#### **2. Selección de Roles y Jerarquía**

- ❖ **Control de Entrada Restringido:** Para el atributo rol, se utiliza un **dropDownList** (lista desplegable).
- ❖ **Definición Estática:** Es relevante notar que los roles (estudiante, gestor, empresa) están definidos estáticamente en la vista ("hardcoded").

- **Nota académica:** En un entorno de producción escalable, sería ideal poblar esta lista desde una tabla de base de datos o una constante en el modelo (`Usuario::getRoles()`) para evitar la dispersión de "números mágicos" o cadenas de texto en las vistas.

### 3. Navegación y Contexto

Utiliza el widget de breadcrumbs (migas de pan) mediante `$this->params['breadcrumbs']`, lo cual es fundamental para la orientación del usuario dentro de la arquitectura de la información del sitio, permitiendo un retorno fácil a la página de inicio.

## BASE DE DATOS

### Creación

La base de datos que se ha creado para este trabajo se ha llamado “`divulgacionacademica_db`”

#### Tabla Usuario

En la tabla `Usuario` están todos perfiles conjuntos en una misma tabla, siendo la clave primaria el `id`, y el usuario será por defecto alumno, pudiendo ser cambiado posteriormente. El atributo `rol` actúa como discriminador, determinando si el usuario es `'admin'`, `'gestor'`, `'cliente'` o `'alumno'`, y controlando así los permisos de acceso en la aplicación.

```
CREATE TABLE Usuario (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    nombre VARCHAR(100) NOT NULL,  
    email VARCHAR(100) NOT NULL UNIQUE,  
    password VARCHAR(255) NOT NULL,  
    rol ENUM('normal', 'colaborador', 'admin') DEFAULT 'normal',  
    estado ENUM('activo', 'inactivo', 'bloqueado') DEFAULT 'activo',  
    fechaAlta DATETIME DEFAULT CURRENT_TIMESTAMP  
);
```



MySQL ha devuelto un conjunto de valores vacíos (se omiten todas las columnas). La consulta tardó 0.0013 segundos.  
CREATE TABLE Usuario ( id INT AUTO\_INCREMENT PRIMARY KEY, nombre VARCHAR(100) NOT NULL, email VARCHAR(100) NOT NULL UNIQUE, password VARCHAR(255) NOT NULL, rol ENUM('normal', 'colaborador', 'admin') DEFAULT 'normal', estado ENUM('activo', 'inactivo', 'bloqueado') DEFAULT 'activo', fechaAlta DATETIME DEFAULT CURRENT\_TIMESTAMP )  
[Editar en línea] [Tabla] [Crear código PHP]

#### Tabla Institución

En lugar de crear una tabla para universidades y otra para academias o similares, se ha optado por hacer una única tabla en la que se generaliza empleando el campo `tipo`.

```
1 CREATE TABLE Institucion (  
2     id INT AUTO_INCREMENT PRIMARY KEY,  
3     nombre VARCHAR(150) NOT NULL,  
4     tipo ENUM('universidad', 'academia') NOT NULL,  
5     ubicacion VARCHAR(255),  
6     contacto VARCHAR(100)  
7 );
```



MySQL ha devuelto un conjunto de valores vacíos (se omiten todas las columnas). La consulta tardó 0.0013 segundos.  
CREATE TABLE Institucion ( id INT AUTO\_INCREMENT PRIMARY KEY, nombre VARCHAR(150) NOT NULL, tipo ENUM('universidad', 'academia') NOT NULL, ubicacion VARCHAR(255), contacto VARCHAR(100) )  
[Editar en línea] [Tabla] [Crear código PHP]

## Tabla Materia

El atributo **parentId** es para indicar que una asignatura está dentro de otra.

```
1 CREATE TABLE Materia (  
2     id INT AUTO_INCREMENT PRIMARY KEY,  
3     nombre VARCHAR(100) NOT NULL,  
4     parentId INT DEFAULT NULL,  
5     FOREIGN KEY (parentId) REFERENCES Materia(id) ON DELETE SET NULL  
6 );
```

MySQL ha devuelto un conjunto de valores vacío (es decir: cero columnas). (La consulta tardó 0.0009 segundos.)

```
CREATE TABLE `Materia` ( `id` INT AUTO_INCREMENT PRIMARY KEY, `nombre` VARCHAR(100) NOT NULL, `parentId` INT DEFAULT NULL, FOREIGN KEY (`parentId`) REFERENCES `Materia`(`id`) ON DELETE SET NULL );  
[Editar en línea] [Editar] [Crear código PHP]
```

## Tabla Personal

```
1 CREATE TABLE Personal (  
2     id INT AUTO_INCREMENT PRIMARY KEY,  
3     nombre VARCHAR(100) NOT NULL,  
4     apellidos VARCHAR(100),  
5     institucionId INT NOT NULL,  
6     FOREIGN KEY (institucionId) REFERENCES Institucion(id) ON DELETE CASCADE  
7 );
```

MySQL ha devuelto un conjunto de valores vacío (es decir: cero columnas). (La consulta tardó 0.0003 segundos.)

```
CREATE TABLE `Personal` ( `id` INT AUTO_INCREMENT PRIMARY KEY, `nombre` VARCHAR(100) NOT NULL, `apellidos` VARCHAR(100), `institucionId` INT NOT NULL, FOREIGN KEY (`institucionId`) REFERENCES `Institucion`(`id`) ON DELETE CASCADE );  
[Editar en línea] [Editar] [Crear código PHP]
```

## Tabla Documento

Hay varias claves foráneas (**materiaId**, **autorId**, **institucionId**) que permiten realizar filtrados y búsquedas cruzadas, cumpliendo con el requisito de clasificar recursos por asignatura, autor y centro de origen.

```
1 CREATE TABLE Documento (  
2     id INT AUTO_INCREMENT PRIMARY KEY,  
3     titulo VARCHAR(200) NOT NULL,  
4     archivo_url VARCHAR(255) NOT NULL,  
5     tipo_acceso ENUM('publico', 'privado') DEFAULT 'publico',  
6     materiaId INT,  
7     autorId INT,  
8     institucionId INT,  
9     FOREIGN KEY (materiaId) REFERENCES Materia(id) ON DELETE SET NULL,  
10    FOREIGN KEY (autorId) REFERENCES Personal(id) ON DELETE SET NULL,  
11    FOREIGN KEY (institucionId) REFERENCES Institucion(id) ON DELETE CASCADE  
12 );
```

MySQL ha devuelto un conjunto de valores vacío (es decir: cero columnas). (La consulta tardó 0.0004 segundos.)

```
CREATE TABLE `Documento` ( `id` INT AUTO_INCREMENT PRIMARY KEY, `titulo` VARCHAR(200) NOT NULL, `archivo_url` VARCHAR(255) NOT NULL, `tipo_acceso` ENUM('publico', 'privado') DEFAULT 'publico', `materiaId` INT, `autorId` INT, `institucionId` INT, FOREIGN KEY (`materiaId`) REFERENCES `Materia`(`id`) ON DELETE SET NULL, FOREIGN KEY (`autorId`) REFERENCES `Personal`(`id`) ON DELETE SET NULL, FOREIGN KEY (`institucionId`) REFERENCES `Institucion`(`id`) ON DELETE CASCADE );  
[Editar en línea] [Editar] [Crear código PHP]
```

## Tabla documento\_comentario

```
1 CREATE TABLE documento_comentario(  
2     id int(11) NOT NULL,  
3     contenido text NOT NULL,  
4     fecha datetime DEFAULT current_timestamp(),  
5     usuarioId int(11) NOT NULL,  
6     documentoId int(11) NOT NULL  
7 )
```

MySQL ha devuelto un conjunto de valores vacío (es decir: cero columnas). (La consulta tardó 0.0004 segundos.)

```
CREATE TABLE `documento_comentario` ( `id` INT(11) NOT NULL, `contenido` text NOT NULL, `fecha` datetime DEFAULT current_timestamp(), `usuarioId` INT(11) NOT NULL, `documentoId` INT(11) NOT NULL );  
[Editar en línea] [Editar] [Crear código PHP]
```

## Tabla documento\_valoracion

```
1 CREATE TABLE documento_valoracion (  
2     usuarioId int(11) NOT NULL,  
3     documentoId int(11) NOT NULL,  
4     puntuacion int(1) NOT NULL CHECK (`puntuacion` between 1 and 5),  
5     fecha datetime DEFAULT current_timestamp()  
6 )
```

✓ MySQL ha devuelto un conjunto de valores vacío (es decir, cero columnas). (La consulta tardó 0.0003 segundos.)

```
CREATE TABLE documento_valoracion : usuarioId int(11) NOT NULL, documentoId int(11) NOT NULL, puntuacion int(1) NOT NULL CHECK (`puntuacion` between 1 and 5), fecha datetime DEFAULT current_timestamp() : :  
[Editar en línea] [Editar] [Crear código PHP]
```

## Tabla pregunta

```
1 CREATE TABLE pregunta (  
2     id int(11) NOT NULL,  
3     titulo varchar(255) NOT NULL,  
4     descripcion text NOT NULL,  
5     fecha_creacion datetime DEFAULT current_timestamp(),  
6     vistas int(11) DEFAULT 0,  
7     votos_utiles int(11) DEFAULT 0,  
8     estado enum('sin_responder', 'respondida', 'resuelta') DEFAULT 'sin_responder',  
9     usuarioId int(11) NOT NULL,  
10    materiaId int(11) DEFAULT NULL  
11 )
```

✓ MySQL ha devuelto un conjunto de valores vacío (es decir, cero columnas). (La consulta tardó 0.0003 segundos.)

```
CREATE TABLE pregunta : id int(11) NOT NULL, titulo varchar(255) NOT NULL, descripcion text NOT NULL, fecha_creacion datetime DEFAULT current_timestamp(), vistas int(11) DEFAULT 0, votos_utiles int(11) DEFAULT 0, estado enum('sin_responder', 'respondida', 'resuelta') DEFAULT 'sin_responder', usuarioId int(11) NOT NULL, materiaId int(11) DEFAULT NULL : :  
[Editar en línea] [Editar] [Crear código PHP]
```

## Tabla pregunta\_guardada

```
1 CREATE TABLE pregunta_guardada (  
2     usuarioId int(11) NOT NULL,  
3     preguntaId int(11) NOT NULL,  
4     fecha_guardado datetime DEFAULT current_timestamp()  
5 )
```

✓ MySQL ha devuelto un conjunto de valores vacío (es decir, cero columnas). (La consulta tardó 0.0003 segundos.)

```
CREATE TABLE pregunta_guardada : usuarioId int(11) NOT NULL, preguntaId int(11) NOT NULL, fecha_guardado datetime DEFAULT current_timestamp() : :  
[Editar en línea] [Editar] [Crear código PHP]
```

## Tabla pregunta\_voto

```
1 CREATE TABLE pregunta_voto (  
2     usuarioId int(11) NOT NULL,  
3     preguntaId int(11) NOT NULL,  
4     fecha_voto datetime DEFAULT current_timestamp()  
5 )
```

✓ MySQL ha devuelto un conjunto de valores vacío (es decir, cero columnas). (La consulta tardó 0.0004 segundos.)

```
CREATE TABLE pregunta_voto : usuarioId int(11) NOT NULL, preguntaId int(11) NOT NULL, fecha_voto datetime DEFAULT current_timestamp() : :  
[Editar en línea] [Editar] [Crear código PHP]
```



## Tabla respuesta

```
1 CREATE TABLE respuesta (  
2   id int(11) NOT NULL,  
3   contenid text NOT NULL,  
4   fecha datetime DEFAULT current_timestamp(),  
5   es_solucion tinyint(1) DEFAULT 0,  
6   preguntaId int(11) NOT NULL,  
7   usuarioId int(11) NOT NULL  
8 )
```

```
MySQL ha devuelto un conjunto de valores vacíos (es decir, cero columnas). (La consulta tardó 0.0012 segundos.)  
  
CREATE TABLE respuesta ( id int(11) NOT NULL, contenid text NOT NULL, fecha datetime DEFAULT current_timestamp(), es_solucion tinyint(1) DEFAULT 0, preguntaId int(11) NOT NULL, usuarioId int(11) NOT NULL )  
Elaboración de la tabla [[ Tabla ]] (Crear columnas PHP)
```

## Tabla user\_log

```
1 CREATE TABLE user_log (  
2   id int(11) NOT NULL,  
3   user_id int(11) DEFAULT NULL,  
4   user_identifier varchar(255) DEFAULT NULL,  
5   action varchar(255) NOT NULL,  
6   ip_address varchar(50) DEFAULT NULL,  
7   user_agent varchar(255) DEFAULT NULL,  
8   created_at datetime DEFAULT current_timestamp()  
9 )
```

```
MySQL ha devuelto un conjunto de valores vacíos (es decir, cero columnas). (La consulta tardó 0.0012 segundos.)  
  
CREATE TABLE user_log ( id int(11) NOT NULL, user_id int(11) DEFAULT NULL, user_identifier varchar(255) DEFAULT NULL, action varchar(255) NOT NULL, ip_address varchar(50) DEFAULT NULL, user_agent varchar(255) DEFAULT NULL, created_at datetime DEFAULT current_timestamp() )  
Elaboración de la tabla [[ Tabla ]] (Crear columnas PHP)
```

## Tabla coleccion

```
1 CREATE TABLE `coleccion` (  
2   `id` int(11) NOT NULL AUTO_INCREMENT,  
3   `titulo` varchar(255) NOT NULL,  
4   `descripcion` text DEFAULT NULL,  
5   `usuarioId` int(11) NOT NULL,  
6   `descargas` int(11) DEFAULT 0,  
7   `tipo_acceso` enum('publico','privado') NOT NULL DEFAULT 'publico',  
8   `fecha_actualizacion` datetime DEFAULT CURRENT_TIMESTAMP,  
9   PRIMARY KEY (`id`),  
10  CONSTRAINT `fk_coleccion_usuario` FOREIGN KEY (`usuarioId`) REFERENCES `usuario` (`id`) ON DELETE CASCADE  
11 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

```
MySQL ha devuelto un conjunto de valores vacíos (es decir, cero columnas). (La consulta tardó 0.0012 segundos.)  
  
CREATE TABLE coleccion ( `id` int(11) NOT NULL AUTO_INCREMENT, `titulo` varchar(255) NOT NULL, `descripcion` text DEFAULT NULL, `usuarioId` int(11) NOT NULL, `descargas` int(11) DEFAULT 0, `tipo_acceso` enum('publico','privado') NOT NULL DEFAULT 'publico', `fecha_actualizacion` datetime DEFAULT CURRENT_TIMESTAMP, PRIMARY KEY (`id`), CONSTRAINT `fk_coleccion_usuario` FOREIGN KEY (`usuarioId`) REFERENCES `usuario` (`id`) ON DELETE CASCADE ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4  
Elaboración de la tabla [[ Tabla ]] (Crear columnas PHP)
```

## Tabla coleccion\_documento

```
1 CREATE TABLE coleccion_documento (  
2   coleccionId int(11) NOT NULL,  
3   documentoId int(11) NOT NULL,  
4   PRIMARY KEY (coleccionId, documentoId),  
5   CONSTRAINT fk_cd_coleccion FOREIGN KEY (coleccionId) REFERENCES coleccion (id) ON DELETE CASCADE,  
6   CONSTRAINT fk_cd_documento FOREIGN KEY (documentoId) REFERENCES documento (id) ON DELETE CASCADE  
7 )
```

```
MySQL ha devuelto un conjunto de valores vacíos (es decir, cero columnas). (La consulta tardó 0.0004 segundos.)  
  
CREATE TABLE coleccion_documento ( coleccionId int(11) NOT NULL, documentoId int(11) NOT NULL, PRIMARY KEY (coleccionId, documentoId), CONSTRAINT fk_cd_coleccion FOREIGN KEY (coleccionId) REFERENCES coleccion (id) ON DELETE CASCADE, CONSTRAINT fk_cd_documento FOREIGN KEY (documentoId) REFERENCES documento (id) ON DELETE CASCADE )  
Elaboración de la tabla [[ Tabla ]] (Crear columnas PHP)
```



## Tabla usuario\_coleccion

```
1 CREATE TABLE usuario_coleccion (  
2     usuarioId int(11) NOT NULL,  
3     coleccionId int(11) NOT NULL,  
4     fecha_union timestamp DEFAULT current_timestamp(),  
5     PRIMARY KEY (usuarioId, coleccionId),  
6     CONSTRAINT fk_uc_usuario FOREIGN KEY (usuarioId) REFERENCES usuario (id) ON DELETE CASCADE,  
7     CONSTRAINT fk_uc_coleccion FOREIGN KEY (coleccionId) REFERENCES coleccion (id) ON DELETE CASCADE  
8 )
```



MySQL ha devuelto un conjunto de valores vacío (es decir: cero columnas). La consulta tardó 0.0003 segundos.  
  
CREATE TABLE usuario\_coleccion (usuarioId int(11) NOT NULL, coleccionId int(11) NOT NULL, fecha\_union timestamp DEFAULT current\_timestamp(), PRIMARY KEY (usuarioId, coleccionId), CONSTRAINT fk\_uc\_usuario FOREIGN KEY (usuarioId) REFERENCES usuario (id) ON DELETE CASCADE, CONSTRAINT fk\_uc\_coleccion FOREIGN KEY (coleccionId) REFERENCES coleccion (id) ON DELETE CASCADE)  
[Escriba en línea] [Escriba] [Cierre código PHP]

## Tabla Servicio

Esta tabla permite a las Universidades y Academias publicar los cursos, seminarios o auditorías que tienen a la venta. Es imprescindible para cumplir el requisito de "Gestión de Productos/Servicios". Mientras la tabla **Documento** guarda archivos académicos (apuntes, tesis), la tabla **Servicio** define la oferta comercial.

```
1 CREATE TABLE Servicio (  
2     id INT AUTO_INCREMENT PRIMARY KEY,  
3     nombre VARCHAR(150) NOT NULL,  
4     precio DECIMAL(10, 2),  
5     institucionId INT NOT NULL,  
6     FOREIGN KEY (institucionId) REFERENCES Institucion(id) ON DELETE CASCADE  
7 );
```



MySQL ha devuelto un conjunto de valores vacío (es decir: cero columnas). La consulta tardó 0.0003 segundos.  
  
CREATE TABLE Servicio (id INT AUTO\_INCREMENT PRIMARY KEY, nombre VARCHAR(150) NOT NULL, precio DECIMAL(10, 2), institucionId INT NOT NULL, FOREIGN KEY (institucionId) REFERENCES Institucion(id) ON DELETE CASCADE)  
[Escriba en línea] [Escriba] [Cierre código PHP]

## Tabla ClienteEmpresa

Extiende la información de la tabla **Usuario** añadiendo **razon\_social** y **cif**, necesarios para la facturación, pero irrelevantes para un alumno común.

```
1 CREATE TABLE ClienteEmpresa (  
2     id INT AUTO_INCREMENT PRIMARY KEY,  
3     razon_social VARCHAR(150) NOT NULL,  
4     cif VARCHAR(20),  
5     usuarioId INT NOT NULL,  
6     FOREIGN KEY (usuarioId) REFERENCES Usuario(id) ON DELETE CASCADE  
7 );
```



MySQL ha devuelto un conjunto de valores vacío (es decir: cero columnas). La consulta tardó 0.0004 segundos.  
  
CREATE TABLE ClienteEmpresa (id INT AUTO\_INCREMENT PRIMARY KEY, razon\_social VARCHAR(150) NOT NULL, cif VARCHAR(20), usuarioId INT NOT NULL, FOREIGN KEY (usuarioId) REFERENCES Usuario(id) ON DELETE CASCADE)  
[Escriba en línea] [Escriba] [Cierre código PHP]

## Tabla Contratación

Esta tabla intermedia es esencial para mantener un historial de transacciones, registrando cuándo se contrató un servicio y en qué estado se encuentra.

```
1 CREATE TABLE Contratacion (  
2     id INT AUTO_INCREMENT PRIMARY KEY,  
3     clienteId INT NOT NULL,  
4     servicioId INT NOT NULL,  
5     fecha DATETIME DEFAULT CURRENT_TIMESTAMP,  
6     FOREIGN KEY (clienteId) REFERENCES ClienteEmpresa(id) ON DELETE CASCADE,  
7     FOREIGN KEY (servicioId) REFERENCES Servicio(id) ON DELETE CASCADE  
8 );
```

```
MySQL ha devuelto un conjunto de valores vacío (se dejó solo columnas). La consulta tardó 0.000 segundos.)

CREATE TABLE Contratacion ( id INT AUTO_INCREMENT PRIMARY KEY, (titulado INT NOT NULL, nomtitulado INT NOT NULL, fecha_inicio DATETIME NOT NULL, FOREIGN KEY (titulado) REFERENCES (tituladogenera) ON DELETE CASCADE, FOREIGN KEY (nomtitulado) REFERENCES (nomtituladogenera) ON DELETE CASCADE ) )

[Editar en línea][Ver][Crear código PHP]
```

## Tabla Evento

Permite la gestión temporal de actividades. Al incluir la clave foránea **institucionId**, se asegura que cada centro gestione su propia agenda académica de forma aislada.

```
1 CREATE TABLE Evento (
2     id INT AUTO_INCREMENT PRIMARY KEY,
3     titulo VARCHAR(150) NOT NULL,
4     fecha_inicio DATETIME NOT NULL,
5     institucionId INT,
6     FOREIGN KEY (institucionId) REFERENCES Institucion(id) ON DELETE CASCADE
7 );
```

```
MySQL ha devuelto un conjunto de valores vacío (se dejó solo columnas). La consulta tardó 0.000 segundos.)

