

# EcuaSistencia 2026

## Documentación Completa del Proyecto PHP

---

Total de archivos documentados: 83

## Estructura de Carpetas

---

```
FilesImport/  
ARCHITECTURE.md  
README.md  
ResumenEjecutivo_04C.md  
composer.json  
ArchImp/  
config/  
config.php  
database.php  
controllers/  
AcademicController.php  
AssignmentController.php  
AttendanceController.php  
AuthController.php  
BackupController.php  
DashboardController.php  
ExportController.php  
InstitutionController.php  
JustificationController.php  
ProfileController.php  
ReportController.php  
ReportController2.php  
RepresentativeController.php  
ScheduleController.php  
SearchController.php  
StatsController.php  
UserController.php  
database/  
ecuasistencia2026_db.sql  
ecuasistencia2026_db0.sql  
ecuasistencia2026_dbX.sql  
helpers/  
Backup.php  
Logger.php  
Mailer.php  
Security.php  
models/  
Attendance.php  
ClassSchedule.php  
Course.php  
Institution.php  
InstitutionShift.php  
Justification.php  
Notification.php  
Representative.php  
Role.php  
SchoolYear.php  
Shift.php  
Subject.php  
TeacherAssignment.php  
User.php  
views/  
academic/  
course_edit.php  
course_students.php  
create.php  
edit.php
```

enroll.php  
index.php  
subject\_edit.php  
assignments/  
index.php  
tutor.php  
view\_course.php  
attendance/  
calendar.php  
course\_view.php  
my\_attendance.php  
register.php  
view.php  
auth/  
forgot.php  
login.php  
register.php  
reset.php  
backup/  
index.php  
dashboard/  
index.php  
institution/  
index.php  
justifications/  
my\_list.php  
pending.php  
submit.php  
partials/  
navbar.php  
profile/  
change\_password.php  
edit.php  
view.php  
reports/  
index.php  
representatives/  
child\_attendance.php  
manage.php  
my\_children.php  
schedule/  
index.php  
manage.php  
stats/  
index.php  
users/  
create.php  
edit.php  
index.php

# Índice de Archivos

---

## ■ ArchImp/config/

- 2. config.php
- 3. database.php

## ■ ArchImp/controllers/

- 4. AcademicController.php
- 5. AssignmentController.php
- 6. AttendanceController.php
- 7. AuthController.php
- 8. BackupController.php
- 9. DashboardController.php
- 10. ExportController.php
- 11. InstitutionController.php
- 12. JustificationController.php
- 13. ProfileController.php
- 14. ReportController.php
- 15. ReportController2.php
- 16. RepresentativeController.php
- 17. ScheduleController.php
- 18. SearchController.php
- 19. StatsController.php
- 20. UserController.php

## ■ ArchImp/database/

- 21. ecuasistencia2026\_db.sql
- 22. ecuasistencia2026\_db0.sql
- 23. ecuasistencia2026\_dbX.sql

## ■ ArchImp/helpers/

- 24. Backup.php
- 25. Logger.php
- 26. Mailer.php
- 27. Security.php

## ■ ArchImp/models/

- 28. Attendance.php
- 29. ClassSchedule.php
- 30. Course.php
- 31. Institution.php
- 32. InstitutionShift.php
- 33. Justification.php
- 34. Notification.php
- 35. Representative.php
- 36. Role.php

- 37. SchoolYear.php
- 38. Shift.php
- 39. Subject.php
- 40. TeacherAssignment.php
- 41. User.php

#### ■ ArchImp/views/academic/

- 42. course\_edit.php
- 43. course\_students.php
- 44. create.php
- 45. edit.php
- 46. enroll.php
- 47. index.php
- 48. subject\_edit.php

#### ■ ArchImp/views/assignments/

- 49. index.php
- 50. tutor.php
- 51. view\_course.php

#### ■ ArchImp/views/attendance/

- 52. calendar.php
- 53. course\_view.php
- 54. my\_attendance.php
- 55. register.php
- 56. view.php

#### ■ ArchImp/views/auth/

- 57. forgot.php
- 58. login.php
- 59. register.php
- 60. reset.php

#### ■ ArchImp/views/backup/

- 61. index.php

#### ■ ArchImp/views/dashboard/

- 62. index.php

#### ■ ArchImp/views/institution/

- 63. index.php

#### ■ ArchImp/views/justifications/

- 64. my\_list.php
- 65. pending.php

66. submit.php

## ■ ArchImp/views/partials/

67. navbar.php

## ■ ArchImp/views/profile/

68. change\_password.php

69. edit.php

70. view.php

## ■ ArchImp/views/reports/

71. index.php

## ■ ArchImp/views/representatives/

72. child\_attendance.php

73. manage.php

74. my\_children.php

## ■ ArchImp/views/schedule/

75. index.php

76. manage.php

## ■ ArchImp/views/stats/

77. index.php

## ■ ArchImp/views/users/

78. create.php

79. edit.php

80. index.php

## ■ raíz/

1. ARCHITECTURE.md

81. README.md

82. ResumenEjecutivo\_04C.md

83. composer.json

# Código Fuente Completo

## 1. ARCHITECTURE.md

Tu sistema tiene estructura tipo MVC clásico en PHP (sin framework pesado), así que para darle continuidad sin perder la perspectiva debes enfocarte en los archivos que controlan:

- Flujo del sistema

- Seguridad

- Acceso a datos

- Lógica de negocio

Te lo explico estratégicamente.

- 1. El CEREBRO del sistema (Lo más importante)

- /config

Aquí está la base del sistema.

Archivos críticos:

database.php → conexión a BD

config.php → constantes globales

posibles rutas base, URL\_BASE, etc.

- Si esto falla, nada funciona.

- Aquí defines entorno (local / producción).

- /controllers ■ (Muy importante)

Este es el flujo principal del sistema.

Aquí están:

AttendanceController.php

UserController.php

AuthController.php

ReportController.php

JustificationController.php

etc.

- Aquí debes mirar:

Cómo se reciben \$\_GET['action']

Cómo se enrutan las vistas

Cómo se llaman los modelos

- Si quieres entender el sistema, empieza leyendo:

public/index.php

Luego el archivo principal que maneje el router

Después los controllers

## ■ 2. La LÓGICA del sistema

■ /models ■

Aquí está la lógica que habla con la base de datos.

Ejemplos:

User.php

Attendance.php

Course.php

Justification.php

Aquí debes revisar:

Consultas SQL

Métodos como:

getById()

create()

update()

delete()

getByStudent()

■ Si quieres ampliar el sistema sin romperlo, este es el lugar más delicado.

## ■■ 3. La BASE DE DATOS

■ /database

Aquí normalmente encuentras:

Script SQL inicial

Migraciones

Estructura de tablas

■ Muy importante revisar:

Relaciones

Foreign Keys

Índices

Campos ENUM (como status: presente, ausente, justificado)

Si no entiendes la BD, no podrás escalar el sistema correctamente.

## ■■ 4. Las VISTAS (UI)

■ /views

Están organizadas por módulos:



attendance

users

reports

justifications

dashboard

etc.

■ Aquí solo hay presentación (HTML + PHP básico).

■ No debería haber lógica compleja.

■■ Si encuentras consultas SQL aquí → mala práctica.

■ 5. Seguridad

■ /helpers

Muy importante revisar:

Security.php

funciones de validación

control de sesiones

permisos por rol

Ejemplo:

```
Security::requireLogin();
```

```
Security::requireRole('admin');
```

Si vas a crecer el sistema (representantes, docentes, estudiantes), esto es CLAVE.

■ 6. Punto de Entrada

■ /public

Aquí normalmente está:

index.php

assets

js

css

■■ Este es el archivo que recibe TODAS las peticiones.

Si quieres entender el flujo completo:

index.php

router

controller

model

view

- 7. Lo que NO debes tocar mucho
- /vendor

Esto es Composer.

PHPMailer

PhpSpreadsheet

TCPDF

ZipStream

- No modifiques nada aquí.
- Solo se actualiza con composer.

- 8. uploads

Aquí guardas:

justificativos

archivos PDF

evidencias

Debes cuidar:

Validación de tipo de archivo

Seguridad (evitar ejecución .php)

Tamaño máximo

- Si quieres continuidad SIN perder perspectiva

Te recomiendo este orden mental:

- 1■■■ Entender el flujo completo

Abre:

public/index.php

Y sigue el recorrido hasta una vista.

Haz un mapa mental:

Request → Controller → Model → View

- 2■■■ Documentar la base de datos

Haz un esquema visual:

users

roles

courses

attendance

justifications

representatives

Si entiendes la BD, entiendes el 70% del sistema.

### ■ 3■■■ Definir módulos del sistema

Tu sistema tiene módulos claros:

#### ■ Autenticación

#### ■■■ Usuarios

#### ■ Académico

#### ■ Asistencia

#### ■ Justificaciones

#### ■ Reportes

#### ■ Backups

Documenta qué hace cada uno.

#### ■ Resumen Estratégico

Si solo pudieras revisar 6 cosas importantes:

#### ■ public/index.php

#### ■ config/database.php

#### ■ controllers (especialmente Attendance y Auth)

#### ■ models (Attendance, User, Justification)

#### ■ helpers/Security.php

#### ■ estructura de la base de datos

#### ■ Consejo Profesional

Para no perder perspectiva:

- ✓■ Crea un archivo ARCHITECTURE.md
- ✓■ Dibuja el flujo
- ✓■ Define responsabilidades por carpeta
- ✓■ No mezcles lógica en vistas

## 2. ArchImp/config/config.php

```
<?php
// Zona horaria Ecuador
date_default_timezone_set('America/Guayaquil');

// Definir constantes ANTES de todo
define('BASE_PATH', __DIR__ . '/../');
define('BASE_URL', 'http://localhost/ecuasistencia2026');
define('EDIT_ATTENDANCE_HOURS', 48);

// Configuración de sesión ANTES de session_start()
```

```

ini_set('session.gc_maxlifetime', 86400); // 24 horas
session_set_cookie_params([
    'lifetime' => 86400, // 24 horas
    'path' => '/',
    'domain' => '',
    'secure' => false, // true si usas HTTPS
    'httponly' => true, // Protección XSS
    'samesite' => 'Lax' // Protección CSRF
]);

// Iniciar sesión DESPUÉS de configurar
session_start();

// Timeout de inactividad: 30 minutos
$inactive_timeout = 1800; // 30 minutos en segundos
if (isset($_SESSION['last_activity']) &&
    (time() - $_SESSION['last_activity'] > $inactive_timeout)) {
    session_unset();
    session_destroy();
    header('Location: ' . BASE_URL . '/public/?action=login&timeout=1');
    exit;
}
$_SESSION['last_activity'] = time();

// SMTP Config
define('SMTP_HOST', 'smtp.gmail.com');
define('SMTP_PORT', 587);
define('SMTP_USER', 'viktorengel@gmail.com');
define('SMTP_PASS', 'Orktvi.5/*83gM');
define('SMTP_FROM', 'noreply@ecuasist.edu.ec');
define('SMTP_NAME', 'EcuAsist2026');

require_once BASE_PATH . '/config/database.php';
require_once BASE_PATH . '/helpers/Security.php';

```

### 3. ArchImp/config/database.php

```

<?php
class Database {
    private $host = 'localhost';
    private $db = 'ecuasistencia2026_db';
    private $user = 'root';
    private $pass = '';
    private $charset = 'utf8mb4';
    private $pdo;

    public function connect() {
        if ($this->pdo === null) {
            $dsn = "mysql:host={$this->host};dbname={$this->db};charset={$this->charset}";
            $options = [
                PDO::ATTR_ERRMODE => PDO::ERRMODE_EXCEPTION,
                PDO::ATTR_DEFAULT_FETCH_MODE => PDO::FETCH_ASSOC,
                PDO::ATTR_EMULATE_PREPARES => false,
            ];

            try {
                $this->pdo = new PDO($dsn, $this->user, $this->pass, $options);
            } catch (PDOException $e) {
                die('Error de conexión: ' . $e->getMessage());
            }
        }
        return $this->pdo;
    }
}

```

```
}
```

## 4. ArchImp/controllers/AcademicController.php

```
<?php
require_once BASE_PATH . '/config/config.php';
require_once BASE_PATH . '/models/Course.php';
require_once BASE_PATH . '/models/Subject.php';
require_once BASE_PATH . '/models/SchoolYear.php';
require_once BASE_PATH . '/models/Shift.php';
require_once BASE_PATH . '/models/User.php';

class AcademicController {
private $courseModel;
private $subjectModel;
private $schoolYearModel;
private $shiftModel;
private $userModel;

public function __construct() {
Security::requireLogin();
if (!Security::hasRole('autoridad')) {
die('Acceso denegado');
}

$db = new Database();
$this->courseModel = new Course($db);
$this->subjectModel = new Subject($db);
$this->schoolYearModel = new SchoolYear($db);
$this->shiftModel = new Shift($db);
$this->userModel = new User($db);
}

public function index() {
$courses = $this->courseModel->getAll();
$subjects = $this->subjectModel->getAll();
$schoolYears = $this->schoolYearModel->getAll();
$shifts = $this->shiftModel->getAll();

include BASE_PATH . '/views/academic/index.php';
}

public function createCourse() {
if ($_SERVER['REQUEST_METHOD'] === 'POST') {
$activeYear = $this->schoolYearModel->getActive();

if (!$activeYear) {
header('Location: ?action=academic&error=no_active_year');
exit;
}

$data = [
':institution_id' => $_SESSION['institution_id'],
':school_year_id' => $activeYear['id'],
':name' => Security::sanitize($_POST['name']),
':grade_level' => Security::sanitize($_POST['grade_level']),
':parallel' => Security::sanitize($_POST['parallel']),
':shift_id' => (int)$_POST['shift_id']
];

$this->courseModel->create($data);
header('Location: ?action=academic&course_success=1');
exit;
}
```

```

}
}

public function createSubject() {
if ($_SERVER['REQUEST_METHOD'] === 'POST') {
$data = [
':institution_id' => $_SESSION['institution_id'],
':name' => Security::sanitize($_POST['name']),
':code' => Security::sanitize($_POST['code'])
];

$this->subjectModel->create($data);
header('Location: ?action=academic&subject_success=1');
exit;
}
}

// =====
// CRUD CURSOS
// =====

public function editCourse() {
$courseId = (int)$_GET['id'];
$course = $this->courseModel->findById($courseId);

if (!$course || $course['institution_id'] != $_SESSION['institution_id']) {
header('Location: ?action=academic&error=course_not_found');
exit;
}

if ($_SERVER['REQUEST_METHOD'] === 'GET') {
$shifts = $this->shiftModel->getAll();
include BASE_PATH . '/views/academic/course_edit.php';
return;
}

if ($_SERVER['REQUEST_METHOD'] === 'POST') {
$data = [
'id' => $courseId,
'name' => Security::sanitize($_POST['name']),
'grade_level' => Security::sanitize($_POST['grade_level']),
'parallel' => Security::sanitize($_POST['parallel']),
'shift_id' => (int)$_POST['shift_id']
];

if ($this->courseModel->update($data)) {
header('Location: ?action=academic&course_updated=1');
exit;
} else {
$errors[] = "Error al actualizar el curso";
$shifts = $this->shiftModel->getAll();
include BASE_PATH . '/views/academic/course_edit.php';
}
}

public function deleteCourse() {
if ($_SERVER['REQUEST_METHOD'] === 'POST') {
$courseId = (int)$_POST['course_id'];
$course = $this->courseModel->findById($courseId);

if (!$course || $course['institution_id'] != $_SESSION['institution_id']) {
header('Location: ?action=academic&error=course_not_found');
exit;
}
}
}

```

```

}

// Verificar si tiene estudiantes matriculados
$students = $this->courseModel->getEnrolledStudents($courseId);
if (count($students) > 0) {
header('Location: ?action=academic&error=course_has_students');
exit;
}

// Verificar si tiene asignaciones docentes
$db = new Database();
$stmt = $db->connect()->prepare("SELECT COUNT(*) as count FROM teacher_assignments WHERE course_id = :id");
$stmt->execute([':id' => $courseId]);
$result = $stmt->fetch();

if ($result['count'] > 0) {
header('Location: ?action=academic&error=course_has_assignments');
exit;
}

if ($this->courseModel->delete($courseId)) {
header('Location: ?action=academic&course_deleted=1');
exit;
} else {
header('Location: ?action=academic&error=delete_failed');
exit;
}
}

// =====
// CRUD ASIGNATURAS
// =====

public function editSubject() {
$subjectId = (int)$_GET['id'];
$subject = $this->subjectModel->findById($subjectId);

if (!$subject || $subject['institution_id'] != $_SESSION['institution_id']) {
header('Location: ?action=academic&error=subject_not_found');
exit;
}

if ($_SERVER['REQUEST_METHOD'] === 'GET') {
include BASE_PATH . '/views/academic/subject_edit.php';
return;
}

if ($_SERVER['REQUEST_METHOD'] === 'POST') {
$data = [
'id' => $subjectId,
'name' => Security::sanitize($_POST['name']),
'code' => Security::sanitize($_POST['code'])
];

if ($this->subjectModel->update($data)) {
header('Location: ?action=academic&subject_updated=1');
exit;
} else {
$errors[] = "Error al actualizar la asignatura";
include BASE_PATH . '/views/academic/subject_edit.php';
}
}
}

```

```

public function deleteSubject() {
    if ($_SERVER['REQUEST_METHOD'] === 'POST') {
        $subjectId = (int)$_POST['subject_id'];
        $subject = $this->subjectModel->findById($subjectId);

        if (!$subject || $subject['institution_id'] != $_SESSION['institution_id']) {
            header('Location: ?action=academic&error=subject_not_found');
            exit;
        }

        // Verificar si tiene asignaciones docentes
        $db = new Database();
        $stmt = $db->connect()->prepare("SELECT COUNT(*) as count FROM teacher_assignments WHERE subject_id = :id");
        $stmt->execute([':id' => $subjectId]);
        $result = $stmt->fetch();

        if ($result['count'] > 0) {
            header('Location: ?action=academic&error=subject_has_assignments');
            exit;
        }

        if ($this->subjectModel->delete($subjectId)) {
            header('Location: ?action=academic&subject_deleted=1');
            exit;
        } else {
            header('Location: ?action=academic&error=delete_failed');
            exit;
        }
    }
}

public function enrollStudents() {
    $activeYear = $this->schoolYearModel->getActive();
    $courses = $this->courseModel->getAll();
    $availableStudents = $this->userModel->getStudentsNotEnrolled($activeYear['id']);
    $allStudents = $this->userModel->getByRole('estudiante');

    if ($_SERVER['REQUEST_METHOD'] === 'POST') {
        $courseId = (int)$_POST['course_id'];
        $studentIds = $_POST['student_ids'] ?? [];

        $enrolled = 0;
        $errors = 0;

        foreach ($studentIds as $studentId) {
            if ($this->courseModel->enrollStudent($courseId, (int)$studentId, $activeYear['id'])) {
                $enrolled++;
            } else {
                $errors++;
            }
        }

        header('Location: ?action=enroll_students&enrolled=' . $enrolled . '&errors=' . $errors);
        exit;
    }

    include BASE_PATH . '/views/academic/enroll.php';
}

public function unenrollStudent() {
    if ($_SERVER['REQUEST_METHOD'] === 'POST') {
        $studentId = (int)$_POST['student_id'];
        $activeYear = $this->schoolYearModel->getActive();

        if (!$activeYear) {

```



```

header('Location: ?action=enroll_students&error=no_active_year');
exit;
}

// Verificar que el estudiante existe y está matriculado
$course = $this->userModel->getStudentCourse($studentId, $activeYear['id']);

if (!$course) {
header('Location: ?action=enroll_students&error=not_enrolled');
exit;
}

// Retirar estudiante
if ($this->courseModel->unenrollStudent($studentId, $activeYear['id'])) {
header('Location: ?action=enroll_students&unenrolled=1');
exit;
} else {
header('Location: ?action=enroll_students&error=unenroll_failed');
exit;
}
}
}

public function viewCourseStudents() {
$courseId = (int)($_GET['course_id'] ?? 0);

if (!$courseId) {
header('Location: ?action=academic');
exit;
}

$course = $this->courseModel->getAll();
$course = array_filter($course, fn($c) => $c['id'] == $courseId);
$course = reset($course);

$students = $this->courseModel->getEnrolledStudents($courseId);

include BASE_PATH . '/views/academic/course_students.php';
}

// =====
// CRUD AÑOS LECTIVOS
// =====

public function createSchoolYear() {
if ($_SERVER['REQUEST_METHOD'] === 'GET') {
include BASE_PATH . '/views/academic/school_year_create.php';
return;
}

if ($_SERVER['REQUEST_METHOD'] === 'POST') {
$errors = [];

// Validaciones
if (empty($_POST['name'])) {
$errors[] = "El nombre es obligatorio";
}
if (empty($_POST['start_date'])) {
$errors[] = "La fecha de inicio es obligatoria";
}
if (empty($_POST['end_date'])) {
$errors[] = "La fecha de fin es obligatoria";
}

if (!empty($_POST['start_date']) && !empty($_POST['end_date'])) {

```

```

if (strtotime($_POST['end_date']) <= strtotime($_POST['start_date'])) {
$errors[] = "La fecha de fin debe ser posterior a la fecha de inicio";
}

// Verificar solapamiento de fechas
if ($this->schoolYearModel->checkOverlap($_POST['start_date'], $_POST['end_date'])) {
$errors[] = "Ya existe un año lectivo con fechas que se solapan";
}
}

if (!empty($errors)) {
include BASE_PATH . '/views/academic/school_year_create.php';
return;
}

// Crear año lectivo
$data = [
'institution_id' => $_SESSION['institution_id'],
'name' => Security::sanitize($_POST['name']),
'start_date' => $_POST['start_date'],
'end_date' => $_POST['end_date'],
'is_active' => isset($_POST['is_active']) ? 1 : 0
];

// Si se marca como activo, desactivar los demás
if ($data['is_active']) {
$this->schoolYearModel->activate(0); // Desactiva todos
}

if ($this->schoolYearModel->create($data)) {
// Si se creó como activo, activar el nuevo
if ($data['is_active']) {
$db = new Database();
$newId = $db->connect()->lastInsertId();
$this->schoolYearModel->activate($newId);
}

header('Location: ?action=academic&sy_created=1');
exit;
} else {
$errors[] = "Error al crear el año lectivo";
include BASE_PATH . '/views/academic/school_year_create.php';
}
}

public function editSchoolYear() {
$yearId = (int)$_GET['id'];
$year = $this->schoolYearModel->findById($yearId);

if (!$year) {
header('Location: ?action=academic&error=year_not_found');
exit;
}

// Verificar institución
if ($year['institution_id'] != $_SESSION['institution_id']) {
die('Acceso denegado');
}

if ($_SERVER['REQUEST_METHOD'] === 'GET') {
include BASE_PATH . '/views/academic/school_year_edit.php';
return;
}
}

```

```

if ($_SERVER['REQUEST_METHOD'] === 'POST') {
$errors = [];

// Validaciones
if (empty($_POST['name'])) {
$errors[] = "El nombre es obligatorio";
}
if (empty($_POST['start_date'])) {
$errors[] = "La fecha de inicio es obligatoria";
}
if (empty($_POST['end_date'])) {
$errors[] = "La fecha de fin es obligatoria";
}

if (!empty($_POST['start_date']) && !empty($_POST['end_date'])) {
if (strtotime($_POST['end_date']) <= strtotime($_POST['start_date'])) {
$errors[] = "La fecha de fin debe ser posterior a la fecha de inicio";
}
}

// Verificar solapamiento (excluyendo el actual)
if ($this->schoolYearModel->checkOverlap($_POST['start_date'], $_POST['end_date'], $yearId)) {
$errors[] = "Ya existe otro año lectivo con fechas que se solapan";
}
}

if (!empty($errors)) {
include BASE_PATH . '/views/academic/school_year_edit.php';
return;
}

// Actualizar año lectivo
$data = [
'id' => $yearId,
'name' => Security::sanitize($_POST['name']),
'start_date' => $_POST['start_date'],
'end_date' => $_POST['end_date']
];

if ($this->schoolYearModel->update($data)) {
header('Location: ?action=academic&sy_updated=1');
exit;
} else {
$errors[] = "Error al actualizar el año lectivo";
include BASE_PATH . '/views/academic/school_year_edit.php';
}
}

public function deleteSchoolYear() {
if ($_SERVER['REQUEST_METHOD'] === 'POST') {
$yearId = (int)$_POST['year_id'];
$year = $this->schoolYearModel->findById($yearId);

if (!$year) {
header('Location: ?action=academic&error=year_not_found');
exit;
}

// Verificar institución
if ($year['institution_id'] != $_SESSION['institution_id']) {
die('Acceso denegado');
}

// No permitir eliminar año activo
if ($year['is_active'] == 1) {

```

```

header('Location: ?action=academic&error=cannot_delete_active');
exit;
}

if ($this->schoolYearModel->delete($yearId)) {
header('Location: ?action=academic&sy_deleted=1');
exit;
} else {
header('Location: ?action=academic&error=has_courses');
exit;
}
}
}

public function activateSchoolYear() {
if ($_SERVER['REQUEST_METHOD'] === 'POST') {
$yearId = (int)$_POST['year_id'];
$year = $this->schoolYearModel->findById($yearId);

if (!$year) {
header('Location: ?action=academic&error=year_not_found');
exit;
}

// Verificar institución
if ($year['institution_id'] != $_SESSION['institution_id']) {
die('Acceso denegado');
}

if ($this->schoolYearModel->activate($yearId)) {
header('Location: ?action=academic&sy_activated=1');
exit;
} else {
header('Location: ?action=academic&error=activate_failed');
exit;
}
}
}

public function deactivateSchoolYear() {
if ($_SERVER['REQUEST_METHOD'] === 'POST') {
$yearId = (int)$_POST['year_id'];
$year = $this->schoolYearModel->findById($yearId);

if (!$year) {
header('Location: ?action=academic&error=year_not_found');
exit;
}

// Verificar institución
if ($year['institution_id'] != $_SESSION['institution_id']) {
die('Acceso denegado');
}

if ($this->schoolYearModel->deactivate($yearId)) {
header('Location: ?action=academic&sy_deactivated=1');
exit;
} else {
header('Location: ?action=academic&error=deactivate_failed');
exit;
}
}
}
}
}

