



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

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 colección | 15 |
| Tabla colección_documento | 15 |
| Tabla usuario_colección | 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 colección | 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 validadación (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/modelsLoginForm.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
);
```

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 );
```

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 (0 fila(s) en la tabla). (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);
[Crear 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 (0 fila(s) en la tabla). (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);
[Crear 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 (0 fila(s) en la tabla). (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);
[Crear 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 (0 fila(s) en la tabla). (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);
[Crear 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: zero columnas). (La consulta tardó 0.0001 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())
[Edita 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: zero 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)
[Edita 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: zero 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())
[Edita 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: zero 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())
[Edita en línea] [Editar] [Crear código PHP]

Tabla respuesta

```
1 CREATE TABLE respuesta (
2     id int(11) NOT NULL,
3     contenido 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 insertado un conjunto de valores vacío (sin decir más información). (La consulta tardó 0.0012 segundos.)

```
CREATE TABLE respuesta ( `id` int(11) NOT NULL, `contenido` text NOT NULL, `fecha` datetime DEFAULT current_timestamp(), `es_solucion` tinyint(1) DEFAULT 0, `preguntaId` int(11) NOT NULL, `usuarioId` int(11) NOT NULL )
```

Elaborado el lunes 11 de febrero del 2019, 11:45

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 insertado un conjunto de valores vacío (sin decir más información). (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() )
```

Elaborado el lunes 11 de febrero del 2019, 11:45

Tabla colección

```
1 CREATE TABLE `colección` (
2     `id` int(11) NOT NULL AUTO_INCREMENT,
3     `titulo` varchar(255) NOT NULL,
4     `descripción` 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_actualización` 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 insertado un conjunto de valores vacío (sin decir más información). (La consulta tardó 0.0012 segundos.)

```
CREATE TABLE `colección` ( `id` int(11) NOT NULL AUTO_INCREMENT, `titulo` varchar(255) NOT NULL, `descripción` text DEFAULT NULL, `usuarioId` int(11) NOT NULL, `descargas` int(11) DEFAULT 0, `tipo_acceso` enum('publico','privado') NOT NULL DEFAULT 'publico', `fecha_actualización` 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
```

Elaborado el lunes 11 de febrero del 2019, 11:45

Tabla colección_documento

```
1 CREATE TABLE colección_documento (
2     colecciónId int(11) NOT NULL,
3     documentoId int(11) NOT NULL,
4     PRIMARY KEY (colecciónId, documentoId),
5     CONSTRAINT fk_cd_colección FOREIGN KEY (colecciónId) REFERENCES colección (id) ON DELETE CASCADE,
6     CONSTRAINT fk_cd_documento FOREIGN KEY (documentoId) REFERENCES documento (id) ON DELETE CASCADE
7 )
```

✓ MySQL ha insertado un conjunto de valores vacío (sin decir más información). (La consulta tardó 0.0004 segundos.)

```
CREATE TABLE colección_documento ( `colecciónId` int(11) NOT NULL, `documentoId` int(11) NOT NULL, PRIMARY KEY (colecciónId, documentoId), CONSTRAINT fk_cd_colección FOREIGN KEY (colecciónId) REFERENCES colección (id) ON DELETE CASCADE, CONSTRAINT fk_cd_documento FOREIGN KEY (documentoId) REFERENCES documento (id) ON DELETE CASCADE )
```

Elaborado el lunes 11 de febrero del 2019, 11:45

Tabla usuario_coleccion

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

MySQL ha devuelto un conjunto de valores vacío (0 decir: zero columnas). La consulta tardó 0.000 segundos.

```
CREATE TABLE usuario_coleccion (usuarioId int(11) NOT NULL,leccionId int(11) NOT NULL,fecha_union timestamp DEFAULT current_timestamp(), PRIMARY KEY (usuarioId,leccionId),CONSTRAINT fk_uc_usuario FOREIGN KEY (usuarioId) REFERENCES usuario (id) ON DELETE CASCADE,CONSTRAINT fk_uc_coleccion FOREIGN KEY (leccionId) REFERENCESleccion (id) ON DELETE CASCADE);
```

{[Edit in browser] [Edit] [Close using 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 (0 decir: zero columnas). La consulta tardó 0.000 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);
```

{[Edit in browser] [Edit] [Close using 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 (0 decir: zero columnas). La consulta tardó 0.000 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);
```

{[Edit in browser] [Edit] [Close using 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 dice: 'cero columnas') (La consulta tardó 0.000 segundos)

CREATE TABLE Contratacion (
    id INT AUTO_INCREMENT PRIMARY KEY,
    estudianteId INT NOT NULL, FOREIGN KEY (estudianteId) REFERENCES Estudiante(id) ON DELETE CASCADE,
    profesorId INT NOT NULL, FOREIGN KEY (profesorId) REFERENCES Profesor(id) ON DELETE CASCADE
);

[Ver en línea] [Deshacer] [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 dice: 'cero 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
);

[Ver en línea] [Deshacer] [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 dice: 'cero 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
);

[Ver en línea] [Deshacer] [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 | Colejamiento | Tamaño | Residuo a depurar |
|----------------------|---|-------|--------|--------------------|----------|-------------------|
| cliente_empresa | ★ Examinar Estructura Buscar Insertar Vaciar Eliminar | 8 | InnoDB | utf8mb4_general_ci | 32.0 KB | |
| colección | ★ Examinar Estructura Buscar Insertar Vaciar Eliminar | 19 | InnoDB | utf8mb4_0900_ai_ci | 32.0 KB | |
| colección_documento | ★ Examinar Estructura Buscar Insertar Vaciar Eliminar | 59 | InnoDB | utf8mb4_0900_ai_ci | 32.0 KB | |
| contratación | ★ Examinar Estructura Buscar Insertar Vaciar Eliminar | 16 | InnoDB | utf8mb4_general_ci | 48.0 KB | |
| documento | ★ Examinar Estructura Buscar Insertar Vaciar Eliminar | 58 | InnoDB | utf8mb4_general_ci | 64.0 KB | |
| documento_comentario | ★ Examinar Estructura Buscar Insertar Vaciar Eliminar | 8 | InnoDB | utf8mb4_general_ci | 48.0 KB | |
| documento_valoracion | ★ Examinar Estructura Buscar Insertar Vaciar Eliminar | 8 | InnoDB | utf8mb4_general_ci | 32.0 KB | |
| evento | ★ Examinar Estructura Buscar Insertar Vaciar Eliminar | 17 | InnoDB | utf8mb4_general_ci | 32.0 KB | |
| incidencia | ★ Examinar Estructura Buscar Insertar Vaciar Eliminar | 11 | InnoDB | utf8mb4_general_ci | 32.0 KB | |
| institución | ★ Examinar Estructura Buscar Insertar Vaciar Eliminar | 7 | InnoDB | utf8mb4_general_ci | 16.0 KB | |
| materia | ★ Examinar Estructura Buscar Insertar Vaciar Eliminar | 61 | InnoDB | utf8mb4_general_ci | 32.0 KB | |
| personal | ★ Examinar Estructura Buscar Insertar Vaciar Eliminar | 19 | InnoDB | utf8mb4_general_ci | 32.0 KB | |
| pregunta | ★ Examinar Estructura Buscar Insertar Vaciar Eliminar | 5 | InnoDB | utf8mb4_general_ci | 48.0 KB | |
| pregunta_guardada | ★ Examinar Estructura Buscar Insertar Vaciar Eliminar | 4 | InnoDB | utf8mb4_general_ci | 32.0 KB | |
| pregunta_voto | ★ Examinar Estructura Buscar Insertar Vaciar Eliminar | 18 | InnoDB | utf8mb4_general_ci | 32.0 KB | |
| respuesta | ★ Examinar Estructura Buscar Insertar Vaciar Eliminar | 8 | InnoDB | utf8mb4_general_ci | 48.0 KB | |
| servicio | ★ Examinar Estructura Buscar Insertar Vaciar Eliminar | 9 | InnoDB | utf8mb4_general_ci | 32.0 KB | |
| user_log | ★ Examinar Estructura Buscar Insertar Vaciar Eliminar | 23 | InnoDB | utf8mb4_general_ci | 32.0 KB | |
| usuario | ★ Examinar Estructura Buscar Insertar Vaciar Eliminar | 27 | InnoDB | utf8mb4_general_ci | 32.0 KB | |
| usuario_colección | ★ Examinar Estructura Buscar Insertar Vaciar Eliminar | 14 | InnoDB | utf8mb4_0900_ai_ci | 32.0 KB | 0 B |
| 20 tablas | Número de filas | 393 | InnoDB | utf8mb4_0900_ai_ci | 794.8 KB | |

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', 'empresario', '2026-01-03 11:21:26', 'contacto'),
7: (6, 'Consultora Global', 'rrhh@consultoraglobal.com', '1234', 'empresario', '2026-01-03 11:21:26', 'rrhh'),
8: (7, 'Startup Innova', 'admin@innova.io', '1234', 'empresario', '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 Nieto', 'elen@mail.com', '1234', 'alumno', '2026-01-03 11:21:26', 'elen'),
14: (13, 'Sofía Rivas', 'sofia@mail.com', '1234', 'alumno', '2026-01-03 11:21:26', 'sofia'),
15: (14, 'Jorge Baez', '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', '2026-01-03 11:21:26', 'juanro'),
18: (17, 'Andrea Colorado Esteban', 'andrea@mail.com', '1234', '', '2026-01-08 10:53:50', 'andreas'),
19: (18, 'Juan Riego Vila', 'juanrv2@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 Aprendiz', 'laura_nau@mail.com', '1234', 'alumno', '2026-01-09 16:01:32', 'lau2024'),
22: (21, 'Empresa TechSoft', 'admin@techsoft.com', '1234', 'empresario', '2026-01-09 16:01:32', 'techsoft_admin'),
23: (22, 'Constructora Civil S.L.', 'info@constructora.com', '1234', 'empresario', '2026-01-09 16:01:32', 'constructora_sl'),
24: (23, 'Academia BioAdmin', 'gestion@biowebadmin.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', 'pedropref'),
26: (25, 'Marta Diseño', 'marta@design.com', '1234', 'empresario', '2026-01-09 16:01:32', 'marta_design'),
27: (26, 'Hospital General', 'compras@hospitalgeneral.com', '1234', 'empresario', '2026-01-09 16:54:16', 'hosp_general'),
27: (27, 'Editorial Saber', 'pedidos@editorialsaber.com', '1234', 'empresario', '2026-01-09 16:54:16', 'edit_saber');
```