CREATE TABLE Evento ( id INT AUTO_INCREMENT PRIMARY KEY, titulo VARCHAR(150) NOT NULL, fecha_inicio DATETIME NOT NULL, institucionId INT, FOREIGN KEY (institucionId) REFERENCES Institucion(id) ON DELETE CASCADE )

[Editar en línea][Ver][Crear código PHP]
```

## Tabla Incidencia

Establece un sistema centralizado de gestión de incidencias que permite a los usuarios reportar fallos o dudas. Esto facilita a los administradores la detección de errores y la organización de las reparaciones según su urgencia.

```
1 CREATE TABLE Incidencia (
2     id INT AUTO_INCREMENT PRIMARY KEY,
3     descripcion TEXT NOT NULL,
4     estado ENUM('abierta', 'cerrada') DEFAULT 'abierta',
5     usuarioId INT NOT NULL,
6     FOREIGN KEY (usuarioId) REFERENCES Usuario(id) ON DELETE CASCADE
7 );
```

```
MySQL ha devuelto un conjunto de valores vacío (se dejó solo columnas). La consulta tardó 0.000 segundos.)

CREATE TABLE Incidencia ( id INT AUTO_INCREMENT PRIMARY KEY, descripcion TEXT NOT NULL, estado ENUM('abierta', 'cerrada') DEFAULT 'abierta', usuarioId INT NOT NULL, FOREIGN KEY (usuarioId) REFERENCES Usuario(id) ON DELETE CASCADE )

[Editar en línea][Ver][Crear código PHP]
```

## Resultado

En total se han creado 20 tablas, por lo que la base de datos tiene el siguiente aspecto:

Tabla	Acción	Filas	Tipo	Estadamiento	Tamaño	Residuo a depurar
<input type="checkbox"/> clienteempresa		8	InnoDB	utf8mb4_general_ci	32.0 KB	-
<input type="checkbox"/> coleccion		35	InnoDB	utf8mb4_0900_ai_ci	32.0 KB	-
<input type="checkbox"/> coleccion_documento		59	InnoDB	utf8mb4_0900_ai_ci	32.0 KB	-
<input type="checkbox"/> contratacion		16	InnoDB	utf8mb4_general_ci	40.0 KB	-
<input type="checkbox"/> documento		58	InnoDB	utf8mb4_general_ci	64.0 KB	-
<input type="checkbox"/> documento_comentario		8	InnoDB	utf8mb4_general_ci	48.0 KB	-
<input type="checkbox"/> documento_valoracion		9	InnoDB	utf8mb4_general_ci	32.0 KB	-
<input type="checkbox"/> evento		17	InnoDB	utf8mb4_general_ci	32.0 KB	-
<input type="checkbox"/> incidencia		11	InnoDB	utf8mb4_general_ci	32.0 KB	-
<input type="checkbox"/> institucion		7	InnoDB	utf8mb4_general_ci	16.0 KB	-
<input type="checkbox"/> materia		61	InnoDB	utf8mb4_general_ci	32.0 KB	-
<input type="checkbox"/> personal		19	InnoDB	utf8mb4_general_ci	32.0 KB	-
<input type="checkbox"/> pregunta		5	InnoDB	utf8mb4_general_ci	48.0 KB	-
<input type="checkbox"/> pregunta_guardada		4	InnoDB	utf8mb4_general_ci	32.0 KB	-
<input type="checkbox"/> pregunta_voto		18	InnoDB	utf8mb4_general_ci	32.0 KB	-
<input type="checkbox"/> respuesta		5	InnoDB	utf8mb4_general_ci	48.0 KB	-
<input type="checkbox"/> servicio		9	InnoDB	utf8mb4_general_ci	32.0 KB	-
<input type="checkbox"/> user_log		21	InnoDB	utf8mb4_general_ci	16.0 KB	-
<input type="checkbox"/> usuario		27	InnoDB	utf8mb4_general_ci	32.0 KB	-
<input type="checkbox"/> usuario_coleccion		14	InnoDB	utf8mb4_0900_ai_ci	32.0 KB	-
20 tablas	Número de filas	393	InnoDB	utf8mb4_0900_ai_ci	784.0 KB	0 B

# Introducción de datos

## Tabla usuario

```
1 INSERT INTO usuario (id, nombre, email, password, rol, fecha_registro, username) VALUES
2 (1, 'Super Admin', 'admin@portal.com', '1234', 'admin', '2026-01-03 11:21:26', 'admin'),
3 (2, 'Gestor Politécnica', 'gestor@upm.es', '1234', 'gestor', '2026-01-03 11:21:26', 'gestor'),
4 (3, 'Gestor CodeMasters', 'director@codemasters.com', '1234', 'gestor', '2026-01-03 11:21:26', 'director'),
5 (4, 'Gestor EnglishNow', 'hello@englishnow.com', '1234', 'gestor', '2026-01-03 11:21:26', 'hello'),
6 (5, 'Tech Solutions CEO', 'contacto@techsolutions.com', '1234', 'empresa', '2026-01-03 11:21:26', 'contacto'),
7 (6, 'Consultora Global', 'rrhh@consultoraglobal.com', '1234', 'empresa', '2026-01-03 11:21:26', 'rrhh'),
8 (7, 'Startup Innova', 'adsis@innova.io', '1234', 'empresa', '2026-01-03 11:21:26', 'innova'),
9 (8, 'Carlos Pérez', 'carlos@mail.com', '1234', 'alumno', '2026-01-03 11:21:26', 'carlos'),
10 (9, 'María García', 'maria@mail.com', '1234', 'alumno', '2026-01-03 11:21:26', 'maria'),
11 (10, 'Lucía Méndez', 'lucia@mail.com', '1234', 'alumno', '2026-01-03 11:21:26', 'lucia'),
12 (11, 'David Torres', 'david@mail.com', '1234', 'alumno', '2026-01-03 11:21:26', 'david'),
13 (12, 'Elena Nito', 'elena@mail.com', '1234', 'alumno', '2026-01-03 11:21:26', 'elena'),
14 (13, 'Sofía Rivas', 'sofia@mail.com', '1234', 'alumno', '2026-01-03 11:21:26', 'sofia'),
15 (14, 'Jorge Bua', 'jorge@mail.com', '1234', 'alumno', '2026-01-03 11:21:26', 'jorge'),
16 (15, 'Ana Bot', 'ana@mail.com', '1234', 'alumno', '2026-01-03 11:21:26', 'ana'),
17 (16, 'Juan Riego Vila', 'juan@mail.com', '1234', 'alumno', '2026-01-08 10:55:50', 'juanrv'),
18 (17, 'Andrea Colorado Esteban', 'andrea@mail.com', '1234', '', '2026-01-08 13:01:31', 'andrea'),
19 (18, 'Juan Riego Vila', 'juanrv@mail.com', '1234', '', '2026-01-08 13:48:35', 'juanrv2'),
20 (19, 'Roberto Estudiante', 'robert@mail.com', '1234', 'alumno', '2026-01-09 16:01:32', 'robert_stud'),
21 (20, 'Laura Agramón', 'laura_nau@mail.com', '1234', 'alumno', '2026-01-09 16:01:32', 'lau2024'),
22 (21, 'Empresa TechSoft', 'admin@techsoft.com', '1234', 'empresa', '2026-01-09 16:01:32', 'techsoft_admin'),
23 (22, 'Constructora Civil S.L.', 'info@constructora.com', '1234', 'empresa', '2026-01-09 16:01:32', 'constructora_sl'),
24 (23, 'Academia BioAdmin', 'gestion@bioadmin.com', '1234', 'gestor', '2026-01-09 16:01:32', 'bio_gestor'),
25 (24, 'Pedro Profesor', 'pedro@mail.com', '1234', 'alumno', '2026-01-09 16:01:32', 'pedroprof'),
26 (25, 'Marta Diseño', 'marta@design.com', '1234', 'empresa', '2026-01-09 16:01:32', 'marta_design'),
27 (26, 'Hospital General', 'compras@hospitalgeneral.com', '1234', 'empresa', '2026-01-09 16:54:16', 'hosp_general'),
28 (27, 'Editorial Saber', 'pedidos@editorialsaber.com', '1234', 'empresa', '2026-01-09 16:54:16', 'edit_saber');
```

7 Resultados (1 a consulta tarda 0.003 segundos)

```
INSERT INTO usuario (id, nombre, email, password, rol, fecha_registro, username) VALUES
(1, 'Super Admin', 'admin@portal.com', '1234', 'admin', '2026-01-03 11:21:26', 'admin'),
(2, 'Gestor Politécnica', 'gestor@upm.es', '1234', 'gestor', '2026-01-03 11:21:26', 'gestor'),
(3, 'Gestor CodeMasters', 'director@codemasters.com', '1234', 'gestor', '2026-01-03 11:21:26', 'director'),
(4, 'Gestor EnglishNow', 'hello@englishnow.com', '1234', 'gestor', '2026-01-03 11:21:26', 'hello'),
(5, 'Tech Solutions CEO', 'contacto@techsolutions.com', '1234', 'empresa', '2026-01-03 11:21:26', 'contacto'),
(6, 'Consultora Global', 'rrhh@consultoraglobal.com', '1234', 'empresa', '2026-01-03 11:21:26', 'rrhh'),
(7, 'Startup Innova', 'adsis@innova.io', '1234', 'empresa', '2026-01-03 11:21:26', 'innova'),
(8, 'Carlos Pérez', 'carlos@mail.com', '1234', 'alumno', '2026-01-03 11:21:26', 'carlos'),
(9, 'María García', 'maria@mail.com', '1234', 'alumno', '2026-01-03 11:21:26', 'maria'),
(10, 'Lucía Méndez', 'lucia@mail.com', '1234', 'alumno', '2026-01-03 11:21:26', 'lucia'),
(11, 'David Torres', 'david@mail.com', '1234', 'alumno', '2026-01-03 11:21:26', 'david'),
(12, 'Elena Nito', 'elena@mail.com', '1234', 'alumno', '2026-01-03 11:21:26', 'elena'),
(13, 'Sofía Rivas', 'sofia@mail.com', '1234', 'alumno', '2026-01-03 11:21:26', 'sofia'),
(14, 'Jorge Bua', 'jorge@mail.com', '1234', 'alumno', '2026-01-03 11:21:26', 'jorge'),
(15, 'Ana Bot', 'ana@mail.com', '1234', 'alumno', '2026-01-03 11:21:26', 'ana'),
(16, 'Juan Riego Vila', 'juan@mail.com', '1234', 'alumno', '2026-01-08 10:55:50', 'juanrv'),
(17, 'Andrea Colorado Esteban', 'andrea@mail.com', '1234', '', '2026-01-08 13:01:31', 'andrea'),
(18, 'Juan Riego Vila', 'juanrv@mail.com', '1234', '', '2026-01-08 13:48:35', 'juanrv2'),
(19, 'Roberto Estudiante', 'robert@mail.com', '1234', 'alumno', '2026-01-09 16:01:32', 'robert_stud'),
(20, 'Laura Agramón', 'laura_nau@mail.com', '1234', 'alumno', '2026-01-09 16:01:32', 'lau2024'),
(21, 'Empresa TechSoft', 'admin@techsoft.com', '1234', 'empresa', '2026-01-09 16:01:32', 'techsoft_admin'),
(22, 'Constructora Civil S.L.', 'info@constructora.com', '1234', 'empresa', '2026-01-09 16:01:32', 'constructora_sl'),
(23, 'Academia BioAdmin', 'gestion@bioadmin.com', '1234', 'gestor', '2026-01-09 16:01:32', 'bio_gestor'),
(24, 'Pedro Profesor', 'pedro@mail.com', '1234', 'alumno', '2026-01-09 16:01:32', 'pedroprof'),
(25, 'Marta Diseño', 'marta@design.com', '1234', 'empresa', '2026-01-09 16:01:32', 'marta_design'),
(26, 'Hospital General', 'compras@hospitalgeneral.com', '1234', 'empresa', '2026-01-09 16:54:16', 'hosp_general'),
(27, 'Editorial Saber', 'pedidos@editorialsaber.com', '1234', 'empresa', '2026-01-09 16:54:16', 'edit_saber');
```

## Tabla institución

```
1 INSERT INTO institucion (id, nombre, tipo, ubicacion, contacto) VALUES
2 (1, 'Universidad Politécnica de Madrid', 'universidad', 'Av. Complutense, 30, Madrid', 'secretaria@upm.es'),
3 (2, 'Academia CodeMasters', 'academia', 'Calle Per 42, Barcelona', 'info@codemasters.com'),
4 (3, 'English Now School', 'academia', 'Gran Vía 12, Valencia', 'contact@englishnow.com'),
5 (4, 'Centro de Investigaciones Biológicas', 'academia', 'Calle Ciencia 88, Sevilla', 'contacto@cib.es'),
6 (5, 'Universidad de las Artes y Diseño', 'universidad', 'Av. Creativa 10, Bilbao', 'admit@uartes.edu'),
7 (6, 'Instituto Tecnológico del Norte', 'universidad', 'Plaza Mayor 5, Santander', 'info@itn.edu'),
8 (7, 'Escuela de Negocios Global', 'academia', 'Paseo de la Castellana 200, Madrid', 'eba@negocios.com');
```

7 Resultados (1 a consulta tarda 0.003 segundos)

```
INSERT INTO institucion (id, nombre, tipo, ubicacion, contacto) VALUES
(1, 'Universidad Politécnica de Madrid', 'universidad', 'Av. Complutense, 30, Madrid', 'secretaria@upm.es'),
(2, 'Academia CodeMasters', 'academia', 'Calle Per 42, Barcelona', 'info@codemasters.com'),
(3, 'English Now School', 'academia', 'Gran Vía 12, Valencia', 'contact@englishnow.com'),
(4, 'Centro de Investigaciones Biológicas', 'academia', 'Calle Ciencia 88, Sevilla', 'contacto@cib.es'),
(5, 'Universidad de las Artes y Diseño', 'universidad', 'Av. Creativa 10, Bilbao', 'admit@uartes.edu'),
(6, 'Instituto Tecnológico del Norte', 'universidad', 'Plaza Mayor 5, Santander', 'info@itn.edu'),
(7, 'Escuela de Negocios Global', 'academia', 'Paseo de la Castellana 200, Madrid', 'eba@negocios.com');
```

## **Tabla materia**

```

1 INSERT INTO materia (id, nombre, parentId) VALUES
2 (3, 'Programación', NULL),
3 (2, 'Idiomas', NULL),
4 (3, 'Matemáticas', NULL),
5 (4, 'Desarrollo Web', 1),
6 (5, 'Ciberseguridad', 1),
7 (6, 'Bases de Datos', 1),
8 (7, 'Inglés', 2),
9 (8, 'Alemán', 2),
10 (9, 'Álgebra', 3),
11 (10, 'Estadística', 3),
12 (13, 'Biología', NULL),
13 (14, 'Arte y Humanidades', NULL),
14 (15, 'Biología Celular', 13),
15 (16, 'Genética Molecular', 13),
16 (17, 'Historia del Arte', 14),
17 (18, 'Diseño Gráfico Digital', 14),
18 (19, 'Inteligencia Artificial', 1),
19 (21, 'Física', NULL),
20 (22, 'Química', NULL),
21 (23, 'Historia', NULL),
22 (24, 'Economía', NULL),
23 (25, 'Introducción a Python', 1),
24 (26, 'Cinemática y Dinámica', 21),
25 (29, 'Electromagnetismo', 21),
26 (30, 'Termodinámico', 21),
27 (31, 'Formulación Inorgánica', 22),
28 (32, 'Tabla Periódica', 22),
29 (33, 'Reacciones Químicas', 22),

```

51 Max overtook the 52 company truck 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161 162 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198 199 200 201 202 203 204 205 206 207 208 209 210 211 212 213 214 215 216 217 218 219 220 221 222 223 224 225 226 227 228 229 230 231 232 233 234 235 236 237 238 239 240 241 242 243 244 245 246 247 248 249 250 251 252 253 254 255 256 257 258 259 260 261 262 263 264 265 266 267 268 269 270 271 272 273 274 275 276 277 278 279 280 281 282 283 284 285 286 287 288 289 290 291 292 293 294 295 296 297 298 299 300 301 302 303 304 305 306 307 308 309 310 311 312 313 314 315 316 317 318 319 320 321 322 323 324 325 326 327 328 329 330 331 332 333 334 335 336 337 338 339 340 341 342 343 344 345 346 347 348 349 350 351 352 353 354 355 356 357 358 359 360 361 362 363 364 365 366 367 368 369 370 371 372 373 374 375 376 377 378 379 380 381 382 383 384 385 386 387 388 389 390 391 392 393 394 395 396 397 398 399 400 401 402 403 404 405 406 407 408 409 410 411 412 413 414 415 416 417 418 419 420 421 422 423 424 425 426 427 428 429 430 431 432 433 434 435 436 437 438 439 440 441 442 443 444 445 446 447 448 449 450 451 452 453 454 455 456 457 458 459 460 461 462 463 464 465 466 467 468 469 470 471 472 473 474 475 476 477 478 479 480 481 482 483 484 485 486 487 488 489 490 491 492 493 494 495 496 497 498 499 500 501 502 503 504 505 506 507 508 509 510 511 512 513 514 515 516 517 518 519 520 521 522 523 524 525 526 527 528 529 530 531 532 533 534 535 536 537 538 539 540 541 542 543 544 545 546 547 548 549 550 551 552 553 554 555 556 557 558 559 560 561 562 563 564 565 566 567 568 569 570 571 572 573 574 575 576 577 578 579 580 581 582 583 584 585 586 587 588 589 590 591 592 593 594 595 596 597 598 599 600 601 602 603 604 605 606 607 608 609 610 611 612 613 614 615 616 617 618 619 620 621 622 623 624 625 626 627 628 629 630 631 632 633 634 635 636 637 638 639 640 641 642 643 644 645 646 647 648 649 650 651 652 653 654 655 656 657 658 659 660 661 662 663 664 665 666 667 668 669 670 671 672 673 674 675 676 677 678 679 680 681 682 683 684 685 686 687 688 689 690 691 692 693 694 695 696 697 698 699 700 701 702 703 704 705 706 707 708 709 710 711 712 713 714 715 716 717 718 719 720 721 722 723 724 725 726 727 728 729 730 731 732 733 734 735 736 737 738 739 740 741 742 743 744 745 746 747 748 749 750 751 752 753 754 755 756 757 758 759 760 761 762 763 764 765 766 767 768 769 770 771 772 773 774 775 776 777 778 779 780 781 782 783 784 785 786 787 788 789 790 791 792 793 794 795 796 797 798 799 800 801 802 803 804 805 806 807 808 809 810 811 812 813 814 815 816 817 818 819 820 821 822 823 824 825 826 827 828 829 830 831 832 833 834 835 836 837 838 839 840 841 842 843 844 845 846 847 848 849 850 851 852 853 854 855 856 857 858 859 860 861 862 863 864 865 866 867 868 869 870 871 872 873 874 875 876 877 878 879 880 881 882 883 884 885 886 887 888 889 890 891 892 893 894 895 896 897 898 899 900 901 902 903 904 905 906 907 908 909 910 911 912 913 914 915 916 917 918 919 920 921 922 923 924 925 926 927 928 929 930 931 932 933 934 935 936 937 938 939 940 941 942 943 944 945 946 947 948 949 950 951 952 953 954 955 956 957 958 959 960 961 962 963 964 965 966 967 968 969 970 971 972 973 974 975 976 977 978 979 980 981 982 983 984 985 986 987 988 989 990 991 992 993 994 995 996 997 998 999 1000 1001 1002 1003 1004 1005 1006 1007 1008 1009 1010 1011 1012 1013 1014 1015 1016 1017 1018 1019 1020 1021 1022 1023 1024 1025 1026 1027 1028 1029 1030 1031 1032 1033 1034 1035 1036 1037 1038 1039 1040 1041 1042 1043 1044 1045 1046 1047 1048 1049 1050 1051 1052 1053 1054 1055 1056 1057 1058 1059 1060 1061 1062 1063 1064 1065 1066 1067 1

[1991] 1991 articles: (1) *Antes, después: 1940-1990*, (2) *Argentinidad*, (3) *El '68*, (4) *El '73*, (5) *El '76*, (6) *El '82*, (7) *El '83*, (8) *El '84*, (9) *El '85*, (10) *El '86*, (11) *El '87*, (12) *El '88*, (13) *El '89*, (14) *El '90*, (15) *El '91*, (16) *El '92*, (17) *El '93*, (18) *El '94*, (19) *El '95*, (20) *El '96*, (21) *El '97*, (22) *El '98*, (23) *El '99*, (24) *El '00*, (25) *El '01*, (26) *El '02*, (27) *El '03*, (28) *El '04*, (29) *El '05*, (30) *El '06*, (31) *El '07*, (32) *El '08*, (33) *El '09*, (34) *El '10*, (35) *El '11*, (36) *El '12*, (37) *El '13*, (38) *El '14*, (39) *El '15*, (40) *El '16*, (41) *El '17*, (42) *El '18*, (43) *El '19*, (44) *El '20*, (45) *El '21*, (46) *El '22*, (47) *El '23*, (48) *El '24*, (49) *El '25*, (50) *El '26*, (51) *El '27*, (52) *El '28*, (53) *El '29*, (54) *El '30*, (55) *El '31*, (56) *El '32*, (57) *El '33*, (58) *El '34*, (59) *El '35*, (60) *El '36*, (61) *El '37*, (62) *El '38*, (63) *El '39*, (64) *El '40*, (65) *El '41*, (66) *El '42*, (67) *El '43*, (68) *El '44*, (69) *El '45*, (70) *El '46*, (71) *El '47*, (72) *El '48*, (73) *El '49*, (74) *El '50*, (75) *El '51*, (76) *El '52*, (77) *El '53*, (78) *El '54*, (79) *El '55*, (80) *El '56*, (81) *El '57*, (82) *El '58*, (83) *El '59*, (84) *El '60*, (85) *El '61*, (86) *El '62*, (87) *El '63*, (88) *El '64*, (89) *El '65*, (90) *El '66*, (91) *El '67*, (92) *El '68*, (93) *El '69*, (94) *El '70*, (95) *El '71*, (96) *El '72*, (97) *El '73*, (98) *El '74*, (99) *El '75*, (100) *El '76*, (101) *El '77*, (102) *El '78*, (103) *El '79*, (104) *El '80*, (105) *El '81*, (106) *El '82*, (107) *El '83*, (108) *El '84*, (109) *El '85*, (110) *El '86*, (111) *El '87*, (112) *El '88*, (113) *El '89*, (114) *El '90*, (115) *El '91*, (116) *El '92*, (117) *El '93*, (118) *El '94*, (119) *El '95*, (120) *El '96*, (121) *El '97*, (122) *El '98*, (123) *El '99*, (124) *El '00*, (125) *El '01*, (126) *El '02*, (127) *El '03*, (128) *El '04*, (129) *El '05*, (130) *El '06*, (131) *El '07*, (132) *El '08*, (133) *El '09*, (134) *El '10*, (135) *El '11*, (136) *El '12*, (137) *El '13*, (138) *El '14*, (139) *El '15*, (140) *El '16*, (141) *El '17*, (142) *El '18*, (143) *El '19*, (144) *El '20*, (145) *El '21*, (146) *El '22*, (147) *El '23*, (148) *El '24*, (149) *El '25*, (150) *El '26*, (151) *El '27*, (152) *El '28*, (153) *El '29*, (154) *El '30*, (155) *El '31*, (156) *El '32*, (157) *El '33*, (158) *El '34*, (159) *El '35*, (160) *El '36*, (161) *El '37*, (162) *El '38*, (163) *El '39*, (164) *El '40*, (165) *El '41*, (166) *El '42*, (167) *El '43*, (168) *El '44*, (169) *El '45*, (170) *El '46*, (171) *El '47*, (172) *El '48*, (173) *El '49*, (174) *El '50*, (175) *El '51*, (176) *El '52*, (177) *El '53*, (178) *El '54*, (179) *El '55*, (180) *El '56*, (181) *El '57*, (182) *El '58*, (183) *El '59*, (184) *El '60*, (185) *El '61*, (186) *El '62*, (187) *El '63*, (188) *El '64*, (189) *El '65*, (190) *El '66*, (191) *El '67*, (192) *El '68*, (193) *El '69*, (194) *El '70*, (195) *El '71*, (196) *El '72*, (197) *El '73*, (198) *El '74*, (199) *El '75*, (200) *El '76*, (201) *El '77*, (202) *El '78*, (203) *El '79*, (204) *El '80*, (205) *El '81*, (206) *El '82*, (207) *El '83*, (208) *El '84*, (209) *El '85*, (210) *El '86*, (211) *El '87*, (212) *El '88*, (213) *El '89*, (214) *El '90*, (215) *El '91*, (216) *El '92*, (217) *El '93*, (218) *El '94*, (219) *El '95*, (220) *El '96*, (221) *El '97*, (222) *El '98*, (223) *El '99*, (224) *El '00*, (225) *El '01*, (226) *El '02*, (227) *El '03*, (228) *El '04*, (229) *El '05*, (230) *El '06*, (231) *El '07*, (232) *El '08*, (233) *El '09*, (234) *El '10*, (235) *El '11*, (236) *El '12*, (237) *El '13*, (238) *El '14*, (239) *El '15*, (240) *El '16*, (241) *El '17*, (242) *El '18*, (243) *El '19*, (244) *El '20*, (245) *El '21*, (246) *El '22*, (247) *El '23*, (248) *El '24*, (249) *El '25*, (250) *El '26*, (251) *El '27*, (252) *El '28*, (253) *El '29*, (254) *El '30*, (255) *El '31*, (256) *El '32*, (257) *El '33*, (258) *El '34*, (259) *El '35*, (260) *El '36*, (261) *El '37*, (262) *El '38*, (263) *El '39*, (264) *El '40*, (265) *El '41*, (266) *El '42*, (267) *El '43*, (268) *El '44*, (269) *El '45*, (270) *El '46*, (271) *El '47*, (272) *El '48*, (273) *El '49*, (274) *El '50*, (275) *El '51*, (276) *El '52*, (277) *El '53*, (278) *El '54*, (279) *El '*

[Case]

### **Tabla personal**