```

## 5. ArchImp/controllers/AssignmentController.php

```
<?php
require_once BASE_PATH . '/config/config.php';
require_once BASE_PATH . '/models/TeacherAssignment.php';
require_once BASE_PATH . '/models/Course.php';
require_once BASE_PATH . '/models/Subject.php';
require_once BASE_PATH . '/models/SchoolYear.php';
require_once BASE_PATH . '/models/User.php';

class AssignmentController {
private $assignmentModel;
private $courseModel;
private $subjectModel;
private $schoolYearModel;
private $userModel;

public function __construct() {
$db = new Database();
$this->assignmentModel = new TeacherAssignment($db);
$this->courseModel = new Course($db);
$this->subjectModel = new Subject($db);
$this->schoolYearModel = new SchoolYear($db);
$this->userModel = new User($db);
}

public function index() {
Security::requireLogin();
if (!Security::hasRole('autoridad')) {
die('Acceso denegado');
}

$courses = $this->courseModel->getAll();
$subjects = $this->subjectModel->getAll();
$teachers = $this->userModel->getByRole('docente');
$activeYear = $this->schoolYearModel->getActive();
$assignments = $this->assignmentModel->getAll();

include BASE_PATH . '/views/assignments/index.php';
}

public function assign() {
Security::requireLogin();
if (!Security::hasRole('autoridad')) {
die('Acceso denegado');
}

if ($_SERVER['REQUEST_METHOD'] === 'POST') {
$activeYear = $this->schoolYearModel->getActive();

$data = [
':teacher_id' => (int)$_POST['teacher_id'],
':course_id' => (int)$_POST['course_id'],
':subject_id' => (int)$_POST['subject_id'],
':school_year_id' => $activeYear['id'],
':is_tutor' => 0
];

$result = $this->assignmentModel->assign($data);

if ($result['success']) {
header('Location: ?action=assignments&success=1');
```

```

} else {
header('Location: ?action=assignments&error=' . urlencode($result['message']));
}
exit;
}
}

public function setTutor() {
Security::requireLogin();
if (!Security::hasRole('autoridad')) {
die('Acceso denegado');
}

if ($_SERVER['REQUEST_METHOD'] === 'POST') {
$activeYear = $this->schoolYearModel->getActive();
$courseId = (int)$_POST['course_id'];
$teacherId = (int)$_POST['teacher_id'];

$result = $this->assignmentModel->setTutor($courseId, $teacherId, $activeYear['id']);

if ($result['success']) {
header('Location: ?action=tutor_management&tutor_success=1');
} else {
header('Location: ?action=tutor_management&tutor_error=' . urlencode($result['message']));
}
exit;
}
}

public function removeTutor() {
Security::requireLogin();
if (!Security::hasRole('autoridad')) {
die('Acceso denegado');
}

if ($_SERVER['REQUEST_METHOD'] === 'POST') {
$activeYear = $this->schoolYearModel->getActive();
$courseId = (int)$_POST['course_id'];

$db = new Database();
$sql = "UPDATE teacher_assignments
SET is_tutor = 0
WHERE course_id = :course_id
AND school_year_id = :school_year_id";

$stmt = $db->connect()->prepare($sql);
$stmt->execute([
':course_id' => $courseId,
':school_year_id' => $activeYear['id']
]);

header('Location: ?action=tutor_management&tutor_removed=1');
exit;
}
}

public function remove() {
Security::requireLogin();
if (!Security::hasRole('autoridad')) {
die('Acceso denegado');
}

if ($_SERVER['REQUEST_METHOD'] === 'POST') {
$assignmentId = (int)$_POST['assignment_id'];
$result = $this->assignmentModel->remove($assignmentId);

```

```

if ($result['success']) {
header('Location: ?action=assignments&removed=1');
} else {
header('Location: ?action=assignments&error=' . urlencode($result['message']));
}
exit;
}
}

public function viewByCourse() {
Security::requireLogin();
if (!Security::hasRole('autoridad')) {
die('Acceso denegado');
}

$courseId = (int)($_GET['course_id'] ?? 0);

if (!$courseId) {
header('Location: ?action=assignments');
exit;
}

$courses = $this->courseModel->getAll();
$course = array_filter($courses, fn($c) => $c['id'] == $courseId);
$course = reset($course);

$activeYear = $this->schoolYearModel->getActive();
$assignments = $this->assignmentModel->getByCourse($courseId);
$tutor = $this->assignmentModel->getTutorByCourse($courseId, $activeYear['id']);

include BASE_PATH . '/views/assignments/view_course.php';
}

public function getCourseTeachers() {
if (!isset($_SESSION['user_id'])) {
header('Content-Type: application/json');
echo json_encode(['error' => 'No autenticado']);
exit;
}

$courseId = (int)($_GET['course_id'] ?? 0);

if (!$courseId) {
header('Content-Type: application/json');
echo json_encode([]);
exit;
}

// Obtener año lectivo activo
$db = new Database();
$sqlYear = "SELECT id FROM school_years WHERE is_active = 1 LIMIT 1";
$stmtYear = $db->connect()->query($sqlYear);
$activeYear = $stmtYear->fetch();

if (!$activeYear) {
header('Content-Type: application/json');
echo json_encode([]);
exit;
}

// Solo docentes del curso que:
// 1. NO son tutores de otro curso
// 2. NO son tutores del curso actual
$sql = "SELECT DISTINCT ta.teacher_id,
CONCAT(u.last_name, ' ', u.first_name) as teacher_name

```

```

FROM teacher_assignments ta
INNER JOIN users u ON ta.teacher_id = u.id
WHERE ta.course_id = :course_id
AND ta.teacher_id NOT IN (
SELECT teacher_id
FROM teacher_assignments
WHERE is_tutor = 1
AND school_year_id = :school_year_id
)
ORDER BY u.last_name, u.first_name";

$stmt = $db->connect()->prepare($sql);
$stmt->execute([
':course_id' => $courseId,
':school_year_id' => $activeYear['id']
]);
$teachers = $stmt->fetchAll(PDO::FETCH_ASSOC);

header('Content-Type: application/json');
echo json_encode($teachers);
exit;
}

public function tutorManagement() {
$courses = $this->courseModel->getAll();
$assignments = $this->assignmentModel->getAll();

include BASE_PATH . '/views/assignments/tutor.php';
}

public function checkCourseTutor() {
if (!isset($_SESSION['user_id'])) {
header('Content-Type: application/json');
echo json_encode(null);
exit;
}

$courseId = (int)($_GET['course_id'] ?? 0);

if (!$courseId) {
header('Content-Type: application/json');
echo json_encode(null);
exit;
}

$db = new Database();
require_once BASE_PATH . '/models/SchoolYear.php';
$yearModel = new SchoolYear($db);
$activeYear = $yearModel->getActive();

$sql = "SELECT CONCAT(u.last_name, ' ', u.first_name) as tutor_name
FROM teacher_assignments ta
INNER JOIN users u ON ta.teacher_id = u.id
WHERE ta.course_id = :course_id
AND ta.school_year_id = :school_year_id
AND ta.is_tutor = 1
LIMIT 1";

$stmt = $db->connect()->prepare($sql);
$stmt->execute([
':course_id' => $courseId,
':school_year_id' => $activeYear['id']
]);

$tutor = $stmt->fetch(PDO::FETCH_ASSOC);

```



```

header('Content-Type: application/json');
echo json_encode($tutor);
exit;
}
}

```

## 6. ArchImp/controllers/AttendanceController.php

```

<?php
require_once BASE_PATH . '/config/config.php';
require_once BASE_PATH . '/models/Attendance.php';
require_once BASE_PATH . '/models/Course.php';
require_once BASE_PATH . '/models/Subject.php';
require_once BASE_PATH . '/models/SchoolYear.php';
require_once BASE_PATH . '/models/Shift.php';

class AttendanceController {
private $attendanceModel;
private $courseModel;
private $subjectModel;
private $schoolYearModel;
private $shiftModel;

public function __construct() {
Security::requireLogin();
$db = new Database();

$this->attendanceModel = new Attendance($db);
$this->courseModel = new Course($db);
$this->subjectModel = new Subject($db);
$this->schoolYearModel = new SchoolYear($db);
$this->shiftModel = new Shift($db);
}

private function isWithin48BusinessHours($date) {
$targetDate = new DateTime($date);
$today = new DateTime();
$today->setTime(0, 0, 0);

// Si es el mismo día, siempre está permitido
if ($targetDate->format('Y-m-d') === $today->format('Y-m-d')) {
return true;
}

// Si la fecha es futura, no permitir
if ($targetDate > $today) {
return false;
}

$businessHours = 0;
$current = clone $targetDate;

// Contar horas hábiles desde la fecha objetivo hasta hoy
while ($current < $today) {
$current->modify('+1 day');
$dayOfWeek = (int)$current->format('N'); // 1=Lunes, 7=Domingo

// Solo contar si no es fin de semana (sábado=6, domingo=7)
if ($dayOfWeek < 6) {
$businessHours += 24;
}
}
}

```

```

return $businessHours <= EDIT_ATTENDANCE_HOURS;
}

// Agregar nueva función para calcular fecha mínima
private function getMinDateAllowed() {
$today = new DateTime();
$businessHoursNeeded = EDIT_ATTENDANCE_HOURS;
$current = clone $today;

while ($businessHoursNeeded > 0) {
$current->modify('-1 day');
$dayOfWeek = (int)$current->format('N');

// Solo restar si no es fin de semana
if ($dayOfWeek < 6) {
$businessHoursNeeded -= 24;
}
}

return $current->format('Y-m-d');
}

public function register() {
if (!Security::hasRole(['docente', 'autoridad'])) {
die('Acceso denegado');
}

if ($_SERVER['REQUEST_METHOD'] === 'POST') {
$activeYear = $this->schoolYearModel->getActive();
$date = $_POST['date'];
$today = date('Y-m-d');

if (strtotime($date) > strtotime($today)) {
header('Location: ?action=attendance_register&error=future');
exit;
}

if (!$this->isWithin48BusinessHours($date)) {
header('Location: ?action=attendance_register&error=toolate');
exit;
}

$scheduleId = (int)$_POST['schedule_id'];

// Obtener datos del horario
$db = new Database();
$sql = "SELECT * FROM class_schedule WHERE id = :id";
$stmt = $db->connect()->prepare($sql);
$stmt->execute([':id' => $scheduleId]);
$scheduleData = $stmt->fetch();

foreach ($_POST as $key => $value) {
if (strpos($key, 'status_') === 0) {
$studentId = (int)str_replace('status_', '', $key);
$status = Security::sanitize($value);
$observation = Security::sanitize($_POST['obs_' . $studentId] ?? '');

$data = [
'student_id' => $studentId,
'course_id' => $scheduleData['course_id'],
'subject_id' => $scheduleData['subject_id'],
'teacher_id' => $_SESSION['user_id'],
'school_year_id' => $activeYear['id'],
'shift_id' => 1, // Obtener del curso

```

```

':date' => $date,
':hour_period' => $scheduleData['period_number'] . 'ra hora',
':status' => $status,
':observation' => $observation
];

$this->attendanceModel->create($data);
}
}

header('Location: ?action=attendance_register&success=1');
exit;
}

// Obtener clases del docente para hoy
$activeYear = $this->schoolYearModel->getActive();

$db = new Database();
require_once BASE_PATH . '/models/ClassSchedule.php';
$scheduleModel = new ClassSchedule($db);

$todayClasses = $scheduleModel->getTeacherScheduleToday($_SESSION['user_id'], $activeYear['id']);
$minDate = $this->calculateMinDate();

include BASE_PATH . '/views/attendance/register.php';
}

private function getTeacherCourses($teacherId) {
    $sql = "SELECT DISTINCT c.*, sh.name as shift_name
    FROM courses c
    INNER JOIN teacher_assignments ta ON c.id = ta.course_id
    INNER JOIN shifts sh ON c.shift_id = sh.id
    WHERE ta.teacher_id = :teacher_id
    ORDER BY c.name";

    $db = new Database();
    $stmt = $db->connect()->prepare($sql);
    $stmt->execute([':teacher_id' => $teacherId]);
    return $stmt->fetchAll();
}

private function calculateMinDate() {
    $date = new DateTime();
    $hoursBack = 0;

    while ($hoursBack < 48) {
        $date->modify('-1 day');
        $dayOfWeek = $date->format('N'); // 1=Lunes, 7=Domingo

        // Solo contar días laborables (lunes a viernes)
        if ($dayOfWeek >= 1 && $dayOfWeek <= 5) {
            $hoursBack += 24;
        }
    }

    return $date->format('Y-m-d');
}

public function getStudents() {
    if ($_SERVER['REQUEST_METHOD'] === 'POST') {
        $courseId = (int)$_POST['course_id'];
        $students = $this->courseModel->getStudents($courseId);

        header('Content-Type: application/json');
    }
}

```

```

echo json_encode($students);
exit;
}
}

public function view() {
if (!Security::hasRole(['docente', 'inspector', 'autoridad'])) {
die('Acceso denegado');
}

$courses = $this->courseModel->getAll();
$attendances = [];

if ($_SERVER['REQUEST_METHOD'] === 'POST') {
$courseId = (int)$_POST['course_id'];
$date = $_POST['date'];
$attendances = $this->attendanceModel->getByCourse($courseId, $date);
}

include BASE_PATH . '/views/attendance/view.php';
}

public function myAttendance() {
if (!Security::hasRole('estudiante')) {
die('Acceso denegado');
}

$studentId = $_SESSION['user_id'];
$attendances = $this->attendanceModel->getByStudent($studentId);

include BASE_PATH . '/views/attendance/my_attendance.php';
}

public function calendar() {
if (!Security::hasRole(['docente', 'autoridad', 'inspector'])) {
die('Acceso denegado');
}

$courses = $this->courseModel->getAll();

if (isset($_GET['course_id'])) {
$courseId = (int)$_GET['course_id'];
$month = $_GET['month'] ?? date('Y-m');

list($year, $monthNum) = explode('-', $month);

$monthNames = [
'01' => 'Enero', '02' => 'Febrero', '03' => 'Marzo', '04' => 'Abril',
'05' => 'Mayo', '06' => 'Junio', '07' => 'Julio', '08' => 'Agosto',
'09' => 'Septiembre', '10' => 'Octubre', '11' => 'Noviembre', '12' => 'Diciembre'
];

$monthName = $monthNames[$monthNum];

// Calcular mes anterior y siguiente
$prevMonth = date('Y-m', strtotime($month . '-01 -1 month'));
$nextMonth = date('Y-m', strtotime($month . '-01 +1 month'));

// Generar calendario
$firstDay = date('w', strtotime($year . '-' . $monthNum . '-01'));
$daysInMonth = date('t', strtotime($year . '-' . $monthNum . '-01'));

$calendarDays = [];

// Días vacíos al inicio

```

```

for ($i = 0; $i < $firstDay; $i++) {
    $calendarDays[] = ['day' => '', 'classes' => ''];
}

// Días del mes
for ($day = 1; $day <= $daysInMonth; $day++) {
    $date = $year . '-' . $monthNum . '-' . str_pad($day, 2, '0', STR_PAD_LEFT);
    $dayOfWeek = date('w', strtotime($date));

    $classes = [];
    if ($date == date('Y-m-d')) $classes[] = 'today';
    if ($dayOfWeek == 0 || $dayOfWeek == 6) $classes[] = 'weekend';

    // Obtener estadísticas de asistencia del día
    $attendance = $this->attendanceModel->getDayStats($courseId, $date);

    if ($attendance && $attendance['total'] > 0) {
        $classes[] = 'has-attendance';
    }

    $calendarDays[] = [
        'day' => $day,
        'classes' => implode(' ', $classes),
        'attendance' => $attendance
    ];
}

include BASE_PATH . '/views/attendance/calendar.php';
} else {
    include BASE_PATH . '/views/attendance/calendar.php';
}
}

public function getCourseSubjects() {
    if (!isset($_SESSION['user_id'])) {
        header('Content-Type: application/json');
        echo json_encode(['error' => 'No autenticado']);
        exit;
    }

    $courseId = (int)($_GET['course_id'] ?? 0);

    if (!$courseId) {
        header('Content-Type: application/json');
        echo json_encode([]);
        exit;
    }

    $sql = "SELECT DISTINCT ta.subject_id, s.name as subject_name
    FROM teacher_assignments ta
    INNER JOIN subjects s ON ta.subject_id = s.id
    WHERE ta.course_id = :course_id

    ORDER BY s.name";

    $db = new Database();
    $stmt = $db->connect()->prepare($sql);
    $stmt->execute([':course_id' => $courseId]);
    $subjects = $stmt->fetchAll(PDO::FETCH_ASSOC);

    header('Content-Type: application/json');
    echo json_encode($subjects);
    exit;
}

public function getTeacherCourseSubjects() {

```

```

if (!isset($_SESSION['user_id'])) {
header('Content-Type: application/json');
echo json_encode(['error' => 'No autenticado']);
exit;
}

$courseId = (int)($_GET['course_id'] ?? 0);

if (!$courseId) {
header('Content-Type: application/json');
echo json_encode([]);
exit;
}

// Si es autoridad: todas las asignaturas del curso
// Si es docente: solo las que él dicta en ese curso
if (Security::hasRole('autoridad')) {
$sql = "SELECT DISTINCT ta.subject_id, s.name as subject_name
FROM teacher_assignments ta
INNER JOIN subjects s ON ta.subject_id = s.id
WHERE ta.course_id = :course_id
ORDER BY s.name";

$params = [':course_id' => $courseId];
} else {
$sql = "SELECT DISTINCT ta.subject_id, s.name as subject_name
FROM teacher_assignments ta
INNER JOIN subjects s ON ta.subject_id = s.id
WHERE ta.course_id = :course_id
AND ta.teacher_id = :teacher_id
ORDER BY s.name";

$params = [
':course_id' => $courseId,
':teacher_id' => $_SESSION['user_id']
];
}

$db = new Database();
$stmt = $db->connect()->prepare($sql);
$stmt->execute($params);
$subjects = $stmt->fetchAll(PDO::FETCH_ASSOC);

header('Content-Type: application/json');
echo json_encode($subjects);
exit;
}

public function getScheduleInfo() {
if (!isset($_SESSION['user_id'])) {
header('Content-Type: application/json');
echo json_encode(['error' => 'No autenticado']);
exit;
}

$scheduleId = (int)($_GET['schedule_id'] ?? 0);

if (!$scheduleId) {
header('Content-Type: application/json');
echo json_encode([]);
exit;
}

$sql = "SELECT * FROM class_schedule WHERE id = :id";

```

```

$db = new Database();
$stmt = $db->connect()->prepare($sql);
$stmt->execute([':id' => $scheduleId]);
$schedule = $stmt->fetch(PDO::FETCH_ASSOC);

header('Content-Type: application/json');
echo json_encode($schedule);
exit;
}
}

```

## 7. ArchImp/controllers/AuthController.php

```

<?php
require_once BASE_PATH . '/config/config.php';
require_once BASE_PATH . '/models/User.php';
require_once BASE_PATH . '/helpers/Mailer.php';

class AuthController {
private $userModel;

public function __construct() {
$db = new Database();
$this->userModel = new User($db);
}

public function register() {
if ($_SERVER['REQUEST_METHOD'] === 'POST') {
if (!Security::validateToken($_POST['csrf_token'] ?? '')) {
die('Token inválido');
}

$data = [
'institution_id' => 1,
'username' => Security::sanitize($_POST['username']),
'email' => Security::sanitize($_POST['email']),
'password' => $_POST['password'],
'first_name' => Security::sanitize($_POST['first_name']),
'last_name' => Security::sanitize($_POST['last_name']),
'dni' => Security::sanitize($_POST['dni'] ?? ''),
'phone' => Security::sanitize($_POST['phone'] ?? '')
];

if ($this->userModel->create($data)) {
header('Location: ' . BASE_URL . '/public/index.php?action=login&registered=1');
exit;
}
}

include BASE_PATH . '/views/auth/register.php';
}

public function login() {
if ($_SERVER['REQUEST_METHOD'] === 'POST') {
if (!Security::validateToken($_POST['csrf_token'] ?? '')) {
die('Token inválido');
}

$emailOrUsername = Security::sanitize($_POST['email']);
$password = $_POST['password'];

// Buscar por email o username
$user = $this->userModel->findByEmailOrUsername($emailOrUsername);

```

```

if ($user && Security::verifyPassword($password, $user['password'])) {
    $_SESSION['user_id'] = $user['id'];
    $_SESSION['username'] = $user['username'];
    $_SESSION['first_name'] = $user['first_name'];
    $_SESSION['last_name'] = $user['last_name'];
    $_SESSION['institution_id'] = $user['institution_id'];
    $_SESSION['roles'] = $this->userModel->getUserRoles($user['id']);

    header('Location: ' . BASE_URL . '/public/index.php?action=dashboard');
    exit;
} else {
    $error = 'Credenciales incorrectas';
}

include BASE_PATH . '/views/auth/login.php';
}

public function logout() {
    session_destroy();
    header('Location: ' . BASE_URL . '/public/index.php?action=login');
    exit;
}

public function forgotPassword() {
    if ($_SERVER['REQUEST_METHOD'] === 'POST') {
        $email = Security::sanitize($_POST['email']);
        $user = $this->userModel->findByEmail($email);

        if ($user) {
            $token = bin2hex(random_bytes(32));
            $this->userModel->setResetToken($email, $token);

            $resetLink = BASE_URL . "/public/index.php?action=reset&token={$token}";
            $body = "Haga clic aquí para restablecer su contraseña: <a href='{$resetLink}'>{$resetLink}</a>";

            Mailer::send($email, 'Restablecer contraseña', $body);
        }

        $message = 'Si el correo existe, recibirá un enlace de recuperación';
    }

    include BASE_PATH . '/views/auth/forgot.php';
}

public function resetPassword() {
    $token = $_GET['token'] ?? '';

    if ($_SERVER['REQUEST_METHOD'] === 'POST') {
        $user = $this->userModel->validateResetToken($token);

        if ($user && $_POST['password'] === $_POST['password_confirm']) {
            $this->userModel->resetPassword($user['id'], $_POST['password']);
            header('Location: ' . BASE_URL . '/public/index.php?action=login&reset=1');
            exit;
        }
    }

    include BASE_PATH . '/views/auth/reset.php';
}

```



## 8. ArchImp/controllers/BackupController.php

```
<?php
require_once BASE_PATH . '/config/config.php';
require_once BASE_PATH . '/helpers/Backup.php';

class BackupController {
    private $backup;

    public function __construct() {
        Security::requireLogin();
        if (!Security::hasRole('autoridad')) {
            die('Acceso denegado');
        }

        $this->backup = new Backup();
    }

    public function index() {
        $backups = $this->backup->getBackups();
        include BASE_PATH . '/views/backup/index.php';
    }

    public function create() {
        $filename = $this->backup->createBackup();

        if ($filename) {
            header('Location: ?action=backups&success=1&file=' . $filename);
        } else {
            header('Location: ?action=backups&error=1');
        }
        exit;
    }

    public function download() {
        $filename = $_GET['file'] ?? '';
        $this->backup->downloadBackup($filename);
    }

    public function cleanup() {
        $deleted = $this->backup->deleteOldBackups(30);
        header('Location: ?action=backups&cleanup=' . $deleted);
        exit;
    }

    public function delete() {
        $filename = $_GET['file'] ?? '';

        if ($this->backup->deleteBackup($filename)) {
            header('Location: ?action=backups&deleted=1');
        } else {
            header('Location: ?action=backups&error_delete=1');
        }
        exit;
    }
}
```

## 9. ArchImp/controllers/DashboardController.php

```
<?php
require_once BASE_PATH . '/config/config.php';
require_once BASE_PATH . '/models/Attendance.php';
require_once BASE_PATH . '/models/User.php';
```

```

require_once BASE_PATH . '/models/Course.php';

class DashboardController {
private $attendanceModel;
private $userModel;
private $courseModel;

public function __construct() {
Security::requireLogin();

$db = new Database();
$this->attendanceModel = new Attendance($db);
$this->userModel = new User($db);
$this->courseModel = new Course($db);
}

public function index() {
$stats = $this->getStats();
include BASE_PATH . '/views/dashboard/index.php';
}

private function getStats() {
$stats = [];

if (Security::hasRole('autoridad')) {
$stats['total_students'] = $this->getTotalByRole('estudiante');
$stats['total_teachers'] = $this->getTotalByRole('docente');
$stats['total_courses'] = $this->getTotalCourses();
$stats['today_attendance'] = $this->getTodayAttendanceStats();
}

if (Security::hasRole('docente')) {
$stats['my_courses'] = $this->getTeacherCourses($_SESSION['user_id']);
$stats['my_students'] = $this->getTeacherStudentsCount($_SESSION['user_id']);
}

if (Security::hasRole('estudiante')) {
$stats['my_attendance'] = $this->getStudentAttendanceStats($_SESSION['user_id']);
}

if (Security::hasRole('representante')) {
$stats['my_children'] = $this->getRepresentativeChildrenCount($_SESSION['user_id']);
}

return $stats;
}

private function getTotalByRole($role) {
$sql = "SELECT COUNT(DISTINCT u.id) as total
FROM users u
INNER JOIN user_roles ur ON u.id = ur.user_id
INNER JOIN roles r ON ur.role_id = r.id
WHERE r.name = :role AND u.institution_id = :institution_id";

$db = new Database();
$stmt = $db->connect()->prepare($sql);

$stmt->execute([
':role' => $role,
':institution_id' => $_SESSION['institution_id']
]);

return $stmt->fetch()['total'];
}

private function getTotalCourses() {