✓ 27 filas insertadas. 0 errores (0.000 segundos)

INSERT INTO usuario (id, nombre, email, password, rol, fecha_registro, username) VALUES
1 (1, 'Super Admin', 'admin@portal.com', '1234', 'admin', '2026-01-03 11:21:26', 'admin'),
2 (2, 'Gestor Politécnica', 'gestor@upm.es', '1234', 'gestor', '2026-01-03 11:21:26', 'gestor'),
3 (3, 'Gestor CodeMasters', 'director@codemasters.com', '1234', 'gestor', '2026-01-03 11:21:26', 'director'),
4 (4, 'Gestor EnglishNow', 'hello@englishnow.com', '1234', 'gestor', '2026-01-03 11:21:26', 'hello'),
5 (5, 'Tech Solutions CEO', 'contacto@techsolutions.com', '1234', 'empresario', '2026-01-03 11:21:26', 'contacto'),
6 (6, 'Consultora Global', 'rrhh@consultoraglobal.com', '1234', 'empresario', '2026-01-03 11:21:26', 'rrhh'),
7 (7, 'Startup Innova', 'admin@innova.io', '1234', 'empresario', '2026-01-03 11:21:26', 'innova'),
8 (8, 'Carlos Pérez', 'carlos@mail.com', '1234', 'alumno', '2026-01-03 11:21:26', 'carlos'),
9 (9, 'María García', 'maria@mail.com', '1234', 'alumno', '2026-01-03 11:21:26', 'maria'),
10 (10, 'Lucía Méndez', 'lucia@mail.com', '1234', 'alumno', '2026-01-03 11:21:26', 'lucia'),
11 (11, 'David Torres', 'david@mail.com', '1234', 'alumno', '2026-01-03 11:21:26', 'david'),
12 (12, 'Elena Nieto', 'elen@mail.com', '1234', 'alumno', '2026-01-03 11:21:26', 'elen'),
13 (13, 'Sofía Rivas', 'sofia@mail.com', '1234', 'alumno', '2026-01-03 11:21:26', 'sofia'),
14 (14, 'Jorge Baez', 'jorge@mail.com', '1234', 'alumno', '2026-01-03 11:21:26', 'jorge'),
15 (15, 'Ana Bot', 'ana@mail.com', '1234', 'alumno', '2026-01-03 11:21:26', 'ana'),
16 (16, 'Juan Riego Vila', 'juan@mail.com', '1234', 'alumno', '2026-01-03 11:21:26', 'juanro'),
17 (17, 'Andrea Colorado Esteban', 'andrea@mail.com', '1234', '', '2026-01-08 10:53:50', 'andreas'),
18 (18, 'Juan Riego Vila', 'juanrv2@mail.com', '1234', '', '2026-01-08 13:48:35', 'juanrv2'),
19 (19, 'Roberto Estudiante', 'robert@mail.com', '1234', 'alumno', '2026-01-09 16:01:32', 'robert_stud'),
20 (20, 'Laura Aprendiz', 'laura_nau@mail.com', '1234', 'alumno', '2026-01-09 16:01:32', 'lau2024'),
21 (21, 'Empresa TechSoft', 'admin@techsoft.com', '1234', 'empresario', '2026-01-09 16:01:32', 'techsoft_admin'),
22 (22, 'Constructora Civil S.L.', 'info@constructora.com', '1234', 'empresario', '2026-01-09 16:01:32', 'constructora_sl'),
23 (23, 'Academia BioAdmin', 'gestion@biowebadmin.com', '1234', 'gestor', '2026-01-09 16:01:32', 'bio_gestor'),
24 (24, 'Pedro Profesor', 'pedro@mail.com', '1234', 'alumno', '2026-01-09 16:01:32', 'pedropref'),
25 (25, 'Marta Diseño', 'marta@design.com', '1234', 'empresario', '2026-01-09 16:01:32', 'marta_design'),
26 (26, 'Hospital General', 'compras@hospitalgeneral.com', '1234', 'empresario', '2026-01-09 16:54:16', 'hosp_general'),
27 (27, 'Editorial Saber', 'pedidos@editorialsaber.com', '1234', 'empresario', '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 Pío 42, Barcelona', 'info@codemasters.com'),
4: (3, 'English Now School', 'academia', 'Gran Vía 12, Valencia', 'contacto@englishnow.com'),
5: (4, 'Centro de Investigaciones Biológicas', 'academia', 'Calle Ciencias 88, Sevilla', 'contacto@cib.es'),
6: (5, 'Universidad de las Artes y Diseño', 'universidad', 'Av. Creativa 10, Bilbao', 'admit@uarts.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', 'mba@negocios.com');
```

✓ 7 filas insertadas. 0 errores (0.000 segundos)

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

Tabla materia

```
1 INSERT INTO materia (id, nombre, parentId) VALUES
2 (1, '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', 3),
10 (9, 'Álgebra', 3),
11 (10, 'Estadística', 3),
12 (13, 'Biología', NULL),
13 (14, 'Arts 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, 'Termodinámica', 21),
27 (31, 'Formulación Inorgánica', 22),
28 (32, 'Tabla Periódica', 22),
29 (33, 'Reacciones Químicas', 22).
```

✓ 13 filas insertadas. (La consulta tardó 0.0007 segundos.)

INSERT 3000 materia (id, nombre, parentId) VALUES (1, 'Programación', NULL), (2, 'Idiomas', NULL), (3, 'Matemáticas', NULL), (4, 'Desarrollo Web', 1), (5, 'Ciberseguridad', 1), (6, 'Bases de Datos', 1), (7, 'Inglés', 2), (8, 'Alemán', 3), (9, 'Álgebra', 3), (10, 'Estadística', 3), (13, 'Biología', NULL), (14, 'Arts y Humanidades', NULL), (15, 'Biología Celular', 13), (16, 'Genética Molecular', 13), (17, 'Historia del Arte', 14), (18, 'Diseño Gráfico Digital', 14), (19, 'Inteligencia Artificial', 1), (21, 'Física', NULL), (22, 'Química', NULL), (23, 'Historia', NULL), (24, 'Economía', NULL), (25, 'Introducción a Python', 1), (26, 'Cinemática y Dinámica', 21), (29, 'Electromagnetismo', 21), (30, 'Termodinámica', 21), (31, 'Formulación Inorgánica', 22), (32, 'Tabla Periódica', 22), (33, 'Reacciones Químicas', 22).

(3000)

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, 'Javier', 'Hacker', 2),
5 (4, 'Sara', 'Coder', 2),
6 (5, 'John', 'Smith', 3),
7 (6, 'Frau', 'Müller', 3),
8 (10, 'Dra. Elena', 'Gómez', 4),
9 (11, 'Prof. Marco', 'Pincel', 5),
10 (12, 'Ing. Sofía', '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);
```

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

INSERT 2000 personal (id, nombre, apellido, institucionId) VALUES (1, 'Dr. Roberto', 'Méndez', 1), (2, 'Dra. Laura', 'Física', 1), (3, 'Javier', 'Hacker', 2), (4, 'Sara', 'Coder', 2), (5, 'John', 'Smith', 3), (6, 'Frau', 'Müller', 3), (10, 'Dra. Elena', 'Gómez', 4), (11, 'Prof. Marco', 'Pincel', 5), (12, 'Ing. Sofía', 'Robótica', 6), (13, 'Dr. Alan', 'Turing', 1), (14, 'Prof. Alberto', 'Einstein', 1), (15, 'Dra. Marie', 'Curie', 1), (16, 'Prof. Adam', 'Smith', 2), (17, 'Lic. Juana', 'Lovelace', 2), (18, 'Dr. Herodoto', 'Pérez', 1), (19, 'Dra. Rosalind', 'Franklin', 4), (20, 'Prof. Isaac', 'Newton', 1), (21, 'Lic. Ada', 'Byron', 2), (22, 'Dr. John', 'Keynes', 7).