```
1 INSERT INTO personal (id, nombre, apellidos, institucionId) VALUES
2 (1, 'Dr. Roberto', 'Méndez', 1),
3 (2, 'Dra. Laura', 'Física', 1),
4 (3, 'Davien', 'Hacker', 2),
5 (4, 'Sara', 'Coder', 2),
6 (5, 'John', 'Smith', 3),
7 (6, 'Frau', 'Müller', 3),
8 (10, 'Dra. Elena', 'Genoma', 4),
9 (11, 'Prof. Marco', 'Píxel', 5),
10 (12, 'Ing. Sofia', 'Robótica', 6),
11 (13, 'Dr. Alan', 'Turing', 1),
12 (14, 'Prof. Alberto', 'Einstein', 1),
13 (15, 'Dra. Marie', 'Curie', 1),
14 (16, 'Prof. Adam', 'Smith', 2),
15 (17, 'Lic. Juana', 'Lovelace', 2),
16 (18, 'Dr. Herodoto', 'Pérez', 1),
17 (19, 'Dra. Rosalind', 'Franklin', 4),
18 (20, 'Prof. Isaac', 'Newton', 1),
19 (21, 'Lic. Ada', 'Byron', 2),
20 (22, 'Dr. John', 'Keynes', 7);
```

© 1998 Macmillan, a division of Macmillan Publishing Co., Inc.

[illegible]

© 2000 Blackwell Science Ltd *Journal of Internal Medicine* 247: 395–401



## Tabla documento

```
1 INSERT INTO documento (id, titulo, archivo_url, tipo_acceso, materialId, autorId, institucionId) VALUES
2 (1, 'Plan de Estudios 2025', 'plan2025.pdf', 'publico', 1, 1, 1),
3 (2, 'Tesis Doctoral: IA Generativa', 'tesis_la.pdf', 'publico', 1, 1, 1),
4 (3, 'Ejercicios de Álgebra Lineal', 'algebra_ej.pdf', 'privado', 3, 2, 1),
5 (4, 'Guía Definitiva de React Hooks', 'react_guide.pdf', 'publico', 4, 4, 2),
6 (5, 'Manual de Hacking Ético', 'hacking_v1.pdf', 'privado', 5, 1, 2),
7 (6, 'Cheat Sheet PHP 8.2', 'php_resumen.pdf', 'publico', 4, 4, 2),
8 (7, 'Laboratorio SQL Avanzado', 'sql_lab.pdf', 'privado', 6, 3, 2),
9 (8, 'Phrasal Verbs List', 'phrasal_verbs.pdf', 'publico', 7, 5, 3),
10 (9, 'German Grammar B1', 'deutsch_b1.pdf', 'privado', 8, 6, 3),
11 (20, 'Introducción a la Edición Genética CRISPR', 'crispr_intro.pdf', 'publico', 16, 10, 4),
12 (21, 'Historia del Renacimiento Italiano', 'renacimiento.pdf', 'privado', 17, 11, 5),
13 (22, 'Redes Neuronales Convolucionales', 'cnn_guide.pdf', 'privado', 19, 12, 6),
14 (23, 'Principios de Composición Visual', 'composicion.pdf', 'publico', 18, 13, 5),
15 (24, 'Atlas Celular 2026', 'atlas_celular.pdf', 'privado', 15, 10, 4),
16 (35, 'Leyes de Newton: Resumen y Ejercicios', 'newton_resumen.pdf', 'publico', 28, 14, 1),
17 (36, 'Formulario de Física para Selectividad', 'formulario_fisica.pdf', 'publico', 23, 14, 1),
18 (37, 'Tabla de Valencia para Imprimir', 'valencias.pdf', 'publico', 31, 15, 1),
19 (38, 'Ejercicios de Ajuste de Reacciones', 'ajuste_reacciones.pdf', 'privado', 33, 15, 1),
20 (39, 'Mi primer programa: Hola Mundo en Python', 'python_intro.pdf', 'publico', 25, 17, 2),
21 (40, 'Guía de Bucles y Condicionales', 'bucles_java.pdf', 'privado', 1, 17, 2),
22 (41, 'Eje Cronológico de la Guerra Civil', 'eje_guerra_civil.pdf', 'publico', 34, 18, 1),
23 (42, 'Resumen Primera Guerra Mundial', 'ww1_resumen.pdf', 'publico', 35, 18, 1),
24 (43, 'Conceptos básicos: Oferta y Demanda', 'oferta_demanda.pdf', 'publico', 38, 5, 2),
25 (44, 'Balance de Situación: Ejercicio Resuelto', 'balance_ejemplo.pdf', 'privado', 37, 5, 2),
36 (45, 'Comportamiento de la Luz', 'optica_intro.pdf', 'publico', 42, 20, 1),
37 (46, 'Energía Nuclear: Ventajas y Riesgos', 'nuclear_debate.pdf', 'privado', 43, 14, 1),
38 (47, 'Nomenclatura de Carbono', 'quimica_organica_1.pdf', 'publico', 45, 19, 4),
39 (48, 'Manual de Seguridad en Laboratorio', 'safety_lab.pdf', 'publico', 47, 15, 1),
```

✓ 50 filas insertadas. (La consulta tomó 0.0001 segundos.)

```
SELECT * FROM documento (id, titulo, archivo_url, tipo_acceso, materialId, autorId, institucionId) WHERE (id, 'Plan de Estudios 2025', 'plan2025.pdf', 'publico', 1, 1, 1), ('Tesis Doctoral: IA Generativa', 'tesis_la.pdf', 'publico', 1, 1, 1), ('Ejercicios de Álgebra Lineal', 'algebra_ej.pdf', 'privado', 3, 2, 1), ('Guía Definitiva de React Hooks', 'react_guide.pdf', 'publico', 4, 4, 2), ('Manual de Hacking Ético', 'hacking_v1.pdf', 'privado', 5, 1, 2), ('Cheat Sheet PHP 8.2', 'php_resumen.pdf', 'publico', 4, 4, 2), ('Laboratorio SQL Avanzado', 'sql_lab.pdf', 'privado', 6, 3, 2), ('Phrasal Verbs List', 'phrasal_verbs.pdf', 'publico', 7, 5, 3), ('German Grammar B1', 'deutsch_b1.pdf', 'privado', 8, 6, 3), ('Introducción a la Edición Genética CRISPR', 'crispr_intro.pdf', 'publico', 16, 10, 4), ('Historia del Renacimiento Italiano', 'renacimiento.pdf', 'privado', 17, 11, 5), ('Redes Neuronales Convolucionales', 'cnn_guide.pdf', 'privado', 19, 12, 6), ('Principios de Composición Visual', 'composicion.pdf', 'publico', 18, 13, 5), ('Atlas Celular 2026', 'atlas_celular.pdf', 'privado', 15, 10, 4), ('Leyes de Newton: Resumen y Ejercicios', 'newton_resumen.pdf', 'publico', 28, 14, 1), ('Formulario de Física para Selectividad', 'formulario_fisica.pdf', 'publico', 23, 14, 1), ('Tabla de Valencia para Imprimir', 'valencias.pdf', 'publico', 31, 15, 1), ('Ejercicios de Ajuste de Reacciones', 'ajuste_reacciones.pdf', 'privado', 33, 15, 1), ('Mi primer programa: Hola Mundo en Python', 'python_intro.pdf', 'publico', 25, 17, 2), ('Guía de Bucles y Condicionales', 'bucles_java.pdf', 'privado', 1, 17, 2), ('Eje Cronológico de la Guerra Civil', 'eje_guerra_civil.pdf', 'publico', 34, 18, 1), ('Resumen Primera Guerra Mundial', 'ww1_resumen.pdf', 'publico', 35, 18, 1), ('Conceptos básicos: Oferta y Demanda', 'oferta_demanda.pdf', 'publico', 38, 5, 2), ('Balance de Situación: Ejercicio Resuelto', 'balance_ejemplo.pdf', 'privado', 37, 5, 2), ('Comportamiento de la Luz', 'optica_intro.pdf', 'publico', 42, 20, 1), ('Energía Nuclear: Ventajas y Riesgos', 'nuclear_debate.pdf', 'privado', 43, 14, 1), ('Nomenclatura de Carbono', 'quimica_organica_1.pdf', 'publico', 45, 19, 4), ('Manual de Seguridad en Laboratorio', 'safety_lab.pdf', 'publico', 47, 15, 1),
```

[Cierre]

## Tabla servicio

```
1 INSERT INTO servicio (id, nombre, precio, institucionId) VALUES
2 (1, 'Máster en Ingeniería de Datos', 4500.00, 1),
3 (2, 'Bootcamp Full Stack Developer', 2900.00, 2),
4 (3, 'Auditoría de Seguridad Web', 1200.00, 2),
5 (4, 'Curso Inglés de Negocios', 800.00, 3),
6 (5, 'Pack Formación Alemán Técnico', 1500.00, 3),
7 (6, 'Curso Avanzado de Genética', 600.00, 4),
8 (7, 'Máster en Diseño UI/UX', 3500.00, 5),
9 (8, 'Bootcamp de Inteligencia Artificial', 2200.00, 6),
10 (9, 'MBA Executive', 12000.00, 7);
```

✓ 9 filas insertadas. (La consulta tomó 0.0010 segundos.)

```
SELECT * FROM servicio (id, nombre, precio, institucionId) WHERE (id, 'Máster en Ingeniería de Datos', 4500.00, 1), ('Bootcamp Full Stack Developer', 2900.00, 2), ('Auditoría de Seguridad Web', 1200.00, 2), ('Curso Inglés de Negocios', 800.00, 3), ('Pack Formación Alemán Técnico', 1500.00, 3), ('Curso Avanzado de Genética', 600.00, 4), ('Máster en Diseño UI/UX', 3500.00, 5), ('Bootcamp de Inteligencia Artificial', 2200.00, 6), ('MBA Executive', 12000.00, 7),
```

[Cierre en línea] [Cierre en PHP]

## Tabla user\_log

```
1 INSERT INTO user_log (id, user_id, user_identifier, action, ip_address, user_agent, created_at) VALUES
2 (1, 10, 'juanv2@mail.com', 'Logout', '111', 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/143.0.0.0 Safari/537.36 Edg/143.0.0.0', '2026-01-08 13:54:13'),
3 (2, 10, 'juanv2@mail.com', 'Login Success', '111', 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/143.0.0.0 Safari/537.36 Edg/143.0.0.0', '2026-01-08 13:54:14'),
4 (3, 10, 'juanv2@mail.com', 'Logout', '111', 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/143.0.0.0 Safari/537.36 Edg/143.0.0.0', '2026-01-08 13:54:14'),
5 (4, 8, 'carlos@mail.com', 'Login Success', '192.168.1.10', 'Mozilla/5.0 (Windows) Chrome/99', '2026-01-09 16:53:02'),
6 (5, 8, 'carlos@mail.com', 'View Document ID 20', '192.168.1.10', 'Mozilla/5.0 (Macintosh; Intel Mac OS) Safari/14', '2026-01-09 16:53:02'),
7 (6, 8, 'carlos@mail.com', 'Download PDF', '192.168.1.10', 'Mozilla/5.0 (Windows) Chrome/99', '2026-01-09 16:53:02'),
8 (7, 10, 'roberta_cruz', 'Login Failed', '10.0.0.0', 'Mozilla/5.0 (Linux) Firefox/98', '2026-01-09 16:53:02'),
9 (8, 10, 'roberta_cruz', 'Login Success', '10.0.0.0', 'Mozilla/5.0 (Linux) Firefox/98', '2026-01-09 16:53:02'),
10 (9, 10, 'juanv2@mail.com', 'Update Profile', '192.168.1.10', 'Mozilla/5.0 (iPhone) Safari/14', '2026-01-09 16:53:02'),
11 (10, 5, 'contacto', 'View Invoice', '90.20.10.5', 'Mozilla/5.0 (Windows) Edge/90', '2026-01-09 16:53:02'),
12 (11, 20, 'luis2024', 'Search query: Python', '192.168.0.13', 'Mozilla/5.0 (Android) Chrome/99', '2026-01-08 16:53:02'),
13 (12, 11, 'david@mail.com', 'Logout', '192.168.1.30', 'Mozilla/5.0 (Windows) Chrome/90', '2026-01-09 16:53:02'),
14 (13, 10, 'juanv2', 'Login Success', '111', 'Mozilla/5.0 (Windows) Chrome/120', '2026-01-09 16:53:02'),
15 (14, 10, 'juanv2', 'View Dashboard', '111', 'Mozilla/5.0 (Windows) Chrome/120', '2026-01-09 16:53:02'),
```

✓ 15 filas insertadas. (La consulta tomó 0.0004 segundos.)

```
SELECT * FROM user_log (id, user_id, user_identifier, action, ip_address, user_agent, created_at) WHERE (id, 10, 'juanv2@mail.com', 'Logout', '111', 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/143.0.0.0 Safari/537.36 Edg/143.0.0.0', '2026-01-08 13:54:13'), (2, 10, 'juanv2@mail.com', 'Login Success', '111', 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/143.0.0.0 Safari/537.36 Edg/143.0.0.0', '2026-01-08 13:54:14'), (3, 10, 'juanv2@mail.com', 'Logout', '111', 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/143.0.0.0 Safari/537.36 Edg/143.0.0.0', '2026-01-08 13:54:14'), (4, 8, 'carlos@mail.com', 'Login Success', '192.168.1.10', 'Mozilla/5.0 (Windows) Chrome/99', '2026-01-09 16:53:02'), (5, 8, 'carlos@mail.com', 'View Document ID 20', '192.168.1.10', 'Mozilla/5.0 (Macintosh; Intel Mac OS) Safari/14', '2026-01-09 16:53:02'), (6, 8, 'carlos@mail.com', 'Download PDF', '192.168.1.10', 'Mozilla/5.0 (Windows) Chrome/99', '2026-01-09 16:53:02'), (7, 10, 'roberta_cruz', 'Login Failed', '10.0.0.0', 'Mozilla/5.0 (Linux) Firefox/98', '2026-01-09 16:53:02'), (8, 10, 'roberta_cruz', 'Login Success', '10.0.0.0', 'Mozilla/5.0 (Linux) Firefox/98', '2026-01-09 16:53:02'), (9, 10, 'juanv2@mail.com', 'Update Profile', '192.168.1.10', 'Mozilla/5.0 (iPhone) Safari/14', '2026-01-09 16:53:02'), (10, 5, 'contacto', 'View Invoice', '90.20.10.5', 'Mozilla/5.0 (Windows) Edge/90', '2026-01-09 16:53:02'), (11, 20, 'luis2024', 'Search query: Python', '192.168.0.13', 'Mozilla/5.0 (Android) Chrome/99', '2026-01-08 16:53:02'), (12, 11, 'david@mail.com', 'Logout', '192.168.1.30', 'Mozilla/5.0 (Windows) Chrome/90', '2026-01-09 16:53:02'), (13, 10, 'juanv2', 'Login Success', '111', 'Mozilla/5.0 (Windows) Chrome/120', '2026-01-09 16:53:02'), (14, 10, 'juanv2', 'View Dashboard', '111', 'Mozilla/5.0 (Windows) Chrome/120', '2026-01-09 16:53:02'),
```

[Cierre]

### ***Tabla pregunta***

INICIO	FIN	pregunta	id	título	descripcion	fechaCreacion	usuario	estado	usuarioId	materiaId	VALORES
1	1	"¿Cómo implementar recursividad en Python?"	1	"No entiendo bien el caso base en las funciones recursivas. ¿Alguien tiene un ejemplo simple con factoriales?"	"2020-01-05 18:58:00"	11	1	"sin_responder"	36	231	
2	2	"Diferencia conceptual entre derivada e integral"	2	"Entiendo cómo calcularlas, pero no entiendo qué significa en la vida real. ¿pueden?"	"2020-01-06 14:55:00"	150	2	"resuelta"	8	72	
3	3	"Truco para memorizar Phrasal Verbs"	3	"Son demasiados y me confundo entre 'get on', 'get in', 'get by'". ¿Alguien consejo?"	"2020-01-04 09:00:00"	342	10	"respondida"	8	7	
4	4	"Demostración visual de Pitágoras"	4	"Busco una explicación geométrica, no solo la fórmula $a^2 + b^2 = c^2$ ".	"2020-02-02 16:20:00"	139	3	"resuelta"	13	77	
5	5	"Organización de la Tabla Periódica"	5	"¿Por qué las gases nobles están a la derecha del todo?"	"2020-01-07 11:43:00"	67	0	"sin_responder"	13	12	

[illegible]

## Tabla coleccion