```

```

$sql = "SELECT COUNT(*) as total FROM courses WHERE institution_id = :institution_id";

$db = new Database();
$stmt = $db->connect()->prepare($sql);
$stmt->execute([':institution_id' => $_SESSION['institution_id']]);

return $stmt->fetch()['total'];
}

private function getTodayAttendanceStats() {
    $sql = "SELECT
COUNT(*) as total,
SUM(CASE WHEN status = 'presente' THEN 1 ELSE 0 END) as presente,
SUM(CASE WHEN status = 'ausente' THEN 1 ELSE 0 END) as ausente
FROM attendances
WHERE date = CURDATE()";

    $db = new Database();
    $stmt = $db->connect()->query($sql);
    return $stmt->fetch();
}

private function getTeacherCourses($teacherId) {
    $sql = "SELECT COUNT(DISTINCT course_id) as total
FROM teacher_assignments
WHERE teacher_id = :teacher_id";

    $db = new Database();
    $stmt = $db->connect()->prepare($sql);
    $stmt->execute([':teacher_id' => $teacherId]);

    return $stmt->fetch()['total'];
}

private function getTeacherStudentsCount($teacherId) {
    $sql = "SELECT COUNT(DISTINCT cs.student_id) as total
FROM teacher_assignments ta
INNER JOIN course_students cs ON ta.course_id = cs.course_id
WHERE ta.teacher_id = :teacher_id";

    $db = new Database();
    $stmt = $db->connect()->prepare($sql);
    $stmt->execute([':teacher_id' => $teacherId]);

    return $stmt->fetch()['total'];
}

private function getStudentAttendanceStats($studentId) {
    $sql = "SELECT
COUNT(*) as total,
SUM(CASE WHEN status = 'presente' THEN 1 ELSE 0 END) as presente,
SUM(CASE WHEN status = 'ausente' THEN 1 ELSE 0 END) as ausente,
SUM(CASE WHEN status = 'tardanza' THEN 1 ELSE 0 END) as tardanza
FROM attendances
WHERE student_id = :student_id
AND date >= DATE_SUB(CURDATE(), INTERVAL 30 DAY)";

    $db = new Database();
    $stmt = $db->connect()->prepare($sql);
    $stmt->execute([':student_id' => $studentId]);

    return $stmt->fetch();
}

private function getRepresentativeChildrenCount($representativeId) {

```

```

$sql = "SELECT COUNT(*) as total
FROM representatives
WHERE representative_id = :representative_id";

$db = new Database();
$stmt = $db->connect()->prepare($sql);
$stmt->execute([':representative_id' => $representativeId]);

return $stmt->fetch()['total'];
}
}

```

## 10. ArchImp/controllers/ExportController.php

```

<?php
require_once BASE_PATH . '/config/config.php';
require_once BASE_PATH . '/vendor/autoload.php';
require_once BASE_PATH . '/models/Course.php';

use PhpOffice\PhpSpreadsheet\Spreadsheet;
use PhpOffice\PhpSpreadsheet\Writer\Xlsx;
use PhpOffice\PhpSpreadsheet\Style\Fill;
use PhpOffice\PhpSpreadsheet\Style\Alignment;

class ExportController {
private $courseModel;

public function __construct() {
Security::requireLogin();
if (!Security::hasRole(['autoridad', 'docente'])) {
die('Acceso denegado');
}

$db = new Database();
$this->courseModel = new Course($db);
}

public function exportStudentList() {
$courseId = (int)$_GET['course_id'];
$students = $this->courseModel->getEnrolledStudents($courseId);

$courses = $this->courseModel->getAll();
$course = array_filter($courses, fn($c) => $c['id'] == $courseId);
$course = reset($course);

$spreadsheet = new Spreadsheet();
$sheet = $spreadsheet->getActiveSheet();

$sheet->setTitle('Nómina');

// Encabezado
$sheet->mergeCells('A1:E1');
$sheet->setCellValue('A1', 'NÓMINA DE ESTUDIANTES');
$sheet->getStyle('A1')->getFont()->setBold(true)->setSize(16);
$sheet->getStyle('A1')->getAlignment()->setHorizontal(Alignment::HORIZONTAL_CENTER);

$sheet->setCellValue('A3', 'Curso:');
$sheet->setCellValue('B3', $course['name'] . ' - ' . $course['shift_name']);
$sheet->getStyle('A3')->getFont()->setBold(true);

$sheet->setCellValue('A4', 'Total:');
$sheet->setCellValue('B4', count($students) . ' estudiantes');
$sheet->getStyle('A4')->getFont()->setBold(true);

```

```

// Cabecera tabla
$headers = ['#', 'Apellidos', 'Nombres', 'Cédula', 'Fecha Matrícula'];
$col = 'A';
foreach ($headers as $header) {
    $sheet->setCellValue($col . '6', $header);
    $sheet->getStyle($col . '6')->getFont()->setBold(true);
    $sheet->getStyle($col . '6')->getFill()
    ->setFillType(Fill::FILL_SOLID)
    ->getStartColor()->setRGB('007bff');
    $sheet->getStyle($col . '6')->getFont()->getColor()->setRGB('FFFFFF');

    $col++;
}

// Datos
$row = 7;
$num = 1;
foreach ($students as $student) {
    $sheet->setCellValue('A' . $row, $num++);
    $sheet->setCellValue('B' . $row, $student['last_name']);
    $sheet->setCellValue('C' . $row, $student['first_name']);
    $sheet->setCellValue('D' . $row, $student['dni'] ?? '-');
    $sheet->setCellValue('E' . $row, date('d/m/Y', strtotime($student['enrollment_date'])));
    $row++;
}

// Ajustar anchos
$sheet->getColumnDimension('A')->setWidth(5);
$sheet->getColumnDimension('B')->setWidth(20);
$sheet->getColumnDimension('C')->setWidth(20);
$sheet->getColumnDimension('D')->setWidth(15);
$sheet->getColumnDimension('E')->setWidth(15);

$writer = new Xlsx($spreadsheet);

header('Content-Type: application/vnd.openxmlformats-officedocument.spreadsheetml.sheet');
header('Content-Disposition: attachment;filename="nomina_' . $course['name'] . '_' . date('YmdHis') .
'.xlsx"');
header('Cache-Control: max-age=0');

$writer->save('php://output');
exit;
}
}

```

## 11. ArchImp/controllers/InstitutionController.php

```

<?php
require_once BASE_PATH . '/config/config.php';
require_once BASE_PATH . '/models/Institution.php';
require_once BASE_PATH . '/models/InstitutionShift.php';
require_once BASE_PATH . '/models/Shift.php';

class InstitutionController {
    private $institutionModel;
    private $institutionShiftModel;
    private $shiftModel;

    public function __construct() {
        Security::requireLogin();
        if (!Security::hasRole('autoridad')) {
            die('Acceso denegado');
        }
    }
}

```

```

}

$db = new Database();
$this->institutionModel = new Institution($db);
$this->institutionShiftModel = new InstitutionShift($db);
$this->shiftModel = new Shift($db);
}

public function index() {
    $institution = $this->institutionModel->getById(1); // Por ahora solo una institución
    $allShifts = $this->shiftModel->getAll();
    $assignedShiftIds = $this->institutionShiftModel->getInstitutionShiftIds(1);

    include BASE_PATH . '/views/institution/index.php';
}

public function update() {
    if ($_SERVER['REQUEST_METHOD'] === 'POST') {
        $logoPath = null;

        // Subir logo si existe
        if (isset($_FILES['logo']) && $_FILES['logo']['error'] == 0) {
            $uploadDir = BASE_PATH . '/uploads/institution/';
            if (!file_exists($uploadDir)) {
                mkdir($uploadDir, 0755, true);
            }

            $extension = pathinfo($_FILES['logo']['name'], PATHINFO_EXTENSION);
            $filename = 'logo_' . time() . '.' . $extension;
            $uploadPath = $uploadDir . $filename;

            if (move_uploaded_file($_FILES['logo']['tmp_name'], $uploadPath)) {
                $logoPath = 'uploads/institution/' . $filename;
            }

            // Eliminar logo anterior si existe
            $current = $this->institutionModel->getById(1);
            if ($current['logo_path'] && file_exists(BASE_PATH . '/' . $current['logo_path'])) {
                unlink(BASE_PATH . '/' . $current['logo_path']);
            }
        }

        $data = [
            'name' => Security::sanitize($_POST['name']),
            'address' => Security::sanitize($_POST['address']),
            'province' => Security::sanitize($_POST['province']),
            'city' => Security::sanitize($_POST['city']),
            'phone' => Security::sanitize($_POST['phone']),
            'email' => Security::sanitize($_POST['email']),
            'director_name' => Security::sanitize($_POST['director_name']),
            'amie_code' => Security::sanitize($_POST['amie_code']),
            'website' => Security::sanitize($_POST['website']),
            'logo_path' => $logoPath
        ];

        $this->institutionModel->update(1, $data);
        header('Location: ?action=institution&success=1');
        exit;
    }
}

public function assignShift() {
    if ($_SERVER['REQUEST_METHOD'] === 'POST') {
        $data = [
            ':institution_id' => 1,

```

```

':shift_id' => (int)$_POST['shift_id']
];

$this->institutionShiftModel->assign($data);
header('Location: ?action=institution&shift_assigned=1');
exit;
}
}

public function removeShift() {
if ($_SERVER['REQUEST_METHOD'] === 'POST') {
$shiftId = (int)$_POST['shift_id'];
$this->institutionShiftModel->remove(1, $shiftId);
header('Location: ?action=institution&shift_removed=1');
exit;
}
}
}
}

```

## 12. ArchImp/controllers/JustificationController.php

```

<?php
require_once BASE_PATH . '/config/config.php';
require_once BASE_PATH . '/models/Justification.php';
require_once BASE_PATH . '/models/Attendance.php';

class JustificationController {
private $justificationModel;
private $attendanceModel;

public function __construct() {
Security::requireLogin();

$db = new Database();
$this->justificationModel = new Justification($db);
$this->attendanceModel = new Attendance($db);
}

public function submit() {
if (!Security::hasRole(['estudiante', 'representante'])) {
die('Acceso denegado');
}

$attendanceId = (int)$_GET['attendance_id'];

if ($_SERVER['REQUEST_METHOD'] === 'POST') {
// Verificar si ya existe una justificación para esta asistencia
if ($this->justificationModel->existsByAttendance($attendanceId)) {
header('Location: ?action=my_attendance&error=already_justified');
exit;
}

$documentPath = null;

// Subir documento si existe
if (isset($_FILES['document']) && $_FILES['document']['error'] == 0) {
$uploadDir = BASE_PATH . '/uploads/justifications/';
if (!file_exists($uploadDir)) {
mkdir($uploadDir, 0755, true);
}

$extension = pathinfo($_FILES['document']['name'], PATHINFO_EXTENSION);
$filename = uniqid() . '.' . $extension;

```

```

$uploadPath = $uploadDir . $filename;

if (move_uploaded_file($_FILES['document']['tmp_name'], $uploadPath)) {
    $documentPath = 'uploads/justifications/' . $filename;
}

}

$data = [
    ':attendance_id' => $attendanceId,
    ':student_id' => Security::hasRole('estudiante') ? $_SESSION['user_id'] : (int)$_POST['student_id'],
    ':submitted_by' => $_SESSION['user_id'],
    ':reason' => Security::sanitize($_POST['reason']),
    ':document_path' => $documentPath
];

$this->justificationModel->create($data);
header('Location: ?action=my_justifications&success=1');
exit;
}

// Verificar si ya existe justificación antes de mostrar el formulario
if ($this->justificationModel->existsByAttendance($attendanceId)) {
    header('Location: ?action=my_attendance&error=already_justified');
    exit;
}

include BASE_PATH . '/views/justifications/submit.php';
}

public function myJustifications() {
    if (!Security::hasRole(['estudiante', 'representante'])) {
        die('Acceso denegado');
    }

    $studentId = $_SESSION['user_id'];
    $justifications = $this->justificationModel->getByStudent($studentId);

    include BASE_PATH . '/views/justifications/my_list.php';
}

public function pending() {
    if (!Security::hasRole(['autoridad', 'inspector'])) {
        die('Acceso denegado');
    }

    $justifications = $this->justificationModel->getPending();

    include BASE_PATH . '/views/justifications/pending.php';
}

public function review() {
    if (!Security::hasRole(['autoridad', 'inspector'])) {
        die('Acceso denegado');
    }

    if ($_SERVER['REQUEST_METHOD'] === 'POST') {
        $justificationId = (int)$_POST['justification_id'];
        $action = $_POST['review_action'];
        $notes = Security::sanitize($_POST['notes'] ?? '');

        if ($action === 'approve') {
            $this->justificationModel->approve($justificationId, $_SESSION['user_id'], $notes);
        } else {
            $this->justificationModel->reject($justificationId, $_SESSION['user_id'], $notes);
        }
    }
}

```



```

header('Location: ?action=pending_justifications&success=1');
exit;
}
}
}

```

### 13. ArchImp/controllers/ProfileController.php

```

<?php
require_once BASE_PATH . '/config/config.php';
require_once BASE_PATH . '/models/User.php';

class ProfileController {
private $userModel;

public function __construct() {
Security::requireLogin();

$db = new Database();
$this->userModel = new User($db);
}

public function view() {
$user = $this->userModel->findById($_SESSION['user_id']);
$roles = $this->userModel->getUserRoles($_SESSION['user_id']);

include BASE_PATH . '/views/profile/view.php';
}

public function edit() {
$user = $this->userModel->findById($_SESSION['user_id']);

if ($_SERVER['REQUEST_METHOD'] === 'POST') {
$data = [
'first_name' => Security::sanitize($_POST['first_name']),
'last_name' => Security::sanitize($_POST['last_name']),
'phone' => Security::sanitize($_POST['phone']),
'email' => Security::sanitize($_POST['email'])
];

if ($this->userModel->updateProfile($_SESSION['user_id'], $data)) {
header('Location: ?action=profile&success=1');
exit;
}
}

include BASE_PATH . '/views/profile/edit.php';
}

public function changePassword() {
if ($_SERVER['REQUEST_METHOD'] === 'POST') {
$currentPassword = $_POST['current_password'];
$newPassword = $_POST['new_password'];
$confirmPassword = $_POST['confirm_password'];

$user = $this->userModel->findById($_SESSION['user_id']);

if (!Security::verifyPassword($currentPassword, $user['password'])) {
$error = 'Contraseña actual incorrecta';
} elseif ($newPassword !== $confirmPassword) {
$error = 'Las contraseñas no coinciden';
} elseif (strlen($newPassword) < 6) {

```

```

$error = 'La contraseña debe tener al menos 6 caracteres';
} else {
if ($this->userModel->updatePassword($_SESSION['user_id'], $newPassword)) {
header('Location: ?action=profile&password_changed=1');
exit;
}
}
}

include BASE_PATH . '/views/profile/change_password.php';
}
}

```

## 14. ArchImp/controllers/ReportController.php

```

<?php
require_once BASE_PATH . '/config/config.php';
require_once BASE_PATH . '/models/Attendance.php';
require_once BASE_PATH . '/models/Course.php';
require_once BASE_PATH . '/models/Subject.php';
require_once BASE_PATH . '/models/SchoolYear.php';
require_once BASE_PATH . '/models/User.php';
require_once BASE_PATH . '/models/Institution.php';
require_once BASE_PATH . '/vendor/autoload.php';

use PhpOffice\PhpSpreadsheet\Spreadsheet;
use PhpOffice\PhpSpreadsheet\Writer\Xlsx;
use PhpOffice\PhpSpreadsheet\Style\Fill;
use PhpOffice\PhpSpreadsheet\Style\Border;
use PhpOffice\PhpSpreadsheet\Style\Alignment;

class ReportController {
private $attendanceModel;
private $courseModel;
private $subjectModel;
private $schoolYearModel;
private $userModel;
private $institutionModel;

public function __construct() {
Security::requireLogin();
if (!Security::hasRole(['autoridad', 'inspector', 'docente'])) {
die('Acceso denegado');
}

$db = new Database();
$this->attendanceModel = new Attendance($db);
$this->courseModel = new Course($db);
$this->subjectModel = new Subject($db);
$this->schoolYearModel = new SchoolYear($db);
$this->userModel = new User($db);
$this->institutionModel = new Institution($db);
}

public function index() {
$courses = $this->courseModel->getAll();
$subjects = $this->subjectModel->getAll();

$data = [];
$course = null;
$startDate = null;
$endDate = null;

```

```

if ($_SERVER['REQUEST_METHOD'] === 'POST' && isset($_POST['preview'])) {
    $courseId = (int)$_POST['course_id'];
    $startDate = $_POST['start_date'];
    $endDate = $_POST['end_date'];

    $data = $this->attendanceModel->getReportData($courseId, $startDate, $endDate);
    $course = $this->getCourseInfo($courseId);
}

include BASE_PATH . '/views/reports/index.php';
}

public function generatePDF() {
    $courseId = (int)$_POST['course_id'];
    $startDate = $_POST['start_date'];
    $endDate = $_POST['end_date'];

    $data = $this->attendanceModel->getReportData($courseId, $startDate, $endDate);
    $course = $this->getCourseInfo($courseId);
    $institution = $this->institutionModel->getById($_SESSION['institution_id']);

    $pdf = new \TCPDF('L', 'mm', 'A4', true, 'UTF-8');

    $pdf->SetCreator('EcuAsist 2026');
    $pdf->SetAuthor('Sistema de Asistencia');
    $pdf->SetTitle('Reporte de Asistencia');

    $pdf->setPrintHeader(false);
    $pdf->setPrintFooter(false);
    $pdf->SetMargins(10, 10, 10);
    $pdf->SetAutoPageBreak(true, 10);

    $pdf->AddPage();

    $pdf->SetFont('helvetica', 'B', 16);
    $pdf->Cell(0, 10, 'REPORTE DE ASISTENCIA', 0, 1, 'C');

    $pdf->SetFont('helvetica', '', 10);
    $pdf->Cell(0, 5, $institution['name'] ?? 'Institución Educativa', 0, 1, 'C');
    $pdf->Ln(5);

    $pdf->SetFont('helvetica', 'B', 11);
    $pdf->Cell(50, 6, 'Curso:', 0, 0);
    $pdf->SetFont('helvetica', '', 11);
    $pdf->Cell(0, 6, html_entity_decode($course['name'], ENT_QUOTES, 'UTF-8'), 0, 1);

    $pdf->SetFont('helvetica', 'B', 11);
    $pdf->Cell(50, 6, 'Período:', 0, 0);
    $pdf->SetFont('helvetica', '', 11);
    $pdf->Cell(0, 6, date('d/m/Y', strtotime($startDate)) . ' al ' . date('d/m/Y', strtotime($endDate)), 0, 1);
    $pdf->Ln(3);

    $html = '<table border="1" cellpadding="4">
    <thead>
    <tr style="background-color: #007bff; color: white; font-weight: bold;">
    <th width="16.666666%">Fecha</th>
    <th width="16.666666%">Estudiante</th>
    <th width="16.666666%">Asignatura</th>
    <th width="16.666666%">Hora</th>
    <th width="16.666666%">Estado</th>
    <th width="16.666666%">Observación</th>
    </tr>
    </thead>
    <tbody>';

```

```

foreach ($data as $row) {
    $statusColor = $this->getStatusColor($row['status']);
    $html .= '<tr>
    <td>' . date('d/m/Y', strtotime($row['date'])) . '</td>
    <td>' . $row['student_name'] . '</td>
    <td>' . $row['subject_name'] . '</td>
    <td>' . $row['hour_period'] . '</td>

    <td style="background-color: ' . $statusColor . ';">' . ucfirst($row['status']) . '</td>
    <td>' . ($row['observation'] ? '-' : '') . '</td>
    </tr>';
}

$html .= '</tbody></table>';

$pdf->SetFont('helvetica', '', 9);

$pdf->writeHTML($html, true, false, true, false, '');

// Generar nombre de archivo con curso
$courseName = $this->sanitizeFilename($course['name']);
$filename = 'reporte_asistencia_' . $courseName . '_' . date('YmdHis') . '.pdf';

$pdf->Output($filename, 'D');
exit;
}

public function generateExcel() {
    $courseId = (int)$_POST['course_id'];
    $startDate = $_POST['start_date'];
    $endDate = $_POST['end_date'];

    $data = $this->attendanceModel->getReportData($courseId, $startDate, $endDate);
    $course = $this->getCourseInfo($courseId);
    $institution = $this->institutionModel->getById($_SESSION['institution_id']);

    $spreadsheet = new Spreadsheet();
    $sheet = $spreadsheet->getActiveSheet();

    $sheet->setTitle('Asistencia');

    // Encabezado
    $sheet->mergeCells('A1:F1');
    $sheet->setCellValue('A1', 'REPORTE DE ASISTENCIA');
    $sheet->getStyle('A1')->getFont()->setBold(true)->setSize(16);
    $sheet->getStyle('A1')->getAlignment()->setHorizontal(Alignment::HORIZONTAL_CENTER);

    $sheet->mergeCells('A2:F2');
    $sheet->setCellValue('A2', $institution['name'] ?? 'Institución Educativa');
    $sheet->getStyle('A2')->getAlignment()->setHorizontal(Alignment::HORIZONTAL_CENTER);

    $sheet->setCellValue('A4', 'Curso:');
    $sheet->setCellValue('B4', html_entity_decode($course['name'], ENT_QUOTES, 'UTF-8'));
    $sheet->getStyle('A4')->getFont()->setBold(true);

    $sheet->setCellValue('A5', 'Período:');
    $sheet->setCellValue('B5', date('d/m/Y', strtotime($startDate)) . ' al ' . date('d/m/Y',
    strtotime($endDate)));
    $sheet->getStyle('A5')->getFont()->setBold(true);

    // Cabecera de tabla
    $row = 7;
    $headers = ['Fecha', 'Estudiante', 'Asignatura', 'Hora', 'Estado', 'Observación'];
    $col = 'A';

```

```

foreach ($headers as $header) {
    $sheet->setCellValue($col . $row, $header);
    $sheet->getStyle($col . $row)->getFont()->setBold(true);
    $sheet->getStyle($col . $row)->getFill()
    ->setFillType(Fill::FILL_SOLID)
    ->getStartColor()->setRGB('007bff');
    $sheet->getStyle($col . $row)->getFont()->getColor()->setRGB('FFFFFF');
    $col++;
}

// Datos
$row = 8;
foreach ($data as $record) {
    $sheet->setCellValue('A' . $row, date('d/m/Y', strtotime($record['date'])));
    $sheet->setCellValue('B' . $row, $record['student_name']);
    $sheet->setCellValue('C' . $row, $record['subject_name']);
    $sheet->setCellValue('D' . $row, $record['hour_period']);
    $sheet->setCellValue('E' . $row, ucfirst($record['status']));
    $sheet->setCellValue('F' . $row, $record['observation'] ? '-' : '');

    $statusColor = $this->getStatusColorHex($record['status']);
    $sheet->getStyle('E' . $row)->getFill()
    ->setFillType(Fill::FILL_SOLID)
    ->getStartColor()->setRGB($statusColor);

    $row++;
}

// Ajustar anchos
$sheet->getColumnDimension('A')->setWidth(12);
$sheet->getColumnDimension('B')->setWidth(25);
$sheet->getColumnDimension('C')->setWidth(20);
$sheet->getColumnDimension('D')->setWidth(15);
$sheet->getColumnDimension('E')->setWidth(12);
$sheet->getColumnDimension('F')->setWidth(25);

// Bordes
$styleArray = [
    'borders' => [
        'allBorders' => [
            'borderStyle' => Border::BORDER_THIN,
        ],
    ],
];
$sheet->getStyle('A7:F' . ($row - 1))->applyFromArray($styleArray);

$writer = new Xlsx($spreadsheet);

// Generar nombre de archivo con curso
$courseName = $this->sanitizeFilename($course['name']);
$filename = 'reporte_asistencia_' . $courseName . '_' . date('YmdHis') . '.xlsx';

header('Content-Type: application/vnd.openxmlformats-officedocument.spreadsheetml.sheet');
header('Content-Disposition: attachment;filename="' . $filename . '"');
header('Cache-Control: max-age=0');

$writer->save('php://output');
exit;
}

private function getCourseInfo($courseId) {
    $courses = $this->courseModel->getAll();
    foreach ($courses as $course) {
        if ($course['id'] == $courseId) {

```

```

return $course;
}
}
return null;
}

private function getStatusColor($status) {
switch ($status) {
case 'presente': return '#d4edda';
case 'ausente': return '#f8d7da';
case 'tardanza': return '#fff3cd';
case 'justificado': return '#d1ecf1';
default: return '#ffffff';
}
}

private function getStatusColorHex($status) {
switch ($status) {
case 'presente': return '28a745';
case 'ausente': return 'dc3545';
case 'tardanza': return 'ffc107';
case 'justificado': return '17a2b8';
default: return 'FFFFFF';
}
}

private function sanitizeFilename($filename) {
// Decodificar HTML entities
$filename = html_entity_decode($filename, ENT_QUOTES, 'UTF-8');

// Reemplazar caracteres no permitidos en nombres de archivo
$filename = str_replace(['"', '/', '\\', ':', '*', '?', '<', '>', '|'], '', $filename);

// Reemplazar espacios con guiones bajos
$filename = str_replace(' ', '_', $filename);

// Reemplazar guiones múltiples
$filename = preg_replace('/-+/', '-', $filename);

// Limitar longitud (máximo 50 caracteres)
if (strlen($filename) > 50) {
$filename = substr($filename, 0, 50);
}

// Limpiar caracteres al inicio y final
$filename = trim($filename, '-_');

return $filename;
}
}