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', 2, 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', 3, 3, 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, 6, 2),
9 (8, 'Phrasal Verbs List', 'phrasal_verbs.pdf', 'publico', 7, 7, 3),
10 (9, 'German Grammar B1', 'deutsch_b1.pdf', 'privado', 8, 8, 3),
11 (20, 'Introducción a la Edición Genética CRISPR', 'crispr_intro.pdf', 'publico', 10, 10, 4),
12 (21, 'Historia del Renacimiento Italiano', 'renacimiento.pdf', 'privado', 11, 11, 5),
13 (22, 'Redes Neuronales Convolucionales', 'cnn_guide.pdf', 'privado', 12, 12, 6),
14 (23, 'Principios de Composición Visual', 'composicion.pdf', 'publico', 13, 13, 5),
15 (24, 'Atlas Celular 2026', 'atlas_celular.pdf', 'privado', 15, 15, 4),
16 (25, 'Leyes de Newton: Resumen y Ejercicios', 'newton_resumen.pdf', 'publico', 20, 14, 1),
17 (26, 'Formulario de Física para Selectividad', 'formulario_fisica.pdf', 'publico', 21, 14, 1),
18 (27, 'Tabla de Valencias para Imprimir', 'valencias.pdf', 'publico', 21, 15, 1),
19 (28, 'Ejercicios de Ajuste de Reacciones', 'ajuste_reacciones.pdf', 'privado', 25, 15, 1),
20 (29, 'Mi primer programa: nula Mundo en Python', 'python_intro.pdf', 'publico', 25, 17, 2),
21 (40, 'Guía de Núcleos y Condicionales', 'nucleos_java.pdf', 'privado', 3, 17, 2),
22 (41, 'Eje Cronológico de la Guerra Civil', 'eje_guerra_civil.pdf', 'publico', 34, 18, 1),
23 (42, 'Resumen Primera Guerra Mundial', 'wwi_resumen.pdf', 'publico', 35, 18, 1),
24 (43, 'Conceptos básicos: Oferta y Demanda', 'oferta_demanda.pdf', 'publico', 36, 5, 2),
25 (44, 'Balance de Situación: Ejercicio Resuelto', 'balance_ejemplo.pdf', 'privado', 37, 5, 2),
26 (45, 'Comportamiento de la Luz', 'optica_intro.pdf', 'publico', 42, 20, 1),
27 (46, 'Energía Nuclear: Ventajas y Riesgos', 'nuclear_debate.pdf', 'privado', 43, 14, 1),
28 (47, 'Nomenclatura de Carbono', 'quimica_organica_1.pdf', 'publico', 45, 19, 4),
29 (48, 'Manual de Seguridad en Laboratorio', 'safety_lab.pdf', 'publico', 47, 15, 1),

```

✓ 51 filas insertadas. (La consulta tardó 0.000 segundos.)

Datos: 51 filas document (id, titulo, archivo_url, tipo_acceso, materialId, autorId, institucionId). MySQL [1]: Pase el cursor al final. ✓ 1. "Tesis Doctoral: IA Generativa", "tesis_la.pdf", "publico", 1, 1, 1); ✓ 2. "Plan de Estudios 2025", "plan2025.pdf", "publico", 1, 1, 1); ✓ 3. "Ejercicios de Álgebra Lineal", "algebra_ej.pdf", "privado", 2, 2, 1); ✓ 4. "Guía Definitiva de React Hooks", "react_guide.pdf", "publico", 4, 4, 2); ✓ 5. "Manual de Hacking Ético", "hacking_v1.pdf", "privado", 3, 3, 2); ✓ 6. "Cheat Sheet PHP 8.2", "php_resumen.pdf", "publico", 4, 4, 2); ✓ 7. "Laboratorio SQL Avanzado", "sql_lab.pdf", "privado", 6, 6, 2); ✓ 8. "Phrasal Verbs List", "phrasal_verbs.pdf", "publico", 7, 7, 3); ✓ 9. "German Grammar B1", "deutsch_b1.pdf", "privado", 8, 8, 3); ✓ 10. "Introducción a la Edición Genética CRISPR", "crispr_intro.pdf", "publico", 10, 10, 4); ✓ 11. "Historia del Renacimiento Italiano", "renacimiento.pdf", "privado", 11, 11, 5); ✓ 12. "Redes Neuronales Convolucionales", "cnn_guide.pdf", "privado", 12, 12, 6); ✓ 13. "Principios de Composición Visual", "composicion.pdf", "publico", 13, 13, 5); ✓ 14. "Atlas Celular 2026", "atlas_celular.pdf", "privado", 15, 15, 4); ✓ 15. "Leyes de Newton: Resumen y Ejercicios", "newton_resumen.pdf", "publico", 20, 14, 1); ✓ 16. "Formulario de Física para Selectividad", "formulario_fisica.pdf", "publico", 21, 14, 1); ✓ 17. "Tabla de Valencias para Imprimir", "valencias.pdf", "publico", 21, 15, 1); ✓ 18. "Ejercicios de Ajuste de Reacciones", "ajuste_reacciones.pdf", "privado", 25, 15, 1); ✓ 19. "Mi primer programa: nula Mundo en Python", "python_intro.pdf", "publico", 25, 17, 2); ✓ 20. "Guía de Núcleos y Condicionales", "nucleos_java.pdf", "privado", 3, 17, 2); ✓ 21. "Eje Cronológico de la Guerra Civil", "eje_guerra_civil.pdf", "publico", 34, 18, 1); ✓ 22. "Resumen Primera Guerra Mundial", "wwi_resumen.pdf", "publico", 35, 18, 1); ✓ 23. "Conceptos básicos: Oferta y Demanda", "oferta_demanda.pdf", "publico", 36, 5, 2); ✓ 24. "Balance de Situación: Ejercicio Resuelto", "balance_ejemplo.pdf", "privado", 37, 5, 2); ✓ 25. "Comportamiento de la Luz", "optica_intro.pdf", "publico", 42, 20, 1); ✓ 26. "Energía Nuclear: Ventajas y Riesgos", "nuclear_debate.pdf", "privado", 43, 14, 1); ✓ 27. "Nomenclatura de Carbono", "quimica_organica_1.pdf", "publico", 45, 19, 4); ✓ 28. "Manual de Seguridad en Laboratorio", "safety_lab.pdf", "publico", 47, 15, 1);

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 tardó 0.000 segundos.)

Datos: 9 filas servicio (id, nombre, precio, institucionId). MySQL [1]: Pase el cursor al final. ✓ 1. "Máster en Ingeniería de Datos", 4500.00, 1); ✓ 2. "Bootcamp Full Stack Developer", 2900.00, 2); ✓ 3. "Auditoría de Seguridad Web", 1200.00, 2); ✓ 4. "Curso Inglés de Negocios", 800.00, 3); ✓ 5. "Pack Formación Alemán Técnico", 1500.00, 3); ✓ 6. "Curso Avanzado de Genética", 600.00, 4); ✓ 7. "Máster en Diseño UI/UX", 3500.00, 5); ✓ 8. "Bootcamp de Inteligencia Artificial", 2200.00, 6); ✓ 9. "MBA Executive", 12000.00, 7);

Tabla user_log