```

1 INSERT INTO coleccion (id_titulo, descripcion, usuarioId, descargas, tipo_acceso) VALUES
2 (1, 'Matemáticas de Ingeniería', 'Cálculo diferencial, integral y vectores en 3D.', 10, 450, 'publico'),
3 (2, 'Desarrollo Frontend Moderno', 'Dominando React Hooks y composición visual.', 4, 1200, 'publico'),
4 (3, 'Ciberseguridad y Redes', 'Hacking ético, SQL avanzado y seguridad en redes.', 3, 850, 'publico'),
5 (4, 'Idiomas y Cultura Global', 'Francés, Alemán, Italiano y cultura japonesa.', 8, 340, 'publico'),
6 (5, 'Historia y Civilizaciones', 'Del Renacimiento a la Guerra Civil Española.', 10, 560, 'publico'),
7 (6, 'Biología y Genética 2024', 'Edición genética CRISPR y Atlas Celular.', 10, 310, 'publico'),
8 (7, 'Física y Química General', 'Leyes de Newton, óptica y nomenclatura de carbono.', 15, 670, 'publico'),
9 (8, 'Economía y Finanzas', 'Oferta, demanda, contabilidad e interés compuesto.', 5, 430, 'publico'),
10 (9, 'Arte y Composición Visual', 'Teoría del color, fotografía y diseño gráfico.', 11, 780, 'publico'),
11 (10, 'IA y Neurociencia', 'Redes neuronales y plasticidad cerebral.', 1, 1500, 'publico'),
12 (11, 'Programación con Python', 'Desde Hola Mundo hasta estructuras de datos.', 17, 920, 'publico'),
13 (12, 'Tecnología Móvil y Apps', 'Desarrollo en Android y estrategias móviles.', 21, 310, 'publico'),
14 (13, 'Producción Musical Pro', 'Mezcla, equalización y teoría musical.', 11, 150, 'publico'),
15 (14, 'Zoología y Ecosistemas', 'Fauna ibérica e invertebrados marinos.', 10, 85, 'publico'),

```

[Ver lista de inventarios](#) | [Ver la consulta herself 0.0001 segunda](#)

## Tabla contratación

```
INSERT INTO contratacion (Id, clienteId, servicioId, fecha) VALUES
1 (1, 1, 2, '2023-09-01 16:00:00'),
2 (2, 1, 3, '2024-01-15 09:30:00'),
3 (3, 2, 4, '2024-02-01 11:00:00'),
4 (4, 3, 2, '2024-03-10 16:45:00'),
5 (5, 4, 8, '2026-01-10 10:00:00'),
6 (6, 6, 7, '2026-01-12 11:30:00'),
7 (7, 1, 9, '2026-01-15 09:00:00'),
8 (8, 2, 1, '2025-11-15 09:00:00'),
9 (9, 3, 8, '2025-12-01 10:30:00'),
10 (10, 5, 9, '2025-12-10 16:00:00'),
11 (11, 1, 9, '2026-01-02 08:45:00'),
12 (12, 4, 2, '2026-01-05 11:15:00'),
13 (13, 6, 6, '2026-01-08 14:20:00'),
14 (14, 2, 3, '2026-01-09 09:00:00'),
15 (15, 7, 6, '2026-01-05 10:00:00'),
16 (16, 8, 4, '2026-01-07 12:30:00');
```

[↗ 50 hits encontrados. La consulta tardó 0.0004 segundos.](#)

[2007-07-01](#) [construccion](#) [34](#), [clase02](#), [arriba0104](#), [fecha](#) [44/02](#) [1](#), [1](#), [2007-09-05 10:00:00](#) [1](#), [1](#), [1](#), [2014-09-15 10:00:00](#) [1](#), [3](#), [4](#), [2014-01-01 11:00:00](#) [1](#), [4](#), [2](#), [2014-09-10 10:47:00](#) [1](#), [3](#), [4](#), [2015-09-18 00:00:00](#) [1](#), [3](#), [4](#), [2015-05-12 11:30:00](#) [1](#), [1](#), [2](#), [2015-09-11 00:00:00](#) [1](#), [1](#), [1](#), [2015-11-15 09:00:00](#) [1](#), [3](#), [3](#), [2017-11-05 09:30:00](#) [1](#), [10](#), [7](#), [2015-11-28 10:00:00](#) [1](#), [11](#), [1](#), [2015-09-01 00:00:00](#) [1](#), [11](#), [4](#), [2015-09-01 12:29:00](#) [1](#), [13](#), [4](#), [2015-09-01 00:00:00](#) [1](#), [19](#), [3](#), [2015-03-01 00:00:00](#) [1](#), [13](#), [3](#), [2015-09-01 10:00:00](#) [1](#), [13](#), [2](#), [2015-09-07 11:30:00](#) [1](#)

[\[ Alterar consulta \]](#) [\[ Ver los 50 hits \]](#) [\[ Descarga CSV \]](#)

### **Tabla documento\_comentario**

```
1 INSERT INTO documento_comentario (id, contenido, fecha, usuarioId, documentoId) VALUES
2 (1, '¡Excelente guía! Por fin entiendo useEffect.', '2026-01-05 10:00:00', 8, 4),
3 (2, '¡Podrías añadir ejemplos con TypeScript?', '2026-01-05 12:30:00', 16, 4),
4 (3, 'Me ha salvado para el proyecto final, gracias.', '2026-01-06 09:15:00', 9, 4),
5 (4, 'El PDF no se ve bien en móvil, ¿pueden revisarlo?', '2026-01-02 18:00:00', 13, 39),
6 (5, 'Muy básico pero bien explicado.', '2026-01-03 11:20:00', 11, 39),
7 (6, 'Los ejercicios del final no tienen solución, ¿dónde puedo verlos?', '2026-01-07 16:45:00', 20, 35)
```

0 Resultados (A consulta levou 0.007 segundos)

## Tabla documento\_valoracion

```
1 INSERT INTO documento_valoracion (usuarioId, documentoId, puntuacion, fecha) VALUES
2 (8, 4, 5, '2026-01-09 20:00:48'),
3 (8, 35, 5, '2026-01-09 20:00:48'),
4 (9, 4, 5, '2026-01-09 20:00:48'),
5 (11, 39, 4, '2026-01-09 20:00:48'),
6 (13, 4, 5, '2026-01-09 20:00:48'),
7 (13, 39, 3, '2026-01-09 20:00:48'),
8 (16, 4, 4, '2026-01-09 20:00:48'),
9 (20, 35, 4, '2026-01-09 20:00:48'),
10 (20, 39, 5, '2026-01-09 20:00:48');
```

✓ 9 filas insertadas. (La consulta tardó 0.0013 segundos.)

```
INSERT INTO documento_valoracion (usuarioId, documentoId, puntuacion, fecha) VALUES (8, 4, 5, '2026-01-09 20:00:48'), (8, 35, 5, '2026-01-09 20:00:48'), (9, 4, 5, '2026-01-09 20:00:48'), (11, 39, 4, '2026-01-09 20:00:48'), (13, 4, 5, '2026-01-09 20:00:48'), (13, 39, 3, '2026-01-09 20:00:48'), (16, 4, 4, '2026-01-09 20:00:48'), (20, 35, 4, '2026-01-09 20:00:48'), (20, 39, 5, '2026-01-09 20:00:48');
```

[Editar en línea] [Editar] [Código PHP]

## Tabla pregunta\_guardada

```
1 INSERT INTO pregunta_guardada (usuarioId, preguntaId, fecha_guardado) VALUES
2 (8, 5, '2026-01-09 19:24:04'),
3 (9, 1, '2026-01-09 19:24:04'),
4 (16, 2, '2026-01-09 19:24:04'),
5 (16, 3, '2026-01-09 19:24:04');
```

✓ 4 filas insertadas. (La consulta tardó 0.0012 segundos.)

```
INSERT INTO pregunta_guardada (usuarioId, preguntaId, fecha_guardado) VALUES (8, 5, '2026-01-09 19:24:04'), (9, 1, '2026-01-09 19:24:04'), (16, 2, '2026-01-09 19:24:04'), (16, 3, '2026-01-09 19:24:04');
```

[Editar en línea] [Editar] [Código PHP]

## Tabla pregunta\_voto

```
1 INSERT INTO pregunta_voto (usuarioId, preguntaId, fecha_voto) VALUES
2 (8, 2, '2026-01-09 19:22:14'),
3 (8, 4, '2026-01-09 19:22:14'),
4 (9, 1, '2026-01-09 19:22:14'),
5 (10, 2, '2026-01-09 19:22:14'),
6 (11, 2, '2026-01-09 19:22:14'),
7 (13, 2, '2026-01-09 19:22:14'),
8 (13, 4, '2026-01-09 19:22:14'),
9 (15, 2, '2026-01-09 19:22:14'),
10 (16, 4, '2026-01-09 19:22:14'),
11 (20, 1, '2026-01-09 19:22:14');
```

✓ 11 filas insertadas. (La consulta tardó 0.0004 segundos.)

```
INSERT INTO pregunta_voto (usuarioId, preguntaId, fecha_voto) VALUES (8, 2, '2026-01-09 19:22:14'), (8, 4, '2026-01-09 19:22:14'), (9, 1, '2026-01-09 19:22:14'), (10, 2, '2026-01-09 19:22:14'), (11, 2, '2026-01-09 19:22:14'), (13, 2, '2026-01-09 19:22:14'), (13, 4, '2026-01-09 19:22:14'), (15, 2, '2026-01-09 19:22:14'), (16, 4, '2026-01-09 19:22:14'), (20, 1, '2026-01-09 19:22:14');
```

[Editar en línea] [Editar] [Código PHP]

## Tabla respuesta

```
1 INSERT INTO respuesta (id, contenido, fecha, es_solucion, preguntaId, usuarioId) VALUES
2 (1, 'La derivada es la velocidad del coche (ritmo de cambio) y la integral es la distancia total recorrida.', '2026-01-09 19:00:00', 1, 2, 14),
3 (2, 'Paramétricamente con operaciones inversas, como la multiplicación y la división.', '2026-01-09 19:30:00', 0, 2, 20),
4 (3, 'Intenta agruparlos por tema en lugar de por verbo. Por ejemplo, todos los que sirven para "viajar".', '2026-01-04 12:00:00', 0, 3, 11),
5 (4, 'Yo uso tarjetas de memoria (Flashcards) con la app Anki, funciona muy bien.', '2026-01-05 09:15:00', 0, 3, 10),
6 (5, 'Imagina un cuadrado construido sobre cada lado del triángulo. El área del cuadrado grande es igual a la suma de las áreas de los pequeños.', '2026-01-03 10:00:00', 1, 4, 24);
```

✓ 5 filas insertadas. (La consulta tardó 0.0004 segundos.)

```
INSERT INTO respuesta (id, contenido, fecha, es_solucion, preguntaId, usuarioId) VALUES (1, 'La derivada es la velocidad del coche (ritmo de cambio) y la integral es la distancia total recorrida.', '2026-01-09 19:00:00', 1, 2, 14), (2, 'Paramétricamente con operaciones inversas, como la multiplicación y la división.', '2026-01-09 19:30:00', 0, 2, 20), (3, 'Intenta agruparlos por tema en lugar de por verbo. Por ejemplo, todos los que sirven para "viajar".', '2026-01-04 12:00:00', 0, 3, 11), (4, 'Yo uso tarjetas de memoria (Flashcards) con la app Anki, funciona muy bien.', '2026-01-05 09:15:00', 0, 3, 10), (5, 'Imagina un cuadrado construido sobre cada lado del triángulo. El área del cuadrado grande es igual a la suma de las áreas de los pequeños.', '2026-01-03 10:00:00', 1, 4, 24);
```

[Editar en línea] [Editar] [Código PHP]



Tabla coleccion\_documento

```
1 INSERT INTO coleccion_documento (coleccionId, documentoId) VALUES
2 (1, 3), (1, 61), (1, 62), (1, 63), (1, 64),
3 (2, 4), (2, 6), (2, 23),
4 (3, 5), (3, 7), (3, 48),
5 (4, 8), (4, 9), (4, 55), (4, 56), (4, 57), (4, 58), (4, 59), (4, 60),
6 (5, 21), (5, 41), (5, 42), (5, 51), (5, 52),
7 (6, 20), (6, 24), (6, 73), (6, 74),
8 (7, 35), (7, 36), (7, 37), (7, 38), (7, 45), (7, 46), (7, 47),
9 (8, 43), (8, 44), (8, 54), (8, 65), (8, 66),
10 (9, 67), (9, 68), (9, 69), (9, 70),
11 (10, 2), (10, 22), (10, 77), (10, 78),
12 (11, 39), (11, 48), (11, 50),
13 (12, 49), (12, 53),
14 (13, 71), (13, 72),
15 (14, 73), (14, 74),
16 (15, 75), (15, 76);
```

15 filas insertadas. La consulta tardó 0.003 segundos.

```
INSERT INTO coleccion_documento (coleccionId, documentoId) VALUES (1, 3), (1, 61), (1, 62), (1, 63), (1, 64), (2, 4), (2, 6), (2, 23), (2, 24), (2, 25), (2, 26), (2, 27), (2, 28), (2, 29), (2, 30), (2, 31), (2, 32), (2, 33), (2, 34), (2, 35), (2, 36), (2, 37), (2, 38), (2, 39), (2, 40), (2, 41), (2, 42), (2, 43), (2, 44), (2, 45), (2, 46), (2, 47), (2, 48), (2, 49), (2, 50), (2, 51), (2, 52), (2, 53), (2, 54), (2, 55), (2, 56), (2, 57), (2, 58), (2, 59), (2, 60), (2, 61), (2, 62), (2, 63), (2, 64), (2, 65), (2, 66), (2, 67), (2, 68), (2, 69), (2, 70), (2, 71), (2, 72), (2, 73), (2, 74), (2, 75), (2, 76), (2, 77), (2, 78), (2, 79), (2, 80), (2, 81), (2, 82), (2, 83), (2, 84), (2, 85), (2, 86), (2, 87), (2, 88), (2, 89), (2, 90), (2, 91), (2, 92), (2, 93), (2, 94), (2, 95), (2, 96), (2, 97), (2, 98), (2, 99), (3, 5), (3, 6), (3, 7), (3, 8), (3, 9), (3, 10), (3, 11), (3, 12), (3, 13), (3, 14), (3, 15), (3, 16), (3, 17), (3, 18), (3, 19), (3, 20), (3, 21), (3, 22), (3, 23), (3, 24), (3, 25), (3, 26), (3, 27), (3, 28), (3, 29), (3, 30), (3, 31), (3, 32), (3, 33), (3, 34), (3, 35), (3, 36), (3, 37), (3, 38), (3, 39), (3, 40), (3, 41), (3, 42), (3, 43), (3, 44), (3, 45), (3, 46), (3, 47), (3, 48), (3, 49), (3, 50), (3, 51), (3, 52), (3, 53), (3, 54), (3, 55), (3, 56), (3, 57), (3, 58), (3, 59), (3, 60), (3, 61), (3, 62), (3, 63), (3, 64), (3, 65), (3, 66), (3, 67), (3, 68), (3, 69), (3, 70), (3, 71), (3, 72), (3, 73), (3, 74), (3, 75), (3, 76), (3, 77), (3, 78), (3, 79), (3, 80), (3, 81), (3, 82), (3, 83), (3, 84), (3, 85), (3, 86), (3, 87), (3, 88), (3, 89), (3, 90), (3, 91), (3, 92), (3, 93), (3, 94), (3, 95), (3, 96), (3, 97), (3, 98), (3, 99), (4, 8), (4, 9), (4, 10), (4, 11), (4, 12), (4, 13), (4, 14), (4, 15), (4, 16), (4, 17), (4, 18), (4, 19), (4, 20), (4, 21), (4, 22), (4, 23), (4, 24), (4, 25), (4, 26), (4, 27), (4, 28), (4, 29), (4, 30), (4, 31), (4, 32), (4, 33), (4, 34), (4, 35), (4, 36), (4, 37), (4, 38), (4, 39), (4, 40), (4, 41), (4, 42), (4, 43), (4, 44), (4, 45), (4, 46), (4, 47), (4, 48), (4, 49), (4, 50), (4, 51), (4, 52), (4, 53), (4, 54), (4, 55), (4, 56), (4, 57), (4, 58), (4, 59), (4, 60), (4, 61), (4, 62), (4, 63), (4, 64), (4, 65), (4, 66), (4, 67), (4, 68), (4, 69), (4, 70), (4, 71), (4, 72), (4, 73), (4, 74), (4, 75), (4, 76), (4, 77), (4, 78), (4, 79), (4, 80), (4, 81), (4, 82), (4, 83), (4, 84), (4, 85), (4, 86), (4, 87), (4, 88), (4, 89), (4, 90), (4, 91), (4, 92), (4, 93), (4, 94), (4, 95), (4, 96), (4, 97), (4, 98), (4, 99), (5, 21), (5, 22), (5, 23), (5, 24), (5, 25), (5, 26), (5, 27), (5, 28), (5, 29), (5, 30), (5, 31), (5, 32), (5, 33), (5, 34), (5, 35), (5, 36), (5, 37), (5, 38), (5, 39), (5, 40), (5, 41), (5, 42), (5, 43), (5, 44), (5, 45), (5, 46), (5, 47), (5, 48), (5, 49), (5, 50), (5, 51), (5, 52), (5, 53), (5, 54), (5, 55), (5, 56), (5, 57), (5, 58), (5, 59), (5, 60), (5, 61), (5, 62), (5, 63), (5, 64), (5, 65), (5, 66), (5, 67), (5, 68), (5, 69), (5, 70), (5, 71), (5, 72), (5, 73), (5, 74), (5, 75), (5, 76), (5, 77), (5, 78), (5, 79), (5, 80), (5, 81), (5, 82), (5, 83), (5, 84), (5, 85), (5, 86), (5, 87), (5, 88), (5, 89), (5, 90), (5, 91), (5, 92), (5, 93), (5, 94), (5, 95), (5, 96), (5, 97), (5, 98), (5, 99), (6, 20), (6, 21), (6, 22), (6, 23), (6, 24), (6, 25), (6, 26), (6, 27), (6, 28), (6, 29), (6, 30), (6, 31), (6, 32), (6, 33), (6, 34), (6, 35), (6, 36), (6, 37), (6, 38), (6, 39), (6, 40), (6, 41), (6, 42), (6, 43), (6, 44), (6, 45), (6, 46), (6, 47), (6, 48), (6, 49), (6, 50), (6, 51), (6, 52), (6, 53), (6, 54), (6, 55), (6, 56), (6, 57), (6, 58), (6, 59), (6, 60), (6, 61), (6, 62), (6, 63), (6, 64), (6, 65), (6, 66), (6, 67), (6, 68), (6, 69), (6, 70), (6, 71), (6, 72), (6, 73), (6, 74), (6, 75), (6, 76), (6, 77), (6, 78), (6, 79), (6, 80), (6, 81), (6, 82), (6, 83), (6, 84), (6, 85), (6, 86), (6, 87), (6, 88), (6, 89), (6, 90), (6, 91), (6, 92), (6, 93), (6, 94), (6, 95), (6, 96), (6, 97), (6, 98), (6, 99), (7, 35), (7, 36), (7, 37), (7, 38), (7, 39), (7, 40), (7, 41), (7, 42), (7, 43), (7, 44), (7, 45), (7, 46), (7, 47), (7, 48), (7, 49), (7, 50), (7, 51), (7, 52), (7, 53), (7, 54), (7, 55), (7, 56), (7, 57), (7, 58), (7, 59), (7, 60), (7, 61), (7, 62), (7, 63), (7, 64), (7, 65), (7, 66), (7, 67), (7, 68), (7, 69), (7, 70), (7, 71), (7, 72), (7, 73), (7, 74), (7, 75), (7, 76), (7, 77), (7, 78), (7, 79), (7, 80), (7, 81), (7, 82), (7, 83), (7, 84), (7, 85), (7, 86), (7, 87), (7, 88), (7, 89), (7, 90), (7, 91), (7, 92), (7, 93), (7, 94), (7, 95), (7, 96), (7, 97), (7, 98), (7, 99), (8, 43), (8, 44), (8, 45), (8, 46), (8, 47), (8, 48), (8, 49), (8, 50), (8, 51), (8, 52), (8, 53), (8, 54), (8, 55), (8, 56), (8, 57), (8, 58), (8, 59), (8, 60), (8, 61), (8, 62), (8, 63), (8, 64), (8, 65), (8, 66), (8, 67), (8, 68), (8, 69), (8, 70), (8, 71), (8, 72), (8, 73), (8, 74), (8, 75), (8, 76), (8, 77), (8, 78), (8, 79), (8, 80), (8, 81), (8, 82), (8, 83), (8, 84), (8, 85), (8, 86), (8, 87), (8, 88), (8, 89), (8, 90), (8, 91), (8, 92), (8, 93), (8, 94), (8, 95), (8, 96), (8, 97), (8, 98), (8, 99), (9, 67), (9, 68), (9, 69), (9, 70), (9, 71), (9, 72), (9, 73), (9, 74), (9, 75), (9, 76), (9, 77), (9, 78), (9, 79), (9, 80), (9, 81), (9, 82), (9, 83), (9, 84), (9, 85), (9, 86), (9, 87), (9, 88), (9, 89), (9, 90), (9, 91), (9, 92), (9, 93), (9, 94), (9, 95), (9, 96), (9, 97), (9, 98), (9, 99), (10, 2), (10, 3), (10, 4), (10, 5), (10, 6), (10, 7), (10, 8), (10, 9), (10, 10), (10, 11), (10, 12), (10, 13), (10, 14), (10, 15), (10, 16), (10, 17), (10, 18), (10, 19), (10, 20), (10, 21), (10, 22), (10, 23), (10, 24), (10, 25), (10, 26), (10, 27), (10, 28), (10, 29), (10, 30), (10, 31), (10, 32), (10, 33), (10, 34), (10, 35), (10, 36), (10, 37), (10, 38), (10, 39), (10, 40), (10, 41), (10, 42), (10, 43), (10, 44), (10, 45), (10, 46), (10, 47), (10, 48), (10, 49), (10, 50), (10, 51), (10, 52), (10, 53), (10, 54), (10, 55), (10, 56), (10, 57), (10, 58), (10, 59), (10, 60), (10, 61), (10, 62), (10, 63), (10, 64), (10, 65), (10, 66), (10, 67), (10, 68), (10, 69), (10, 70), (10, 71), (10, 72), (10, 73), (10, 74), (10, 75), (10, 76), (10, 77), (10, 78), (10, 79), (10, 80), (10, 81), (10, 82), (10, 83), (10, 84), (10, 85), (10, 86), (10, 87), (10, 88), (10, 89), (10, 90), (10, 91), (10, 92), (10, 93), (10, 94), (10, 95), (10, 96), (10, 97), (10, 98), (10, 99), (11, 39), (11, 40), (11, 41), (11, 42), (11, 43), (11, 44), (11, 45), (11, 46), (11, 47), (11, 48), (11, 49), (11, 50), (11, 51), (11, 52), (11, 53), (11, 54), (11, 55), (11, 56), (11, 57), (11, 58), (11, 59), (11, 60), (11, 61), (11, 62), (11, 63), (11, 64), (11, 65), (11, 66), (11, 67), (11, 68), (11, 69), (11, 70), (11, 71), (11, 72), (11, 73), (11, 74), (11, 75), (11, 76), (11, 77), (11, 78), (11, 79), (11, 80), (11, 81), (11, 82), (11, 83), (11, 84), (11, 85), (11, 86), (11, 87), (11, 88), (11, 89), (11, 90), (11, 91), (11, 92), (11, 93), (11, 94), (11, 95), (11, 96), (11, 97), (11, 98), (11, 99), (12, 49), (12, 50), (12, 51), (12, 52), (12, 53), (12, 54), (12, 55), (12, 56), (12, 57), (12, 58), (12, 59), (12, 60), (12, 61), (12, 62), (12, 63), (12, 64), (12, 65), (12, 66), (12, 67), (12, 68), (12, 69), (12, 70), (12, 71), (12, 72), (12, 73), (12, 74), (12, 75), (12, 76), (12, 77), (12, 78), (12, 79), (12, 80), (12, 81), (12, 82), (12, 83), (12, 84), (12, 85), (12, 86), (12, 87), (12, 88), (12, 89), (12, 90), (12, 91), (12, 92), (12, 93), (12, 94), (12, 95), (12, 96), (12, 97), (12, 98), (12, 99), (13, 71), (13, 72), (13, 73), (13, 74), (13, 75), (13, 76), (13, 77), (13, 78), (13, 79), (13, 80), (13, 81), (13, 82), (13, 83), (13, 84), (13, 85), (13, 86), (13, 87), (13, 88), (13, 89), (13, 90), (13, 91), (13, 92), (13, 93), (13, 94), (13, 95), (13, 96), (13, 97), (13, 98), (13, 99), (14, 73), (14, 74), (14, 75), (14, 76), (14, 77), (14, 78), (14, 79), (14, 80), (14, 81), (14, 82), (14, 83), (14, 84), (14, 85), (14, 86), (14, 87), (14, 88), (14, 89), (14, 90), (14, 91), (14, 92), (14, 93), (14, 94), (14, 95), (14, 96), (14, 97), (14, 98), (14, 99), (15, 75), (15, 76), (15, 77), (15, 78), (15, 79), (15, 80), (15, 81), (15, 82), (15, 83), (15, 84), (15, 85), (15, 86), (15, 87), (15, 88), (15, 89), (15, 90), (15, 91), (15, 92), (15, 93), (15, 94), (15, 95), (15, 96), (15, 97), (15, 98), (15, 99).
```

15 filas insertadas. La consulta tardó 0.003 segundos.

15 filas insertadas. La consulta tardó 0.003 segundos.

Tabla usuario\_coleccion