```

## 15. ArchImp/controllers/ReportController2.php

```

<?php
require_once BASE_PATH . '/config/config.php';
require_once BASE_PATH . '/vendor/autoload.php';
require_once BASE_PATH . '/models/Attendance.php';
require_once BASE_PATH . '/models/Course.php';
require_once BASE_PATH . '/models/Subject.php';
require_once BASE_PATH . '/models/SchoolYear.php';
require_once BASE_PATH . '/models/User.php';

use PhpOffice\PhpSpreadsheet\Spreadsheet;
use PhpOffice\PhpSpreadsheet\Writer\Xlsx;

```

```

use PhpOffice\PhpSpreadsheet\Style\Fill;
use PhpOffice\PhpSpreadsheet\Style\Border;
use PhpOffice\PhpSpreadsheet\Style\Alignment;

class ReportController {
private $attendanceModel;
private $courseModel;
private $subjectModel;
private $schoolYearModel;
private $userModel;

public function __construct() {
Security::requireLogin();
if (!Security::hasRole(['autoridad', 'inspector', 'docente'])) {
die('Acceso denegado');
}

$db = new Database();
$this->attendanceModel = new Attendance($db);
$this->courseModel = new Course($db);
$this->subjectModel = new Subject($db);
$this->schoolYearModel = new SchoolYear($db);
$this->userModel = new User($db);
}

public function index() {
$courses = $this->courseModel->getAll();
$subjects = $this->subjectModel->getAll();

include BASE_PATH . '/views/reports/index.php';
}

public function generatePDF() {
$courseId = (int)$_POST['course_id'];
$startDate = $_POST['start_date'];
$endDate = $_POST['end_date'];

$data = $this->attendanceModel->getReportData($courseId, $startDate, $endDate);
$course = $this->getCourseInfo($courseId);

try {
$pdf = new \TCPDF('L', 'mm', 'A4', true, 'UTF-8');

$pdf->SetCreator('EcuAsist 2026');
$pdf->SetAuthor('Sistema de Asistencia');
$pdf->SetTitle('Reporte de Asistencia');

$pdf->setPrintHeader(false);
$pdf->setPrintFooter(false);

$pdf->SetMargins(10, 10, 10);
$pdf->SetAutoPageBreak(true, 10);

$pdf->AddPage();

$pdf->SetFont('helvetica', 'B', 16);
$pdf->Cell(0, 10, 'REPORTE DE ASISTENCIA', 0, 1, 'C');

$pdf->SetFont('helvetica', '', 10);
$pdf->Cell(0, 5, 'Unidad Educativa Demo', 0, 1, 'C');
$pdf->Ln(5);

$pdf->SetFont('helvetica', 'B', 11);
$pdf->Cell(50, 6, 'Curso:', 0, 0);
$pdf->SetFont('helvetica', '', 11);
$pdf->Cell(0, 6, $course['name'] . ' - ' . $course['shift_name'], 0, 1);

```

```

$pdf->SetFont('helvetica', 'B', 11);
$pdf->Cell(50, 6, 'Período:', 0, 0);
$pdf->SetFont('helvetica', '', 11);
$pdf->Cell(0, 6, date('d/m/Y', strtotime($startDate)) . ' al ' . date('d/m/Y', strtotime($endDate)), 0, 1);
$pdf->Ln(3);

$html = '<table border="1" cellpadding="4">
<thead>
<tr style="background-color: #007bff; color: white; font-weight: bold;">
<th width="10%">Fecha</th>
<th width="25%">Estudiante</th>
<th width="20%">Asignatura</th>
<th width="15%">Hora</th>
<th width="12%">Estado</th>
<th width="18%">Observación</th>
</tr>
</thead>
<tbody>';

foreach ($data as $row) {
    $statusColor = $this->getStatusColor($row['status']);
    $html .= '<tr>
<td>' . date('d/m/Y', strtotime($row['date'])) . '</td>
<td>' . htmlspecialchars($row['student_name']) . '</td>
<td>' . htmlspecialchars($row['subject_name']) . '</td>
<td>' . htmlspecialchars($row['hour_period']) . '</td>
<td style="background-color: ' . $statusColor . ';">' . ucfirst($row['status']) . '</td>
<td>' . htmlspecialchars($row['observation'] ?? '-') . '</td>
</tr>';
}

$html .= '</tbody></table>';

$pdf->writeHTML($html, true, false, true, false, '');

$pdf->Output('reporte_asistencia_' . date('YmdHis') . '.pdf', 'D');
exit;

} catch (Exception $e) {
    die('Error generando PDF: ' . $e->getMessage());
}

}

public function generateExcel() {
    $courseId = (int)$_POST['course_id'];
    $startDate = $_POST['start_date'];
    $endDate = $_POST['end_date'];

    $data = $this->attendanceModel->getReportData($courseId, $startDate, $endDate);
    $course = $this->getCourseInfo($courseId);

    try {
        $spreadsheet = new Spreadsheet();
        $sheet = $spreadsheet->getActiveSheet();

        $sheet->setTitle('Asistencia');

        // Encabezado
        $sheet->mergeCells('A1:F1');
        $sheet->setCellValue('A1', 'REPORTE DE ASISTENCIA');
        $sheet->getStyle('A1')->getFont()->setBold(true)->setSize(16);
        $sheet->getStyle('A1')->getAlignment()->setHorizontal(Alignment::HORIZONTAL_CENTER);

        $sheet->mergeCells('A2:F2');

```



```

$sheet->setCellValue('A2', 'Unidad Educativa Demo');
$sheet->getStyle('A2')->getAlignment()->setHorizontal(Alignment::HORIZONTAL_CENTER);

$sheet->setCellValue('A4', 'Curso:');
$sheet->setCellValue('B4', $course['name'] . ' - ' . $course['shift_name']);
$sheet->getStyle('A4')->getFont()->setBold(true);

$sheet->setCellValue('A5', 'Período:');
$sheet->setCellValue('B5', date('d/m/Y', strtotime($startDate)) . ' al ' . date('d/m/Y',
strtotime($endDate)));
$sheet->getStyle('A5')->getFont()->setBold(true);

// Cabecera de tabla
$row = 7;
$headers = ['Fecha', 'Estudiante', 'Asignatura', 'Hora', 'Estado', 'Observación'];
$col = 'A';

foreach ($headers as $header) {
    $sheet->setCellValue($col . $row, $header);
    $sheet->getStyle($col . $row)->getFont()->setBold(true);
    $sheet->getStyle($col . $row)->getFill()
    ->setFillType(Fill::FILL_SOLID)
    ->getStartColor()->setRGB('007bff');
    $sheet->getStyle($col . $row)->getFont()->getColor()->setRGB('FFFFFF');
    $col++;
}

// Datos
$row = 8;
foreach ($data as $record) {
    $sheet->setCellValue('A' . $row, date('d/m/Y', strtotime($record['date'])));
    $sheet->setCellValue('B' . $row, $record['student_name']);
    $sheet->setCellValue('C' . $row, $record['subject_name']);
    $sheet->setCellValue('D' . $row, $record['hour_period']);
    $sheet->setCellValue('E' . $row, ucfirst($record['status']));
    $sheet->setCellValue('F' . $row, $record['observation'] ? : '-');

    $statusColor = $this->getStatusColorHex($record['status']);
    $sheet->getStyle('E' . $row)->getFill()
    ->setFillType(Fill::FILL_SOLID)
    ->getStartColor()->setRGB($statusColor);

    $row++;
}

// Ajustar anchos
$sheet->getColumnDimension('A')->setWidth(12);
$sheet->getColumnDimension('B')->setWidth(25);
$sheet->getColumnDimension('C')->setWidth(20);
$sheet->getColumnDimension('D')->setWidth(15);
$sheet->getColumnDimension('E')->setWidth(12);
$sheet->getColumnDimension('F')->setWidth(25);

// Bordes
$styleArray = [
    'borders' => [
        'allBorders' => [
            'borderStyle' => Border::BORDER_THIN,
        ],
    ],
];
$sheet->getStyle('A7:F' . ($row - 1))->applyFromArray($styleArray);

$writer = new Xlsx($spreadsheet);

```

```

header('Content-Type: application/vnd.openxmlformats-officedocument.spreadsheetml.sheet');
header('Content-Disposition: attachment;filename="reporte_asistencia_' . date('YmdHis') . '.xlsx"');
header('Cache-Control: max-age=0');

$writer->save('php://output');
exit;

} catch (Exception $e) {
die('Error generando Excel: ' . $e->getMessage());
}
}

private function getCourseInfo($courseId) {
$courses = $this->courseModel->getAll();
foreach ($courses as $course) {
if ($course['id'] == $courseId) {
return $course;
}
}
return null;
}

private function getStatusColor($status) {
switch ($status) {
case 'presente': return '#d4edda';
case 'ausente': return '#f8d7da';
case 'tardanza': return '#fff3cd';
case 'justificado': return '#d1ecf1';
default: return '#ffffff';
}
}

private function getStatusColorHex($status) {
switch ($status) {
case 'presente': return 'd4edda';
case 'ausente': return 'f8d7da';
case 'tardanza': return 'fff3cd';
case 'justificado': return 'd1ecf1';
default: return 'FFFFFF';
}
}
}

```

## 16. ArchImp/controllers/RepresentativeController.php

```

<?php
require_once BASE_PATH . '/config/config.php';
require_once BASE_PATH . '/models/Representative.php';
require_once BASE_PATH . '/models/User.php';
require_once BASE_PATH . '/models/Attendance.php';

class RepresentativeController {
private $representativeModel;
private $userModel;
private $attendanceModel;

public function __construct() {
Security::requireLogin();

$db = new Database();
$this->representativeModel = new Representative($db);
$this->userModel = new User($db);

```

```

$this->attendanceModel = new Attendance($db);
}

public function manageRepresentatives() {
    if (!Security::hasRole('autoridad')) {
        die('Acceso denegado');
    }

    $representatives = $this->userModel->getByRole('representante');
    $students = $this->userModel->getByRole('estudiante');

    if ($_SERVER['REQUEST_METHOD'] === 'POST') {
        $repId = (int)$_POST['representative_id'];
        $studentId = (int)$_POST['student_id'];
        $relationship = Security::sanitize($_POST['relationship']);
        $isPrimary = isset($_POST['is_primary']) ? 1 : 0;

        $this->representativeModel->assignStudent($repId, $studentId, $relationship, $isPrimary);
        header('Location: ?action=manage_representatives&success=1');
        exit;
    }

    include BASE_PATH . '/views/representatives/manage.php';
}

public function myChildren() {
    if (!Security::hasRole('representante')) {
        die('Acceso denegado');
    }

    $children = $this->representativeModel->getStudentsByRepresentative($_SESSION['user_id']);

    include BASE_PATH . '/views/representatives/my_children.php';
}

public function childAttendance() {
    if (!Security::hasRole('representante')) {
        die('Acceso denegado');
    }

    $studentId = (int)($_GET['student_id'] ?? 0);

    // Verificar que el estudiante sea hijo del representante
    $children = $this->representativeModel->getStudentsByRepresentative($_SESSION['user_id']);
    $authorized = false;
    $student = null;

    foreach ($children as $child) {
        if ($child['id'] == $studentId) {
            $authorized = true;
            $student = $child;
            break;
        }
    }

    if (!$authorized) {
        die('No autorizado para ver este estudiante');
    }

    $attendances = $this->attendanceModel->getByStudent($studentId);

    include BASE_PATH . '/views/representatives/child_attendance.php';
}

public function removeRelation() {

```

```

if (!Security::hasRole('autoridad')) {
die('Acceso denegado');
}

$repId = (int)($_GET['rep_id'] ?? 0);
$studentId = (int)($_GET['student_id'] ?? 0);

if ($repId && $studentId) {
if ($this->representativeModel->removeStudent($repId, $studentId)) {
header('Location: ?action=manage_representatives&removed=1');
exit;
}
}

header('Location: ?action=manage_representatives&error=1');
exit;
}
}

```

## 17. ArchImp/controllers/ScheduleController.php

```

<?php
require_once BASE_PATH . '/config/config.php';
require_once BASE_PATH . '/models/ClassSchedule.php';
require_once BASE_PATH . '/models/Course.php';
require_once BASE_PATH . '/models/Subject.php';
require_once BASE_PATH . '/models/User.php';
require_once BASE_PATH . '/models/SchoolYear.php';

class ScheduleController {
private $scheduleModel;
private $courseModel;
private $subjectModel;
private $userModel;
private $schoolYearModel;

public function __construct() {
Security::requireLogin();
if (!Security::hasRole('autoridad')) {
die('Acceso denegado');
}

$db = new Database();
$this->scheduleModel = new ClassSchedule($db);
$this->courseModel = new Course($db);
$this->subjectModel = new Subject($db);
$this->userModel = new User($db);
$this->schoolYearModel = new SchoolYear($db);
}

public function index() {
$courses = $this->courseModel->getAll();
include BASE_PATH . '/views/schedule/index.php';
}

public function manageCourse() {
$courseId = (int)($_GET['course_id'] ?? 0);

if (!$courseId) {
header('Location: ?action=schedules');
exit;
}
}

```

```

if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    $activeYear = $this->schoolYearModel->getActive();

    $data = [
        ':course_id' => $courseId,
        ':subject_id' => (int)$_POST['subject_id'],
        ':teacher_id' => (int)$_POST['teacher_id'],
        ':school_year_id' => $activeYear['id'],
        ':day_of_week' => $_POST['day_of_week'],
        ':period_number' => (int)$_POST['period_number']
    ];

    $result = $this->scheduleModel->create($data);

    if ($result['success']) {
        header('Location: ?action=manage_schedule&course_id=' . $courseId . '&success=1');
    } else {
        header('Location: ?action=manage_schedule&course_id=' . $courseId . '&error=' .
            urlencode($result['message']));
    }
    exit;
}

$activeYear = $this->schoolYearModel->getActive();
$courses = $this->courseModel->getAll();
$course = array_filter($courses, fn($c) => $c['id'] == $courseId);
$course = reset($course);

$schedule = $this->scheduleModel->getByCourse($courseId, $activeYear['id']);
$subjects = $this->subjectModel->getAll();
$teachers = $this->userModel->getByRole('docente');

include BASE_PATH . '/views/schedule/manage.php';
}

public function deleteClass() {
    if ($_SERVER['REQUEST_METHOD'] === 'POST') {
        $scheduleId = (int)$_POST['schedule_id'];
        $courseId = (int)$_POST['course_id'];

        $this->scheduleModel->delete($scheduleId);
        header('Location: ?action=manage_schedule&course_id=' . $courseId . '&deleted=1');
        exit;
    }
}

public function getCourseSubjectsSchedule() {
    if (!isset($_SESSION['user_id'])) {
        header('Content-Type: application/json');
        echo json_encode([]);
        exit;
    }

    $courseId = (int)($_GET['course_id'] ?? 0);

    if (!$courseId) {
        header('Content-Type: application/json');
        echo json_encode([]);
        exit;
    }

    // Obtener asignaturas con sus docentes asignados al curso
    $sql = "SELECT DISTINCT ta.subject_id, s.name as subject_name,
        ta.teacher_id, CONCAT(u.last_name, ' ', u.first_name) as teacher_name
    FROM teacher_assignments ta

```

```

INNER JOIN subjects s ON ta.subject_id = s.id
INNER JOIN users u ON ta.teacher_id = u.id
WHERE ta.course_id = :course_id
ORDER BY s.name";

$db = new Database();
$stmt = $db->connect()->prepare($sql);
$stmt->execute([':course_id' => $courseId]);
$subjects = $stmt->fetchAll(PDO::FETCH_ASSOC);

header('Content-Type: application/json');
echo json_encode($subjects);
exit;
}

public function checkScheduleConflict() {
if (!isset($_SESSION['user_id'])) {
header('Content-Type: application/json');
echo json_encode(['exists' => false]);
exit;
}

$courseId = (int)($_GET['course_id'] ?? 0);
$day = $_GET['day'] ?? '';
$period = (int)($_GET['period'] ?? 0);

if (!$courseId || !$day || !$period) {
header('Content-Type: application/json');
echo json_encode(['exists' => false]);
exit;
}

$db = new Database();
$activeYear = $this->schoolYearModel->getActive();

$sql = "SELECT s.name as subject_name, CONCAT(u.last_name, ' ', u.first_name) as teacher_name
FROM class_schedule cs
INNER JOIN subjects s ON cs.subject_id = s.id
INNER JOIN users u ON cs.teacher_id = u.id
WHERE cs.course_id = :course_id
AND cs.day_of_week = :day
AND cs.period_number = :period
AND cs.school_year_id = :school_year_id
LIMIT 1";

$stmt = $db->connect()->prepare($sql);
$stmt->execute([
':course_id' => $courseId,
':day' => $day,
':period' => $period,
':school_year_id' => $activeYear['id']
]);

$existing = $stmt->fetch(PDO::FETCH_ASSOC);

$result = [
'exists' => $existing ? true : false,
'subject_name' => $existing['subject_name'] ?? null,
'teacher_name' => $existing['teacher_name'] ?? null
];

header('Content-Type: application/json');
echo json_encode($result);
exit;
}

```

```
}
```

## 18. ArchImp/controllers/SearchController.php

```
<?php
require_once BASE_PATH . '/config/config.php';
require_once BASE_PATH . '/models/User.php';
require_once BASE_PATH . '/models/Course.php';
require_once BASE_PATH . '/models/Attendance.php';

class SearchController {
private $userModel;
private $courseModel;
private $attendanceModel;

public function __construct() {
Security::requireLogin();

$db = new Database();
$this->userModel = new User($db);
$this->courseModel = new Course($db);
$this->attendanceModel = new Attendance($db);
}

public function searchStudents() {
$query = $_GET['q'] ?? '';

if (strlen($query) < 2) {
echo json_encode([]);
exit;
}

$sql = "SELECT u.id, u.first_name, u.last_name, u.dni, u.email,
c.name as course_name
FROM users u
INNER JOIN user_roles ur ON u.id = ur.user_id
INNER JOIN roles r ON ur.role_id = r.id
LEFT JOIN course_students cs ON u.id = cs.student_id
LEFT JOIN courses c ON cs.course_id = c.id
WHERE r.name = 'estudiante'
AND u.institution_id = :institution_id
AND (u.first_name LIKE :query
OR u.last_name LIKE :query
OR u.dni LIKE :query
OR u.email LIKE :query)
LIMIT 10";

$db = new Database();
$stmt = $db->connect()->prepare($sql);
$stmt->execute([
':institution_id' => $_SESSION['institution_id'],
':query' => '%' . $query . '%'
]);

header('Content-Type: application/json');
echo json_encode($stmt->fetchAll());
exit;
}

public function advancedAttendanceReport() {
if (!Security::hasRole(['autoridad', 'inspector'])) {
die('Acceso denegado');
}
}
```

```

$filters = [
'course_id' => $_POST['course_id'] ?? null,
'student_id' => $_POST['student_id'] ?? null,
'subject_id' => $_POST['subject_id'] ?? null,
'status' => $_POST['status'] ?? null,
'start_date' => $_POST['start_date'] ?? null,
'end_date' => $_POST['end_date'] ?? null
];

$results = $this->attendanceModel->getFiltered($filters);

include BASE_PATH . '/views/reports/advanced.php';
}
}

```

## 19. ArchImp/controllers/StatsController.php

```

<?php
require_once BASE_PATH . '/config/config.php';
require_once BASE_PATH . '/models/Attendance.php';
require_once BASE_PATH . '/models/Course.php';
require_once BASE_PATH . '/models/User.php';

class StatsController {
private $attendanceModel;
private $courseModel;
private $userModel;

public function __construct() {
Security::requireLogin();
if (!Security::hasRole('autoridad')) {
die('Acceso denegado');
}

$db = new Database();
$this->attendanceModel = new Attendance($db);
$this->courseModel = new Course($db);
$this->userModel = new User($db);
}

public function index() {
$startDate = $_GET['start_date'] ?? date('Y-m-01');
$endDate = $_GET['end_date'] ?? date('Y-m-d');

$stats = [
'overall' => $this->getOverallStats($startDate, $endDate),
'by_course' => $this->getStatsByCourse($startDate, $endDate),
'by_student' => $this->getTopAbsentStudents($startDate, $endDate),
'by_day' => $this->getStatsByDay($startDate, $endDate)
];

include BASE_PATH . '/views/stats/index.php';
}

private function getOverallStats($startDate, $endDate) {
$sql = "SELECT
COUNT(*) as total_records,
COUNT(DISTINCT student_id) as total_students,
SUM(CASE WHEN status = 'presente' THEN 1 ELSE 0 END) as presente,
SUM(CASE WHEN status = 'ausente' THEN 1 ELSE 0 END) as ausente,
SUM(CASE WHEN status = 'tardanza' THEN 1 ELSE 0 END) as tardanza,

```



```

SUM(CASE WHEN status = 'justificado' THEN 1 ELSE 0 END) as justificado
FROM attendances
WHERE date BETWEEN :start_date AND :end_date";

$db = new Database();
$stmt = $db->connect()->prepare($sql);
$stmt->execute([':start_date' => $startDate, ':end_date' => $endDate]);
return $stmt->fetch();
}

private function getStatsByCourse($startDate, $endDate) {
    $sql = "SELECT
    c.name as course_name,
    COUNT(*) as total,
    SUM(CASE WHEN a.status = 'presente' THEN 1 ELSE 0 END) as presente,
    SUM(CASE WHEN a.status = 'ausente' THEN 1 ELSE 0 END) as ausente,
    ROUND(SUM(CASE WHEN a.status = 'presente' THEN 1 ELSE 0 END) * 100.0 / COUNT(*), 2) as percentage
    FROM attendances a
    INNER JOIN courses c ON a.course_id = c.id
    WHERE a.date BETWEEN :start_date AND :end_date
    GROUP BY c.id, c.name
    ORDER BY percentage DESC";

    $db = new Database();
    $stmt = $db->connect()->prepare($sql);
    $stmt->execute([':start_date' => $startDate, ':end_date' => $endDate]);
    return $stmt->fetchAll();
}

private function getTopAbsentStudents($startDate, $endDate, $limit = 10) {
    $sql = "SELECT
    CONCAT(u.last_name, ' ', u.first_name) as student_name,
    c.name as course_name,
    COUNT(*) as total_absences
    FROM attendances a
    INNER JOIN users u ON a.student_id = u.id
    INNER JOIN courses c ON a.course_id = c.id
    WHERE a.status = 'ausente'
    AND a.date BETWEEN :start_date AND :end_date
    GROUP BY a.student_id, u.last_name, u.first_name, c.name
    ORDER BY total_absences DESC
    LIMIT :limit";

    $db = new Database();
    $stmt = $db->connect()->prepare($sql);
    $stmt->bindValue(':start_date', $startDate);
    $stmt->bindValue(':end_date', $endDate);
    $stmt->bindValue(':limit', $limit, PDO::PARAM_INT);
    $stmt->execute();
    return $stmt->fetchAll();
}

private function getStatsByDay($startDate, $endDate) {
    $sql = "SELECT
    date,
    COUNT(*) as total,
    SUM(CASE WHEN status = 'presente' THEN 1 ELSE 0 END) as presente,
    ROUND(SUM(CASE WHEN status = 'presente' THEN 1 ELSE 0 END) * 100.0 / COUNT(*), 2) as percentage
    FROM attendances
    WHERE date BETWEEN :start_date AND :end_date
    GROUP BY date
    ORDER BY date ASC";

    $db = new Database();
    $stmt = $db->connect()->prepare($sql);

```

```

$stmt->execute([':start_date' => $startDate, ':end_date' => $endDate]);
return $stmt->fetchAll();
}
}

```

## 20. ArchImp/controllers/UserController.php

```

<?php
require_once BASE_PATH . '/config/config.php';
require_once BASE_PATH . '/models/User.php';
require_once BASE_PATH . '/models/Role.php';

class UserController {
    private $userModel;
    private $roleModel;

    public function __construct() {
        Security::requireLogin();
        if (!Security::hasRole('autoridad')) {
            die('Acceso denegado');
        }

        $db = new Database();
        $this->userModel = new User($db);
        $this->roleModel = new Role($db);
    }

    public function index() {
        $users = $this->userModel->getAll();
        $roles = $this->roleModel->getAll();
        include BASE_PATH . '/views/users/index.php';
    }

    public function assignRole() {
        if ($_SERVER['REQUEST_METHOD'] === 'POST') {
            $userId = (int)$_POST['user_id'];
            $roleId = (int)$_POST['role_id'];

            $this->userModel->assignRole($userId, $roleId);
            $filter = isset($_GET['filter_role']) ? '&filter_role=' . $_GET['filter_role'] : '';
            header('Location: ?action=users&success=1' . $filter);
            exit;
        }
    }

    public function removeRole() {
        if ($_SERVER['REQUEST_METHOD'] === 'POST') {
            $userId = (int)$_POST['user_id'];
            $roleId = (int)$_POST['role_id'];

            // Verificar si el rol es "docente"
            $db = new Database();
            $stmt = $db->connect()->prepare("SELECT name FROM roles WHERE id = :role_id");
            $stmt->execute([':role_id' => $roleId]);
            $role = $stmt->fetch();

            if ($role && $role['name'] === 'docente') {
                // Eliminar asignaciones de materias
                $stmt = $db->connect()->prepare("DELETE FROM teacher_assignments WHERE teacher_id = :user_id");
                $stmt->execute([':user_id' => $userId]);
            }

            $this->userModel->removeRole($userId, $roleId);
        }
    }
}