```

1 INSERT INTO user_log (id, user_id, user_identifier, action, ip_address, user_agent, created_at) VALUES
2 (1, 10, "juanro2@mail.com", "Logout", "192", "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/104.0.0.0 Safari/537.36 Edg/104.0.0.0", "2026-01-09 13:54:11"),
3 (2, 10, "juanro2@mail.com", "Login Success", "192", "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/104.0.0.0 Safari/537.36 Edg/104.0.0.0", "2026-01-09 13:54:24"),
4 (3, 10, "juanro2@mail.com", "Logout", "192", "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/104.0.0.0 Safari/537.36 Edg/104.0.0.0", "2026-01-09 13:54:38"),
5 (4, 8, "carlos@mail.com", "Login Success", "192.168.1.10", "Mozilla/5.0 (Windows) Chrome/98", "2026-01-09 16:53:02"),
6 (5, 9, "maria@mail.com", "View Document ID 28", "192.168.1.11", "Mozilla/5.0 (Macintosh) Safari/14", "2026-01-09 16:53:02"),
7 (6, 9, "carlos@mail.com", "Download PDF", "192.168.1.10", "Mozilla/5.0 (Windows) Chrome/99", "2026-01-09 16:53:02"),
8 (7, 10, "roberto_stud", "Login Failed", "10.0.0.0", "Mozilla/5.0 (Linux) Firefox/98", "2026-01-09 16:53:02"),
9 (8, 10, "roberto_stud", "Login Success", "10.0.0.5", "Mozilla/5.0 (Linux) Firefox/98", "2026-01-09 16:53:02"),
10 (9, 10, "lucia@mail.com", "Update Profile", "192.168.1.10", "Mozilla/5.0 (iPhone) Safari/14", "2026-01-09 16:53:02"),
11 (10, 7, "Contacto", "View Events", "192.168.1.9", "Mozilla/5.0 (Windows) Edge/98", "2026-01-09 16:53:02"),
12 (11, 20, "luis004", "Search query: 'Python'", "192.168.0.12", "Mozilla/5.0 (Android) Chrome/99", "2026-01-09 16:53:02"),
13 (12, 11, "diego@mail.com", "Logout", "192.168.1.9", "Mozilla/5.0 (Windows) Chrome/99", "2026-01-09 20:53:02"),
14 (13, 10, "juanro", "Login Success", "192", "Mozilla/5.0 (Windows) Chrome/120", "2026-01-09 16:53:02"),
15 (14, 10, "juanro", "View Dashboard", "192", "Mozilla/5.0 (Windows) Chrome/120", "2026-01-09 16:53:02");

```

✓ 15 filas insertadas. (La consulta tardó 0.000 segundos.)

Datos: 15 filas user_log (id, user_id, user_identifier, action, ip_address, user_agent, created_at). MySQL [1]: Pase el cursor al final. ✓ 1. "Logout", "192", "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/104.0.0.0 Safari/537.36 Edg/104.0.0.0", "2026-01-09 13:54:11"); ✓ 2. "Login Success", "192", "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/104.0.0.0 Safari/537.36 Edg/104.0.0.0", "2026-01-09 13:54:24"); ✓ 3. "Logout", "192", "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/104.0.0.0 Safari/537.36 Edg/104.0.0.0", "2026-01-09 13:54:38"); ✓ 4. "Login Success", "192.168.1.10", "Mozilla/5.0 (Windows) Chrome/98", "2026-01-09 16:53:02"); ✓ 5. "View Document ID 28", "192.168.1.11", "Mozilla/5.0 (Macintosh) Safari/14", "2026-01-09 16:53:02"); ✓ 6. "Download PDF", "192.168.1.10", "Mozilla/5.0 (Windows) Chrome/99", "2026-01-09 16:53:02"); ✓ 7. "Login Failed", "10.0.0.0", "Mozilla/5.0 (Linux) Firefox/98", "2026-01-09 16:53:02"); ✓ 8. "Login Success", "10.0.0.5", "Mozilla/5.0 (Linux) Firefox/98", "2026-01-09 16:53:02"); ✓ 9. "Update Profile", "192.168.1.10", "Mozilla/5.0 (iPhone) Safari/14", "2026-01-09 16:53:02"); ✓ 10. "View Events", "192.168.1.9", "Mozilla/5.0 (Windows) Edge/98", "2026-01-09 16:53:02"); ✓ 11. "Search query: 'Python'", "192.168.0.12", "Mozilla/5.0 (Android) Chrome/99", "2026-01-09 16:53:02"); ✓ 12. "Logout", "192.168.1.9", "Mozilla/5.0 (Windows) Chrome/99", "2026-01-09 20:53:02"); ✓ 13. "Login Success", "192", "Mozilla/5.0 (Windows) Chrome/120", "2026-01-09 16:53:02"); ✓ 14. "View Dashboard", "192", "Mozilla/5.0 (Windows) Chrome/120", "2026-01-09 16:53:02"); ✓ 15. "Logout", "192", "Mozilla/5.0 (Windows) Chrome/120", "2026-01-09 16:53:02");

Tabla pregunta

```
1 INSERT INTO pregunta (id, titulo, descripcion, fecha_creacion, vistas, votos_utiles, estatus, usuarioId, materialId) VALUES
2 (1, '¿Cómo implementar recursividad en Python?', 'No entiendo bien el caso base en las funciones recursivas. ¿Alguno tiene un ejemplo simple con factorial?', '2020-01-01 10:00:00', 10, 2, 'abi_responder', 10, 2),
3 (2, 'Diferencia conceptual entre Derivada e Integral', 'Entiendo como calcularlas, pero no entiendo que significa en la vida real. ¿Ayuda?', '2020-01-02 14:15:00', 10, 2, 'reusulta', 8, 7),
4 (3, 'Trucos para memorizar Phrasal Verbs', 'Son demasiados y me confundes entre 'get in', 'get by'. ¿Algún consejo?', '2020-01-04 09:00:00', 10, 10, 'respondida', 8, 7),
5 (4, 'Demostración visual de Pitágoras', 'Buenas una explicación geométrica, no solo la fórmula  $a^2 + b^2 = c^2$ ', '2020-02-02 15:20:00', 10, 3, 'reusulta', 10, 7),
6 (5, 'Organización de la Tabla Periódica', '¿Por qué los gases nobles están a la derecha del todo?', '2020-01-07 11:45:00', 47, 0, 'sin_responder', 10, 3);
```

✓ 5 filas insertadas. La consulta tardó 0.003 segundos.

INSERT INTO pregunta (id, titulo, descripcion, vistas, votos_utiles, estatus, usuarioId, materialId) VALUES
(1, '¿Cómo implementar recursividad en Python?', 'No entiendo bien el caso base en las funciones recursivas. ¿Alguno tiene un ejemplo simple con factorial?', '2020-01-01 10:00:00', 10, 2, 'abi_responder', 10, 2), (2, 'Diferencia conceptual entre Derivada e Integral', 'Entiendo como calcularlas, pero no entiendo que significa en la vida real. ¿Ayuda?', '2020-01-02 14:15:00', 10, 2, 'reusulta', 8, 7), (3, 'Trucos para memorizar Phrasal Verbs', 'Son demasiados y me confundes entre "get in", "get by". ¿Algún consejo?', '2020-01-04 09:00:00', 10, 10, 'respondida', 8, 7), (4, 'Demostración visual de Pitágoras', 'Buenas una explicación geométrica, no solo la fórmula $a^2 + b^2 = c^2$ ', '2020-02-02 15:20:00', 10, 3, 'reusulta', 10, 7), (5, 'Organización de la Tabla Periódica', '¿Por qué los gases nobles están a la derecha del todo?', '2020-01-07 11:45:00', 47, 0, 'sin_responder', 10, 3);

Datos:

Tabla colección