```
1 INSERT INTO usuario_coleccion (usuarioId, coleccionId, fecha_union) VALUES
2 (16, 1, '2026-01-10 10:00:00'), (16, 2, '2026-01-10 10:05:00'),
3 (16, 10, '2026-01-10 10:10:00'), (16, 11, '2026-01-10 10:15:00'),
4 (8, 2, '2026-01-09 09:00:00'), (8, 3, '2026-01-09 09:30:00'),
5 (8, 12, '2026-01-09 10:00:00'),
6 (9, 6, '2026-01-08 11:00:00'), (9, 7, '2026-01-08 11:30:00'),
7 (9, 14, '2026-01-08 12:00:00'),
8 (12, 5, '2026-01-07 15:00:00'), (12, 8, '2026-01-07 15:30:00'),
9 (12, 9, '2026-01-07 16:00:00'), (12, 15, '2026-01-07 16:30:00');
```

14 filas insertadas. La consulta tardó 0.003 segundos.

```
INSERT INTO usuario_coleccion (usuarioId, coleccionId, fecha_union) VALUES (16, 1, '2026-01-10 10:00:00'), (16, 2, '2026-01-10 10:05:00'), (16, 10, '2026-01-10 10:10:00'), (16, 11, '2026-01-10 10:15:00'), (8, 2, '2026-01-09 09:00:00'), (8, 3, '2026-01-09 09:30:00'), (8, 12, '2026-01-09 10:00:00'), (9, 6, '2026-01-08 11:00:00'), (9, 7, '2026-01-08 11:30:00'), (9, 14, '2026-01-08 12:00:00'), (12, 5, '2026-01-07 15:00:00'), (12, 8, '2026-01-07 15:30:00'), (12, 9, '2026-01-07 16:00:00'), (12, 15, '2026-01-07 16:30:00').
```

14 filas insertadas. La consulta tardó 0.003 segundos.

Tabla Cliente-Empresa

```
1 INSERT INTO clienteempresa (id, razon_social, cif, usuarioId) VALUES
2 (1, 'Tech Solutions S.L.', 'B999999991', 5),
3 (2, 'Consultora Global S.A.', 'A888888882', 6),
4 (3, 'Innova StartUp', 'B777777773', 7),
5 (4, 'TechSoft Development', 'B12345678', 21),
6 (5, 'Construcciones Civiles Norte', 'A98765432', 22),
7 (6, 'Estudio de Diseño Marta', 'B11223344', 25),
8 (7, 'Hospital General Universitario', 'Q2812345G', 26),
9 (8, 'Editorial El Saber S.L.', 'B55443322', 27);
```

8 filas insertadas. La consulta tardó 0.004 segundos.

```
INSERT INTO clienteempresa (id, razon_social, cif, usuarioId) VALUES (1, 'Tech Solutions S.L.', 'B999999991', 5), (2, 'Consultora Global S.A.', 'A888888882', 6), (3, 'Innova StartUp', 'B777777773', 7), (4, 'TechSoft Development', 'B12345678', 21), (5, 'Construcciones Civiles Norte', 'A98765432', 22), (6, 'Estudio de Diseño Marta', 'B11223344', 25), (7, 'Hospital General Universitario', 'Q2812345G', 26), (8, 'Editorial El Saber S.L.', 'B55443322', 27).
```

8 filas insertadas. La consulta tardó 0.004 segundos.

Tabla Contratación

```
1 INSERT INTO Contratacion (clienteId, servicioId, fecha) VALUES
2 (1, 2, '2023-09-01 10:00:00'),
3 (1, 3, '2024-01-15 09:30:00'),
4 (2, 4, '2024-02-01 11:00:00'),
5 (3, 2, '2024-03-10 16:45:00');
```

4 filas insertadas. La 1a de las insertadas es: 6. La consulta tardó 0.000 segundos.

```
INSERT INTO Contratacion (clienteId, servicioId, fecha) VALUES (1, 2, '2023-09-01 10:00:00'), (1, 3, '2024-01-15 09:30:00'), (2, 4, '2024-02-01 11:00:00'), (3, 2, '2024-03-10 16:45:00');
```

4 filas insertadas. La consulta tardó 0.000 segundos.

23



## Tabla Evento

```
1 INSERT INTO evento (id, titulo, fecha_inicio, institucionId) VALUES
2 (1, 'Examen Parcial Álgebra', '2024-05-15 09:00:00', 1),
3 (2, 'Feria de Empleo UPM', '2024-06-20 10:00:00', 1),
4 (3, 'Webinar: React vs Vue', '2024-04-10 18:00:00', 2),
5 (4, 'Hackathon de Seguridad', '2024-11-05 09:00:00', 2),
6 (5, 'Open Day: English for Business', '2024-09-01 17:00:00', 3),
7 (6, 'Simposio de Biotecnología', '2026-03-15 09:00:00', 4),
8 (7, 'Exposición de Arte Digital', '2026-04-20 18:00:00', 5),
9 (8, 'Hackathon AI for Good', '2026-05-10 08:00:00', 6),
10 (9, 'Networking Empresarial', '2026-02-28 19:00:00', 7),
11 (10, 'Simposio de Biotecnología', '2026-03-15 09:00:00', 4),
12 (11, 'Exposición de Arte Digital', '2026-04-20 18:00:00', 5),
13 (12, 'Hackathon AI for Good', '2026-05-10 08:00:00', 6),
14 (13, 'Networking Empresarial', '2026-02-28 19:00:00', 7),
15 (14, 'Simposio de Biotecnología', '2026-03-15 09:00:00', 4),
16 (15, 'Exposición de Arte Digital', '2026-04-20 18:00:00', 5),
17 (16, 'Hackathon AI for Good', '2026-05-10 08:00:00', 6),
18 (17, 'Networking Empresarial', '2026-02-28 19:00:00', 7);
```



## Tabla Incidencia

```
1 INSERT INTO incidencia (id, descripcion, estado, usuarioId) VAL
2 (1, 'No puedo restablecer mi contraseña', 'cerrada', 8),
3 (2, 'El PDF de React da error 404 al descargar', 'abierta', 9),
4 (3, 'Quiero factura del último curso contratado', 'abierta', 5),
5 (4, 'Error al subir documento nuevo', 'abierta', 2),
6 (5, 'La plataforma va lenta por las tardes', 'abierta', 10),
7 (6, 'No puedo acceder al curso de Genética', 'abierta', 19),
8 (7, 'Error en la pasarela de pago', 'cerrada', 21),
9 (8, 'Solicito cambio de contraseña manual', 'abierta', 20),
10 (9, 'Solicito reembolso del curso de Java', 'cerrada', 11),
11 (10, 'No me llegó el correo de confirmación', 'abierta', 13),
12 (11, 'Error visual en el menú de perfil', 'abierta', 16);
```



## Implementar base de datos

En phpMyAdmin, al intentar implementar el archivo [sql/divulgacion\\_cientifica.sql](#), da el siguiente error:



Para solucionarlo, se edita el archivo `/etc/php/8.3/apache2/php.ini` con el siguiente comando: `sudo nano /etc/php/8.3/apache2/php.ini`. Tras ejecutar el comando, se abre el archivo y en él hay que buscar la línea `upload_max_filesize = 2M`, que se cambiará a `128M`, y `post_max_size = 8M`, que se cambiará a `128M`.

```

GNU nano 2.7.2 /etc/php/8.3/apache2/php.ini
; Note that the line below will be ignored if the directive is turned on.
; https://php.net/cgi-check-uidbug-line
;cgi-check_uidbug_line=1

;
; File Uploads
;
; Whether to allow HTTP file uploads.
; https://php.net/file-uploads
file_uploads = On

; Temporary directory for HTTP uploaded files (will use system default if not
; specified).
; https://php.net/upload-temp-dir
upload_tmp_dir =

; Maximum allowed size for uploaded files.
; https://php.net/upload-max-filesize
upload_max_filesize = 128M

; Maximum number of files that can be uploaded via a single request
max_file_uploads = 20

;
; Fopen wrappers
;
; Whether to allow the treatment of URLs (like http:// or ftp://) as files.
; https://php.net/allow-url-fopen

```

```

GNU nano 2.7.2 /etc/php/8.3/apache2/php.ini *
; Maximum size of POST data that PHP will accept.
; Its value may be 0 to disable the limit. It is ignored if POST data reading
; is disabled through enable_post_data_reading.
; https://php.net/post-max-size
post_max_size = 128M

; Automatically add files before PHP document.
; https://php.net/auto-prepend-file
auto_prepend_file =

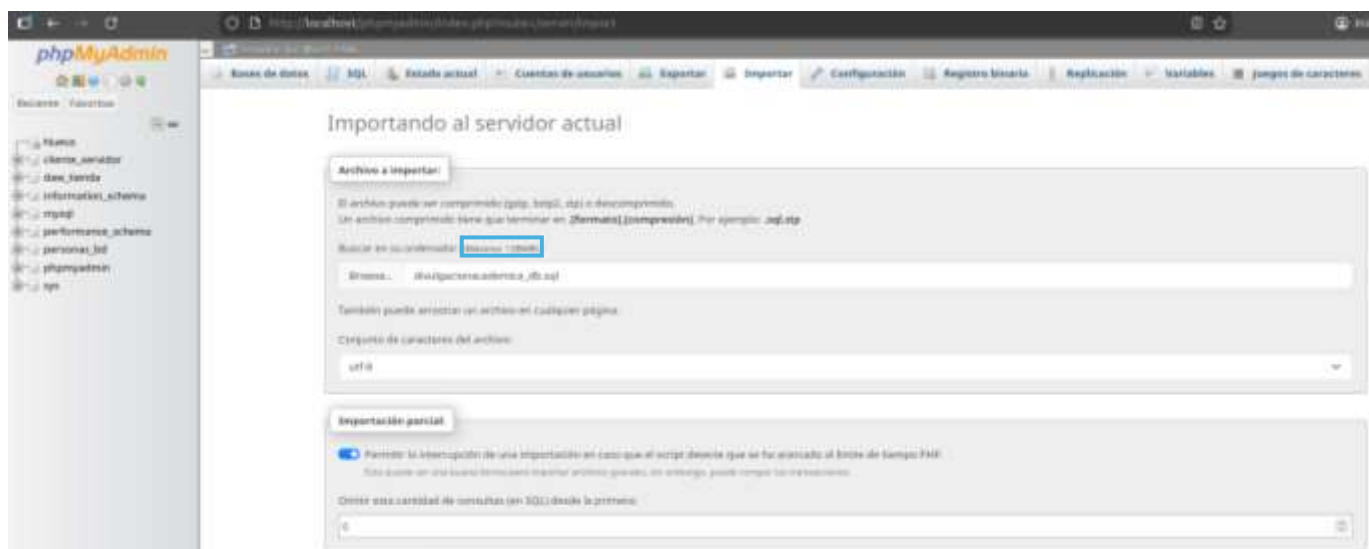
; Automatically add files after PHP document.
; https://php.net/auto-append-file
auto_append_file =

; By default, PHP will output a media type using the Content-Type header. To
; disable this, simply set it to be empty.
; PHP's built-in default media type is set to text/html.
; https://php.net/default-mimetype

```

Una vez guardados los cambios, se ejecuta el comando `sudo systemctl restart apache2` para reiniciar Apache y que los cambios tengan efecto.

Si se vuelve a intentar importar la base de datos en phpMyAdmin, se verá que el tamaño máximo por archivo es 128 MB.



## BACKEND

### Colecciones

El módulo de Colecciones permite a los usuarios (principalmente alumnos) agrupar diversos recursos académicos bajo un eje temático común, gestionar su visibilidad y permitir que otros usuarios se unan a estas agrupaciones para el aprendizaje colaborativo.

### 1. Características Principales

- ❖ **Gestión de Agrupaciones:** creación, edición y eliminación de colecciones temáticas.
- ❖ **Control de Privacidad:** soporte para colecciones **Públicas** (visibles para todos) y **Privadas** (solo para el autor).
- ❖ **Sistema de Membresía:** los alumnos pueden unirse a colecciones de otros para seguir sus materiales.
- ❖ **Gestión de Materiales:** vinculación dinámica de documentos existentes a una o varias colecciones propias.

- ❖ **Exportación a PDF:** generación automática de un catálogo de la colección en formato PDF mediante la librería Dompdf.
- ❖ **Buscador en Tiempo Real:** filtrado dinámico de colecciones y usuarios mediante JavaScript en las vistas de administración y exploración.

## 2. Modelo de Datos y Relaciones

El componente central es el modelo **Colección**, que interactúa con las siguientes entidades:

- ❖ **Usuario (Autor):** relación muchos a uno. Una colección pertenece a un único creador (**usuarioId**).
- ❖ **Documentos:** relación muchos a muchos gestionada a través de la tabla intermedia **coleccion\_documento**.
- ❖ **Miembros (Usuarios Unidos):** relación muchos a muchos a través de la tabla **usuario\_coleccion**.
- ❖ **Contador de Descargas:** atributo persistente que rastrea cuántas veces se ha generado el PDF de la colección.

## 3. Seguridad y Control de Acceso

El acceso está estrictamente regulado mediante **AccessControl** en el **ColeccionController** y validaciones manuales de propiedad.

### Permisos por Rol

ACCIÓN	INVITADO	ALUMNO	GESTOR	ADMINISTRADOR
Index (Explorar)	Solo Públicas	Públicas + Propias	Vista Admin	Vista Admin
View (Ver detalle)	Sí	Sí	Sí	Sí
Create	No	Sí	Sí	Sí
Update / Quitar Material	No	Solo Propias	No	Sí
Unirse / Abandonar	No	Sí	Sí	Sí
Usuarios (Gestionar)	No	No	No	Sí
Delete	No	Solo Propias	No	Sí

### Reglas de Validación Críticas

- ❖ **Propiedad:** solo el autor original o el administrador puede editar o eliminar materiales de una colección específica.
- ❖ **Regla de Abandono:** un autor no puede abandonar su propia colección para evitar que esta quede "huérfana".
- ❖ **Eliminación Automática:** si el último miembro abandona una colección, el sistema la elimina automáticamente de la base de datos para mantener la limpieza del sistema.

## 4. Flujos de Trabajo

### Creación y Auto Unión

Al crear una nueva colección, el sistema no solo guarda el registro, sino que ejecuta una "unión automática" insertando al creador en la tabla `usuario_coleccion`. Esto permite que el autor aparezca en la lista de miembros y gestione su contenido como un usuario activo.

### Gestión de Materiales

Un usuario puede navegar a cualquier Documento y, mediante la acción `agregar-material`, ver una lista de las colecciones que ha creado. El sistema permite la selección múltiple y evita duplicados mediante una verificación de existencia previa antes de insertar en `coleccion_documento`.

### Generación de PDF

El sistema utiliza `Dompdf` para transformar vistas parciales en documentos descargables:

1. Se renderiza la plantilla `_pdf_template.php` con los datos del modelo.
2. Se configura el papel en tamaño A4 y orientación vertical.
3. Se incrementa el contador de descargas del modelo de forma atómica mediante `updateCounters()`.
4. El archivo se envía al navegador con el nombre del título de la colección (sanitizado con guiones bajos).

## models/Documento.php

El archivo `Documento.php` representa la entidad central de la aplicación. Actúa como una capa de abstracción entre el controlador y la tabla documento de la base de datos MySQL, ya que cualquier información pasa antes por este documento. Tiene como objetivo asegurar que la información sea válida y coherente antes de interactuar con la Base de Datos.

La variable `$archivoFile` se usa para recibir o manejar archivos.

### Reglas de validación (rules)

Antes de guardar cualquier cosa, el framework ejecuta estas reglas.

```
public function rules()
{
    return [
        [['titulo', 'materialId', 'institucionId'], 'required'],
        [['materialId', 'autorId', 'institucionId', 'usuario_subida_id'], 'integer'],
        [['tipo_acceso'], 'string'],
        [['archivo_url'], 'string', 'max' => 255],

        [['archivoFile'], 'file',
            'skipOnEmpty' => false,
            'extensions' => 'pdf, doc, docx, xls,xlsx, ppt, pptx, mp4',
            'maxSize' => 1024 * 1024 * 200 // límite subido para los videos
        ],
    ];
}
```

- ❖ **Campos Requeridos (required)**: impide que se creen documentos sin título o sin asignar a una materia.
- ❖ **Tipos de Datos (integer, string)**: asegura la integridad referencial. Por ejemplo, `materialId` debe ser un número entero para poder enlazar correctamente con la tabla de materias.
- ❖ **Validación de Archivos (file)**:

- **extensions:** implementa una lista blanca de formatos, permitiendo en este caso solo archivos pdf, doc, docx, xls, xlsx, ppt, pptx, mp4. Cualquier otro formato (como .exe o .php) será bloqueado.
- **maxSize:** establece una cuota de almacenamiento por archivo de 200 MB. Este tamaño ha sido aumentado por la posibilidad de subir videos.

```
public function attributeLabels()
{
    return [
        'titulo' => 'Título del Documento',
        'materiaId' => 'Asignatura / Materia',
        'autorId' => 'Autor Original (Profesor)',
        'institucionId' => 'Institución',
        'tipo_acceso' => 'Visibilidad',
        'archivoFile' => 'Archivo',
    ];
}
```

Este método es esencial ya que permite mantener nombres de variables limpios y técnicos en el backend, ofreciendo una interfaz descriptiva y profesional en el frontend. Pasando de mostrar al usuario “titulo” a “Título del Documento”.