```

```

$filter = isset($_GET['filter_role']) ? '&filter_role=' . $_GET['filter_role'] : '';
header('Location: ?action=users&removed=1' . $filter);
exit;
}

}

public function create() {
if ($_SERVER['REQUEST_METHOD'] === 'GET') {
$roles = $this->roleModel->getAll();
include BASE_PATH . '/views/users/create.php';
return;
}

if ($_SERVER['REQUEST_METHOD'] === 'POST') {
$errors = [];

// Validaciones
if (empty($_POST['username'])) {
$errors[] = "El nombre de usuario es obligatorio";
}
if (empty($_POST['email']) || !filter_var($_POST['email'], FILTER_VALIDATE_EMAIL)) {
$errors[] = "Email inválido";
}
if (empty($_POST['password']) || strlen($_POST['password']) < 6) {
$errors[] = "La contraseña debe tener al menos 6 caracteres";
}
if ($_POST['password'] !== $_POST['confirm_password']) {
$errors[] = "Las contraseñas no coinciden";
}
if (empty($_POST['first_name'])) {
$errors[] = "El nombre es obligatorio";
}
if (empty($_POST['last_name'])) {
$errors[] = "El apellido es obligatorio";
}

// Verificar username único
if ($this->userModel->findByUsername($_POST['username'])) {
$errors[] = "El nombre de usuario ya está registrado";
}

// Verificar email único
if ($this->userModel->findByEmail($_POST['email'])) {
$errors[] = "El email ya está registrado";
}

if (!empty($errors)) {
$roles = $this->roleModel->getAll();
include BASE_PATH . '/views/users/create.php';
return;
}

// Crear usuario
$userData = [
'institution_id' => $_SESSION['institution_id'],
'username' => Security::sanitize($_POST['username']),
'email' => Security::sanitize($_POST['email']),
'password' => $_POST['password'],
'first_name' => Security::sanitize($_POST['first_name']),
'last_name' => Security::sanitize($_POST['last_name']),
'dni' => !empty($_POST['dni']) ? Security::sanitize($_POST['dni']) : null,
'phone' => !empty($_POST['phone']) ? Security::sanitize($_POST['phone']) : null
];

```

```

if ($this->userModel->create($userData)) {
    // Obtener ID del usuario recién creado
    $newUserId = $this->userModel->db->lastInsertId();

    // Asignar roles si se seleccionaron
    if (!empty($_POST['roles'])) {
        foreach ($_POST['roles'] as $roleId) {
            $this->userModel->assignRole($newUserId, (int)$roleId);
        }
    }

    header('Location: ?action=users&created=1');
    exit;
} else {
    $errors[] = "Error al crear el usuario";
    $roles = $this->roleModel->getAll();
    include BASE_PATH . '/views/users/create.php';
}
}

public function edit() {
    $userId = (int)$_GET['id'];
    $user = $this->userModel->findById($userId);

    if (!$user) {
        header('Location: ?action=users&error=not_found');
        exit;
    }

    // Verificar que pertenece a la misma institución
    if ($user['institution_id'] != $_SESSION['institution_id']) {
        die('Acceso denegado');
    }

    if ($_SERVER['REQUEST_METHOD'] === 'GET') {
        $roles = $this->roleModel->getAll();
        $userRoles = $this->userModel->getUserRoleIds($userId);
        include BASE_PATH . '/views/users/edit.php';
        return;
    }

    if ($_SERVER['REQUEST_METHOD'] === 'POST') {
        $errors = [];

        // Validaciones
        if (empty($_POST['username'])) {
            $errors[] = "El nombre de usuario es obligatorio";
        }
        if (empty($_POST['email']) || !filter_var($_POST['email'], FILTER_VALIDATE_EMAIL)) {
            $errors[] = "Email inválido";
        }
        if (empty($_POST['first_name'])) {
            $errors[] = "El nombre es obligatorio";
        }
        if (empty($_POST['last_name'])) {
            $errors[] = "El apellido es obligatorio";
        }

        // Verificar username único (excepto el actual)
        $existingUser = $this->userModel->findByUsername($_POST['username']);
        if ($existingUser && $existingUser['id'] != $userId) {
            $errors[] = "El nombre de usuario ya está registrado";
        }

        // Verificar email único (excepto el actual)
    }
}

```

```

$existingEmail = $this->userModel->findByEmail($_POST['email']);
if ($existingEmail && $existingEmail['id'] != $userId) {
$errors[] = "El email ya está registrado";
}

// Si hay nueva contraseña, validarla
if (!empty($_POST['password'])) {
if (strlen($_POST['password']) < 6) {
$errors[] = "La contraseña debe tener al menos 6 caracteres";
}
if ($_POST['password'] != $_POST['confirm_password']) {
$errors[] = "Las contraseñas no coinciden";
}
}

if (!empty($errors)) {
$roles = $this->roleModel->getAll();
$userRoles = $this->userModel->getUserRoleIds($userId);
include BASE_PATH . '/views/users/edit.php';
return;
}

// Actualizar usuario
$updateData = [
'id' => $userId,
'username' => Security::sanitize($_POST['username']),
'email' => Security::sanitize($_POST['email']),
'first_name' => Security::sanitize($_POST['first_name']),
'last_name' => Security::sanitize($_POST['last_name']),
'dni' => !empty($_POST['dni']) ? Security::sanitize($_POST['dni']) : null,
'phone' => !empty($_POST['phone']) ? Security::sanitize($_POST['phone']) : null
];

// Solo actualizar contraseña si se proporcionó una nueva
if (!empty($_POST['password'])) {
$updateData['password'] = $_POST['password'];
}

if ($this->userModel->update($updateData)) {
header('Location: ?action=users&updated=1');
exit;
} else {
$errors[] = "Error al actualizar el usuario";
$roles = $this->roleModel->getAll();
$userRoles = $this->userModel->getUserRoleIds($userId);
include BASE_PATH . '/views/users/edit.php';
}
}

public function delete() {
if ($_SERVER['REQUEST_METHOD'] === 'POST') {
$userId = (int)$_POST['user_id'];
$user = $this->userModel->findById($userId);

if (!$user) {
header('Location: ?action=users&error=not_found');
exit;
}

// Verificar que pertenece a la misma institución
if ($user['institution_id'] != $_SESSION['institution_id']) {
die('Acceso denegado');
}
}

```

```

// No permitir eliminar al propio usuario
if ($userId == $_SESSION['user_id']) {
header('Location: ?action=users&error=self_delete');
exit;
}

// Verificar si tiene asistencias registradas
$db = new Database();
$stmt = $db->connect()->prepare("SELECT COUNT(*) as count FROM attendances WHERE student_id = :user_id");
$stmt->execute([':user_id' => $userId]);
$result = $stmt->fetch();

if ($result['count'] > 0) {
// Soft delete - desactivar en lugar de eliminar
if ($this->userModel->deactivate($userId)) {
header('Location: ?action=users&deactivated=1');
exit;
}
} else {
// Hard delete - eliminar completamente
if ($this->userModel->delete($userId)) {
header('Location: ?action=users&deleted=1');
exit;
}
}

header('Location: ?action=users&error=delete_failed');
exit;
}
}
}

```

## 21. ArchImp/database/ecuasistencia2026\_db.sql

```

-- =====
-- ECUASIST 2026 - BASE DE DATOS COMPLETA
-- =====

-- Crear base de datos
CREATE DATABASE IF NOT EXISTS ecuasistencia2026_db
CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci;

USE ecuasistencia2026_db;

-- =====
-- ESTRUCTURA DE TABLAS
-- =====

-- Instituciones
CREATE TABLE institutions (
id INT PRIMARY KEY AUTO_INCREMENT,
name VARCHAR(255) NOT NULL,
code VARCHAR(50) UNIQUE NOT NULL,
address TEXT,
phone VARCHAR(20),
email VARCHAR(100),
logo VARCHAR(255),
is_active TINYINT(1) DEFAULT 1,
created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP
) ENGINE=InnoDB;

-- Usuarios

```

```

CREATE TABLE users (
  id INT PRIMARY KEY AUTO_INCREMENT,
  institution_id INT NOT NULL,
  username VARCHAR(50) UNIQUE NOT NULL,
  email VARCHAR(100) UNIQUE NOT NULL,
  password VARCHAR(255) NOT NULL,
  first_name VARCHAR(100) NOT NULL,
  last_name VARCHAR(100) NOT NULL,
  dni VARCHAR(20) UNIQUE,
  phone VARCHAR(20),
  reset_token VARCHAR(255),
  reset_expires DATETIME,
  is_active TINYINT(1) DEFAULT 1,
  created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
  updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
  FOREIGN KEY (institution_id) REFERENCES institutions(id)
) ENGINE=InnoDB;

-- Roles
CREATE TABLE roles (
  id INT PRIMARY KEY AUTO_INCREMENT,
  name VARCHAR(50) UNIQUE NOT NULL,
  description TEXT,
  created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
  updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP
) ENGINE=InnoDB;

-- Usuarios-Roles
CREATE TABLE user_roles (
  id INT PRIMARY KEY AUTO_INCREMENT,
  user_id INT NOT NULL,
  role_id INT NOT NULL,
  assigned_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
  FOREIGN KEY (user_id) REFERENCES users(id) ON DELETE CASCADE,
  FOREIGN KEY (role_id) REFERENCES roles(id) ON DELETE CASCADE,
  UNIQUE KEY unique_user_role (user_id, role_id)
) ENGINE=InnoDB;

-- Permisos especiales
CREATE TABLE permissions (
  id INT PRIMARY KEY AUTO_INCREMENT,
  user_id INT NOT NULL,
  permission_type VARCHAR(50) NOT NULL,
  entity_id INT,
  created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
  FOREIGN KEY (user_id) REFERENCES users(id) ON DELETE CASCADE
) ENGINE=InnoDB;

-- Años lectivos
CREATE TABLE school_years (
  id INT PRIMARY KEY AUTO_INCREMENT,
  institution_id INT NOT NULL,
  name VARCHAR(100) NOT NULL,
  start_date DATE NOT NULL,
  end_date DATE NOT NULL,
  is_active TINYINT(1) DEFAULT 0,
  created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
  updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
  FOREIGN KEY (institution_id) REFERENCES institutions(id)
) ENGINE=InnoDB;

-- Jornadas
CREATE TABLE shifts (
  id INT PRIMARY KEY AUTO_INCREMENT,
  name VARCHAR(50) NOT NULL,

```

```

created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
) ENGINE=InnoDB;

-- Cursos
CREATE TABLE courses (
id INT PRIMARY KEY AUTO_INCREMENT,
institution_id INT NOT NULL,
school_year_id INT NOT NULL,
name VARCHAR(100) NOT NULL,
grade_level VARCHAR(50),
parallel VARCHAR(10),
shift_id INT NOT NULL,
created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
FOREIGN KEY (institution_id) REFERENCES institutions(id),
FOREIGN KEY (school_year_id) REFERENCES school_years(id),
FOREIGN KEY (shift_id) REFERENCES shifts(id)
) ENGINE=InnoDB;

-- Asignaturas
CREATE TABLE subjects (
id INT PRIMARY KEY AUTO_INCREMENT,
institution_id INT NOT NULL,
name VARCHAR(100) NOT NULL,
code VARCHAR(20),
created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
FOREIGN KEY (institution_id) REFERENCES institutions(id)
) ENGINE=InnoDB;

-- Asignaciones docente
CREATE TABLE teacher_assignments (
id INT PRIMARY KEY AUTO_INCREMENT,
teacher_id INT NOT NULL,
course_id INT NOT NULL,
subject_id INT NOT NULL,
school_year_id INT NOT NULL,
is_tutor TINYINT(1) DEFAULT 0,
created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
FOREIGN KEY (teacher_id) REFERENCES users(id),
FOREIGN KEY (course_id) REFERENCES courses(id),
FOREIGN KEY (subject_id) REFERENCES subjects(id),
FOREIGN KEY (school_year_id) REFERENCES school_years(id)
) ENGINE=InnoDB;

-- Estudiantes en cursos
CREATE TABLE course_students (
id INT PRIMARY KEY AUTO_INCREMENT,
student_id INT NOT NULL,
course_id INT NOT NULL,
school_year_id INT NOT NULL,
enrollment_date DATE,
created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
FOREIGN KEY (student_id) REFERENCES users(id),
FOREIGN KEY (course_id) REFERENCES courses(id),
FOREIGN KEY (school_year_id) REFERENCES school_years(id),
UNIQUE KEY unique_enrollment (student_id, course_id, school_year_id)
) ENGINE=InnoDB;

-- Representantes
CREATE TABLE representatives (
id INT PRIMARY KEY AUTO_INCREMENT,
representative_id INT NOT NULL,
student_id INT NOT NULL,
relationship VARCHAR(50),

```



```

is_primary TINYINT(1) DEFAULT 0,
created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
FOREIGN KEY (representative_id) REFERENCES users(id),
FOREIGN KEY (student_id) REFERENCES users(id),
UNIQUE KEY unique_rep_student (representative_id, student_id)
) ENGINE=InnoDB;

-- Asistencias
CREATE TABLE attendances (
id INT PRIMARY KEY AUTO_INCREMENT,
student_id INT NOT NULL,
course_id INT NOT NULL,
subject_id INT NOT NULL,
teacher_id INT NOT NULL,
school_year_id INT NOT NULL,
shift_id INT NOT NULL,
date DATE NOT NULL,
hour_period VARCHAR(20),
status ENUM('presente', 'ausente', 'tardanza', 'justificado') NOT NULL,
observation TEXT,
created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
FOREIGN KEY (student_id) REFERENCES users(id),
FOREIGN KEY (course_id) REFERENCES courses(id),
FOREIGN KEY (subject_id) REFERENCES subjects(id),
FOREIGN KEY (teacher_id) REFERENCES users(id),
FOREIGN KEY (school_year_id) REFERENCES school_years(id),
FOREIGN KEY (shift_id) REFERENCES shifts(id),
INDEX idx_date (date),
INDEX idx_student (student_id)
) ENGINE=InnoDB;

-- Justificaciones
CREATE TABLE justifications (
id INT PRIMARY KEY AUTO_INCREMENT,
attendance_id INT NOT NULL,
student_id INT NOT NULL,
submitted_by INT NOT NULL,
reason TEXT NOT NULL,
document_path VARCHAR(255),
status ENUM('pendiente', 'aprobado', 'rechazado') DEFAULT 'pendiente',
reviewed_by INT,
review_notes TEXT,
created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
FOREIGN KEY (attendance_id) REFERENCES attendances(id) ON DELETE CASCADE,
FOREIGN KEY (student_id) REFERENCES users(id),
FOREIGN KEY (submitted_by) REFERENCES users(id),
FOREIGN KEY (reviewed_by) REFERENCES users(id),
INDEX idx_status (status),
INDEX idx_student (student_id)
) ENGINE=InnoDB;

-- Notificaciones
CREATE TABLE notifications (
id INT PRIMARY KEY AUTO_INCREMENT,
user_id INT NOT NULL,
title VARCHAR(255) NOT NULL,
message TEXT NOT NULL,
type ENUM('info', 'warning', 'success', 'danger') DEFAULT 'info',
is_read TINYINT(1) DEFAULT 0,
created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
FOREIGN KEY (user_id) REFERENCES users(id) ON DELETE CASCADE,
INDEX idx_user_read (user_id, is_read)
) ENGINE=InnoDB;

```

```

-- Logs de actividad
CREATE TABLE activity_logs (
id INT PRIMARY KEY AUTO_INCREMENT,
user_id INT NOT NULL,
action VARCHAR(100) NOT NULL,
entity_type VARCHAR(50),
entity_id INT,
description TEXT,
ip_address VARCHAR(45),
user_agent VARCHAR(255),
created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
FOREIGN KEY (user_id) REFERENCES users(id),
INDEX idx_user (user_id),
INDEX idx_date (created_at)
) ENGINE=InnoDB;

-- =====
-- DATOS INICIALES
-- =====

-- Institución
INSERT INTO institutions (name, code, address, phone, email, is_active) VALUES
('Unidad Educativa Pomasqui', 'UEP001', 'Av. Manuel Cordova Galarza 123, Quito, Ecuador', '02-2351072',
'contacto@uep.edu.ec', 1);

-- Roles base
INSERT INTO roles (name, description) VALUES
('docente', 'Profesor de la institución'),
('estudiante', 'Estudiante activo'),
('inspector', 'Inspector de disciplina'),
('autoridad', 'Autoridad institucional'),
('representante', 'Representante legal de estudiante');

-- Jornadas
INSERT INTO shifts (name) VALUES
('matutina'),
('vespertina'),
('nocturna');

-- Año lectivo activo
INSERT INTO school_years (institution_id, name, start_date, end_date, is_active) VALUES
(1, '2025-2026', '2025-09-01', '2026-07-30', 1);

-- =====
-- USUARIOS DE PRUEBA
-- Contraseña para todos: "password"
-- =====

-- Usuario Administrador
INSERT INTO users (institution_id, username, email, password, first_name, last_name, dni, phone, is_active)
VALUES
(1, 'admin', 'admin@uep.edu.ec', '$2y$10$92IXUNpkj00rOQ5byMi.Ye4oKoEa3Ro9llC/.og/at2.uheWG/igi',
'Administrador', 'Sistema', '1700000000', '0999999999', 1);

-- Docentes (6)
INSERT INTO users (institution_id, username, email, password, first_name, last_name, dni, phone, is_active)
VALUES
(1, 'prof.garcia', 'garcia@uep.edu.ec', '$2y$10$92IXUNpkj00rOQ5byMi.Ye4oKoEa3Ro9llC/.og/at2.uheWG/igi',
'Roberto', 'García Mendoza', '1751234567', '0991234571', 1),
(1, 'prof.martinez', 'martinez@uep.edu.ec', '$2y$10$92IXUNpkj00rOQ5byMi.Ye4oKoEa3Ro9llC/.og/at2.uheWG/igi',
'Ana', 'Martínez Silva', '1751234568', '0991234572', 1),
(1, 'prof.rodriquez', 'rodriquez@uep.edu.ec', '$2y$10$92IXUNpkj00rOQ5byMi.Ye4oKoEa3Ro9llC/.og/at2.uheWG/igi',
'Luis', 'Rodríguez Cano', '1751234569', '0991234573', 1),

```

```

(1, 'prof.fernandez', 'fernandez@uep.edu.ec',
'$2y$10$92IXUNpkj00rOQ5byMi.Ye4oKoEa3Ro9llC/.og/at2.uheWG/igi', 'Carmen', 'Fernández Ortiz', '1751234570',
'0991234574', 1),
(1, 'prof.gomez', 'gomez@uep.edu.ec', '$2y$10$92IXUNpkj00rOQ5byMi.Ye4oKoEa3Ro9llC/.og/at2.uheWG/igi',
'Miguel', 'Gómez Torres', '1751234571', '0991234575', 1),
(1, 'prof.diaz', 'diaz@uep.edu.ec', '$2y$10$92IXUNpkj00rOQ5byMi.Ye4oKoEa3Ro9llC/.og/at2.uheWG/igi',
'Patricia', 'Díaz Ramírez', '1751234572', '0991234576', 1);

-- Estudiantes (15)
INSERT INTO users (institution_id, username, email, password, first_name, last_name, dni, phone, is_active)
VALUES
(1, 'juan.perez', 'juan.perez@estudiante.edu.ec',
'$2y$10$92IXUNpkj00rOQ5byMi.Ye4oKoEa3Ro9llC/.og/at2.uheWG/igi', 'Juan', 'Pérez García', '1750123456',
'0991234567', 1),
(1, 'maria.lopez', 'maria.lopez@estudiante.edu.ec',
'$2y$10$92IXUNpkj00rOQ5byMi.Ye4oKoEa3Ro9llC/.og/at2.uheWG/igi', 'María', 'López Sánchez', '1750234567',
'0991234568', 1),
(1, 'carlos.mora', 'carlos.mora@estudiante.edu.ec',
'$2y$10$92IXUNpkj00rOQ5byMi.Ye4oKoEa3Ro9llC/.og/at2.uheWG/igi', 'Carlos', 'Mora Ruiz', '1750345678',
'0991234569', 1),
(1, 'ana.torres', 'ana.torres@estudiante.edu.ec',
'$2y$10$92IXUNpkj00rOQ5byMi.Ye4oKoEa3Ro9llC/.og/at2.uheWG/igi', 'Ana', 'Torres Vega', '1750456789',
'0991234570', 1),
(1, 'sofia.castro', 'sofia.castro@estudiante.edu.ec',
'$2y$10$92IXUNpkj00rOQ5byMi.Ye4oKoEa3Ro9llC/.og/at2.uheWG/igi', 'Sofía', 'Castro Morales', '1750567890',
'0991234577', 1),
(1, 'diego.herrera', 'diego.herrera@estudiante.edu.ec',
'$2y$10$92IXUNpkj00rOQ5byMi.Ye4oKoEa3Ro9llC/.og/at2.uheWG/igi', 'Diego', 'Herrera Ríos', '1750567891',
'0991234578', 1),
(1, 'valentina.lopez', 'valentina.lopez@estudiante.edu.ec',
'$2y$10$92IXUNpkj00rOQ5byMi.Ye4oKoEa3Ro9llC/.og/at2.uheWG/igi', 'Valentina', 'López Suárez', '1750567892',
'0991234579', 1),
(1, 'mateo.silva', 'mateo.silva@estudiante.edu.ec',
'$2y$10$92IXUNpkj00rOQ5byMi.Ye4oKoEa3Ro9llC/.og/at2.uheWG/igi', 'Mateo', 'Silva Navarro', '1750567893',
'0991234580', 1),
(1, 'isabella.rojas', 'isabella.rojas@estudiante.edu.ec',
'$2y$10$92IXUNpkj00rOQ5byMi.Ye4oKoEa3Ro9llC/.og/at2.uheWG/igi', 'Isabella', 'Rojas Pérez', '1750567894',
'0991234581', 1),
(1, 'sebastian.cruz', 'sebastian.cruz@estudiante.edu.ec',
'$2y$10$92IXUNpkj00rOQ5byMi.Ye4oKoEa3Ro9llC/.og/at2.uheWG/igi', 'Sebastián', 'Cruz Méndez', '1750567895',
'0991234582', 1),
(1, 'camila.ramos', 'camila.ramos@estudiante.edu.ec',
'$2y$10$92IXUNpkj00rOQ5byMi.Ye4oKoEa3Ro9llC/.og/at2.uheWG/igi', 'Camila', 'Ramos Flores', '1750567896',
'0991234583', 1),
(1, 'nicolas.vargas', 'nicolas.vargas@estudiante.edu.ec',
'$2y$10$92IXUNpkj00rOQ5byMi.Ye4oKoEa3Ro9llC/.og/at2.uheWG/igi', 'Nicolás', 'Vargas Castro', '1750567897',
'0991234584', 1),
(1, 'martina.gil', 'martina.gil@estudiante.edu.ec',
'$2y$10$92IXUNpkj00rOQ5byMi.Ye4oKoEa3Ro9llC/.og/at2.uheWG/igi', 'Martina', 'Gil Núñez', '1750567898',
'0991234585', 1),
(1, 'lucas.medina', 'lucas.medina@estudiante.edu.ec',
'$2y$10$92IXUNpkj00rOQ5byMi.Ye4oKoEa3Ro9llC/.og/at2.uheWG/igi', 'Lucas', 'Medina Ortega', '1750567899',
'0991234586', 1),
(1, 'emma.santos', 'emma.santos@estudiante.edu.ec',
'$2y$10$92IXUNpkj00rOQ5byMi.Ye4oKoEa3Ro9llC/.og/at2.uheWG/igi', 'Emma', 'Santos Molina', '1750567900',
'0991234587', 1);

-- Representantes (5)
INSERT INTO users (institution_id, username, email, password, first_name, last_name, dni, phone, is_active)
VALUES
(1, 'rep.castro', 'lcastro@gmail.com', '$2y$10$92IXUNpkj00rOQ5byMi.Ye4oKoEa3Ro9llC/.og/at2.uheWG/igi',
'Laura', 'Castro Vega', '1752345678', '0991234588', 1),
(1, 'rep.herrera', 'jherrera@gmail.com', '$2y$10$92IXUNpkj00rOQ5byMi.Ye4oKoEa3Ro9llC/.og/at2.uheWG/igi',
'Jorge', 'Herrera Palacios', '1752345679', '0991234589', 1),
(1, 'rep.lopez', 'mlopez@gmail.com', '$2y$10$92IXUNpkj00rOQ5byMi.Ye4oKoEa3Ro9llC/.og/at2.uheWG/igi',

```

```

'María', 'López Benítez', '1752345680', '0991234590', 1),
(1, 'rep.silva', 'psilva@gmail.com', '$2y$10$92IXUNpkj00rOQ5byMi.Ye4oKoEa3Ro9llC/.og/at2.uheWG/igi',
'Pedro', 'Silva Moreno', '1752345681', '0991234591', 1),
(1, 'rep.rojas', 'crojas@gmail.com', '$2y$10$92IXUNpkj00rOQ5byMi.Ye4oKoEa3Ro9llC/.og/at2.uheWG/igi',
'Carolina', 'Rojas Delgado', '1752345682', '0991234592', 1);

-- Inspector
INSERT INTO users (institution_id, username, email, password, first_name, last_name, dni, phone, is_active)
VALUES
(1, 'inspector', 'inspector@uep.edu.ec', '$2y$10$92IXUNpkj00rOQ5byMi.Ye4oKoEa3Ro9llC/.og/at2.uheWG/igi',
'José', 'Ramírez Campos', '1753456789', '0991234593', 1);

-- =====
-- ASIGNAR ROLES
-- =====

-- Admin (autoridad)
INSERT INTO user_roles (user_id, role_id) VALUES (1, 4);

-- Docentes
INSERT INTO user_roles (user_id, role_id) VALUES
(2, 1), (3, 1), (4, 1), (5, 1), (6, 1), (7, 1);

-- Estudiantes
INSERT INTO user_roles (user_id, role_id) VALUES
(8, 2), (9, 2), (10, 2), (11, 2), (12, 2), (13, 2), (14, 2), (15, 2),
(16, 2), (17, 2), (18, 2), (19, 2), (20, 2), (21, 2), (22, 2);

-- Representantes
INSERT INTO user_roles (user_id, role_id) VALUES
(23, 5), (24, 5), (25, 5), (26, 5), (27, 5);

-- Inspector
INSERT INTO user_roles (user_id, role_id) VALUES (28, 3);

-- =====
-- ASIGNATURAS
-- =====

INSERT INTO subjects (institution_id, name, code) VALUES
(1, 'Matemáticas', 'MAT'),
(1, 'Lengua y Literatura', 'LEN'),
(1, 'Ciencias Naturales', 'CCN'),
(1, 'Estudios Sociales', 'ESS'),
(1, 'Inglés', 'ING'),
(1, 'Educación Física', 'EDF'),
(1, 'Arte y Cultura', 'ART'),
(1, 'Informática', 'INF');

-- =====
-- CURSOS
-- =====

INSERT INTO courses (institution_id, school_year_id, name, grade_level, parallel, shift_id) VALUES
(1, 1, 'Octavo A', 'Octavo', 'A', 1),
(1, 1, 'Séptimo A', 'Séptimo', 'A', 1),
(1, 1, 'Octavo B', 'Octavo', 'B', 1),
(1, 1, 'Noveno A', 'Noveno', 'A', 2),
(1, 1, 'Décimo A', 'Décimo', 'A', 2);

-- =====
-- MATRICULAR ESTUDIANTES
-- =====

-- Octavo A (curso 1)
INSERT INTO course_students (student_id, course_id, school_year_id, enrollment_date) VALUES