```
1 INSERT INTO colección (id, título, descripción, usuarioId, descargas, tipo_acceso) VALUES
2 (1, 'Matemáticas de Ingeniería', 'Cálculo diferencial, integral y vectores en 3D.', 10, 450, 'público'),
3 (2, 'Desarrolla Frontend Moderno', 'Dominando React Hooks y composición visual.', 4, 1200, 'público'),
4 (3, 'Ciberseguridad y Redes', 'Hacking ético, SQL avanzado y seguridad en redes.', 3, 800, 'público'),
5 (4, 'Idiomas y Cultura Global', 'Francés, Alemán, Italiano y cultura japonesa.', 8, 340, 'público'),
6 (5, 'Historia y Civilizaciones', 'Del Renacimiento a la Guerra Civil Española.', 10, 500, 'público'),
7 (6, 'Biológia y Genética 2026', 'Edición genética CRISPR y Atlas Celular.', 10, 210, 'público'),
8 (7, 'Física y Química General', 'Leyes de Newton, óptica y conservación de carbono.', 15, 670, 'público'),
9 (8, 'Economía y Finanzas', 'Oferta, demanda, contabilidad e Interés compuesto.', 5, 430, 'público'),
10 (9, 'Arte y Composición Visual', 'Teoría del color, fotografía y diseño gráfico.', 11, 700, 'público'),
11 (10, 'IA y Neurociencia', 'Redes neuronales y plasticidad cerebral.', 1, 2500, 'público'),
12 (11, 'Programación con Python', 'DesdeHola Mundo hasta estructuras de datos.', 17, 920, 'público'),
13 (12, 'Tecnología Móvil y Apps', 'Desarrollo en Android y estrategias móviles.', 21, 310, 'público'),
14 (13, 'Producción Musical Pro', 'Mezcla, equalización y teoría musical.', 11, 150, 'público'),
15 (14, 'Zoología y Ecosistemas', 'Fauna ibérica e invertebrados marinos.', 10, 85, 'público');
```

✓ 15 filas insertadas. La consulta tardó 0.003 segundos.

INSERT INTO colección (id, título, descripción, usuarioId, descargas, tipo_acceso) VALUES
(1, 'Matemáticas de Ingeniería', 'Cálculo diferencial, integral y vectores en 3D.', 10, 450, 'público'), (2, 'Desarrolla Frontend Moderno', 'Dominando React Hooks y composición visual.', 4, 1200, 'público'), (3, 'Ciberseguridad y Redes', 'Hacking ético, SQL avanzado y seguridad en redes.', 3, 800, 'público'), (4, 'Idiomas y Cultura Global', 'Francés, Alemán, Italiano y cultura japonesa.', 8, 340, 'público'), (5, 'Historia y Civilizaciones', 'Del Renacimiento a la Guerra Civil Española.', 10, 500, 'público'), (6, 'Biológia y Genética 2026', 'Edición genética CRISPR y Atlas Celular.', 10, 210, 'público'), (7, 'Física y Química General', 'Leyes de Newton, óptica y conservación de carbono.', 15, 670, 'público'), (8, 'Economía y Finanzas', 'Oferta, demanda, contabilidad e Interés compuesto.', 5, 430, 'público'), (9, 'Arte y Composición Visual', 'Teoría del color, fotografía y diseño gráfico.', 11, 700, 'público'), (10, 'IA y Neurociencia', 'Redes neuronales y plasticidad cerebral.', 1, 2500, 'público'), (11, 'Programación con Python', 'DesdeHola Mundo hasta estructuras de datos.', 17, 920, 'público'), (12, 'Tecnología Móvil y Apps', 'Desarrollo en Android y estrategias móviles.', 21, 310, 'público'), (13, 'Producción Musical Pro', 'Mezcla, equalización y teoría musical.', 11, 150, 'público'), (14, 'Zoología y Ecosistemas', 'Fauna ibérica e invertebrados marinos.', 10, 85, 'público');

Datos:

Tabla contratacion

```
1 INSERT INTO contratacion (id, clienteId, servicioId, fecha) VALUES
2 (1, 1, 2, '2020-09-01 10:00:00'),
3 (2, 1, 3, '2020-01-15 09:30:00'),
4 (3, 2, 4, '2020-02-01 11:00:00'),
5 (4, 3, 2, '2020-03-10 16:45:00'),
6 (5, 4, 8, '2020-01-10 10:00:00'),
7 (6, 6, 7, '2020-01-12 11:30:00'),
8 (7, 1, 9, '2020-01-15 09:00:00'),
9 (8, 2, 1, '2020-11-15 09:00:00'),
10 (9, 3, 8, '2020-12-01 10:30:00'),
11 (10, 9, 9, '2020-12-30 16:00:00'),
12 (11, 1, 9, '2020-01-02 00:45:00'),
13 (12, 4, 2, '2020-01-05 11:15:00'),
14 (13, 6, 6, '2020-01-08 14:20:00'),
15 (14, 2, 3, '2020-01-09 09:00:00'),
16 (15, 7, 6, '2020-01-05 10:00:00'),
17 (16, 8, 4, '2020-01-07 12:30:00');
```

✓ 17 filas insertadas. La consulta tardó 0.004 segundos.

INSERT INTO contratacion (id, clienteId, servicioId, fecha) VALUES
(1, 1, 2, '2020-09-01 10:00:00'), (2, 1, 3, '2020-01-15 09:30:00'), (3, 2, 4, '2020-02-01 11:00:00'), (4, 3, 2, '2020-03-10 16:45:00'), (5, 4, 8, '2020-01-10 10:00:00'), (6, 6, 7, '2020-01-12 11:30:00'), (7, 1, 9, '2020-01-15 09:00:00'), (8, 2, 1, '2020-11-15 09:00:00'), (9, 3, 8, '2020-12-01 10:30:00'), (10, 9, 9, '2020-12-30 16:00:00'), (11, 1, 9, '2020-01-02 00:45:00'), (12, 4, 2, '2020-01-05 11:15:00'), (13, 6, 6, '2020-01-08 14:20:00'), (14, 2, 3, '2020-01-09 09:00:00'), (15, 7, 6, '2020-01-05 10:00:00'), (16, 8, 4, '2020-01-07 12:30:00');

Datos:

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 verlas?', '2026-01-07 16:45:00', 20, 35);
```

✓ 6 filas insertadas. La consulta tardó 0.013 segundos.

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

Datos:

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');
```

✓ 10 filas insertadas. La consulta tardó 0.0023 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');

[Deshacer] [Editar] [Ver código SQL]

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');
```

✓ 5 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');

[Deshacer] [Editar] [Ver código SQL]

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 0.0024 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');

[Deshacer] [Editar] [Ver código SQL]

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, 18),
3 (2, 'Matemáticamente son operaciones inversas, como la multiplicación y la división.', '2026-01-09 19:00:00', 1, 2, 10),
4 (3, 'Intenta agruparlos por tema en lugar de por verso. Por ejemplo, todos los que sirven para "velocidad".', '2026-01-09 19:00:00', 1, 3, 11),
5 (4, 'Yo uso tarjetas de memoria (Flashcards) con la app Anki. Funciona muy bien.', '2026-01-09 09:15:00', 1, 3, 10),
6 (5, 'Imagínate un cuadrado construido sobre cada lado del triángulo. El área del cuadrado grande es igual a la suma de las áreas de los pequeños.', '2026-01-09 19:00:00', 1, 4, 24);
```

✓ 6 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, 18), (2, 'Matemáticamente son operaciones inversas, como la multiplicación y la división.', '2026-01-09 19:00:00', 1, 2, 10), (3, 'Intenta agruparlos por tema en lugar de por verso. Por ejemplo, todos los que sirven para "velocidad".', '2026-01-09 19:00:00', 1, 3, 11), (4, 'Yo uso tarjetas de memoria (Flashcards) con la app Anki. Funciona muy bien.', '2026-01-09 09:15:00', 1, 3, 10), (5, 'Imagínate un cuadrado construido sobre cada lado del triángulo. El área del cuadrado grande es igual a la suma de las áreas de los pequeños.', '2026-01-09 19:00:00', 1, 4, 24);

[Deshacer] [Editar] [Ver código SQL]

Tabla colección_documento

```
1 INSERT INTO colección_documento (colecciónId, 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)
```

✓ 16 filas insertadas. La consulta tardó 0.003 segundos.

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

Elaborado por [] el [] [Crear código HTML]

Tabla usuario_colección

```
1 INSERT INTO usuario_colección (usuarioId, colecciónId, 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');
```

✓ 9 filas insertadas. La consulta tardó 0.003 segundos.

INSERT 9 INTO usuario_colección (usuarioId, colecciónId, 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')

Elaborado por [] el [] [Crear código HTML]

Tabla Cliente-Empresa

```
1 INSERT INTO clienteempresa (id, razon_social, cif, usuarioId) VALUES
2 (1, 'Tech Solutions S.L.', 'B99999991', 5),
3 (2, 'Consultora Global S.A.', 'A88888882', 6),
4 (3, 'Innova StartUp', 'B77777773', 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 8 INTO clienteempresa (id, razon_social, cif, usuarioId) VALUES (1, 'Tech Solutions S.L.', 'B99999991', 5), (2, 'Consultora Global S.A.', 'A88888882', 6), (3, 'Innova StartUp', 'B77777773', 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)

Elaborado por [] el [] [Crear código HTML]

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');
```

✓ 5 filas insertadas. La fila de la fila insertada se: 6 (La consulta tardó 0.006 segundos.)

INSERT 5 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')

Elaborado por [] el [] [Crear código HTML]

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);
```

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

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

✓ 11 filas insertadas. (La consulta tardó 0.0007 segundos.)