```
// --- Relaciones ---
public function getMateria() { return $this->hasOne(Materia::class, ['id' => 'materiaId']); }
public function getAutor() { return $this->hasOne(Personal::class, ['id' => 'autorId']); }
public function getInstitucion() { return $this->hasOne(Institucion::class, ['id' => 'institucionId']); }

// Funciones extra
public function getComentarios() { return $this->hasMany(DocumentoComentario::class, ['documentId' => 'id'], orderBy(['fecha' => SORT_DESC]); }
public function getValoraciones() { return $this->hasMany(DocumentoValoracion::class, ['documentId' => 'id']); }
```

Estos métodos definen cómo se conecta este documento con el resto del sistema, permitiendo navegar por los datos:

- ❖ **Relaciones hasOne (Uno a Uno):** un documento pertenece a una Materia y a un Autor (Profesor), permite acceder al nombre de la materia con `$documento->materia->nombre`.
- ❖ **Relaciones hasMany (Uno a Muchos):** un documento tiene múltiples Comentarios y Valoraciones, facilita obtener todos los comentarios de los usuarios asociado a un archivo.

```
public function getRatingPromedio() {
    $total = $this->getValoraciones()->count();
    if ($total == 0) return 0;
    return round($this->getValoraciones()->sum('puntuacion') / $total, 1);
}
```

Esta función no guarda un dato estático, si no que hace un cálculo en tiempo real.

1. Cuenta cuántos votos tiene el documento.
2. Suma las puntuaciones.
3. Realiza la división para obtener la media.
4. Redondea el resultado a 1 decimal.

## **controllers/MaterialesController.php**

Este componente centraliza la lógica de cuatro procesos críticos: el buscador facetado, la visualización detallada, el sistema de votación y la gestión de subidas de archivos.



```
// 1. CONSULTA BASE
$queryBase = Documento::find()->joinWith(['materia', 'autor', 'institucion']);

// A. TEXTO
if ($q = $request->get('q')) {
    $queryBase->andWhere(['or',
        ['like', 'documento.titulo', $q],
        ['like', 'materia.nombre', $q],
        ['like', 'personal.nombre', $q],
        ['like', 'institucion.nombre', $q]
    ]);
}

// B. TIPO Y ACCESO
$types = $request->get('type');
$access = $request->get('access');
$filtrosAcceso = [];

if (!empty($types) && is_array($types)) {
    if (in_array('apuntes', $types)) $filtrosAcceso[] = 'publico';
    if (in_array('exámenes', $types)) $filtrosAcceso[] = 'privado';
}

if (!empty($access) && is_array($access)) {
    $accessLimpios = array_filter($access);
    if (!empty($accessLimpios)) {
        $filtrosAcceso = array_merge($filtrosAcceso, $accessLimpios);
    }
}

if (!empty($filtrosAcceso)) {
    $queryBase->andWhere(['in', 'documento.tipo_acceso', array_unique($filtrosAcceso)]);
}
```

El código evalúa los parámetros **GET** (q, type). Implementa una búsqueda recursiva. Si el usuario selecciona una categoría "Padre" como "Ciencias", el sistema busca automáticamente todas sus sub-categorías para incluir también esos documentos.

```
// VER DETALLE Y PROCESAR COMENTARIOS
public function actionView($id)
{
    $model = Documento::findOne($id);
    if (!$model) throw new \yii\web\NotFoundException("El documento no existe.");

    $nuevoComentario = new DocumentoComentario();

    // Solo se procesa el comentario si el usuario ha iniciado sesión
    if (!Yii::$app->user->isGuest && $nuevoComentario->load(Yii::$app->request->post())) {

        $nuevoComentario->documentoId = $model->id;
        $nuevoComentario->usuarioId = Yii::$app->user->id; // ID real del usuario
        $nuevoComentario->fecha = date('Y-m-d H:i:s');

        if ($nuevoComentario->save()) {
            Yii::$app->session->setFlash('success', '¡Comentario publicado con éxito!');
            return $this->refresh();
        } else {
            Yii::$app->session->setFlash('error', 'Error al guardar el comentario.');
```

En **actionView**, detecta si la petición es de tipo **POST**. Si un usuario registrado envía un comentario, el controlador captura su ID de sesión (**Yii::\$app->user->id**), lo vincula al comentario y lo guarda.

```
// ACCIÓN PARA GUARDAR VOTOS
public function actionValorar($id, $puntuacion)
{
    // Si no está logueado, se va al login
    if (Yii::$app->user->isGuest) {
        Yii::$app->session->setFlash('error', 'Debes iniciar sesión para valorar documentos.');
```

return \$this->redirect(['site/login']);

```
    }

    $userId = Yii::$app->user->id;

    $valoracion = DocumentoValoracion::findOne(['usuarioId' => $userId, 'documentoId' => $id]);

    if (!$valoracion) {
        $valoracion = new DocumentoValoracion();
        $valoracion->usuarioId = $userId;
        $valoracion->documentoId = $id;

        $valoracion->puntuacion = (int)$puntuacion;
        $valoracion->fecha = date('Y-m-d H:i:s');

        if($valoracion->save()) {
            Yii::$app->session->setFlash('success', '¡Gracias por valorar el documento!');
        }

        return $this->redirect(['view', 'id' => $id]);
    }
}
```

La función **actionValorar** comprueba primero si ya existe un voto de ese usuario para ese documento. Si existe actualiza la puntuación antigua, si no existe crea un nuevo registro; así se evita que el mismo usuario vote varias veces.

```
// ACCIÓN PARA SUBIR NUEVO DOCUMENTO
public function actionSubir()
{
    $model = new Documento();

    // Datos para los desplegables del formulario
    $materias = ArrayHelper::map(Materia::find()->orderBy('nombre')->all(), 'id', 'nombre');
    $instituciones = ArrayHelper::map($app\models\Institucion::find()->orderBy('nombre')->all(), 'id', 'nombre');
    $autores = ArrayHelper::map($app\models\Personal::find()->orderBy('nombre')->all(), 'id', function($p){ return $p->nombre . ' ' . $p->apellidos; });

    if ($request = Yii::$app->request->post()) {
        // 1. Cargar datos del formulario
        if ($model->load($request)) {
            // 2. Obtener la instancia del archivo subido
            $model->archivoFile = UploadedFile::getInstance($model, 'archivoFile');

            if ($model->archivoFile && $model->validate()) {
                // 3. Generar nombre único
                // Evita problemas si dos personas suben el mismo archivo
                $nombreUnico = preg_replace('/[^\wA-Z0-9]/', '_', strtolower($model->titulo)) . '_' . time() . '_' . $model->archivoFile->extension;

                // 4. Mover el archivo a la carpeta 'web/uploads'
                $rutaGuardada = 'uploads/' . $nombreUnico;

                if ($model->archivoFile->saveAs($rutaGuardada)) {
                    // 5. Guardar solo el nombre en la base de datos
                    $model->archivo_url = $nombreUnico;

                    // Guardar el registro completo
                    if ($model->save(false)) { // false porque ya se validó antes
                        Yii::$app->session->setFlash('success', 'Documento subido correctamente!');
                        return $this->redirect(['view', 'id' => $model->id]);
                    }
                }
            }
        }
    }
}
```

Se encarga de:

- ❖ Preparar y mostrar el formulario con los datos necesarios (listas de materias, autores, etc.).
- ❖ Recibir los datos, validar la seguridad del archivo, guardarlo físicamente en el disco y registrar sus metadatos en la base de datos.

Se utiliza **ArrayHelper::map**. Esta función transforma una colección de objetos (ej: todas las materias de la BD) en un array asociativo simple [**id => nombre**]. Porque la vista lo necesita así para poder generar las etiquetas **<option value="ID">NOMBRE</option>** dentro del formulario HTML.

**UploadedFile::getInstance**: Convierte el archivo en un objeto manipulable con propiedades como extension, size, tempName, etc.

**\$model->validate()**: Ejecuta las reglas definidas en Documento.php (las extensiones y el peso). Si algo falla, el código se detiene aquí y muestra los errores al usuario.

**preg\_replace:** Limpia el título del documento, eliminando espacios, tildes o caracteres raros que podrían romper la URL.

**time():** Añade la marca de tiempo actual (timestamp).

**saveAs(...):** Mueve el archivo desde la carpeta temporal de PHP a la carpeta **pública** /web/uploads.

## FRONTEND

### CSS

Primero, se creó el archivo de diseño ya existente (**styles.css**) dentro de la carpeta pública de la web. Luego, se configuró el archivo **AppAsset.php**.

```
public $css = [
    '//css/site.css', // Estilos base
    'css/styles.css', // Estilos página
];
```

Además, se hicieron cambios en el archivo **main.php**, de forma que se borró la estructura predefinida de Yii:

- ❖ Se quitó la barra de navegación automática (Widgets).
- ❖ Se eliminó los contenedores que limitaban el ancho de la página.

### Construcción de la Portada (site/index.php)

La página de inicio mostraba un mensaje de bienvenida genérico, por lo que se trasladó todo el código HTML del diseño original a la vista de portada de Yii.

- ❖ Se recreó el Buscador con su lupa y sus despleables.
- ❖ Se hicieron las Tarjetas de Categorías (Matemáticas, Física, etc.).
- ❖ Se añadieron las secciones de "Últimas subidas" y "Estadísticas".
- ❖ Se cambiaron los enlaces normales (**<a href="html">**) por enlaces de Yii (**Url::to**).

## VISTA MATERIALES

### Documento.php (Modelo Principal)

Es el núcleo del módulo. Representa la tabla documento y recupera los datos de los archivos (título, ruta, tipo de acceso).

- ❖ **getMateria():** conecta con la tabla **materia** para saber de qué asignatura es el archivo.
- ❖ **getAutor():** conecta con la tabla **personal** para obtener el nombre del profesor.
- ❖ **getInstitucion():** conecta con la tabla **institucion** para saber a qué centro pertenece.

### Materia.php, Personal.php, Institucion.php (Modelos Auxiliares)

Representan las tablas (**materia**, **personal**, **institucion**). Sirven como catálogos. Permiten que, al filtrar por "Matemáticas", el sistema pueda traducir esos nombres a los IDs correspondientes en la tabla de documentos.



## Capa Lógica (Controlador)

El archivo [codigo/controllers/MaterialesController.php](#).

### Método `actionIndex()`

Este método maneja todas las peticiones de la página de materiales.

- ❖ **Búsqueda por texto:** filtra por coincidencias parciales en el título del documento (`LIKE %texto%`).
- ❖ **Filtro por categoría (Materias):**
  - El usuario selecciona categorías visuales (Matemáticas, Física, etc.).
  - El controlador realiza un `JOIN` con la tabla materia y filtra los resultados cuyo nombre coincida con la selección.
- ❖ **Filtro por Tipo de Material:**
  - **Desafío:** la interfaz visual solicita múltiples tipos (Apuntes, Exámenes, Prácticas...), pero la base de datos actual solo distingue entre público y privado.
  - **Solución Lógica:** se ha implementado un mapeo en el controlador:
    - Selección "Apuntes" -> Busca documentos público.
    - Selección "Exámenes" -> Busca documentos privado.
- ❖ **Paginación:** se gestiona la carga de datos mediante un `ActiveDataProvider`, limitando el número de tarjetas por página para optimizar el rendimiento.

## Desarrollo del Frontend (Vista y Diseño)

La vista [views/materiales/index.php](#) ha sido construida para ser igual al prototipo HTML que ya teníamos.

### Barra Lateral de Filtros (<aside>)

Se ha construido un formulario HTML integrado con PHP que incluye los tres bloques que anteriormente teníamos:

- ❖ **Categoría:** lista de asignaturas con sus iconos correspondientes.
- ❖ **Tipo de Material:** listado de checkboxes para "Apuntes", "Exámenes", "Prácticas", etc.
- ❖ **Valoración Mínima:** selector de estrellas (Radio buttons) para filtrar por calidad percibida.

### Tarjetas de Resultados (Grid)

La visualización de cada documento adapta los datos reales de la BD al diseño de la tarjeta:

- ❖ **Adaptación de Badges:**
  - Se usa el campo `tipo_acceso` para determinar el estilo del badge.
  - 'publico' se renderiza visualmente como la etiqueta "Apuntes" (Azul).
  - 'privado' se renderiza visualmente como la etiqueta "Examen" (Gris/Rojo).
- ❖ **Metadatos:** se muestran dinámicamente el título del archivo, el nombre real de la materia y el autor recuperados de la base de datos.

# VISTA COLECCIONES

## 1. Nomenclatura de Componentes

- ❖ **Base de Datos (Tabla):** se llama **coleccion** y es la fuente de verdad donde se almacenan el título, descripción y el ID del autor.
- ❖ **Modelo (app\models\Coleccion):** es la representación lógica de la tabla. Utiliza **Active Record** para interactuar con la DB.
- ❖ **Controlador (app\controllers\ColeccionController):** es el "cerebro" que gestiona las peticiones.
- ❖ **Vista (views/coleccion/index.php):** es el archivo de diseño. La carpeta debe llamarse igual que el controlador (en minúsculas) y el archivo igual que la acción.

## 2. Obtención y Muestreo de Datos

El flujo de datos desde la base de datos hasta el HTML del usuario es el siguiente:

- ❖ **Consulta (Query):** el controlador ejecuta **Coleccion::find()->all()**. Esto traduce una orden de PHP a un **SELECT \* FROM coleccion** de SQL de forma automática.
- ❖ **Relaciones (Joins):** gracias a la función **getUsuario()** definida en el modelo, Yii2 sabe que cada colección está ligada a un registro de la tabla usuario. Esto permite acceder al nombre del autor mediante **\$coleccion->usuario->nombre**.
- ❖ **Bucle en la Vista:** en el archivo **index.php**, se utiliza un bucle **foreach** para recorrer el array de objetos recibidos. Por cada registro en la DB, se genera una tarjeta HTML (**<article class="collection-card">**) usando las clases de nuestro archivo **styles.css**.
- ❖ **Seguridad:** se utiliza el **helper Html::encode()** al mostrar los datos para evitar ataques de inyección de scripts (XSS), asegurando que el contenido de la base de datos se visualice sólo como texto plano.

## views/materiales/view.php

```
// --- LÓGICA DE ARCHIVO ---
$rutaArchivo = 'uploads/' . $model->archivo_url;
$existeArchivo = file_exists($rutaArchivo);
$tamano = $existeArchivo ? round(filesize($rutaArchivo) / 1024 / 1024, 2) . ' MB' : 'Desconocido';
$fechaSubida = $existeArchivo ? date('d/m/Y', filemtime($rutaArchivo)) : date('d/m/Y');
```

Este código se encarga de que antes de mostrar nada en pantalla se realice un chequeo del estado físico del archivo en el servidor.

- ❖ **Verificación de Integridad (file\_exists):** El código verifica físicamente en el disco duro si el fichero existe para poder evitar el Error 404 (enlaces mal).
- ❖ **Si el archivo existe,** calcula su peso en Megabytes y su fecha de última modificación.

```
// Detectar extensión para iconos y texto
$text = strtolower(pathinfo($model->archivo_url, PATHINFO_EXTENSION));
$tipoInfo = [
    'pdf' => ['icon' => '📄', 'label' => 'PDF (Documento)'],
    'doc' => ['icon' => '📄', 'label' => 'Word (Documento)'],
    'docx' => ['icon' => '📄', 'label' => 'Word (Documento)'],
    'xls' => ['icon' => '📊', 'label' => 'Excel (Hoja de cálculo)'],
    'xlsx' => ['icon' => '📊', 'label' => 'Excel (Hoja de cálculo)'],
    'ppt' => ['icon' => '📄', 'label' => 'PowerPoint (Presentación)'],
    'pptx' => ['icon' => '📄', 'label' => 'PowerPoint (Presentación)'],
    'mp4' => ['icon' => '📺', 'label' => 'Video MP4'],
];
$tipoInfo = $tipoInfo[$text] ?? ['icon' => '📁', 'label' => strtoupper($text) . ' (Archivo)'];
```

Esta parte del código utiliza la función **pathinfo** para extraer la extensión del archivo. Cruza la extensión con un **\$tipoInfo** para asignar el icono y la etiqueta correctos. Si no es ninguno de los definidos aparte coge por defecto el icono de la carpeta.

```

<?php if ($existeArchivo): ?>
    <a href="<?> Url::to(['web/uploads/' . $model->archivo_url] ?>" class="btn btn-primary btn-lg px-4" target="_blank" download>
        Descargar <?> strtoupper($text) ?> (<?> $tamano ?>)
    </a>
<?php else: ?>
    <button class="btn btn-secondary btn-lg px-4" disabled>⚠ No disponible</button>
<?php endif; ?>

```

**Si el archivo es correcto:** Muestra un botón azul de descargar, indicando el formato y el peso. En caso de que no esté el archivo, el botón se pone de color gris y avisa de que no está disponible.

Para los comentarios:

```

<?php foreach ($model->comentarios as $comentario): ?>
    <div class="d-flex mb-3 pl-3 border-bottom">
        <div class="flex-shrink 0 mr-3">
            <div class="bg-light rounded-circle d-flex align-items-center justify-content-center fw-bold text-primary" style="width:40px; height:40px; font-size:1.2em; line-height:1">
                <?> strtoupper(substr($comentario->usuario ? $comentario->usuario : nombre : '?', 0, 1)) ?>
            </div>
        </div>
        <div>
            <div class="fw-bold ms-1">
                <?> html::encode($comentario->usuario ? $comentario->usuario : nombre : 'usuario') ?>
            </div>
            <small class="text-muted d-block mb-2">
                <?> $comment->formatar->date($comentario->fecha, 'long') ?>
            </small>
            <p class="mb-0 text-dark"><?> nl2br(html::encode($comentario->contenido)) ?></p>
        </div>
    </div>
<?php endforeach; ?>

```

Se utiliza un bucle **foreach** para recorrer el array de objetos **DocumentoComentario**.

**Si es Invitado (isGuest):** Se oculta el formulario y se muestra un botón para iniciar sesión.

**Si es Usuario Registrado:** Se muestra el área de texto de los comentarios. Al enviar, se almacena el ID del usuario logueado.

Los comentarios se guardan en una tabla dedicada llamada **documento\_comentario**

**Para las estrellas:** Se envía dos parámetros al controlador: el id del documento y la puntuación (1, 2, 3, 4 o 5).

```

<?php foreach([1, 2, 3, 4, 5] as $num): ?>
    <a href="<?> Url::to(['materiales/valorar', 'id' => $model->id, 'puntuacion' => $num]) ?>"
        class="btn btn-outline-warning btn-sm mx-1 text-dark shadow-sm d-flex align-items-center justify-content-center"
        title="Dar <?> $num ?> estrellas"
        style="width: 35px; height: 35px; font-size: 1.2rem; padding: 0;"
    >
        ★
    </a>
<?php endforeach; ?>

```

Los votos se almacenan en la tabla **documento\_valoracion**.

El identificador del documento actual (**\$model->id**), para saber qué se está votando.

Se utiliza **btn-outline-warning** para dar la apariencia de estrellas doradas al pasar el ratón (**hover**).

```

<div class="text-warning small ms-auto" title="Valoración media: <?> $rating ?>">
    <?php
        for($i=1; $i<=5; $i++) {
            echo ($i <= round($rating)) ? '★' : '☆';
        }
    >
    <span class="text-muted ms-1 text-dark">(<?> $rating ?>/5 - <?> $votos ?> votos)</span>
</div>

```

En esta parte del código es donde se “colorean” las estrellas, se la posición actual (**\$i**) es menor o igual a la nota media se colorea la estrella, si no se queda vacía.

## **views/materiales/index.php**

La función principal de este archivo es listar, filtrar y presentar los recursos disponibles en la plataforma.

La mayoría de esta página se basa en formularios HTML.

```

<?php Html::beginForm(['/materiales/index'], 'get', ['class' => 'search-form mb-8']) ?>
<div class="flex gap-4 flex-wrap">
    <div class="form-group flex-1 min-w-300">
        <label for="search-input" class="visually-hidden">Buscar</label>
        <div class="form-search">
            <span class="form-search__icon">&#128269;</span>
            <input type="search" id="search-input" name="q" class="form-input form-search__input"
                placeholder="Buscar por título, autor..."
                value="<?php Html::encode($request->get('q')) ?>" />
        </div>
    </div>
    <button type="submit" class="btn btn--primary">Buscar</button>
</div>
<?php Html::endForm() ?>

```

Al usar **GET** los filtros que aplica el usuario (búsqueda de texto, categorías seleccionadas) se reflejan en la barra de direcciones del navegador, por ejemplo **?q=matematicas&category[]=5**, permitiendo así a los usuarios compartir enlaces de búsquedas específicas.

El código genera la lista de asignaturas disponibles en el lateral izquierdo con una asignación de iconos basada en el nombre de la materia.