```

```

(8, 1, 1, '2025-09-01'),
(9, 1, 1, '2025-09-01'),
(10, 1, 1, '2025-09-01'),
(11, 1, 1, '2025-09-01');

-- Séptimo A (curso 2)
INSERT INTO course_students (student_id, course_id, school_year_id, enrollment_date) VALUES
(12, 2, 1, '2025-09-01'),
(13, 2, 1, '2025-09-01'),
(14, 2, 1, '2025-09-01');

-- Octavo B (curso 3)
INSERT INTO course_students (student_id, course_id, school_year_id, enrollment_date) VALUES
(15, 3, 1, '2025-09-01'),
(16, 3, 1, '2025-09-01'),
(17, 3, 1, '2025-09-01');

-- Noveno A (curso 4)
INSERT INTO course_students (student_id, course_id, school_year_id, enrollment_date) VALUES
(18, 4, 1, '2025-09-01'),
(19, 4, 1, '2025-09-01'),
(20, 4, 1, '2025-09-01');

-- Décimo A (curso 5)
INSERT INTO course_students (student_id, course_id, school_year_id, enrollment_date) VALUES
(21, 5, 1, '2025-09-01'),
(22, 5, 1, '2025-09-01');

-- =====
-- ASIGNACIONES DOCENTES
-- =====

-- Profesor García (2) - Matemáticas, TUTOR Octavo A
INSERT INTO teacher_assignments (teacher_id, course_id, subject_id, school_year_id, is_tutor) VALUES
(2, 1, 1, 1, 1),
(2, 2, 1, 1, 0),
(2, 3, 1, 1, 0);

-- Profesora Martínez (3) - Lengua, TUTORA Séptimo A
INSERT INTO teacher_assignments (teacher_id, course_id, subject_id, school_year_id, is_tutor) VALUES
(3, 1, 2, 1, 0),
(3, 2, 2, 1, 1),
(3, 4, 2, 1, 0);

-- Profesor Rodríguez (4) - Ciencias, TUTOR Octavo B
INSERT INTO teacher_assignments (teacher_id, course_id, subject_id, school_year_id, is_tutor) VALUES
(4, 1, 3, 1, 0),
(4, 3, 3, 1, 1),
(4, 5, 3, 1, 0);

-- Profesora Fernández (5) - Estudios Sociales, TUTORA Noveno A
INSERT INTO teacher_assignments (teacher_id, course_id, subject_id, school_year_id, is_tutor) VALUES
(5, 2, 4, 1, 0),
(5, 3, 4, 1, 0),
(5, 4, 4, 1, 1);

-- Profesor Gómez (6) - Inglés, TUTOR Décimo A
INSERT INTO teacher_assignments (teacher_id, course_id, subject_id, school_year_id, is_tutor) VALUES
(6, 1, 5, 1, 0),
(6, 2, 5, 1, 0),
(6, 4, 5, 1, 0),
(6, 5, 5, 1, 1);

-- Profesora Díaz (7) - Educación Física
INSERT INTO teacher_assignments (teacher_id, course_id, subject_id, school_year_id, is_tutor) VALUES

```

```

(7, 1, 6, 1, 0),
(7, 3, 6, 1, 0),
(7, 5, 6, 1, 0);

-- =====
-- REPRESENTANTES
-- =====

INSERT INTO representatives (representative_id, student_id, relationship, is_primary) VALUES
(23, 12, 'Madre', 1),
(23, 13, 'Madre', 1),
(24, 14, 'Padre', 1),
(25, 15, 'Madre', 1),
(25, 16, 'Madre', 1),
(26, 17, 'Padre', 1),
(26, 18, 'Padre', 1),
(27, 19, 'Madre', 1),
(27, 20, 'Madre', 1),
(27, 21, 'Madre', 1);

-- =====
-- ASISTENCIAS DE PRUEBA
-- =====

-- Semana pasada - Octavo A - Matemáticas
INSERT INTO attendances (student_id, course_id, subject_id, teacher_id, school_year_id, shift_id, date,
hour_period, status, observation) VALUES
-- Lunes
(8, 1, 1, 2, 1, 1, DATE_SUB(CURDATE(), INTERVAL 4 DAY), '1ra hora', 'presente', ''),
(9, 1, 1, 2, 1, 1, DATE_SUB(CURDATE(), INTERVAL 4 DAY), '1ra hora', 'presente', ''),
(10, 1, 1, 2, 1, 1, DATE_SUB(CURDATE(), INTERVAL 4 DAY), '1ra hora', 'ausente', 'Sin justificación'),
(11, 1, 1, 2, 1, 1, DATE_SUB(CURDATE(), INTERVAL 4 DAY), '1ra hora', 'presente', ''),
-- Martes
(8, 1, 1, 2, 1, 1, DATE_SUB(CURDATE(), INTERVAL 3 DAY), '1ra hora', 'tardanza', 'Llegó 10 min tarde'),
(9, 1, 1, 2, 1, 1, DATE_SUB(CURDATE(), INTERVAL 3 DAY), '1ra hora', 'presente', ''),
(10, 1, 1, 2, 1, 1, DATE_SUB(CURDATE(), INTERVAL 3 DAY), '1ra hora', 'justificado', 'Cita médica'),
(11, 1, 1, 2, 1, 1, DATE_SUB(CURDATE(), INTERVAL 3 DAY), '1ra hora', 'presente', ''),
-- Miércoles
(8, 1, 1, 2, 1, 1, DATE_SUB(CURDATE(), INTERVAL 2 DAY), '1ra hora', 'presente', ''),
(9, 1, 1, 2, 1, 1, DATE_SUB(CURDATE(), INTERVAL 2 DAY), '1ra hora', 'presente', ''),
(10, 1, 1, 2, 1, 1, DATE_SUB(CURDATE(), INTERVAL 2 DAY), '1ra hora', 'presente', ''),
(11, 1, 1, 2, 1, 1, DATE_SUB(CURDATE(), INTERVAL 2 DAY), '1ra hora', 'ausente', 'Enfermo'),
-- Jueves
(8, 1, 1, 2, 1, 1, DATE_SUB(CURDATE(), INTERVAL 1 DAY), '1ra hora', 'presente', ''),
(9, 1, 1, 2, 1, 1, DATE_SUB(CURDATE(), INTERVAL 1 DAY), '1ra hora', 'presente', ''),
(10, 1, 1, 2, 1, 1, DATE_SUB(CURDATE(), INTERVAL 1 DAY), '1ra hora', 'presente', ''),
(11, 1, 1, 2, 1, 1, DATE_SUB(CURDATE(), INTERVAL 1 DAY), '1ra hora', 'tardanza', 'Tráfico'),
-- Hoy
(8, 1, 1, 2, 1, 1, CURDATE(), '1ra hora', 'presente', ''),
(9, 1, 1, 2, 1, 1, CURDATE(), '1ra hora', 'presente', ''),
(10, 1, 1, 2, 1, 1, CURDATE(), '1ra hora', 'presente', ''),
(11, 1, 1, 2, 1, 1, CURDATE(), '1ra hora', 'presente', '');

-- Lengua y Literatura - Octavo A
INSERT INTO attendances (student_id, course_id, subject_id, teacher_id, school_year_id, shift_id, date,
hour_period, status, observation) VALUES
(8, 1, 2, 3, 1, 1, DATE_SUB(CURDATE(), INTERVAL 2 DAY), '2da hora', 'presente', ''),
(9, 1, 2, 3, 1, 1, DATE_SUB(CURDATE(), INTERVAL 2 DAY), '2da hora', 'presente', 'Participación activa'),
(10, 1, 2, 3, 1, 1, DATE_SUB(CURDATE(), INTERVAL 2 DAY), '2da hora', 'ausente', ''),
(11, 1, 2, 3, 1, 1, DATE_SUB(CURDATE(), INTERVAL 2 DAY), '2da hora', 'presente', '');

-- =====
-- ESTADÍSTICAS FINALES
-- =====

```

```

SELECT 'BASE DE DATOS CREADA EXITOSAMENTE' as '■';
SELECT '' as '';
SELECT 'RESUMEN:' as '';
SELECT COUNT(*) as 'Total Usuarios' FROM users;
SELECT COUNT(*) as 'Docentes' FROM user_roles WHERE role_id = 1;
SELECT COUNT(*) as 'Estudiantes' FROM user_roles WHERE role_id = 2;
SELECT COUNT(*) as 'Inspectores' FROM user_roles WHERE role_id = 3;
SELECT COUNT(*) as 'Autoridades' FROM user_roles WHERE role_id = 4;
SELECT COUNT(*) as 'Representantes' FROM user_roles WHERE role_id = 5;
SELECT COUNT(*) as 'Cursos' FROM courses;
SELECT COUNT(*) as 'Asignaturas' FROM subjects;
SELECT COUNT(*) as 'Matrículas' FROM course_students;
SELECT COUNT(*) as 'Asignaciones Docentes' FROM teacher_assignments;
SELECT COUNT(*) as 'Relaciones Rep-Estudiante' FROM representatives;
SELECT COUNT(*) as 'Registros Asistencia' FROM attendances;

SELECT '' as '';
SELECT 'CREDENCIALES DE ACCESO:' as '';
SELECT 'Usuario: admin | Contraseña: password' as 'Administrador';
SELECT 'Usuario: prof.garcia | Contraseña: password' as 'Docente';
SELECT 'Usuario: juan.perez | Contraseña: password' as 'Estudiante';
SELECT 'Usuario: rep.castro | Contraseña: password' as 'Representante';
SELECT 'Usuario: inspector | Contraseña: password' as 'Inspector';

```

## 22. ArchImp/database/ecuasistencia2026\_db0.sql

```

-- =====
-- ECUASIST 2026 - Base de Datos Limpia
-- Solo estructura y usuario administrador
-- =====

DROP DATABASE IF EXISTS ecuasistencia2026_db;
CREATE DATABASE ecuasistencia2026_db CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci;
USE ecuasistencia2026_db;

-- =====
-- TABLA: institutions
-- =====
CREATE TABLE institutions (
id INT PRIMARY KEY AUTO_INCREMENT,
name VARCHAR(255) NOT NULL,
address VARCHAR(255),
phone VARCHAR(20),
email VARCHAR(100),
logo_path VARCHAR(255),
created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;

INSERT INTO institutions (name, address, phone, email) VALUES
('Unidad Educativa Demo', 'Quito, Ecuador', '(02) 2800000', 'info@uedemo.edu.ec');

-- =====
-- TABLA: users
-- =====
CREATE TABLE users (
id INT PRIMARY KEY AUTO_INCREMENT,
institution_id INT NOT NULL DEFAULT 1,
username VARCHAR(50) UNIQUE NOT NULL,
email VARCHAR(100) UNIQUE NOT NULL,
password VARCHAR(255) NOT NULL,
first_name VARCHAR(100) NOT NULL,
last_name VARCHAR(100) NOT NULL,

```

```

dni VARCHAR(20) UNIQUE,
phone VARCHAR(20),
address TEXT,
created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
FOREIGN KEY (institution_id) REFERENCES institutions(id)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;

-- Usuario administrador
-- username: admin
-- password: password
INSERT INTO users (institution_id, username, email, password, first_name, last_name, dni) VALUES
(1, 'admin', 'admin@ecuasist.edu.ec', '$2y$10$92IXUNpkj0rOQ5byMi.Ye4oKoEa3Ro9llC/.og/at2.uheWG/igi',
'Administrador', 'Sistema', '0000000000');

-- =====
-- TABLA: roles
-- =====
CREATE TABLE roles (
id INT PRIMARY KEY AUTO_INCREMENT,
name VARCHAR(50) UNIQUE NOT NULL,
description TEXT,
created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;

INSERT INTO roles (name, description) VALUES
('autoridad', 'Acceso completo al sistema'),
('docente', 'Registro de asistencia y consulta'),
('estudiante', 'Visualización de asistencia propia'),
('inspector', 'Revisión de asistencias y justificaciones'),
('representante', 'Visualización de asistencia de representados');

-- =====
-- TABLA: user_roles (relación muchos a muchos)
-- =====
CREATE TABLE user_roles (
id INT PRIMARY KEY AUTO_INCREMENT,
user_id INT NOT NULL,
role_id INT NOT NULL,
created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
FOREIGN KEY (user_id) REFERENCES users(id) ON DELETE CASCADE,
FOREIGN KEY (role_id) REFERENCES roles(id) ON DELETE CASCADE,
UNIQUE KEY unique_user_role (user_id, role_id)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;

-- Asignar rol de autoridad al administrador
INSERT INTO user_roles (user_id, role_id) VALUES
(1, 1);

-- =====
-- TABLA: permissions
-- =====
CREATE TABLE permissions (
id INT PRIMARY KEY AUTO_INCREMENT,
name VARCHAR(100) NOT NULL,
description TEXT,
created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;

-- =====
-- TABLA: school_years
-- =====
CREATE TABLE school_years (
id INT PRIMARY KEY AUTO_INCREMENT,
institution_id INT NOT NULL DEFAULT 1,

```



```

name VARCHAR(50) NOT NULL,
start_date DATE NOT NULL,
end_date DATE NOT NULL,
is_active BOOLEAN DEFAULT 0,
created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
FOREIGN KEY (institution_id) REFERENCES institutions(id)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;

-- Año lectivo activo
INSERT INTO school_years (institution_id, name, start_date, end_date, is_active) VALUES
(1, '2025-2026', '2025-09-01', '2026-07-31', 1);

-- =====
-- TABLA: shifts (jornadas)
-- =====
CREATE TABLE shifts (
id INT PRIMARY KEY AUTO_INCREMENT,
institution_id INT NOT NULL DEFAULT 1,
name VARCHAR(50) NOT NULL,
created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
FOREIGN KEY (institution_id) REFERENCES institutions(id)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;

INSERT INTO shifts (institution_id, name) VALUES
(1, 'matutina'),
(1, 'Vespertina'),
(1, 'nocturna');

-- =====
-- TABLA: courses
-- =====
CREATE TABLE courses (
id INT PRIMARY KEY AUTO_INCREMENT,
institution_id INT NOT NULL DEFAULT 1,
school_year_id INT NOT NULL,
shift_id INT NOT NULL,
name VARCHAR(100) NOT NULL,
grade_level VARCHAR(50) NOT NULL,
parallel VARCHAR(10) NOT NULL,
created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
FOREIGN KEY (institution_id) REFERENCES institutions(id),
FOREIGN KEY (school_year_id) REFERENCES school_years(id),
FOREIGN KEY (shift_id) REFERENCES shifts(id)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;

-- =====
-- TABLA: subjects
-- =====
CREATE TABLE subjects (
id INT PRIMARY KEY AUTO_INCREMENT,
institution_id INT NOT NULL DEFAULT 1,
name VARCHAR(100) NOT NULL,
description TEXT,
created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
FOREIGN KEY (institution_id) REFERENCES institutions(id)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;

-- =====
-- TABLA: teacher_assignments
-- =====
CREATE TABLE teacher_assignments (
id INT PRIMARY KEY AUTO_INCREMENT,
teacher_id INT NOT NULL,

```

```

course_id INT NOT NULL,
subject_id INT NOT NULL,
school_year_id INT NOT NULL,
is_tutor BOOLEAN DEFAULT 0,
created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
FOREIGN KEY (teacher_id) REFERENCES users(id) ON DELETE CASCADE,
FOREIGN KEY (course_id) REFERENCES courses(id) ON DELETE CASCADE,
FOREIGN KEY (subject_id) REFERENCES subjects(id),
FOREIGN KEY (school_year_id) REFERENCES school_years(id)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;

-- =====
-- TABLA: class_schedule (horarios)
-- =====
CREATE TABLE class_schedule (
id INT PRIMARY KEY AUTO_INCREMENT,
course_id INT NOT NULL,
subject_id INT NOT NULL,
teacher_id INT NOT NULL,
school_year_id INT NOT NULL,
day_of_week ENUM('lunes','martes','miercoles','jueves','viernes','sabado') NOT NULL,
period_number INT NOT NULL COMMENT 'Número de hora: 1-8',
created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
FOREIGN KEY (course_id) REFERENCES courses(id) ON DELETE CASCADE,
FOREIGN KEY (subject_id) REFERENCES subjects(id),
FOREIGN KEY (teacher_id) REFERENCES users(id),
FOREIGN KEY (school_year_id) REFERENCES school_years(id),
UNIQUE KEY unique_schedule (course_id, day_of_week, period_number, school_year_id)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;

-- =====
-- TABLA: course_students
-- =====
CREATE TABLE course_students (
id INT PRIMARY KEY AUTO_INCREMENT,
course_id INT NOT NULL,
student_id INT NOT NULL,
school_year_id INT NOT NULL,
enrollment_date DATE DEFAULT (CURRENT_DATE),
created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
FOREIGN KEY (course_id) REFERENCES courses(id) ON DELETE CASCADE,
FOREIGN KEY (student_id) REFERENCES users(id) ON DELETE CASCADE,
FOREIGN KEY (school_year_id) REFERENCES school_years(id),
UNIQUE KEY unique_student_year (student_id, school_year_id)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;

-- =====
-- TABLA: attendances
-- =====
CREATE TABLE attendances (
id INT PRIMARY KEY AUTO_INCREMENT,
student_id INT NOT NULL,
course_id INT NOT NULL,
subject_id INT NOT NULL,
teacher_id INT NOT NULL,
school_year_id INT NOT NULL,
shift_id INT NOT NULL,
date DATE NOT NULL,
hour_period VARCHAR(50) NOT NULL,
status ENUM('presente', 'ausente', 'tardanza', 'justificado') NOT NULL,
observation TEXT,
created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,

```

```

FOREIGN KEY (student_id) REFERENCES users(id) ON DELETE CASCADE,
FOREIGN KEY (course_id) REFERENCES courses(id),
FOREIGN KEY (subject_id) REFERENCES subjects(id),
FOREIGN KEY (teacher_id) REFERENCES users(id),
FOREIGN KEY (school_year_id) REFERENCES school_years(id),
FOREIGN KEY (shift_id) REFERENCES shifts(id),
INDEX idx_student_date (student_id, date),
INDEX idx_course_date (course_id, date)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;

-- =====
-- TABLA: justifications
-- =====
CREATE TABLE justifications (
id INT PRIMARY KEY AUTO_INCREMENT,
attendance_id INT NOT NULL,
student_id INT NOT NULL,
submitted_by INT NOT NULL,
reason TEXT NOT NULL,
document_path VARCHAR(255),
status ENUM('pendiente', 'aprobado', 'rechazado') DEFAULT 'pendiente',
reviewed_by INT,
review_notes TEXT,
created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
FOREIGN KEY (attendance_id) REFERENCES attendances(id) ON DELETE CASCADE,
FOREIGN KEY (student_id) REFERENCES users(id),
FOREIGN KEY (submitted_by) REFERENCES users(id),
FOREIGN KEY (reviewed_by) REFERENCES users(id)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;

-- =====
-- TABLA: representatives
-- =====
CREATE TABLE representatives (
id INT PRIMARY KEY AUTO_INCREMENT,
representative_id INT NOT NULL,
student_id INT NOT NULL,
relationship VARCHAR(50) NOT NULL,
is_primary BOOLEAN DEFAULT 1,
created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
FOREIGN KEY (representative_id) REFERENCES users(id) ON DELETE CASCADE,
FOREIGN KEY (student_id) REFERENCES users(id) ON DELETE CASCADE,
UNIQUE KEY unique_rep_student (representative_id, student_id)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;

-- =====
-- TABLA: notifications
-- =====
CREATE TABLE notifications (
id INT PRIMARY KEY AUTO_INCREMENT,
user_id INT NOT NULL,
title VARCHAR(255) NOT NULL,
message TEXT NOT NULL,
type VARCHAR(50) DEFAULT 'info',
is_read BOOLEAN DEFAULT 0,
created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
FOREIGN KEY (user_id) REFERENCES users(id) ON DELETE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;

-- =====
-- TABLA: activity_logs
-- =====
CREATE TABLE activity_logs (
id INT PRIMARY KEY AUTO_INCREMENT,

```

```

user_id INT NOT NULL,
action VARCHAR(100) NOT NULL,
table_name VARCHAR(50),
record_id INT,

description TEXT,
ip_address VARCHAR(45),
created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
FOREIGN KEY (user_id) REFERENCES users(id) ON DELETE CASCADE,
INDEX idx_user_date (user_id, created_at),
INDEX idx_action (action)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;

-- =====
-- FIN DE LA ESTRUCTURA
-- =====

```

## 23. ArchImp/database/ecuasistencia2026\_dbX.sql

```

-- phpMyAdmin SQL Dump
-- version 5.2.1
-- https://www.phpmyadmin.net/
--
-- Servidor: 127.0.0.1
-- Tiempo de generación: 12-02-2026 a las 06:04:12
-- Versión del servidor: 10.4.32-MariaDB
-- Versión de PHP: 8.2.12

SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
START TRANSACTION;
SET time_zone = "+00:00";

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8mb4 */;

--
-- Base de datos: `ecuasistencia2026_db`
--

--
-- Estructura de tabla para la tabla `activity_logs`
--

CREATE TABLE `activity_logs` (
  `id` int(11) NOT NULL,
  `user_id` int(11) NOT NULL,
  `action` varchar(100) NOT NULL,
  `entity_type` varchar(50) DEFAULT NULL,
  `entity_id` int(11) DEFAULT NULL,
  `description` text DEFAULT NULL,
  `ip_address` varchar(45) DEFAULT NULL,
  `user_agent` varchar(255) DEFAULT NULL,
  `created_at` timestamp NOT NULL DEFAULT current_timestamp()
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

--
--

```

```

-- Estructura de tabla para la tabla `attendances`
--

CREATE TABLE `attendances` (
  `id` int(11) NOT NULL,
  `student_id` int(11) NOT NULL,
  `course_id` int(11) NOT NULL,
  `subject_id` int(11) NOT NULL,
  `teacher_id` int(11) NOT NULL,
  `school_year_id` int(11) NOT NULL,
  `shift_id` int(11) NOT NULL,
  `date` date NOT NULL,
  `hour_period` varchar(20) DEFAULT NULL,
  `status` enum('presente','ausente','tardanza','justificado') NOT NULL,
  `observation` text DEFAULT NULL,
  `created_at` timestamp NOT NULL DEFAULT current_timestamp(),
  `updated_at` timestamp NOT NULL DEFAULT current_timestamp() ON UPDATE current_timestamp()
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

--
-- Volcado de datos para la tabla `attendances`
--

INSERT INTO `attendances` (`id`, `student_id`, `course_id`, `subject_id`, `teacher_id`, `school_year_id`,
`shift_id`, `date`, `hour_period`, `status`, `observation`, `created_at`, `updated_at`) VALUES
(1, 8, 1, 1, 2, 1, 1, '2026-02-07', '1ra hora', 'presente', '', '2026-02-11 23:56:54', '2026-02-11
23:56:54'),
(2, 9, 1, 1, 2, 1, 1, '2026-02-07', '1ra hora', 'presente', '', '2026-02-11 23:56:54', '2026-02-11
23:56:54'),
(3, 10, 1, 1, 2, 1, 1, '2026-02-07', '1ra hora', 'ausente', 'Sin justificación', '2026-02-11 23:56:54',
'2026-02-11 23:56:54'),
(4, 11, 1, 1, 2, 1, 1, '2026-02-07', '1ra hora', 'presente', '', '2026-02-11 23:56:54', '2026-02-11
23:56:54'),
(5, 8, 1, 1, 2, 1, 1, '2026-02-08', '1ra hora', 'tardanza', 'Llegó 10 min tarde', '2026-02-11 23:56:54',
'2026-02-11 23:56:54'),
(6, 9, 1, 1, 2, 1, 1, '2026-02-08', '1ra hora', 'presente', '', '2026-02-11 23:56:54', '2026-02-11
23:56:54'),
(7, 10, 1, 1, 2, 1, 1, '2026-02-08', '1ra hora', 'justificado', 'Cita médica', '2026-02-11 23:56:54',
'2026-02-11 23:56:54'),
(8, 11, 1, 1, 2, 1, 1, '2026-02-08', '1ra hora', 'presente', '', '2026-02-11 23:56:54', '2026-02-11
23:56:54'),
(9, 8, 1, 1, 2, 1, 1, '2026-02-09', '1ra hora', 'presente', '', '2026-02-11 23:56:54', '2026-02-11
23:56:54'),
(10, 9, 1, 1, 2, 1, 1, '2026-02-09', '1ra hora', 'presente', '', '2026-02-11 23:56:54', '2026-02-11
23:56:54'),
(11, 10, 1, 1, 2, 1, 1, '2026-02-09', '1ra hora', 'presente', '', '2026-02-11 23:56:54', '2026-02-11
23:56:54'),
(12, 11, 1, 1, 2, 1, 1, '2026-02-09', '1ra hora', 'ausente', 'Enfermo', '2026-02-11 23:56:54', '2026-02-11
23:56:54'),
(13, 8, 1, 1, 2, 1, 1, '2026-02-10', '1ra hora', 'presente', '', '2026-02-11 23:56:54', '2026-02-11
23:56:54'),
(14, 9, 1, 1, 2, 1, 1, '2026-02-10', '1ra hora', 'presente', '', '2026-02-11 23:56:54', '2026-02-11
23:56:54'),
(15, 10, 1, 1, 2, 1, 1, '2026-02-10', '1ra hora', 'presente', '', '2026-02-11 23:56:54', '2026-02-11
23:56:54'),
(16, 11, 1, 1, 2, 1, 1, '2026-02-10', '1ra hora', 'tardanza', 'Tráfico', '2026-02-11 23:56:54', '2026-02-11
23:56:54'),
(17, 8, 1, 1, 2, 1, 1, '2026-02-11', '1ra hora', 'presente', '', '2026-02-11 23:56:54', '2026-02-11
23:56:54'),
(18, 9, 1, 1, 2, 1, 1, '2026-02-11', '1ra hora', 'presente', '', '2026-02-11 23:56:54', '2026-02-11
23:56:54'),
(19, 10, 1, 1, 2, 1, 1, '2026-02-11', '1ra hora', 'presente', '', '2026-02-11 23:56:54', '2026-02-11
23:56:54'),
(20, 11, 1, 1, 2, 1, 1, '2026-02-11', '1ra hora', 'presente', '', '2026-02-11 23:56:54', '2026-02-11
23:56:54'),