INSERT INTO incidencia (id, descripcion, estado, usuarioId) VALUES (1, 'No puedo restablecer mi contraseña', 'cerrada', 8), (2, 'El PDF de React da error 404 al descargar', 'abierta', 9), (3, 'Quiero factura del último curso contratado', 'abierta', 5), (4, 'Error al subir documento nuevo', 'abierta', 2), (5, 'La plataforma va lenta por las tardes', 'abierta', 10), (6, 'No puedo acceder al curso de Genética', 'abierta', 19), (7, 'Error en la pasarela de pago', 'cerrada', 21), (8, 'Solicito cambio de contraseña manual', 'abierta', 20), (9, 'Solicito reembolso del curso de Java', 'cerrada', 11), (10, 'No me llegó el correo de confirmación', 'abierta', 13), (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 7.2                                /etc/php/8.3/apache2/php.ini
; mode skips this line and ignores its content if this directive is turned on.
; https://php.net/ini.core.disable_classes
;cgi.check_shebang_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-tmp-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/urldecode
allow_url_fopen = On

```

```

GNU nano 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 =

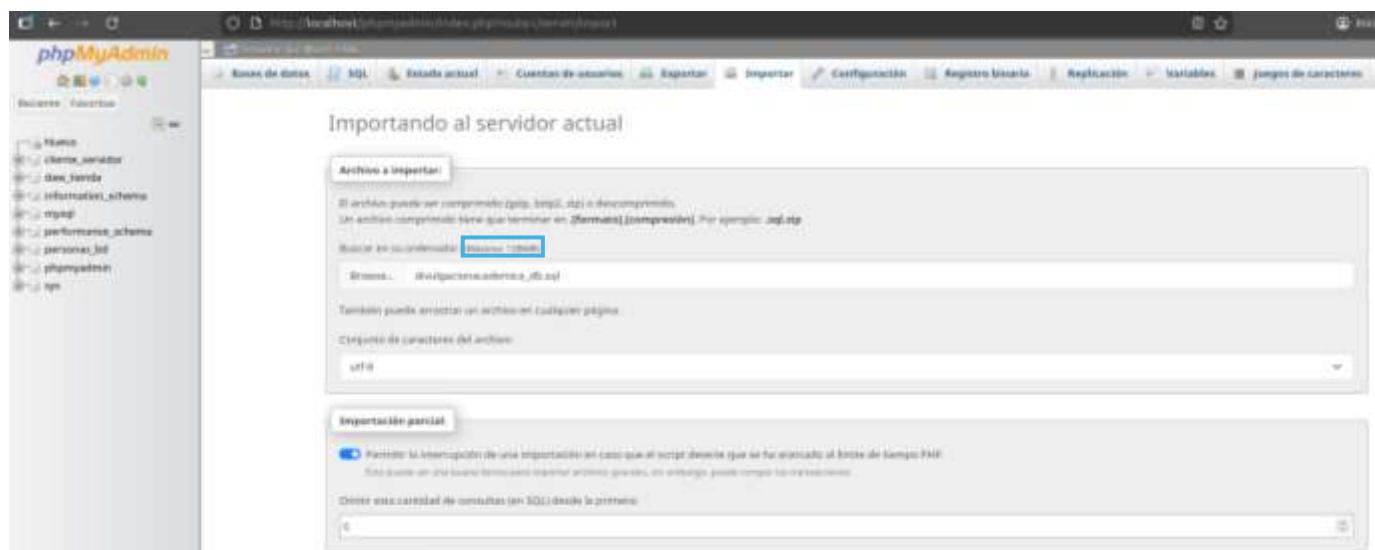
; Automatically add files after PHP document.
; https://php.net/auto-append-file
auto_append_file =

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

```

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 [Coleccion](#), 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 [colección_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 **colección_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 validadación (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, ppxt, 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, ['documentoId' => 'id'])->orderBy(['fecha' => SORT_DESC]); }
public function getValoraciones() { return $this->hasMany(DocumentoValoracion::class, ['documentoId' => '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 asociados 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');
$filtritosAcceso = [];

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

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

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

```

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\NotFoundHttpException('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.');
        }
    }

    return $this->render('view', [
        'model' => $model,
        'nuevoComentario' => $nuevoComentario,
    ]);
}

```

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 actionsabir()
{
    $model = new Documento();

    // Inicio para los despliegues en formulario
    $materias = ArrayHelper::map(Materia::find()->orderBy('nombre')->all(), 'id', 'nombre');
    $instituciones = ArrayHelper::map(App\models\Institucion::find()->orderBy('nombre')->all(), 'id', 'nombre');
    $autores = ArrayHelper::map(App\models\Personas::find()->orderBy('nombre')->all(), 'id', function($p){ return $p->nombres . ' ' . $p->apellidos; });

    if ($request = Yii::$app->request->post()) {

        // 1. Carga 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
                $nombreunico = preg_replace('/[^a-zA-Z0-9]+/','_',$model->título) . '-' . time() . '.' . $model->archivofile->extension;

                // 4. mover el archivo a la carpeta "uploads"
                $rutaGuardada = 'uploads/' . $nombreunico;

                if ($model->archivofile->save($rutaGuardada)) {
                    // 5. Guardar este id dentro en la base de datos
                    $model->archiv_uei = $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, guardarla 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 desplegables.
- ❖ Se hicieron las Tarjetas de Categorías (Matemáticas, Física, etc.).
- ❖ Se añadieron las secciones de "Últimas subidas" y "Estadísticas".
- ❖ Se cambiaron los enlaces normales (``) 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 **colección** y es la fuente de verdad donde se almacenan el título, descripción y el ID del autor.
- ❖ **Modelo (app\models\Colección):** es la representación lógica de la tabla. Utiliza **Active Record** para interactuar con la DB.
- ❖ **Controlador (app\controllers\ColecciónController):** es el "cerebro" que gestiona las peticiones.
- ❖ **Vista (views/colección/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 **Colección::find()->all()**. Esto traduce una orden de PHP a un **SELECT * FROM colección** 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 **\$colección->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);
$stamano = $existeArchivo ? round(filesize($rutaArchivo) / 1024 / 1024, 2) : 'No' : '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
$ext = strtolower(pathinfo($model->archivo_url, PATHINFO_EXTENSION));
$tipoInfo = [
    'pdf' => ['icon' => 'pdf', 'label' => 'PDF (Documento)'],
    'doc' => ['icon' => 'doc', 'label' => 'Word (Documento)'],
    'docx' => ['icon' => 'docx', 'label' => 'Word (Documento)'],
    'xls' => ['icon' => 'xls', 'label' => 'Excel (Hoja de cálculo)'],
    'xlsx' => ['icon' => 'xlsx', 'label' => 'Excel (Hoja de cálculo)'],
    'ppt' => ['icon' => 'ppt', 'label' => 'PowerPoint (Presentación)'],
    'pptx' => ['icon' => 'pptx', 'label' => 'PowerPoint (Presentación)'],
    'mp4' => ['icon' => 'mp4', 'label' => 'Video MP4'],
];
$info = $tipoInfo[$ext] ?? ['icon' => 'default', 'label' => strtoupper($ext) . ' (Archivo)'];
>>>
```

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

```

<?php if ($existeArchivo): ?>
    <a href=<?= Url::to('web/uploads/' . $model->archivo_url) ?> class="btn btn-primary btn-lg px-4" target="_blank" download>
        Descargar <?= strtoupper($ext) ?> (<?= $tamaño ?>)
    </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 $comentario): ?>
    <div class="d-flex align-items-center border-bottom">
        <div class="link-primer-9 ms-2">
            <div class="bg-light rounded-circle d-flex align-items-center justify-content-center text-primary" style="width: 30px; height: 30px; font-size: 10px; border-radius: 50%;">
                <?= strtoupper(substr($comentario->usuario ? $comentario->usuario->nombre : '', 0, 1)) ?>
            </div>
        </div>
        <div>
            <hr class="hr-bold mb-1" />
            <?php if($model->isGuest): ?>
                <div><?php echo($comentario->usuario ? $comentario->usuario->nombre : 'Invitado') ?></div>
            <?php else: ?>
                <small class="text-muted d-block mb-2">
                    <?php $app->formatter->format($comentario->fecha, 'long') ?>
                </small>
                <p class="mb-0 text-dark"><?php echo($comentario->contenido) ?></p>
            </?php endif: ?>
        </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](#)).

```


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


```

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.