```

<?php foreach ($maincategories as $cat): ?>
    <?php
        $num = $categoryCounts[$cat->id] ?? 0;
        $isChecked = in_array($cat->id, $catsSel);
        $icon = '🔍';
        if (stripos($cat->nombre, 'Programación') !== false) $icon = '💻';
        elseif (stripos($cat->nombre, 'Arte y humanidades') !== false) $icon = '🎨';
        elseif (stripos($cat->nombre, 'Matemáticas') !== false) $icon = '📐';
        elseif (stripos($cat->nombre, 'Química') !== false) $icon = '🧪';
        elseif (stripos($cat->nombre, 'Idiomas') !== false) $icon = '🗣️';
        elseif (stripos($cat->nombre, 'Historia') !== false) $icon = '📖';
        elseif (stripos($cat->nombre, 'Física') !== false) $icon = '⚡';
        elseif (stripos($cat->nombre, 'Economía') !== false) $icon = '💰';
        elseif (stripos($cat->nombre, 'Biología') !== false) $icon = '🌱';
    <?php
    <div class="form-checkbox">
        <input type="checkbox" id="cat-<?php $cat->id ?>" name="category[]" value="<?php $cat->id ?>" <?php $isChecked ? 'checked' : '' ?> />
        <label for="cat-<?php $cat->id ?>">
            <?php $icon ?> <?php Html::encode($cat->nombre) ?>
            <span class="text-muted small"><?php $num ?></span>
        </label>
    </div>
<?php endforeach ?>

```

El sistema recorre las categorías. Mediante la función **stripos** (búsqueda de texto sin tener en cuenta las mayúsculas), detecta palabras clave y asigna un icono representativo.

**El filtro:** Verifica si la categoría está marcada con **in\_array** para mantener la casilla activada (**checked**) tras recargar la página.

Diseño de tarjetas

```

<?php foreach ($dataProvider->getModels() as $model): ?>
    <article class="material-card card">
        <div class="card__body">
            <div class="flex justify-between align-center mb-4">
                <span class="badge <?php ($model->tipo_acceso == 'privado') ? 'badge--secondary' : 'badge--primary' ?>">
                    <?php ($model->tipo_acceso == 'privado') ? 'Examen' : 'Apuntes' ?>
                </span>
            </div>
            <div class="rating">★ <?php $model->getRatingPromedio() ?></div>
        </div>
        <h3 class="material-card__title">
            <a href="<?php Url::to(['materiales/view', 'id' => $model->id]) ?>">
                <?php Html::encode($model->título) ?>
            </a>
        </h3>
        <p class="material-card__description">Archivo: <em><?php Html::encode($model->archivo_url) ?></em></p>
        <div class="material-card__meta">
            <span class="material-card__category">
                <?php $model->materia ? Html::encode($model->materia->nombre) : 'General' ?>
            </span>
        </div>
        <div class="card__footer flex justify-between align-center">
            <span class="material-card__author">👤 <?php $model->autor ? Html::encode($model->autor->nombre) : 'Andrino' ?></span>
            <span>📄 PDF</span>
        </div>
    </article>
<?php endforeach ?>

```



Utiliza un bucle **foreach** para procesar el bloque de documentos, por defecto, 9 por página. Dentro de cada tarjeta, invoca métodos del modelo como **getRatingPromedio()** para mostrar las estrellas actualizadas al instante.

Si el documento es privado, le asigna la clase **badge--secondary** (gris) y escribe "Examen". Si es público, usa **badge--primary** (azul) y escribe "Apuntes". Esto permite al usuario distinguir el tipo de contenido solo con ver el color.

Números de página

```
<div class="mt-8 flex justify-center">
  <?php LinkPager::widget([
    'pagination' => $dataProvider->pagination,
    'options' => ['class' => 'pagination-custom'], // Clase contenedora UL
    'linkContainerOptions' => ['class' => 'page-item'], // Clases para el LI
    'linkOptions' => ['class' => 'page-link'], // Clases para el enlace A
    'disabledListItemSubTagOptions' => ['class' => 'page-link'],

    // Textos bonitos
    'prevPageLabel' => '← Anterior',
    'nextPageLabel' => 'Siguiendo →',

    // Ocultar primero/último
    'firstPageLabel' => false,
    'lastPageLabel' => false,

    // Limitar número de botones
    'maxButtonCount' => 5,
  ]) ?>
</div>
```

- ❖ **Limitación de Botones (maxButtonCount => 5):** Se restringe el número de páginas visibles a 5., para evitar que la barra de paginación se desborde o rompa el diseño.
- ❖ Se han añadido textos visuales ("← Anterior", "Siguiendo →"), mejorando la accesibilidad cognitiva.
- ❖ Se han desactivado los botones de "Primera" y "Última" página (**firstPageLabel => false**).

## views/materiales/subir.php

El archivo **subir.php** representa la interfaz de entrada de datos del sistema. Se permite a los usuarios autenticados contribuir con nuevo material. Es un formulario HTML.

```
<?php $form = ActiveForm::begin([
    'options' => ['enctype' => 'multipart/form-data']
]); ?>
```

Por defecto, los formularios HTML solo envían texto. Para poder subir archivos, hay que configurar el atributo **enctype** como **multipart/form-data**. Se permite así que el archivo vaya adjunto en la petición **POST** hacia el controlador.

El formulario no utiliza cajas, si no listas desplegables (**<select>**)

```
<div class="row">
  <div class="col-md-6">
    <?php $form->field($model, 'material_id')->dropDownList($materias, ['prompt' => 'Selecciona Materia...', 'class' => 'form-select mb-3']) ?>
  </div>
  <div class="col-md-6">
    <?php $form->field($model, 'tipo_acceso')->dropDownList([
      'publico' => '🔓 Público (apuntes)',
      'privado' => '🔒 Privado (examen)'
    ], ['class' => 'form-select mb-3']) ?>
  </div>
</div>
```

Área de Carga de Archivos:

```
<div class="mb-4 p-3 bg-light rounded border border-dashed text-center">
  <label class="form-label fw-bold d-block">Selecciona el archivo</label>
  <?=$form->field($model, 'archivoFile')->fileInput(['class' => 'form-control']->label(false) ?>
  <small class="text-muted d-block mt-2">
    Formatos: <strong>PDF, Word, Excel, PowerPoint, MP4</strong>
  </small>
  <small class="text-muted">Máximo 50 MB</small>
</div>
```

Se incluyen textos de ayuda (**<small>**) que recuerdan al usuario los formatos permitidos y el límite de peso.

## views/layouts/main.php

Este archivo se encarga de que se visualice la cabecera y el pie de la página.

```
// Registrer estilos
AppAsset::register($this);
```

Injecta automáticamente todas las hojas de estilo y scripts definidos en el paquete de recursos. Haciendo esto se logra evitar tener que escribir etiquetas **<link>** o **<script>** en cada página.

El menú de navegación (**<nav>**) implementa una lógica condicional basada en el estado de autenticación del usuario.

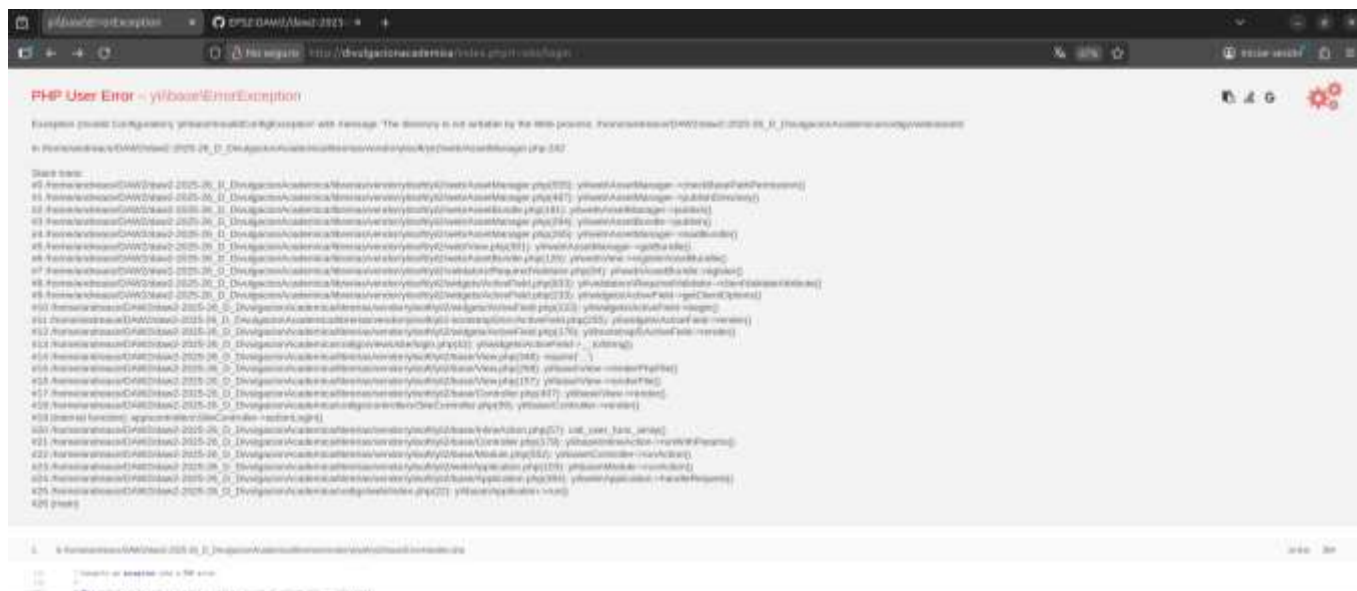
- ❖ A los que no han iniciado sesión se les invita a que lo hagan.
- ❖ A los usuarios se les da acceso a herramientas de creación (Botón "Subir") y gestión de perfil.

```
<main id="main-content" class="main" role="main">
  <?=$content ?>
</main>
```

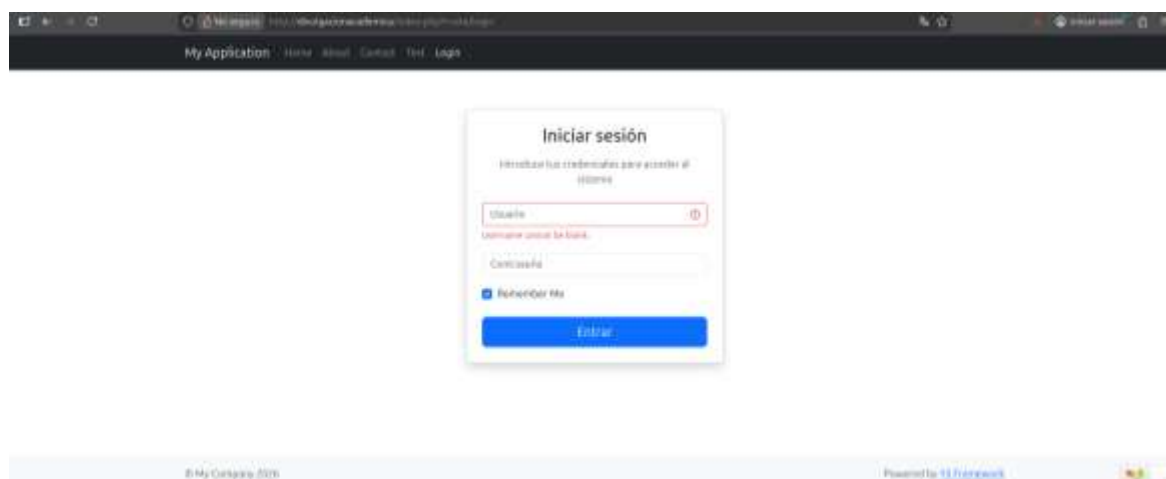
La variable **\$content** es el marcador de posición donde Yii2 inserta el resultado de la vista específica que se está visitando como el formulario de subida o el catálogo. Esto permite que el contenido cambie mientras la cabecera y el footer permanecen fijos.

## Iniciar sesión

Lo primero que se hace es hacer que Apache (www-data) sea el propietario de la carpeta **assets** y de todo su contenido, permitiéndole crear, modificar y borrar archivos dentro de ella. Para ello, se usa el comando **sudo chown -R www-data:www-data /home/andreace/DAW2/daw2-2025-26\_D\_DivulgacionAcademica/codigo/web/assets**. Esto es muy importante, ya que, si no se hace, al intentar acceder a la URL **http://divulgacionacademica/index.php?r=site/login** saldrá lo siguiente:



Para cambiar la forma en la que se muestra la página de iniciar sesión en la web, se debe modificar el código del archivo [codigo/views/site/login.php](#).



## Q&A

La página de Q&A, o preguntas y respuestas en español, necesita una URL desde la que se pueda acceder a ella. Para esto, se tiene que añadir al archivo [codigo/controllers/SiteController.php](#) el siguiente método:

```

105 // Opcional: Proteger esta ruta para que sólo usuarios logueados (o admins) lo vean
106 if (Yii::$app->user->isGuest) {
107     return $this->redirect(['site/login']);
108 }
109
110 // Preparamos los datos para el GridView usando el modelo Userlog
111 $dataProvider = new yii\data\ActiveDataProvider([
112     'query' => \app\models\Userlog::find()->orderBy(['created_at' => SORT_DESC]),
113     'pagination' => ['pagesize' => 20],
114 ]);
115
116 return $this->render('logs', [
117     'dataProvider' => $dataProvider,
118 ]);
119 }
120
121 /**
122  * Displays QA page.
123  *
124  * @return string
125  */
126 public function actionQa()
127 {
128     return $this->render('qa');
129 }
130
131 }

```

Con esto, se podrá acceder a la página con la URL <http://divulgacionacademica/index.php?r=qand-a%2Findex>. Sin embargo, hay que crear un archivo para la vista, cuyo nombre será [codigo/views/qand-a/index.php](#).

```

1 <?php
2 use yii\helpers\Html;
3
4 $this->title = 'Preguntas y Respuestas';
5 ?>
6
7 <div class="site-qa">
8
9     <h1><?php Html::encode($this->title) ?></h1>
10
11     <!-- PREGUNTAS Y RESPUESTAS -->
12     <div class="qa-filters">
13
14         <form class="row g-3 align-items-end">
15
16             <!-- BUSCAR -->
17             <div class="col-md-4">
18                 <label class="form-label">Buscar duda</label>
19                 <input
20                     type="text"
21                     class="form-control"
22                     placeholder="Buscar por texto..."
23                 >
24             </div>
25
26             <!-- CATEGORÍA -->
27             <div class="col-md-4">
28                 <label class="form-label">Categoría</label>
29                 <select class="form-select">
30                     <option selected>Todas las categorías</option>
31                     <option>Programación</option>
32                     <option>Matemáticas</option>
33                     <option>Bases de Datos</option>
34                     <option>Sistemas</option>
35                 </select>
36             </div>
37
38             <!-- ESTADO -->
39             <div class="col-md-4">
40                 <label class="form-label">Estado</label>
41                 <select class="form-select">
42                     <option selected>Todos</option>
43                     <option>Sin responder</option>
44                     <option>Respondida</option>
45                     <option>Resuelta</option>
46                 </select>
47             </div>
48         </form>
49     </div>
50
51     <!-- RESUMEN DE DATOS -->
52     <div class="qa-summary">
53
54         <button class="qa-summary-btn" disabled>
55             <div class="qa-number">12</div>
56             <div>Total</div>
57         </button>
58
59         <button class="qa-summary-btn" disabled>
60             <div class="qa-number">5</div>
61             <div>Sin responder</div>
62         </button>
63     </div>
64

```

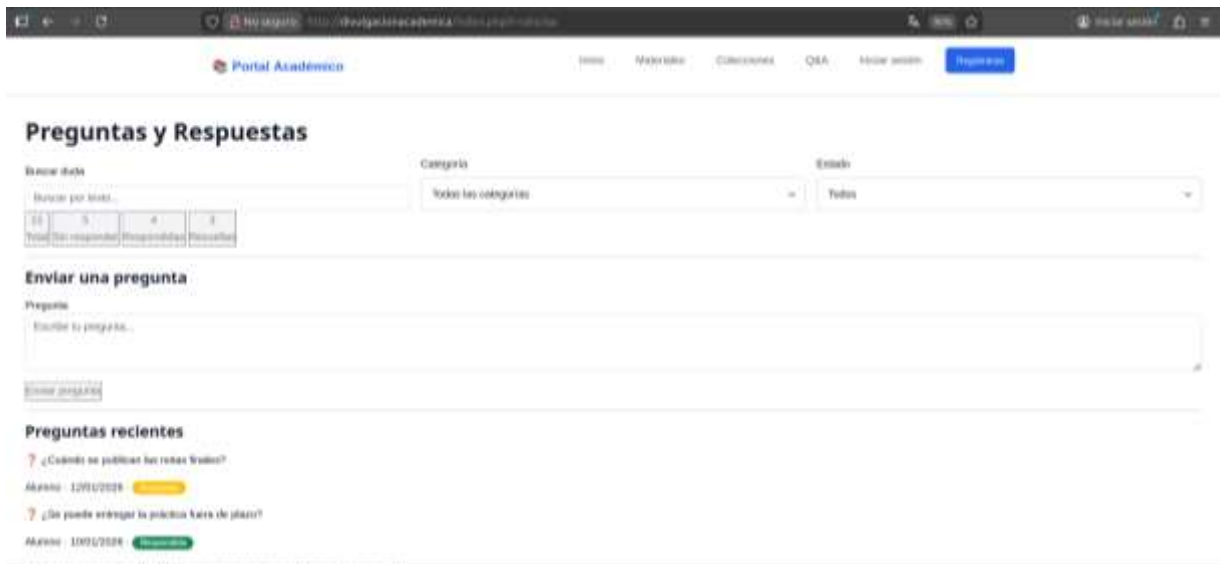


```

66     <button class="qa-summary-btn" disabled>
67       <div class="qa-number">4</div>
68       <div>Respondidas</div>
69     </button>
70
71     <button class="qa-summary-btn" disabled>
72       <div class="qa-number">3</div>
73       <div>Resueltas</div>
74     </button>
75
76   </div>
77
78   <hr>
79
80   <!-- FORMULARIO ALUMNO -->
81   <div class="qa-form">
82     <h3>Enviar una pregunta</h3>
83
84     <form>
85       <div class="form-group mb-3">
86         <label>Pregunta</label>
87         <textarea class="form-control" rows="3" placeholder="Escribe tu pregunta..."></textarea>
88       </div>
89
90       <button class="qa-summary-btn" disabled>
91         Enviar pregunta
92       </button>
93     </form>
94   </div>
95
96   <hr>
97
98   <!-- LISTADO Q&A -->
99   <div class="qa-list">
100
101     <h3>Preguntas recientes</h3>
102
103     <!-- PREGUNTA SIN RESPUESTA -->
104     <div class="qa-item pending">
105       <p class="qa-question">
106         ? ¿Cuándo se publican las notas finales?
107       </p>
108       <p class="qa-meta">
109         Alumno - 12/01/2026 - <span class="badge bg-warning">Pendiente</span>
110       </p>
111     </div>
112
113     <!-- PREGUNTA RESPONDIDA -->
114     <div class="qa-item answered">
115       <p class="qa-question">
116         ? ¿Se puede entregar la práctica fuera de plazo?
117       </p>
118       <p class="qa-meta">
119         Alumno - 10/01/2026 - <span class="badge bg-success">Respondida</span>
120       </p>
121       <div class="qa-answer">
122         <p>✓ SI, con una penalización del 20% si se entrega dentro de la semana siguiente. </p>
123         <p class="qa-teacher">Profesor: Juan Pérez </p>
124       </div>
125     </div>
126   </div>
127 </div>
128 </div>

```

Tras esto, el aspecto de la página será el siguiente:



Para mejorar el aspecto visual de la página, se usarán estilos CSS, por lo que se editará el archivo `codigo/web/css/site.css`. Además, en el archivo `codigo/assets/AppAsset.php` se quitan dos barras que causan que los estilos no se cargarán:

```
<?php
/**
 * @link http://www.yiiframework.com/
 * @copyright Copyright (c) 2008 Yii Software LLC
 * @license http://www.yiiframework.com/license/
 */

namespace app\assets;
use yii\web\AssetBundle;

/**
 * Main application asset bundle.
 *
 * @author Qiang Xue <qiang.xue@gmail.com>
 * @since 2.0
 */

class AppAsset extends AssetBundle
{
    public $basePath = '@webroot';
    public $baseUrl = '@web';

    public $css = [
        //css/site.css, // Estilos base
        'css/styles.css', // Estilos pagina
    ];

    public $js = [
    ];

    public $depends = [
        'yii\web\YiiAsset',
        'yii\bootstrap5\BootstrapAsset'
    ];
}
```

Lo siguiente que se hará es conectar la página a la base de datos, de forma que las preguntas y respuestas no sean simple texto HTML. Para ello, primero se crean los modelos para las preguntas y respuestas, llamando a los archivos `models/Pregunta.php` y `models/Respuesta.php`, respectivamente.