```

```

(21, 8, 1, 2, 3, 1, 1, '2026-02-09', '2da hora', 'presente', '', '2026-02-11 23:56:54', '2026-02-11
23:56:54'),
(22, 9, 1, 2, 3, 1, 1, '2026-02-09', '2da hora', 'presente', 'Participación activa', '2026-02-11 23:56:54',
'2026-02-11 23:56:54'),
(23, 10, 1, 2, 3, 1, 1, '2026-02-09', '2da hora', 'ausente', '', '2026-02-11 23:56:54', '2026-02-11
23:56:54'),
(24, 11, 1, 2, 3, 1, 1, '2026-02-09', '2da hora', 'presente', '', '2026-02-11 23:56:54', '2026-02-11
23:56:54');

-- -----

--
-- Estructura de tabla para la tabla `courses`
--

CREATE TABLE `courses` (
  `id` int(11) NOT NULL,
  `institution_id` int(11) NOT NULL,
  `school_year_id` int(11) NOT NULL,
  `name` varchar(100) NOT NULL,
  `grade_level` varchar(50) DEFAULT NULL,
  `parallel` varchar(10) DEFAULT NULL,
  `shift_id` int(11) NOT NULL,
  `created_at` timestamp NOT NULL DEFAULT current_timestamp(),
  `updated_at` timestamp NOT NULL DEFAULT current_timestamp() ON UPDATE current_timestamp()
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

--
-- Volcado de datos para la tabla `courses`
--

INSERT INTO `courses` (`id`, `institution_id`, `school_year_id`, `name`, `grade_level`, `parallel`,
`shift_id`, `created_at`, `updated_at`) VALUES
(1, 1, 1, 'Octavo A', 'Octavo', 'A', 1, '2026-02-11 23:56:54', '2026-02-11 23:56:54'),
(2, 1, 1, 'Séptimo A', 'Séptimo', 'A', 1, '2026-02-11 23:56:54', '2026-02-11 23:56:54'),
(3, 1, 1, 'Octavo B', 'Octavo', 'B', 1, '2026-02-11 23:56:54', '2026-02-11 23:56:54'),
(4, 1, 1, 'Noveno A', 'Noveno', 'A', 2, '2026-02-11 23:56:54', '2026-02-11 23:56:54'),
(5, 1, 1, 'Décimo A', 'Décimo', 'A', 2, '2026-02-11 23:56:54', '2026-02-11 23:56:54');

-- -----

--
-- Estructura de tabla para la tabla `course_students`
--

CREATE TABLE `course_students` (
  `id` int(11) NOT NULL,
  `student_id` int(11) NOT NULL,
  `course_id` int(11) NOT NULL,
  `school_year_id` int(11) NOT NULL,
  `enrollment_date` date DEFAULT NULL,
  `created_at` timestamp NOT NULL DEFAULT current_timestamp()
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

--
-- Volcado de datos para la tabla `course_students`
--

INSERT INTO `course_students` (`id`, `student_id`, `course_id`, `school_year_id`, `enrollment_date`,
`created_at`) VALUES
(1, 8, 1, 1, '2025-09-01', '2026-02-11 23:56:54'),
(2, 9, 1, 1, '2025-09-01', '2026-02-11 23:56:54'),
(3, 10, 1, 1, '2025-09-01', '2026-02-11 23:56:54'),
(4, 11, 1, 1, '2025-09-01', '2026-02-11 23:56:54'),
(5, 12, 2, 1, '2025-09-01', '2026-02-11 23:56:54'),

```

```

(6, 13, 2, 1, '2025-09-01', '2026-02-11 23:56:54'),
(7, 14, 2, 1, '2025-09-01', '2026-02-11 23:56:54'),
(8, 15, 3, 1, '2025-09-01', '2026-02-11 23:56:54'),
(9, 16, 3, 1, '2025-09-01', '2026-02-11 23:56:54'),
(10, 17, 3, 1, '2025-09-01', '2026-02-11 23:56:54'),
(11, 18, 4, 1, '2025-09-01', '2026-02-11 23:56:54'),
(12, 19, 4, 1, '2025-09-01', '2026-02-11 23:56:54'),
(13, 20, 4, 1, '2025-09-01', '2026-02-11 23:56:54'),
(14, 21, 5, 1, '2025-09-01', '2026-02-11 23:56:54'),
(15, 22, 5, 1, '2025-09-01', '2026-02-11 23:56:54');

-- -----

--
-- Estructura de tabla para la tabla `institutions`
--

CREATE TABLE `institutions` (
  `id` int(11) NOT NULL,
  `name` varchar(255) NOT NULL,
  `code` varchar(50) NOT NULL,
  `address` text DEFAULT NULL,
  `phone` varchar(20) DEFAULT NULL,
  `email` varchar(100) DEFAULT NULL,
  `logo` varchar(255) DEFAULT NULL,
  `is_active` tinyint(1) DEFAULT 1,
  `created_at` timestamp NOT NULL DEFAULT current_timestamp(),
  `updated_at` timestamp NOT NULL DEFAULT current_timestamp() ON UPDATE current_timestamp()
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

--
-- Volcado de datos para la tabla `institutions`
--

INSERT INTO `institutions` (`id`, `name`, `code`, `address`, `phone`, `email`, `logo`, `is_active`,
`created_at`, `updated_at`) VALUES
(1, 'Unidad Educativa Pomasqui', 'UEP001', 'Av. Manuel Cordova Galarza 123, Quito, Ecuador', '02-2351072',
'contacto@uep.edu.ec', NULL, 1, '2026-02-11 23:56:53', '2026-02-11 23:56:53');

-- -----

--
-- Estructura de tabla para la tabla `justifications`
--

CREATE TABLE `justifications` (
  `id` int(11) NOT NULL,
  `attendance_id` int(11) NOT NULL,
  `student_id` int(11) NOT NULL,
  `submitted_by` int(11) NOT NULL,
  `reason` text NOT NULL,
  `document_path` varchar(255) DEFAULT NULL,
  `status` enum('pendiente','aprobado','rechazado') DEFAULT 'pendiente',
  `reviewed_by` int(11) DEFAULT NULL,
  `review_notes` text DEFAULT NULL,
  `created_at` timestamp NOT NULL DEFAULT current_timestamp(),
  `updated_at` timestamp NOT NULL DEFAULT current_timestamp() ON UPDATE current_timestamp()
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

-- -----

--
-- Estructura de tabla para la tabla `notifications`
--

```

```

CREATE TABLE `notifications` (
  `id` int(11) NOT NULL,
  `user_id` int(11) NOT NULL,
  `title` varchar(255) NOT NULL,
  `message` text NOT NULL,
  `type` enum('info','warning','success','danger') DEFAULT 'info',
  `is_read` tinyint(1) DEFAULT 0,
  `created_at` timestamp NOT NULL DEFAULT current_timestamp()
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

-- -----

--
-- Estructura de tabla para la tabla `permissions`
--

CREATE TABLE `permissions` (
  `id` int(11) NOT NULL,
  `user_id` int(11) NOT NULL,
  `permission_type` varchar(50) NOT NULL,
  `entity_id` int(11) DEFAULT NULL,
  `created_at` timestamp NOT NULL DEFAULT current_timestamp()
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

-- -----

--
-- Estructura de tabla para la tabla `representatives`
--

CREATE TABLE `representatives` (
  `id` int(11) NOT NULL,
  `representative_id` int(11) NOT NULL,
  `student_id` int(11) NOT NULL,
  `relationship` varchar(50) DEFAULT NULL,
  `is_primary` tinyint(1) DEFAULT 0,
  `created_at` timestamp NOT NULL DEFAULT current_timestamp()
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

--
-- Volcado de datos para la tabla `representatives`
--

INSERT INTO `representatives` (`id`, `representative_id`, `student_id`, `relationship`, `is_primary`,
`created_at`) VALUES
(1, 23, 12, 'Madre', 1, '2026-02-11 23:56:54'),
(2, 23, 13, 'Madre', 1, '2026-02-11 23:56:54'),
(3, 24, 14, 'Padre', 1, '2026-02-11 23:56:54'),
(4, 25, 15, 'Madre', 1, '2026-02-11 23:56:54'),
(5, 25, 16, 'Madre', 1, '2026-02-11 23:56:54'),
(6, 26, 17, 'Padre', 1, '2026-02-11 23:56:54'),
(7, 26, 18, 'Padre', 1, '2026-02-11 23:56:54'),
(8, 27, 19, 'Madre', 1, '2026-02-11 23:56:54'),
(9, 27, 20, 'Madre', 1, '2026-02-11 23:56:54'),
(10, 27, 21, 'Madre', 1, '2026-02-11 23:56:54');

-- -----

--
-- Estructura de tabla para la tabla `roles`
--

CREATE TABLE `roles` (
  `id` int(11) NOT NULL,
  `name` varchar(50) NOT NULL,

```



```

`description` text DEFAULT NULL,
`created_at` timestamp NOT NULL DEFAULT current_timestamp(),
`updated_at` timestamp NOT NULL DEFAULT current_timestamp() ON UPDATE current_timestamp()
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

--
-- Volcado de datos para la tabla `roles`
--

INSERT INTO `roles` (`id`, `name`, `description`, `created_at`, `updated_at`) VALUES
(1, 'docente', 'Profesor de la institución', '2026-02-11 23:56:54', '2026-02-11 23:56:54'),
(2, 'estudiante', 'Estudiante activo', '2026-02-11 23:56:54', '2026-02-11 23:56:54'),
(3, 'inspector', 'Inspector de disciplina', '2026-02-11 23:56:54', '2026-02-11 23:56:54'),
(4, 'autoridad', 'Autoridad institucional', '2026-02-11 23:56:54', '2026-02-11 23:56:54'),
(5, 'representante', 'Representante legal de estudiante', '2026-02-11 23:56:54', '2026-02-11 23:56:54');

-- -----

--
-- Estructura de tabla para la tabla `school_years`
--

CREATE TABLE `school_years` (
`id` int(11) NOT NULL,
`institution_id` int(11) NOT NULL,
`name` varchar(100) NOT NULL,
`start_date` date NOT NULL,
`end_date` date NOT NULL,
`is_active` tinyint(1) DEFAULT 0,
`created_at` timestamp NOT NULL DEFAULT current_timestamp(),
`updated_at` timestamp NOT NULL DEFAULT current_timestamp() ON UPDATE current_timestamp()
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

--
-- Volcado de datos para la tabla `school_years`
--

INSERT INTO `school_years` (`id`, `institution_id`, `name`, `start_date`, `end_date`, `is_active`,
`created_at`, `updated_at`) VALUES
(1, 1, '2025-2026', '2025-09-01', '2026-07-30', 1, '2026-02-11 23:56:54', '2026-02-11 23:56:54');

-- -----

--
-- Estructura de tabla para la tabla `shifts`
--

CREATE TABLE `shifts` (
`id` int(11) NOT NULL,
`name` varchar(50) NOT NULL,
`created_at` timestamp NOT NULL DEFAULT current_timestamp()
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

--
-- Volcado de datos para la tabla `shifts`
--

INSERT INTO `shifts` (`id`, `name`, `created_at`) VALUES
(1, 'matutina', '2026-02-11 23:56:54'),
(2, 'vespertina', '2026-02-11 23:56:54'),
(3, 'nocturna', '2026-02-11 23:56:54');

-- -----

--

```

```
-- Estructura de tabla para la tabla `subjects`
--

CREATE TABLE `subjects` (
  `id` int(11) NOT NULL,
  `institution_id` int(11) NOT NULL,
  `name` varchar(100) NOT NULL,
  `code` varchar(20) DEFAULT NULL,
  `created_at` timestamp NOT NULL DEFAULT current_timestamp(),
  `updated_at` timestamp NOT NULL DEFAULT current_timestamp() ON UPDATE current_timestamp()
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

--

-- Volcado de datos para la tabla `subjects`
--

INSERT INTO `subjects` (`id`, `institution_id`, `name`, `code`, `created_at`, `updated_at`) VALUES
(1, 1, 'Matemáticas', 'MAT', '2026-02-11 23:56:54', '2026-02-11 23:56:54'),
(2, 1, 'Lengua y Literatura', 'LEN', '2026-02-11 23:56:54', '2026-02-11 23:56:54'),
(3, 1, 'Ciencias Naturales', 'CCN', '2026-02-11 23:56:54', '2026-02-11 23:56:54'),
(4, 1, 'Estudios Sociales', 'ESS', '2026-02-11 23:56:54', '2026-02-11 23:56:54'),
(5, 1, 'Inglés', 'ING', '2026-02-11 23:56:54', '2026-02-11 23:56:54'),
(6, 1, 'Educación Física', 'EDF', '2026-02-11 23:56:54', '2026-02-11 23:56:54'),
(7, 1, 'Arte y Cultura', 'ART', '2026-02-11 23:56:54', '2026-02-11 23:56:54'),
(8, 1, 'Informática', 'INF', '2026-02-11 23:56:54', '2026-02-11 23:56:54');

-- -----

--

-- Estructura de tabla para la tabla `teacher_assignments`
--

CREATE TABLE `teacher_assignments` (
  `id` int(11) NOT NULL,
  `teacher_id` int(11) NOT NULL,
  `course_id` int(11) NOT NULL,
  `subject_id` int(11) NOT NULL,
  `school_year_id` int(11) NOT NULL,
  `is_tutor` tinyint(1) DEFAULT 0,
  `created_at` timestamp NOT NULL DEFAULT current_timestamp()
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

--

-- Volcado de datos para la tabla `teacher_assignments`
--

INSERT INTO `teacher_assignments` (`id`, `teacher_id`, `course_id`, `subject_id`, `school_year_id`,
`is_tutor`, `created_at`) VALUES
(1, 2, 1, 1, 1, 1, '2026-02-11 23:56:54'),
(2, 2, 2, 1, 1, 0, '2026-02-11 23:56:54'),
(3, 2, 3, 1, 1, 0, '2026-02-11 23:56:54'),
(4, 3, 1, 2, 1, 0, '2026-02-11 23:56:54'),
(5, 3, 2, 2, 1, 1, '2026-02-11 23:56:54'),
(6, 3, 4, 2, 1, 0, '2026-02-11 23:56:54'),
(7, 4, 1, 3, 1, 0, '2026-02-11 23:56:54'),
(8, 4, 3, 3, 1, 1, '2026-02-11 23:56:54'),
(9, 4, 5, 3, 1, 0, '2026-02-11 23:56:54'),
(10, 5, 2, 4, 1, 0, '2026-02-11 23:56:54'),
(11, 5, 3, 4, 1, 0, '2026-02-11 23:56:54'),
(12, 5, 4, 4, 1, 1, '2026-02-11 23:56:54'),
(13, 6, 1, 5, 1, 0, '2026-02-11 23:56:54'),
(14, 6, 2, 5, 1, 0, '2026-02-11 23:56:54'),
(15, 6, 4, 5, 1, 0, '2026-02-11 23:56:54'),
(16, 6, 5, 5, 1, 1, '2026-02-11 23:56:54'),
(17, 7, 1, 6, 1, 0, '2026-02-11 23:56:54'),
```

```

(18, 7, 3, 6, 1, 0, '2026-02-11 23:56:54'),
(19, 7, 5, 6, 1, 0, '2026-02-11 23:56:54');

-- -----

--
-- Estructura de tabla para la tabla `users`
--

CREATE TABLE `users` (
  `id` int(11) NOT NULL,
  `institution_id` int(11) NOT NULL,
  `username` varchar(50) NOT NULL,
  `email` varchar(100) NOT NULL,
  `password` varchar(255) NOT NULL,
  `first_name` varchar(100) NOT NULL,
  `last_name` varchar(100) NOT NULL,
  `dni` varchar(20) DEFAULT NULL,
  `phone` varchar(20) DEFAULT NULL,
  `reset_token` varchar(255) DEFAULT NULL,
  `reset_expires` datetime DEFAULT NULL,
  `is_active` tinyint(1) DEFAULT 1,
  `created_at` timestamp NOT NULL DEFAULT current_timestamp(),
  `updated_at` timestamp NOT NULL DEFAULT current_timestamp() ON UPDATE current_timestamp()
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

--
-- Volcado de datos para la tabla `users`
--

INSERT INTO `users` (`id`, `institution_id`, `username`, `email`, `password`, `first_name`, `last_name`,
`dni`, `phone`, `reset_token`, `reset_expires`, `is_active`, `created_at`, `updated_at`) VALUES
(1, 1, 'admin', 'admin@uep.edu.ec', '$2y$10$92IXUNpkj00rOQ5byMi.Ye4oKoEa3Ro9llC/.og/at2.uheWG/igi',
'Administrador', 'Sistema', '1700000000', '0999999999', NULL, NULL, 1, '2026-02-11 23:56:54', '2026-02-11
23:56:54'),
(2, 1, 'prof.garcia', 'garcia@uep.edu.ec', '$2y$10$92IXUNpkj00rOQ5byMi.Ye4oKoEa3Ro9llC/.og/at2.uheWG/igi',
'Roberto', 'García Mendoza', '1751234567', '0991234571', NULL, NULL, 1, '2026-02-11 23:56:54', '2026-02-11
23:56:54'),
(3, 1, 'prof.martinez', 'martinez@uep.edu.ec',
'$2y$10$92IXUNpkj00rOQ5byMi.Ye4oKoEa3Ro9llC/.og/at2.uheWG/igi', 'Ana', 'Martínez Silva', '1751234568',
'0991234572', NULL, NULL, 1, '2026-02-11 23:56:54', '2026-02-11 23:56:54'),
(4, 1, 'prof.rodriguez', 'rodriguez@uep.edu.ec',
'$2y$10$92IXUNpkj00rOQ5byMi.Ye4oKoEa3Ro9llC/.og/at2.uheWG/igi', 'Luis', 'Rodríguez Cano', '1751234569',
'0991234573', NULL, NULL, 1, '2026-02-11 23:56:54', '2026-02-11 23:56:54'),
(5, 1, 'prof.fernandez', 'fernandez@uep.edu.ec',
'$2y$10$92IXUNpkj00rOQ5byMi.Ye4oKoEa3Ro9llC/.og/at2.uheWG/igi', 'Carmen', 'Fernández Ortiz', '1751234570',
'0991234574', NULL, NULL, 1, '2026-02-11 23:56:54', '2026-02-11 23:56:54'),
(6, 1, 'prof.gomez', 'gomez@uep.edu.ec', '$2y$10$92IXUNpkj00rOQ5byMi.Ye4oKoEa3Ro9llC/.og/at2.uheWG/igi',
'Miguel', 'Gómez Torres', '1751234571', '0991234575', NULL, NULL, 1, '2026-02-11 23:56:54', '2026-02-11
23:56:54'),
(7, 1, 'prof.diaz', 'diaz@uep.edu.ec', '$2y$10$92IXUNpkj00rOQ5byMi.Ye4oKoEa3Ro9llC/.og/at2.uheWG/igi',
'Patricia', 'Díaz Ramírez', '1751234572', '0991234576', NULL, NULL, 1, '2026-02-11 23:56:54', '2026-02-11
23:56:54'),
(8, 1, 'juan.perez', 'juan.perez@estudiante.edu.ec',
'$2y$10$92IXUNpkj00rOQ5byMi.Ye4oKoEa3Ro9llC/.og/at2.uheWG/igi', 'Juan', 'Pérez García', '1750123456',
'0991234567', NULL, NULL, 1, '2026-02-11 23:56:54', '2026-02-11 23:56:54'),
(9, 1, 'maria.lopez', 'maria.lopez@estudiante.edu.ec',
'$2y$10$92IXUNpkj00rOQ5byMi.Ye4oKoEa3Ro9llC/.og/at2.uheWG/igi', 'María', 'López Sánchez', '1750234567',
'0991234568', NULL, NULL, 1, '2026-02-11 23:56:54', '2026-02-11 23:56:54'),
(10, 1, 'carlos.mora', 'carlos.mora@estudiante.edu.ec',
'$2y$10$92IXUNpkj00rOQ5byMi.Ye4oKoEa3Ro9llC/.og/at2.uheWG/igi', 'Carlos', 'Mora Ruiz', '1750345678',
'0991234569', NULL, NULL, 1, '2026-02-11 23:56:54', '2026-02-11 23:56:54'),
(11, 1, 'ana.torres', 'ana.torres@estudiante.edu.ec',
'$2y$10$92IXUNpkj00rOQ5byMi.Ye4oKoEa3Ro9llC/.og/at2.uheWG/igi', 'Ana', 'Torres Vega', '1750456789',
'0991234570', NULL, NULL, 1, '2026-02-11 23:56:54', '2026-02-11 23:56:54'),