```

<?= 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="<?= Html::encode($request->get('q')) ?>">
            </div>
        </div>
        <button type="submit" class="btn btn--primary">Buscar</button>
    </div>
<?= 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, $catSel);
    $icon = '■';
    if (stripos($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 = '🦠';
    ?>
    <div class="form-checkbox">
        <input type="checkbox" id="cat-<?= $cat->id ?>" name="category[]" value="<?= $cat->id ?>" <?= $isChecked ? 'checked' : '' ?>>
        <label for="cat-<?= $cat->id ?>">
            <?= $icon ?> <?= Html::encode($cat->nombre) ?>
            <span class="text-muted small">(<?= $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 ícono 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 $model): ?>
    <article class="material-card card">
        <div class="card__body">
            <div class="flex justify-between align-center mb-4">
                <span class="badge <?= ($model->tipo_acceso == 'privado') ? 'badge--secondary' : 'badge--primary' ?>>
                <?= ($model->tipo_acceso == 'privado') ? 'Examen' : 'Apuntes' ?>
            </span>
            <div class="rating">★ <?= $model->getRatingPromedio() ?></div>
        </div>
        <h3 class="material-card__title">
            <a href="<?= Url::to(['materiales/view', 'id' => $model->id]) ?>">
                <?= Html::encode($model->título) ?>
            </a>
        </h3>
        <p class="material-card__description">Archivo: <?= Html::encode($model->archivo_url) ?></p>
        <div class="material-card__meta">
            <span class="material-card__category">
                <?= $model->materia ? Html::encode($model->materia->nombree) : 'General' ?>
            </span>
        </div>
        <div class="card__footer flex justify-between align-center">
            <span class="material-card__author" style="font-size: 0.8em; color: #666;"><?= $model->autor ? Html::encode($model->autor->nombree) : 'Andelmo' ?></span>
            <span style="font-size: 0.8em; color: #666;">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">
    <?= linkPager::widget([
        'pagination' => $dataProvider->pagination,
        'options' => ['class' => 'pagination-custom'],
        'linkContainerOptions' => ['class' => 'page-item'],
        'linkOptions' => ['class' => 'page-link'],
        'disabledListItemSubTagOptions' => ['class' => 'page-link'],
        // Textos bonitos
        'prevPageLabel' => '< Anterior',
        'nextPageLabel' => 'Siguiente >',
        // 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", "Siguiente >"), 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">
        <?= $form->field($model, 'materia_id')->dropdownList($materias, ['prompt' => 'Selecciona Materia...', 'class' => 'form-select mb-3']) ?>
    </div>
    <div class="col-md-6">
        <?= $form->field($model, 'tipo_acceso')->dropdownList([
            'publico' => 'Público (Apuntes)',
            'privado' => 'Privado (Examenes)'
        ], ['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.

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

Inyecta 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>
```

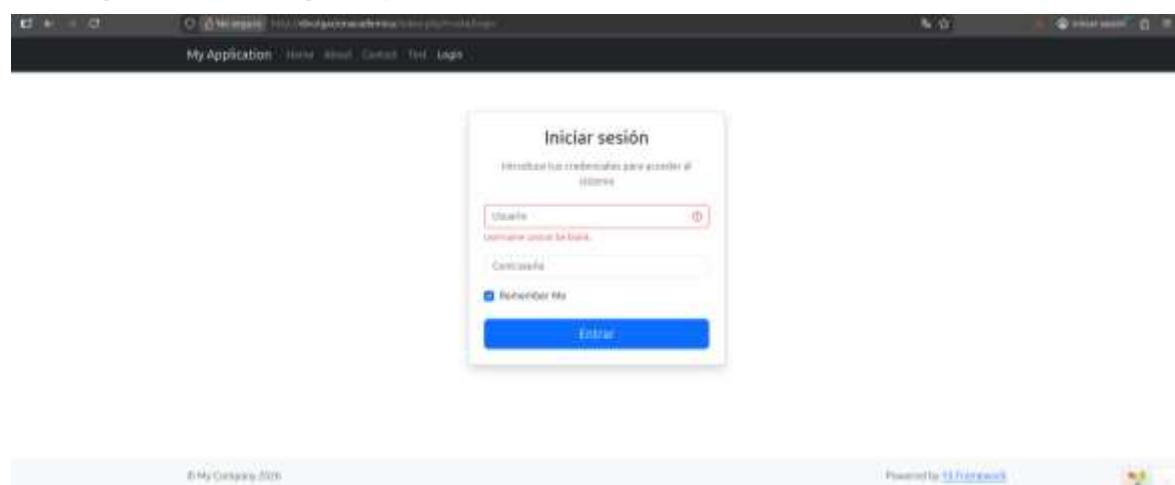
La variable `$content` es el marcador de posición donde Yii2 inserta el resultado de la vista específica que se está visitando como el formulario de subida o el catálogo. Esto permite que el contenido cambie mientras la cabecera y el footer permanecen fijos.

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:

```

1 // Preparamos los datos para el gridview cuando el modelo es null
2 if(Yii::$app->user->isGuest) {
3     return $this->redirect(['site/login']);
4 }
5
6 // Preparamos los datos para el gridview cuando el modelo no es null
7 $dataProvider = new yii\data\ActiveDataProvider([
8     'query' => \app\models\UserLog::find()->orderBy(['created_at' => SORT_DESC]),
9     'pagination' => ['pageSize' => 20],
10 ];
11
12
13 return $this->render('logos', [
14     'dataProvider' => $dataProvider,
15 ]);
16
17 /**
18 * @return \yii\http\Response
19 */
20 public function actionqa()
21 {
22     return $this->render('qa');
23 }
24

```

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

```

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

```

```

18 <button class="qa-summary-btn" disabled>
19   <div class="qa-number">4</div>
20   <div>Respondidas</div>
21 </button>
22
23 <button class="qa-summary-btn" disabled>
24   <div class="qa-number">3</div>
25   <div>Resueltas</div>
26 </button>
27
28 </div>
29
30 <br>
31
32 <!-- FORMULARIO ALUMNO -->
33 <div class="qa-form">
34   <h3>Enviar una pregunta</h3>
35
36   <form>
37     <div class="form-group mb-3">
38       <label>Pregunta:</label>
39       <textarea class="form-control" rows="3" placeholder="Escribe tu pregunta..."></textarea>
40     </div>
41
42     <button class="qa-summary-btn" disabled>
43       Enviar pregunta
44     </button>
45   </form>
46 </div>
47
48 <br>
49
50 <!-- LISTADO DE PREGUNTAS -->
51 <div class="qa-list">
52
53   <h3>Preguntas recientes:</h3>
54
55   <!-- PREGUNTA SIN RESPUESTA -->
56   <div class="qa-item pending">
57     <p class="qa-question">
58       ? ¿Cuándo se publican las notas finales?
59     </p>
60     <p class="qa-meta">
61       Alumno - 12/01/2020 - <span class="badge bg-warning">Pendiente</span>
62     </p>
63   </div>
64
65
66   <!-- PREGUNTA RESPUESTA -->
67   <div class="qa-item answered">
68     <p class="qa-question">
69       ? ¿Se puede entregar la práctica fuera de plazo?
70     </p>
71     <p class="qa-meta">
72       Alumno - 10/01/2020 - <span class="badge bg-success">Respondida</span>
73     </p>
74
75     <div class="qa-answer">
76       <p>✓ Sí, con una penalización del 20% si se entrega dentro de la semana siguiente. </p>
77       <p class="qa-teacher"> Profesor: Juan Pérez </p>
78     </div>
79   </div>
80 </div>
81
82 </div>

```

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

Preguntas y Respuestas

Buscar título
Buscar por autor

| | | | |
|-----------------------|-------------|----------|---|
| 10 | 3 | 4 | 8 |
| Todos los respondidos | Respondidas | Necesita | |

Enviar una pregunta

Prepón tu pregunta...

Preguntas recientes

7 ¿Cuándo se publican las revistas científicas?
Alumno - 12/11/2019 Resuelta

7 ¿Qué puede entregar la policía fuera de plazo?
Alumno - 10/11/2019 Resuelta

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 página
    ];

    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.

```
1 <?php
2
3 namespace app\models;
4
5 use yii\db\ActiveRecord;
6
7
8 class Pregunta extends ActiveRecord
9 {
10     public static function tableName()
11     {
12         return 'pregunta';
13     }
14
15     public function getRespuestas()
16     {
17         return $this->hasMany(Respuesta::class, ['preguntaId' => 'id']);
18     }
19
20     public function getMateria()
21     {
22         return $this->hasOne(Materia::class, ['id' => 'materiaId']);
23     }
24
25     public function getUsuario()
26     {
27         return $this->hasOne(Usuario::class, ['id' => 'usuarioId']);
28     }
29 }
```

```
1 Pregunta.php
2 Respuesta.php X
3 site.css
4
5 <?php
6
7 namespace app\models;
8
9 use yii\db\ActiveRecord;
10
11
12 class Respuesta extends ActiveRecord
13 {
14     public static function tableName()
15     {
16         return 'respuestas';
17     }
18
19     public function getUsuario()
20     {
21         return $this->hasOne(Usuario::class, ['id' => 'usuarioId']);
22     }
23 }
```

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 x QuController.php Pregunta.php Request.php site.css

<?php

namespace app\controllers;

use yii;
use yii\filters\AccessControl;
use yii\web\Controller;
use yii\web\Response;
use yii\filters\VerbFilter;
use app\models\LoginForm;
use app\models\ContactForm;
use app\models\Registrarsuarios; // Necesario para el formulario de registro
use app\models\Pregunta; // Importa el modelo Pregunta para acceder a los registros almacenados en la base de datos

class SiteController extends Controller
{
    /**
     * @inheritdoc
     */
    public function behaviors()
    {
        return [
            'access' => [
                'class' => AccessControl::class,
                'only' => ['logout', 'login'], // Activar 'login' si control de acceso al sistema protegido
                'rules' => [
                    ...
                ]
            ]
        ];
    }
}
```