```

Pregunta.php * site.css qa.php
1 <?php
2
3 namespace app\models;
4
5 use yii\db\ActiveRecord;
6
7 class Pregunta extends ActiveRecord
8 {
9     public static function tableName()
10     {
11         return 'pregunta';
12     }
13
14     public function getRespuestas()
15     {
16         return $this->hasMany(Respuesta::class, ['preguntaId' => 'id']);
17     }
18
19     public function getMateria()
20     {
21         return $this->hasOne(Materia::class, ['id' => 'materiaId']);
22     }
23
24     public function getUsuario()
25     {
26         return $this->hasOne(Usuario::class, ['id' => 'usuarioId']);
27     }
28 }

```

```

Pregunta.php Respuesta.php * site.css
1 <?php
2
3 namespace app\models;
4
5 use yii\db\ActiveRecord;
6
7 class Respuesta extends ActiveRecord
8 {
9     public static function tableName()
10     {
11         return 'respuesta';
12     }
13
14     public function getUsuario()
15     {
16         return $this->hasOne(Usuario::class, ['id' => 'usuarioId']);
17     }
18 }

```

Ahora, se conecta el HTML a la base de datos. Para ello, se editará el archivo [codigo/controllers/SiteController.php](#) para obtener las preguntas de la base de datos:

```

SiteController.php * QaController.php Pregunta.php Respuesta.php site.css
1 <?php
2
3 namespace app\controllers;
4
5 use Yii;
6 use yii\filters\AccessControl;
7 use yii\web\Controller;
8 use yii\web\Response;
9 use yii\filters\VerbFilter;
10 use app\models\LoginForm;
11 use app\models\ContactForm;
12 use app\models\RegistroUsuarios; // Necesario para el formulario de registro
13 use app\models\Pregunta; // Importa el modelo Pregunta para acceder a las preguntas almacenadas en la base de datos
14
15 class SiteController extends Controller
16 {
17     /**
18      * {@inheritdoc}
19      */
20
21     public function behaviors()
22     {
23         return [
24             'access' => [
25                 'class' => AccessControl::class,
26                 'only' => ['logout', 'logs'], // Anotamos "logs" al control de acceso si desemos protegerlo
27                 'rules' => [

```

Después, se creará un controlador para el Q&A, al que se llamará [codigo/controllers/QandAController.php](#):

```

1 <?php
2
3 namespace app\controllers;
4
5 use Yii;
6 use yii\web\Controller;
7 use app\models\Pregunta; // Importar modelo
8 use app\models\Respuesta; // Importar modelo
9 use yii\filters\AccessControl;
10
11 class QandAController extends Controller
12 {
13     /**
14      * Configuración de acceso (opcional): define quién puede ver esto
15      */
16     public function behaviors()
17     {
18         return [
19             'access' => [
20                 'class' => AccessControl::class,
21                 'only' => ['create', 'respond'], // solo proteger estas acciones
22                 'rules' => [
23                     // crear y responder solo para usuarios logueados
24                     [
25                         'actions' => ['create', 'respond'],
26                         'allow' => true,
27                         'roles' => ['@'],
28                     ],
29                     // index y view accesibles para todos
30                     [
31                         'actions' => ['index', 'view'],
32                         'allow' => true,
33                     ],
34                 ],
35             ],
36         ];
37     }
38 }

```

```

13         'rules' => ['g', 't'],
14     ],
15 ],
16 ],
17 ];
18 }
19
20
21 /**
22  * Muestra la página principal de gñ
23  * URL: index.php/respuestas/index
24  */
25 public function actionIndex()
26 {
27     // 1. Obtener preguntas
28     $preguntas = Pregunta::find()
29         ->with(['usuario', 'respuestas.usuario'])
30         ->orderBy(['fecha_creacion' => SORT_DESC])
31         ->all();
32
33     // 2. Generar resúmenes
34     $resumen = [
35         'total' => Pregunta::find()->count(),
36         'sin_responder' => Pregunta::find()->where(['estado' => 'sin_responder'])->count(),
37         'respondida' => Pregunta::find()->where(['estado' => 'respondida'])->count(),
38         'resuelta' => Pregunta::find()->where(['estado' => 'resuelta'])->count(),
39     ];
40
41     // 3. Renderizar la vista
42     return $this->render('index', [
43         'preguntas' => $preguntas,
44         'resumen' => $resumen,
45     ]);
46 }
47
48 }
49
50 }
51
52 }
53
54 }
55
56 }
57
58 }
59
60 }
61
62 }
63
64 }
65
66 }
67
68 }
69
70 }
71
72 }
73
74 }
75
76 }
77
78 }
79
80 }
81
82 }
83
84 }
85
86 }
87
88 }
89
90 }
91
92 }
93
94 }
95
96 }
97
98 }
99
100 }
101
102 }
103
104 }
105
106 }
107
108 }
109
110 }
111
112 }
113
114 }
115
116 }
117
118 }
119
120 }
121
122 }
123
124 }
125
126 }
127
128 }
129
130 }
131
132 }
133
134 }
135
136 }
137
138 }
139
140 }
141
142 }
143
144 }
145
146 }
147
148 }
149
150 }
151
152 }
153
154 }
155
156 }
157
158 }
159
160 }
161
162 }
163
164 }
165
166 }
167
168 }
169
170 }
171
172 }
173
174 }
175
176 }
177
178 }
179
180 }
181
182 }
183
184 }
185
186 }
187
188 }
189
190 }
191
192 }
193
194 }
195
196 }
197
198 }
199
200 }
201
202 }
203
204 }
205
206 }
207
208 }
209
210 }
211
212 }
213
214 }
215
216 }
217
218 }
219
220 }
221
222 }
223
224 }
225
226 }
227
228 }
229
230 }
231
232 }
233
234 }
235
236 }
237
238 }
239
240 }
241
242 }
243
244 }
245
246 }
247
248 }
249
250 }
251
252 }
253
254 }
255
256 }
257
258 }
259
260 }
261
262 }
263
264 }
265
266 }
267
268 }
269
270 }
271
272 }
273
274 }
275
276 }
277
278 }
279
280 }
281
282 }
283
284 }
285
286 }
287
288 }
289
290 }
291
292 }
293
294 }
295
296 }
297
298 }
299
300 }
301
302 }
303
304 }
305
306 }
307
308 }
309
310 }
311
312 }
313
314 }
315
316 }
317
318 }
319
320 }
321
322 }
323
324 }
325
326 }
327
328 }
329
330 }
331
332 }
333
334 }
335
336 }
337
338 }
339
340 }
341
342 }
343
344 }
345
346 }
347
348 }
349
350 }
351
352 }
353
354 }
355
356 }
357
358 }
359
360 }
361
362 }
363
364 }
365
366 }
367
368 }
369
370 }
371
372 }
373
374 }
375
376 }
377
378 }
379
380 }
381
382 }
383
384 }
385
386 }
387
388 }
389
390 }
391
392 }
393
394 }
395
396 }
397
398 }
399
400 }
401
402 }
403
404 }
405
406 }
407
408 }
409
410 }
411
412 }
413
414 }
415
416 }
417
418 }
419
420 }
421
422 }
423
424 }
425
426 }
427
428 }
429
430 }
431
432 }
433
434 }
435
436 }
437
438 }
439
440 }
441
442 }
443
444 }
445
446 }
447
448 }
449
450 }
451
452 }
453
454 }
455
456 }
457
458 }
459
460 }
461
462 }
463
464 }
465
466 }
467
468 }
469
470 }
471
472 }
473
474 }
475
476 }
477
478 }
479
480 }
481
482 }
483
484 }
485
486 }
487
488 }
489
490 }
491
492 }
493
494 }
495
496 }
497
498 }
499
500 }
501
502 }
503
504 }
505
506 }
507
508 }
509
510 }
511
512 }
513
514 }
515
516 }
517
518 }
519
520 }
521
522 }
523
524 }
525
526 }
527
528 }
529
530 }
531
532 }
533
534 }
535
536 }
537
538 }
539
540 }
541
542 }
543
544 }
545
546 }
547
548 }
549
550 }
551
552 }
553
554 }
555
556 }
557
558 }
559
560 }
561
562 }
563
564 }
565
566 }
567
568 }
569
570 }
571
572 }
573
574 }
575
576 }
577
578 }
579
580 }
581
582 }
583
584 }
585
586 }
587
588 }
589
590 }
591
592 }
593
594 }
595
596 }
597
598 }
599
600 }
601
602 }
603
604 }
605
606 }
607
608 }
609
610 }
611
612 }
613
614 }
615
616 }
617
618 }
619
620 }
621
622 }
623
624 }
625
626 }
627
628 }
629
630 }
631
632 }
633
634 }
635
636 }
637
638 }
639
640 }
641
642 }
643
644 }
645
646 }
647
648 }
649
650 }
651
652 }
653
654 }
655
656 }
657
658 }
659
660 }
661
662 }
663
664 }
665
666 }
667
668 }
669
670 }
671
672 }
673
674 }
675
676 }
677
678 }
679
680 }
681
682 }
683
684 }
685
686 }
687
688 }
689
690 }
691
692 }
693
694 }
695
696 }
697
698 }
699
700 }
701
702 }
703
704 }
705
706 }
707
708 }
709
710 }
711
712 }
713
714 }
715
716 }
717
718 }
719
720 }
721
722 }
723
724 }
725
726 }
727
728 }
729
730 }
731
732 }
733
734 }
735
736 }
737
738 }
739
740 }
741
742 }
743
744 }
745
746 }
747
748 }
749
750 }
751
752 }
753
754 }
755
756 }
757
758 }
759
760 }
761
762 }
763
764 }
765
766 }
767
768 }
769
770 }
771
772 }
773
774 }
775
776 }
777
778 }
779
780 }
781
782 }
783
784 }
785
786 }
787
788 }
789
790 }
791
792 }
793
794 }
795
796 }
797
798 }
799
800 }
801
802 }
803
804 }
805
806 }
807
808 }
809
810 }
811
812 }
813
814 }
815
816 }
817
818 }
819
820 }
821
822 }
823
824 }
825
826 }
827
828 }
829
830 }
831
832 }
833
834 }
835
836 }
837
838 }
839
840 }
841
842 }
843
844 }
845
846 }
847
848 }
849
850 }
851
852 }
853
854 }
855
856 }
857
858 }
859
860 }
861
862 }
863
864 }
865
866 }
867
868 }
869
870 }
871
872 }
873
874 }
875
876 }
877
878 }
879
880 }
881
882 }
883
88
```

```

1 create.php      QandAController.php  x      main.php
20
21  * Accede el detalle de una pregunta.
22  * @param int $id
23  * @return string
24  */
25  public function action/view($id)
26  {
27      $model = Pregunta::findWhere($id);
28
29      if (!$model) {
30          throw new yii\web\HttpFoundHttpException("La pregunta no existe.");
31      }
32
33      // Redirigimos la vista 'view.php' que acabamos de crear
34      return $this->render('view', [
35          'model' => $model,
36      ]);
37  }
38  }

```

```

create.php  QandAController.php  main.php  index.php
71 public function actionResponder($id)
72 {
73     $pregunta = Pregunta::findOne($id);
74     if (!$pregunta) {
75         throw new \yii\web\NotFoundException();
76     }
77
78     if (Yii::$app->request->isPost) {
79         $contenido = Yii::$app->request->post('contenido_respuesta');
80
81         if (!empty($contenido)) {
82             $respuesta = new Respuesta();
83             $respuesta->pregunta_id = $pregunta->id;
84             $respuesta->usuario_id = Yii::$app->user->id;
85             $respuesta->contenido = $contenido;
86             // $respuesta->fecha_creacion = date('Y-m-d H:i:s'); // si tu BD no la pone automatico
87
88             if ($respuesta->save()) {
89                 Yii::$app->session->setFlash('success', '¡Respuesta publicada!');
90
91                 // Cambiar estado a 'answered' si estaba pendiente
92                 if ($pregunta->estado == 'pending') {
93                     $pregunta->estado = 'answered';
94                     $pregunta->save();
95                 }
96             } else {
97                 Yii::$app->session->setFlash('error', 'Error al guardar la respuesta.');

```

Por último, se modifica de nuevo el archivo [codigo/views/qand-a/index.php](#) para que muestre los datos reales de la base de datos:

```

create.php  QandAController.php  main.php  index.php  Pregunta.p
70 <!-- Lista de preguntas -->
71 <div class="qa-list">
72     <?php if (empty($preguntas)): ?>
73         <div class="alert alert-info text-center py-3">
74             <i class="bi bi-inbox fs-1"></i><br>
75             No se encontraron preguntas con estos criterios.
76         </div>
77     <?php else: ?>
78         <?php foreach ($preguntas as $model): ?>
79             <div class="card mb-3 shadow-sm hover-effect">
80                 <div class="card-body">
81                     <div class="d-flex justify-content-between align-items-start flex-wrap">
82                         <div>
83                             <div class="mb-2">
84                                 <?php
85                                 $claseEstado = match($model->estado) {
86                                     'resolved' => 'success',
87                                     'answered' => 'info',
88                                     default => 'warning text-dark'
89                                 };
90                             <?>
91                             <span class="badge bg-= $claseEstado ?&gt;"&gt;
92                                 &lt;?&gt; ucfirst($model-&gt;estado) &lt;?&gt;
93                             &lt;/span&gt;
94                         &lt;/div&gt;
</pre

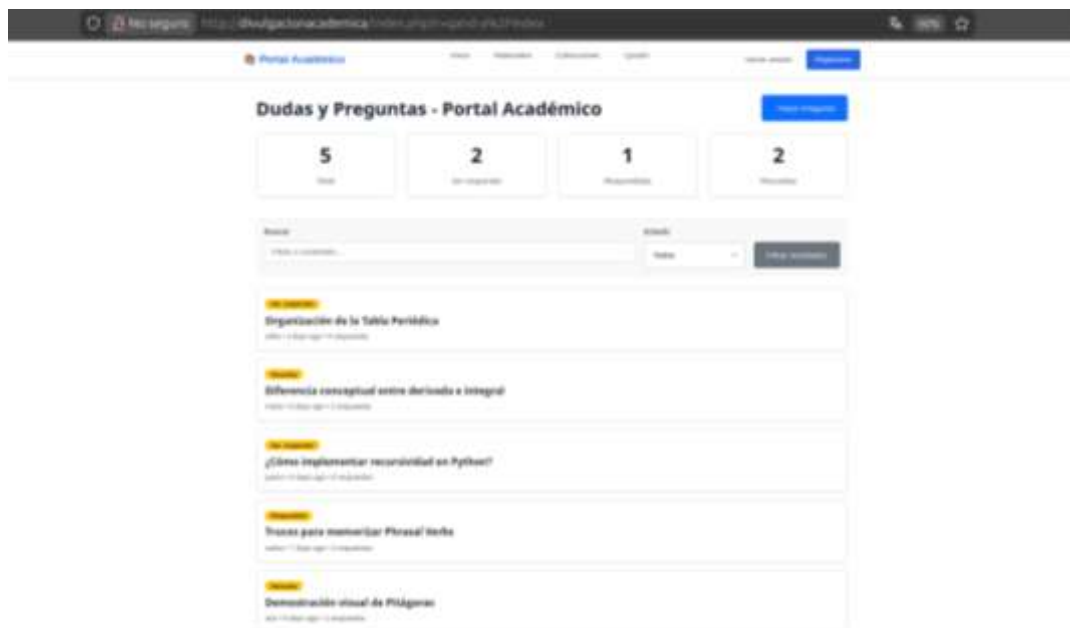
```

```

96         <h4 class="card-title MB-1">
97             <a href="{> Url::to(['view', 'id' => $model->id]) ?>" class="text-decoration: none text-dark stretched-link">
98                 <?> Html::encode($model->titulo) ?>
99             </a>
100         </h4>
101
102         <div class="text-muted small mt-2">
103             <i class="bi bi-person"></i> <?> Html::encode($model->usuario->username ?? 'Anónimo') ?> &bull;
104             <i class="bi bi-calendar"></i> <?> Yii::$app->formatter->asRelativeTime($model->fecha_creacion) ?> &bull;
105             <i class="bi bi-chat-dots"></i> </i> <?> count($model->respuestas) ?> respuestas
106         </div>
107     </div>
108 </div>
109 </div>
110 </div>
111 <?php endforeach; ?>
112 <?php endif; ?>
113 </div>
114
115 </div>
116 </main>

```

Tras todo esto, el resultado final de la página es el siguiente:



Para hacer una pregunta, una vez se ha iniciado sesión, en `codigo/controllers/QandAController.php`, se añade `actionCreate()`.

```

112         return $this->redirect(['view', 'id' => $id]);
113     }
114
115     /**
116      * Crear una nueva pregunta
117      * URL: index.php?r=gonda/create
118      */
119     public function actionCreate()
120     {
121         $model = new Pregunta();
122
123         if ($model->load(Yii::$app->request->post()) && $model->save()) {
124             Yii::$app->session->setFlash('success', '¡Pregunta publicada!');
125             return $this->redirect(['view', 'id' => $model->id]);
126         }
127
128         return $this->render('create', [
129             'model' => $model,
130         ]);
131     }
132 }
133
134 }

```

A continuación, se debe crear una vista, a la que se llamará `codigo/views/qand-a/create.php`:



```

1 <?php
2 use yii\helpers\Html;
3 use yii\widgets\ActiveForm;
4
5 /* @var $this yii\web\View */
6 /* @var $model app\models\Pregunta */
7
8 $this->title = 'Hacer una pregunta';
9 ?>
10
11 <h1><? Html::encode($this->title) ?></h1>
12
13 <?php $form = ActiveForm::begin(); ?>
14
15 <?= $form->field($model, 'titulo')->textInput(['maxlength' => 500]) ?>
16 <?= $form->field($model, 'descripcion')->textareas(['rows' => 6]) ?>
17
18 <div class="form-group">
19     <?= Html::submitButton('Publicar', ['class' => 'btn btn-primary']) ?>
20 </div>
21
22 <?php ActiveForm::end(); ?>

```

El resultado será el siguiente:

The screenshot shows a web browser window with the address bar displaying 'http://dev.guionacademico.com.php?control=pregunta'. The page has a header with the logo 'Portal Académico' and navigation links: 'Inicio', 'Materiales', 'Colecciones', 'Pregunta', 'Hacer una pregunta', 'Mi Perfil', and a red 'Cerrar Sesión' button. The main content area is titled 'Hacer una pregunta' and contains a form with two fields: 'Titulo' (a text input) and 'Descripcion' (a text area). Below the form is a blue button labeled 'Publicar'. The footer is dark blue and contains three columns of links: 'Sobre nosotros' (Plataforma colaborativa para la divulgación académica y científica. Compartiendo conocimiento), 'Enlaces rápidos' (Guía de uso, Normas de la comunidad, Preguntas frecuentes), and 'Legal' (Términos de uso, Política de privacidad, Licencias).

## ARQUITECTURA DEL PROYECTO

- ❖ código/
  - assets/
    - AppAsset.php
  - commands/
    - HelloController.php
  - config/
    - \_\_autocomplete.php
    - console.php
    - db.php
    - db\_orig.php
    - db\_proyecto\_A.php

- db\_proyecto\_B.php
- db\_proyecto\_C.php
- db\_proyecto\_D.php
- params.php
- test.php
- test\_db.php
- web.php
- controllers/
  - ColeccionController.php
  - DocumentoController.php
  - Institucion.php
  - MaterialesController.php
  - QandAController.php
  - SiteController.php
  - TestController.php
  - UsuarioController.php
- mail/
  - layouts/
    - html.php
    - text.php
- models/
  - Coleccion.php
  - ContactForm.php
  - Documento.php
  - DocumentoComentario.php
  - DocumentoValoracion.php
  - Institucion.php
  - LoginForm.php
  - LogVisitaController.php
  - Materia.php
  - Personal.php
  - Pregunta.php
  - RegistroUsuarios.php
  - Respuesta.php
  - UserLog.php
  - Usuario.php
  - UsuarioSearch.php
  - xxx\_User.php
- runtime/
  - cache/
  - debug/
  - logs/
    - app.log
- tests/
  - acceptance/
    - AboutCest.php
    - \_bootstrap.php
    - ContactCest.php
    - HomeCest.php
    - LoginCest.php

- bin/
  - yii
  - yii.bat
- \_data/
- functional/
  - \_bootstrap.php
  - ContactFormCest.php
  - LoginFormCest.php
- \_output/
- \_support/
  - AcceptanceTester.php
  - FunctionalTester.php
  - UnitTester.php
- unit/
  - models/
    - ContactFormTest.php
    - LoginFormTest.php
    - UserTest.php
  - widgets/
    - AlertTest.php
  - \_bootstrap.php
- acceptance.suite.yml.example
- \_bootstrap.php
- functional.suite.yml
- unit.suite.yml
- vagrat/
  - config/
    - vagrant-local.example.yml
  - nginx/
    - log/
    - app.conf
  - provision/
    - always-as-root.sh
    - once-as-root.sh
    - once-as-vagrant.sh
    - provision.awk
- views/
  - coleccion/
    - admin.php
    - agregar\_material.php
    - create.php
    - \_form.php
    - index.php
    - mis\_colecciones.php
    - \_pdf\_template.php
    - update.php
    - usuarios.php
    - view.php
  - documento/

- \_pdf\_documento.php
  - view.php
- layouts/
  - main.php
- materiales/
  - index.php
  - subir.php
  - view.php
- qanda/
  - index.php
  - view.php
- site/
  - about.php
  - contact.php
  - error.php
  - index.php
  - login.php
  - register.php
- test/
  - index.php
- usuario/
  - create.php
  - \_form.php
  - index.php
  - perfil.php
  - \_search.php
  - update.php
  - view.php
- web/
  - assets/
    - 492f9e5c
    - c5bcdab6
    - f1f95a58
  - css/
    - site.css
  - uploads/
    - documento de prueba.pdf
    - excel\_1768064652.xlsx
    - sdasdfasdf\_1768038007.pdf
    - valencias\_1767992955.pdf
    - valencias\_1768040316.pdf
    - valencias\_1768040695.pdf
  - favicon.ico
  - index.php
  - index-test.php
  - robots.txt
- widgets/
  - Alert.php
- LICENSE.md

- README.md
- requirements.php
- yii
- yii.bat
- ❖ librerias/
  - vendor/
  - codeception.yml
  - composer.json
  - composer.lock
  - docker-compose.yml
  - README.md
  - Vagrantfile
- ❖ proyecto/
  - Diagrama.drawio
  - DivulgacionAcademica.pdf
  - README.md
- ❖ sql/
  - divulgacionacademica\_db.sql
  - README.md
- ❖ estructura.txt
- ❖ LICENSE
- ❖ problema\_con\_el\_cambio\_de\_datos\_de\_usuario.txt
- ❖ README.md
- ❖ todo\_juan.md

## **BIBLIOGRAFÍA**