Después, se creará un controlador para el O&A, al que se llamará `codigo/controllers/OandAController.php`:

```

create.php QandaController.php main.php
<?php

namespace app\controllers;

use yii;
use yii\web\Controller;
use app\models\Preguntas; // Importar modelo
use app\models\Respuestas; // Importar modelo
use yii\filters\AccessControl;

class QandaController extends Controller
{
    /**
     * configuración de acceso (permissions) definir quien puede ver esto
     */
    public function behaviors()
    {
        return [
            'access' => [
                'class' => AccessControl::class,
                'only' => ['create', 'responder'], // solo proteger estas acciones
                'rules' => [
                    // create y responder solo para usuarios logueados
                    [
                        'actions' => ['create', 'responder'],
                        'allow' => true,
                        'roles' => ['@'],
                    ],
                    // index y view accesibles para todos
                    [
                        'actions' => ['index', 'view'],
                        'allow' => true,
                    ],
                ],
            ],
        ];
    }

    /**
     * Muestra la página principal de preguntas
     * URL: index.php/qanda/index
     */
    public function actionIndex()
    {
        // 1. obtener preguntas
        $spreguntas = Preguntas::find()
            ->with(['user_id', 'respuestas.user_id'])
            ->orderBy(['fecha_creacion' => SORT_DESC])
            ->all();

        // 2. Generar resumen
        $sresumen = [
            'total' => Preguntas::find()->count(),
            'sin_responder' => Preguntas::find()->where(['estado' => 'sin_responder'])->count(),
            'respondida' => Preguntas::find()->where(['estado' => 'respondida'])->count(),
            'resuelta' => Preguntas::find()->where(['estado' => 'resuelta'])->count(),
        ];

        // 3. renderizar la vista
        return $this->render('index', [
            'preguntas' => $spreguntas,
            'resumen' => $sresumen,
        ]);
    }

    /**
     * Muestra el detalle de una pregunta
     * URL: index.php/qanda/view/1
     */
    public function actionView($id)
    {
        $model = Preguntas::findOne($id);

        if (!$model) {
            throw new \yii\web\NotFoundHttpException('La pregunta no existe');
        }

        // Redirecciona la vista 'view.php' que admite los count
        return $this->render('view', [
            'model' => $model,
        ]);
    }
}

```

```
create.php          QandAController.php      main.php        index.php

92     public function actionResponder($id)
93     {
94         $pregunta = Pregunta::findOne($id);
95         if (!$pregunta) {
96             throw new \yii\web\NotFoundHttpException();
97         }
98
99         if (Yii::$app->request->isPost) {
100             $contenido = Yii::$app->request->post('contenido_resuesta');
101
102             if (!empty($contenido)) {
103                 $respuesta = new Respuesta();
104                 $respuesta->pregunta_id = $pregunta->id;
105                 $respuesta->usuario_id = Yii::$app->user->id;
106                 $respuesta->contenido = $contenido;
107                 // $respuesta->fecha_creacion = date('Y-m-d H:i:s'); // Si lo uso no lo pone automáticamente
108
109                 if ($respuesta->save()) {
110                     Yii::$app->session->setFlash('success', '(Respuesta publicada)');
111
112                     // Cambiar estado a 'answered' si estaba pendiente
113                     if ($pregunta->estado == 'pending') {
114                         $pregunta->estado = 'answered';
115                         $pregunta->save();
116                     }
117                 } else {
118                     Yii::$app->session->setFlash('error', 'Error al guardar la respuesta.');
119                 }
120             }
121
122             return $this->redirect(['view', 'id' => $id]);
123         }
124     }
```

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:

```
79 //--- Listado de preguntas --->
80 <div class="qa-list">
81     <?php if (empty($preguntas)) : ?>
82         <div class="alert alert-info text-center py-5">
83             <ul class="list-group list-group-flush fs-1"></ul><br>
84             No se encontraron preguntas con estos criterios.
85         </div>
86     <?php else: ?>
87         <?php foreach ($preguntas as $model): ?>
88             <div class="card mb-3 shadow-sm hover-effect">
89                 <div class="card-body">
90                     <div class="d-flex justify-content-between align-items-start flex-wrap">
91                         <div>
92                             <div class="mb-2">
93                                 <?php
94                                     $claseEstado = match($model->estado) {
95                                         'resolved' => 'success',
96                                         'answered' => 'info',
97                                         default => 'warning text-dark'
98                                     };
99                                 ?>
100                                 <span class="badge bg- <?= $claseEstado ?>">
101                                     <?= ucfirst($model->estado) ?>
102                                 </span>
103                             </div>
104                         </div>
105                     </div>
106                 </div>
107             </div>
108         <?php endforeach; ?>
109     </div>
110 
```

```

96     <h4 class="card-title mb-1">
97         <a href="#{$this->url(['view', 'id' => $model->id])}">{$model->título}</a>
98     </h4>
99
100
101     <div class="text-muted small mt-2">
102         <i class="bi bi-person"></i> <?= Html::encode($model->usuariousername ?? 'Anónimo') ?: null;
103         <i class="bi bi-calendar"></i> <?= Yii::$app->formatter->asRelativetime($model->fecha_creacion) ?: null;
104         <i class="bi bi-chat-dots"></i> <?= count($model->respuestas) ?: respuestas
105     </div>
106     </div>
107     </div>
108     </div>
109     <?php endforeach; ?>
110     <?php endif; ?>
111 </div>
112
113     </div>
114 </div>

```

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

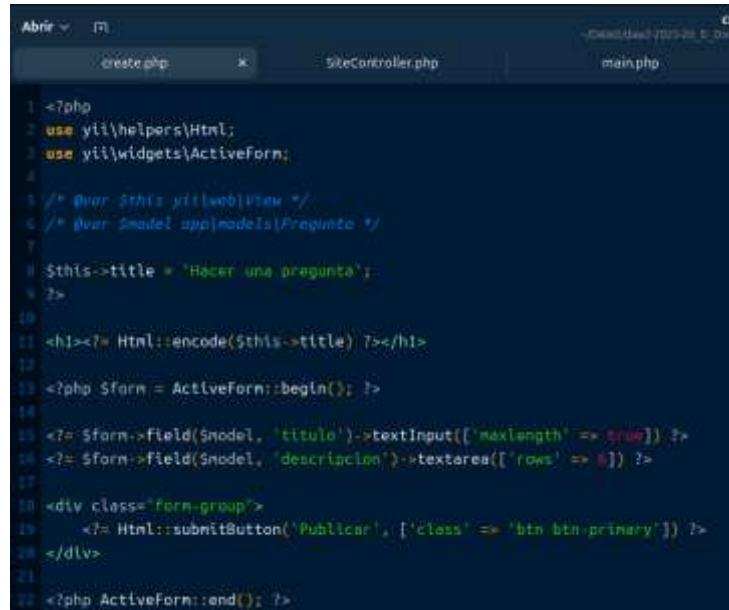
Para hacer una pregunta, una vez se ha iniciado sesión, en [codigo/controllers/QandAController.php](#), se añade [actionCreate\(\)](#).

```

QandAController.php      create.php      SiteController.php
111     return $this->redirect(['view', 'id' => $id]);
112 }
113
114 /**
115 * Crear una nueva pregunta
116 * URL: index.php?r=qanda/create
117 */
118 public function actionCreate()
119 {
120     $model = new Pregunta();
121
122     if ($model->load(Yii::$app->request->post()) && $model->save()) {
123         Yii::$app->session->setFlash('success', '(Pregunta publicada)');
124         return $this->redirect(['view', 'id' => $model->id]);
125     }
126
127     return $this->render('create', [
128         'model' => $model,
129     ]);
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 /* Over class yii\web\View */
6 /* Over 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' => true]) ?>
16 <?= $form->field($model, 'descripcion')->textarea(['rows' => 8]) ?>
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 page titled "Hacer una pregunta". The page has two input fields: "Título" and "Descripción". Below the fields is a blue "Publicar" button. At the bottom of the page is a dark footer bar with three sections: "Sobre nosotros", "Enlaces rápidos", and "Legal".

- Sobre nosotros:** Plataforma colaborativa para la divulgación académica y científica. Colegiando conocimiento.
- Enlaces rápidos:**
 - Guía de uso
 - Normas de la comunidad
 - Preguntas frecuentes
- 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/

- colección/
 - 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

❖