```

```
(12, 1, 'sofia.castro', 'sofia.castro@estudiante.edu.ec',
'$2y$10$92IXUNpkj00rOQ5byMi.Ye4oKoEa3Ro9llC/.og/at2.uheWG/igi', 'Sofia', 'Castro Morales', '1750567890',
'0991234577', NULL, NULL, 1, '2026-02-11 23:56:54', '2026-02-11 23:56:54'),
(13, 1, 'diego.herrera', 'diego.herrera@estudiante.edu.ec',
'$2y$10$92IXUNpkj00rOQ5byMi.Ye4oKoEa3Ro9llC/.og/at2.uheWG/igi', 'Diego', 'Herrera Ríos', '1750567891',
'0991234578', NULL, NULL, 1, '2026-02-11 23:56:54', '2026-02-11 23:56:54'),
(14, 1, 'valentina.lopez', 'valentina.lopez@estudiante.edu.ec',
'$2y$10$92IXUNpkj00rOQ5byMi.Ye4oKoEa3Ro9llC/.og/at2.uheWG/igi', 'Valentina', 'López Suárez', '1750567892',
'0991234579', NULL, NULL, 1, '2026-02-11 23:56:54', '2026-02-11 23:56:54'),
(15, 1, 'mateo.silva', 'mateo.silva@estudiante.edu.ec',
'$2y$10$92IXUNpkj00rOQ5byMi.Ye4oKoEa3Ro9llC/.og/at2.uheWG/igi', 'Mateo', 'Silva Navarro', '1750567893',
'0991234580', NULL, NULL, 1, '2026-02-11 23:56:54', '2026-02-11 23:56:54'),
(16, 1, 'isabella.rojas', 'isabella.rojas@estudiante.edu.ec',
'$2y$10$92IXUNpkj00rOQ5byMi.Ye4oKoEa3Ro9llC/.og/at2.uheWG/igi', 'Isabella', 'Rojas Pérez', '1750567894',
'0991234581', NULL, NULL, 1, '2026-02-11 23:56:54', '2026-02-11 23:56:54'),
(17, 1, 'sebastian.cruz', 'sebastian.cruz@estudiante.edu.ec',
'$2y$10$92IXUNpkj00rOQ5byMi.Ye4oKoEa3Ro9llC/.og/at2.uheWG/igi', 'Sebastián', 'Cruz Méndez', '1750567895',
'0991234582', NULL, NULL, 1, '2026-02-11 23:56:54', '2026-02-11 23:56:54'),
(18, 1, 'camila.ramos', 'camila.ramos@estudiante.edu.ec',
'$2y$10$92IXUNpkj00rOQ5byMi.Ye4oKoEa3Ro9llC/.og/at2.uheWG/igi', 'Camila', 'Ramos Flores', '1750567896',
'0991234583', NULL, NULL, 1, '2026-02-11 23:56:54', '2026-02-11 23:56:54'),
(19, 1, 'nicolas.vargas', 'nicolas.vargas@estudiante.edu.ec',
'$2y$10$92IXUNpkj00rOQ5byMi.Ye4oKoEa3Ro9llC/.og/at2.uheWG/igi', 'Nicolás', 'Vargas Castro', '1750567897',
'0991234584', NULL, NULL, 1, '2026-02-11 23:56:54', '2026-02-11 23:56:54'),
(20, 1, 'martina.gil', 'martina.gil@estudiante.edu.ec',
'$2y$10$92IXUNpkj00rOQ5byMi.Ye4oKoEa3Ro9llC/.og/at2.uheWG/igi', 'Martina', 'Gil Núñez', '1750567898',
'0991234585', NULL, NULL, 1, '2026-02-11 23:56:54', '2026-02-11 23:56:54'),
(21, 1, 'lucas.medina', 'lucas.medina@estudiante.edu.ec',
'$2y$10$92IXUNpkj00rOQ5byMi.Ye4oKoEa3Ro9llC/.og/at2.uheWG/igi', 'Lucas', 'Medina Ortega', '1750567899',
'0991234586', NULL, NULL, 1, '2026-02-11 23:56:54', '2026-02-11 23:56:54'),
(22, 1, 'emma.santos', 'emma.santos@estudiante.edu.ec',
'$2y$10$92IXUNpkj00rOQ5byMi.Ye4oKoEa3Ro9llC/.og/at2.uheWG/igi', 'Emma', 'Santos Molina', '1750567900',
'0991234587', NULL, NULL, 1, '2026-02-11 23:56:54', '2026-02-11 23:56:54'),
(23, 1, 'rep.castro', 'lcastro@gmail.com', '$2y$10$92IXUNpkj00rOQ5byMi.Ye4oKoEa3Ro9llC/.og/at2.uheWG/igi',
'Laura', 'Castro Vega', '1752345678', '0991234588', NULL, NULL, 1, '2026-02-11 23:56:54', '2026-02-11
23:56:54'),
(24, 1, 'rep.herrera', 'jherrera@gmail.com', '$2y$10$92IXUNpkj00rOQ5byMi.Ye4oKoEa3Ro9llC/.og/at2.uheWG/igi',
'Jorge', 'Herrera Palacios', '1752345679', '0991234589', NULL, NULL, 1, '2026-02-11 23:56:54', '2026-02-11
23:56:54'),
(25, 1, 'rep.lopez', 'mlopez@gmail.com', '$2y$10$92IXUNpkj00rOQ5byMi.Ye4oKoEa3Ro9llC/.og/at2.uheWG/igi',
'María', 'López Benítez', '1752345680', '0991234590', NULL, NULL, 1, '2026-02-11 23:56:54', '2026-02-11
23:56:54'),
(26, 1, 'rep.silva', 'psilva@gmail.com', '$2y$10$92IXUNpkj00rOQ5byMi.Ye4oKoEa3Ro9llC/.og/at2.uheWG/igi',
'Pedro', 'Silva Moreno', '1752345681', '0991234591', NULL, NULL, 1, '2026-02-11 23:56:54', '2026-02-11
23:56:54'),
(27, 1, 'rep.rojas', 'crojas@gmail.com', '$2y$10$92IXUNpkj00rOQ5byMi.Ye4oKoEa3Ro9llC/.og/at2.uheWG/igi',
'Carolina', 'Rojas Delgado', '1752345682', '0991234592', NULL, NULL, 1, '2026-02-11 23:56:54', '2026-02-11
23:56:54'),
(28, 1, 'inspector', 'inspector@uep.edu.ec', '$2y$10$92IXUNpkj00rOQ5byMi.Ye4oKoEa3Ro9llC/.og/at2.uheWG/igi',
'José', 'Ramírez Campos', '1753456789', '0991234593', NULL, NULL, 1, '2026-02-11 23:56:54', '2026-02-11
23:56:54');
```

```
-- -----
```

```
--
-- Estructura de tabla para la tabla `user_roles`
--
```

```
CREATE TABLE `user_roles` (
  `id` int(11) NOT NULL,
  `user_id` int(11) NOT NULL,
  `role_id` int(11) NOT NULL,
  `assigned_at` timestamp NOT NULL DEFAULT current_timestamp()
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
```

```
--
-- Volcado de datos para la tabla `user_roles`
--

INSERT INTO `user_roles` (`id`, `user_id`, `role_id`, `assigned_at`) VALUES
(1, 1, 4, '2026-02-11 23:56:54'),
(2, 2, 1, '2026-02-11 23:56:54'),
(3, 3, 1, '2026-02-11 23:56:54'),
(4, 4, 1, '2026-02-11 23:56:54'),
(5, 5, 1, '2026-02-11 23:56:54'),
(6, 6, 1, '2026-02-11 23:56:54'),
(7, 7, 1, '2026-02-11 23:56:54'),
(8, 8, 2, '2026-02-11 23:56:54'),
(9, 9, 2, '2026-02-11 23:56:54'),
(10, 10, 2, '2026-02-11 23:56:54'),
(11, 11, 2, '2026-02-11 23:56:54'),
(12, 12, 2, '2026-02-11 23:56:54'),
(13, 13, 2, '2026-02-11 23:56:54'),
(14, 14, 2, '2026-02-11 23:56:54'),
(15, 15, 2, '2026-02-11 23:56:54'),
(16, 16, 2, '2026-02-11 23:56:54'),
(17, 17, 2, '2026-02-11 23:56:54'),
(18, 18, 2, '2026-02-11 23:56:54'),
(19, 19, 2, '2026-02-11 23:56:54'),
(20, 20, 2, '2026-02-11 23:56:54'),
(21, 21, 2, '2026-02-11 23:56:54'),
(22, 22, 2, '2026-02-11 23:56:54'),
(23, 23, 5, '2026-02-11 23:56:54'),
(24, 24, 5, '2026-02-11 23:56:54'),
(25, 25, 5, '2026-02-11 23:56:54'),
(26, 26, 5, '2026-02-11 23:56:54'),
(27, 27, 5, '2026-02-11 23:56:54'),
(28, 28, 3, '2026-02-11 23:56:54'),
(29, 1, 3, '2026-02-12 05:00:01'),
(30, 1, 1, '2026-02-12 05:00:28');

--
-- Índices para tablas volcadas
--

--
-- Indices de la tabla `activity_logs`
--
ALTER TABLE `activity_logs`
ADD PRIMARY KEY (`id`),
ADD KEY `idx_user` (`user_id`),
ADD KEY `idx_date` (`created_at`);

--
-- Indices de la tabla `attendances`
--
ALTER TABLE `attendances`
ADD PRIMARY KEY (`id`),
ADD KEY `course_id` (`course_id`),
ADD KEY `subject_id` (`subject_id`),
ADD KEY `teacher_id` (`teacher_id`),
ADD KEY `school_year_id` (`school_year_id`),
ADD KEY `shift_id` (`shift_id`),
ADD KEY `idx_date` (`date`),
ADD KEY `idx_student` (`student_id`);

--
-- Indices de la tabla `courses`
--
ALTER TABLE `courses`
```

```

ADD PRIMARY KEY (`id`),
ADD KEY `institution_id` (`institution_id`),
ADD KEY `school_year_id` (`school_year_id`),
ADD KEY `shift_id` (`shift_id`);

--
-- Indices de la tabla `course_students`
--
ALTER TABLE `course_students`
ADD PRIMARY KEY (`id`),
ADD UNIQUE KEY `unique_enrollment` (`student_id`,`course_id`,`school_year_id`),
ADD KEY `course_id` (`course_id`),
ADD KEY `school_year_id` (`school_year_id`);

--
-- Indices de la tabla `institutions`
--
ALTER TABLE `institutions`
ADD PRIMARY KEY (`id`),
ADD UNIQUE KEY `code` (`code`);

--
-- Indices de la tabla `justifications`
--
ALTER TABLE `justifications`
ADD PRIMARY KEY (`id`),
ADD KEY `attendance_id` (`attendance_id`),
ADD KEY `submitted_by` (`submitted_by`),
ADD KEY `reviewed_by` (`reviewed_by`),
ADD KEY `idx_status` (`status`),
ADD KEY `idx_student` (`student_id`);

--
-- Indices de la tabla `notifications`
--
ALTER TABLE `notifications`
ADD PRIMARY KEY (`id`),
ADD KEY `idx_user_read` (`user_id`,`is_read`);

--
-- Indices de la tabla `permissions`
--
ALTER TABLE `permissions`
ADD PRIMARY KEY (`id`),
ADD KEY `user_id` (`user_id`);

--
-- Indices de la tabla `representatives`
--
ALTER TABLE `representatives`
ADD PRIMARY KEY (`id`),
ADD UNIQUE KEY `unique_rep_student` (`representative_id`,`student_id`),
ADD KEY `student_id` (`student_id`);

--
-- Indices de la tabla `roles`
--
ALTER TABLE `roles`
ADD PRIMARY KEY (`id`),
ADD UNIQUE KEY `name` (`name`);

--
-- Indices de la tabla `school_years`
--
ALTER TABLE `school_years`
ADD PRIMARY KEY (`id`),

```

```

ADD KEY `institution_id` (`institution_id`);

--
-- Indices de la tabla `shifts`
--
ALTER TABLE `shifts`
ADD PRIMARY KEY (`id`);

--
-- Indices de la tabla `subjects`
--
ALTER TABLE `subjects`
ADD PRIMARY KEY (`id`),
ADD KEY `institution_id` (`institution_id`);

--
-- Indices de la tabla `teacher_assignments`
--
ALTER TABLE `teacher_assignments`
ADD PRIMARY KEY (`id`),
ADD KEY `teacher_id` (`teacher_id`),
ADD KEY `course_id` (`course_id`),
ADD KEY `subject_id` (`subject_id`),
ADD KEY `school_year_id` (`school_year_id`);

--
-- Indices de la tabla `users`
--
ALTER TABLE `users`
ADD PRIMARY KEY (`id`),
ADD UNIQUE KEY `username` (`username`),
ADD UNIQUE KEY `email` (`email`),
ADD UNIQUE KEY `dni` (`dni`),
ADD KEY `institution_id` (`institution_id`);

--
-- Indices de la tabla `user_roles`
--
ALTER TABLE `user_roles`
ADD PRIMARY KEY (`id`),
ADD UNIQUE KEY `unique_user_role` (`user_id`,`role_id`),
ADD KEY `role_id` (`role_id`);

--
-- AUTO_INCREMENT de las tablas volcadas
--

--
-- AUTO_INCREMENT de la tabla `activity_logs`
--
ALTER TABLE `activity_logs`
MODIFY `id` int(11) NOT NULL AUTO_INCREMENT;

--
-- AUTO_INCREMENT de la tabla `attendances`
--
ALTER TABLE `attendances`
MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=25;

--
-- AUTO_INCREMENT de la tabla `courses`
--
ALTER TABLE `courses`
MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=6;

```

```

--
-- AUTO_INCREMENT de la tabla `course_students`
--
ALTER TABLE `course_students`
MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=16;

--
-- AUTO_INCREMENT de la tabla `institutions`
--
ALTER TABLE `institutions`
MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=2;

--
-- AUTO_INCREMENT de la tabla `justifications`
--
ALTER TABLE `justifications`
MODIFY `id` int(11) NOT NULL AUTO_INCREMENT;

--
-- AUTO_INCREMENT de la tabla `notifications`
--
ALTER TABLE `notifications`
MODIFY `id` int(11) NOT NULL AUTO_INCREMENT;

--
-- AUTO_INCREMENT de la tabla `permissions`
--
ALTER TABLE `permissions`
MODIFY `id` int(11) NOT NULL AUTO_INCREMENT;

--
-- AUTO_INCREMENT de la tabla `representatives`
--
ALTER TABLE `representatives`
MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=11;

--
-- AUTO_INCREMENT de la tabla `roles`
--
ALTER TABLE `roles`
MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=6;

--
-- AUTO_INCREMENT de la tabla `school_years`
--
ALTER TABLE `school_years`
MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=2;

--
-- AUTO_INCREMENT de la tabla `shifts`
--
ALTER TABLE `shifts`
MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=4;

--
-- AUTO_INCREMENT de la tabla `subjects`
--
ALTER TABLE `subjects`
MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=9;

--
-- AUTO_INCREMENT de la tabla `teacher_assignments`
--
ALTER TABLE `teacher_assignments`
MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=20;

```



```

--
-- AUTO_INCREMENT de la tabla `users`
--
ALTER TABLE `users`
MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=29;

--
-- AUTO_INCREMENT de la tabla `user_roles`
--
ALTER TABLE `user_roles`
MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=31;

--
-- Restricciones para tablas volcadas
--

--
-- Filtros para la tabla `activity_logs`
--
ALTER TABLE `activity_logs`
ADD CONSTRAINT `activity_logs_ibfk_1` FOREIGN KEY (`user_id`) REFERENCES `users` (`id`);

--
-- Filtros para la tabla `attendances`
--
ALTER TABLE `attendances`
ADD CONSTRAINT `attendances_ibfk_1` FOREIGN KEY (`student_id`) REFERENCES `users` (`id`),
ADD CONSTRAINT `attendances_ibfk_2` FOREIGN KEY (`course_id`) REFERENCES `courses` (`id`),
ADD CONSTRAINT `attendances_ibfk_3` FOREIGN KEY (`subject_id`) REFERENCES `subjects` (`id`),
ADD CONSTRAINT `attendances_ibfk_4` FOREIGN KEY (`teacher_id`) REFERENCES `users` (`id`),
ADD CONSTRAINT `attendances_ibfk_5` FOREIGN KEY (`school_year_id`) REFERENCES `school_years` (`id`),
ADD CONSTRAINT `attendances_ibfk_6` FOREIGN KEY (`shift_id`) REFERENCES `shifts` (`id`);

--
-- Filtros para la tabla `courses`
--
ALTER TABLE `courses`
ADD CONSTRAINT `courses_ibfk_1` FOREIGN KEY (`institution_id`) REFERENCES `institutions` (`id`),
ADD CONSTRAINT `courses_ibfk_2` FOREIGN KEY (`school_year_id`) REFERENCES `school_years` (`id`),
ADD CONSTRAINT `courses_ibfk_3` FOREIGN KEY (`shift_id`) REFERENCES `shifts` (`id`);

--
-- Filtros para la tabla `course_students`
--
ALTER TABLE `course_students`
ADD CONSTRAINT `course_students_ibfk_1` FOREIGN KEY (`student_id`) REFERENCES `users` (`id`),
ADD CONSTRAINT `course_students_ibfk_2` FOREIGN KEY (`course_id`) REFERENCES `courses` (`id`),
ADD CONSTRAINT `course_students_ibfk_3` FOREIGN KEY (`school_year_id`) REFERENCES `school_years` (`id`);

--
-- Filtros para la tabla `justifications`
--
ALTER TABLE `justifications`
ADD CONSTRAINT `justifications_ibfk_1` FOREIGN KEY (`attendance_id`) REFERENCES `attendances` (`id`) ON
DELETE CASCADE,
ADD CONSTRAINT `justifications_ibfk_2` FOREIGN KEY (`student_id`) REFERENCES `users` (`id`),
ADD CONSTRAINT `justifications_ibfk_3` FOREIGN KEY (`submitted_by`) REFERENCES `users` (`id`),
ADD CONSTRAINT `justifications_ibfk_4` FOREIGN KEY (`reviewed_by`) REFERENCES `users` (`id`);

--
-- Filtros para la tabla `notifications`
--
ALTER TABLE `notifications`
ADD CONSTRAINT `notifications_ibfk_1` FOREIGN KEY (`user_id`) REFERENCES `users` (`id`) ON DELETE CASCADE;

```

```

--
-- Filtros para la tabla `permissions`
--
ALTER TABLE `permissions`
ADD CONSTRAINT `permissions_ibfk_1` FOREIGN KEY (`user_id`) REFERENCES `users` (`id`) ON DELETE CASCADE;

--
-- Filtros para la tabla `representatives`
--
ALTER TABLE `representatives`
ADD CONSTRAINT `representatives_ibfk_1` FOREIGN KEY (`representative_id`) REFERENCES `users` (`id`),
ADD CONSTRAINT `representatives_ibfk_2` FOREIGN KEY (`student_id`) REFERENCES `users` (`id`);

--
-- Filtros para la tabla `school_years`
--
ALTER TABLE `school_years`
ADD CONSTRAINT `school_years_ibfk_1` FOREIGN KEY (`institution_id`) REFERENCES `institutions` (`id`);

--
-- Filtros para la tabla `subjects`
--
ALTER TABLE `subjects`
ADD CONSTRAINT `subjects_ibfk_1` FOREIGN KEY (`institution_id`) REFERENCES `institutions` (`id`);

--
-- Filtros para la tabla `teacher_assignments`
--
ALTER TABLE `teacher_assignments`
ADD CONSTRAINT `teacher_assignments_ibfk_1` FOREIGN KEY (`teacher_id`) REFERENCES `users` (`id`),
ADD CONSTRAINT `teacher_assignments_ibfk_2` FOREIGN KEY (`course_id`) REFERENCES `courses` (`id`),
ADD CONSTRAINT `teacher_assignments_ibfk_3` FOREIGN KEY (`subject_id`) REFERENCES `subjects` (`id`),
ADD CONSTRAINT `teacher_assignments_ibfk_4` FOREIGN KEY (`school_year_id`) REFERENCES `school_years` (`id`);

--
-- Filtros para la tabla `users`
--
ALTER TABLE `users`
ADD CONSTRAINT `users_ibfk_1` FOREIGN KEY (`institution_id`) REFERENCES `institutions` (`id`);

--
-- Filtros para la tabla `user_roles`
--
ALTER TABLE `user_roles`
ADD CONSTRAINT `user_roles_ibfk_1` FOREIGN KEY (`user_id`) REFERENCES `users` (`id`) ON DELETE CASCADE,
ADD CONSTRAINT `user_roles_ibfk_2` FOREIGN KEY (`role_id`) REFERENCES `roles` (`id`) ON DELETE CASCADE;
COMMIT;

/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;

```

## 24. ArchImp/helpers/Backup.php

```

<?php
class Backup {
private $dbHost = 'localhost';
private $dbUser = 'root';
private $dbPass = '';
private $dbName = 'ecuasistencia2026_db';

```

```

private $backupDir;

public function __construct() {
    $this->backupDir = BASE_PATH . '/backups/';
    if (!file_exists($this->backupDir)) {
        mkdir($this->backupDir, 0755, true);
    }
}

public function createBackup() {
    $filename = 'backup_' . date('Y-m-d_H-i-s') . '.sql';
    $filepath = $this->backupDir . $filename;

    // Verificar que mysqldump esté disponible
    $mysqldumpPath = 'C:\\xampp\\mysql\\bin\\mysqldump.exe'; // XAMPP Windows
    if (!file_exists($mysqldumpPath)) {
        $mysqldumpPath = 'mysqldump'; // Linux/Mac o PATH configurado
    }

    // Comando mejorado con opciones
    if (empty($this->dbPass)) {
        $command = sprintf(
            "%s --host=%s --user=%s --skip-password %s > %s" 2>&1',
            $mysqldumpPath,
            $this->dbHost,
            $this->dbUser,
            $this->dbName,
            $filepath
        );
    } else {
        $command = sprintf(
            "%s --host=%s --user=%s --password=%s %s > %s" 2>&1',
            $mysqldumpPath,
            $this->dbHost,
            $this->dbUser,
            $this->dbPass,
            $this->dbName,
            $filepath
        );
    }

    exec($command, $output, $result);

    // Verificar que el archivo existe y tiene contenido
    if (file_exists($filepath) && filesize($filepath) > 0) {
        return $filename;
    }

    // Si falló, intentar eliminar archivo vacío
    if (file_exists($filepath)) {
        unlink($filepath);
    }

    return false;
}

public function getBackups() {
    $files = glob($this->backupDir . 'backup_*.sql');
    $backups = [];

    foreach ($files as $file) {
        $size = filesize($file);
        $backups[] = [
            'filename' => basename($file),
            'path' => $file,

```

```

'size' => $this->formatBytes($size),
'date' => date('d/m/Y H:i:s', filetime($file)),
'timestamp' => filetime($file)
];
}

usort($backups, function($a, $b) {
return $b['timestamp'] - $a['timestamp'];
});

return $backups;
}

private function formatBytes($bytes, $precision = 2) {
$units = ['B', 'KB', 'MB', 'GB'];
$bytes = max($bytes, 0);
$pow = floor(($bytes ? log($bytes) : 0) / log(1024));
$pow = min($pow, count($units) - 1);
$bytes /= (1 << (10 * $pow));
return round($bytes, $precision) . ' ' . $units[$pow];
}

public function deleteOldBackups($daysToKeep = 30) {
$files = glob($this->backupDir . 'backup_*.sql');
$deleted = 0;

foreach ($files as $file) {
if (time() - filetime($file) > ($daysToKeep * 24 * 60 * 60)) {
unlink($file);
$deleted++;
}
}

return $deleted;
}

public function downloadBackup($filename) {
$filepath = $this->backupDir . $filename;

if (!file_exists($filepath)) {
return false;
}

header('Content-Type: application/octet-stream');
header('Content-Disposition: attachment; filename="' . $filename . '"');
header('Content-Length: ' . filesize($filepath));
readfile($filepath);
exit;
}

public function deleteBackup($filename) {
// Validar que sea un archivo de backup válido
if (!preg_match('/^backup_[\d\_]+\..sql$/', $filename)) {
return false;
}

$filepath = $this->backupDir . $filename;

if (file_exists($filepath)) {
return unlink($filepath);
}

return false;
}
}

```

## 25. ArchImp/helpers/Logger.php

```
<?php
class Logger {
private $db;

public function __construct() {
    $database = new Database();
    $this->db = $database->connect();
}

public function log($action, $entityType = null, $entityId = null, $description = null) {
    if (!isset($_SESSION['user_id'])) {
        return false;
    }

    $sql = "INSERT INTO activity_logs
    (user_id, action, entity_type, entity_id, description, ip_address, user_agent)
    VALUES (:user_id, :action, :entity_type, :entity_id, :description, :ip_address, :user_agent)";

    $stmt = $this->db->prepare($sql);
    return $stmt->execute([
        ':user_id' => $_SESSION['user_id'],
        ':action' => $action,
        ':entity_type' => $entityType,
        ':entity_id' => $entityId,
        ':description' => $description,
        ':ip_address' => $_SERVER['REMOTE_ADDR'] ?? null,
        ':user_agent' => $_SERVER['HTTP_USER_AGENT'] ?? null
    ]);
}

public function getLogs($limit = 100, $filters = []) {
    $sql = "SELECT l.*, CONCAT(u.last_name, ' ', u.first_name) as user_name
    FROM activity_logs l
    INNER JOIN users u ON l.user_id = u.id
    WHERE 1=1";

    $params = [];

    if (!empty($filters['user_id'])) {
        $sql .= " AND l.user_id = :user_id";
        $params[':user_id'] = $filters['user_id'];
    }

    if (!empty($filters['action'])) {
        $sql .= " AND l.action = :action";
        $params[':action'] = $filters['action'];
    }

    if (!empty($filters['start_date'])) {
        $sql .= " AND l.created_at >= :start_date";
        $params[':start_date'] = $filters['start_date'];
    }

    if (!empty($filters['end_date'])) {
        $sql .= " AND l.created_at <= :end_date";
        $params[':end_date'] = $filters['end_date'] . ' 23:59:59';
    }

    $sql .= " ORDER BY l.created_at DESC LIMIT :limit";
```

```

$stmt = $this->db->prepare($sql);
foreach ($params as $key => $value) {
    $stmt->bindValue($key, $value);
}
$stmt->bindValue(':limit', $limit, PDO::PARAM_INT);
$stmt->execute();

return $stmt->fetchAll();
}
}

```

## 26. ArchImp/helpers/Mailer.php

```

<?php
use PHPMailer\PHPMailer\PHPMailer;
use PHPMailer\PHPMailer\Exception;

require BASE_PATH . '/vendor/autoload.php';

class Mailer {
    public static function send($to, $subject, $body) {
        $mail = new PHPMailer(true);

        try {
            $mail->isSMTP();
            $mail->Host = SMTP_HOST;
            $mail->SMTPAuth = true;
            $mail->Username = SMTP_USER;
            $mail->Password = SMTP_PASS;
            $mail->SMTPSecure = PHPMailer::ENCRYPTION_STARTTLS;
            $mail->Port = SMTP_PORT;
            $mail->CharSet = 'UTF-8';

            $mail->setFrom(SMTP_FROM, SMTP_NAME);
            $mail->addAddress($to);

            $mail->isHTML(true);
            $mail->Subject = $subject;
            $mail->Body = $body;

            $mail->send();
            return true;
        } catch (Exception $e) {
            error_log("Error de correo: {$mail->ErrorInfo}");
            return false;
        }
    }
}

```

## 27. ArchImp/helpers/Security.php

```

<?php
class Security {
    public static function generateToken() {
        if (!isset($_SESSION['csrf_token'])) {
            $_SESSION['csrf_token'] = bin2hex(random_bytes(32));
        }
        return $_SESSION['csrf_token'];
    }

    public static function validateToken($token) {
        return isset($_SESSION['csrf_token']) && hash_equals($_SESSION['csrf_token'], $token);
    }
}

```

```

}

public static function sanitize($data) {
return htmlspecialchars(strip_tags(trim($data)), ENT_QUOTES, 'UTF-8');
}

public static function validateEmail($email) {
return filter_var($email, FILTER_VALIDATE_EMAIL);
}

public static function hashPassword($password) {
return password_hash($password, PASSWORD_BCRYPT);
}

public static function verifyPassword($password, $hash) {
return password_verify($password, $hash);
}

public static function requireLogin() {
if (!isset($_SESSION['user_id'])) {
header('Location: ' . BASE_URL . '/public/index.php?action=login');
exit;
}
}

public static function hasRole($roleNames) {
if (!isset($_SESSION['roles'])) return false;
$roleNames = (array) $roleNames;
return count(array_intersect($roleNames, $_SESSION['roles'])) > 0;
}

public static function startSession($rememberMe = false) {
if ($rememberMe) {
// Sesión de 7 días
$lifetime = 7 * 24 * 60 * 60; // 7 días
} else {
// Sesión expira al cerrar navegador
$lifetime = 0;
}

session_set_cookie_params([
'lifetime' => $lifetime,
'path' => '/',
'httponly' => true,
'samesite' => 'Lax'
]);

session_start();

// Regenerar ID por seguridad
if (!isset($_SESSION['initiated'])) {
session_regenerate_id(true);
$_SESSION['initiated'] = true;
}

// Timeout de inactividad (30 minutos)
if (isset($_SESSION['last_activity']) &&
(time() - $_SESSION['last_activity'] > 1800)) {
session_unset();
session_destroy();
return false;
}

$_SESSION['last_activity'] = time();
return true;
}

```

```
}  
}
```

## 28. ArchImp/models/Attendance.php

```
<?php  
class Attendance {  
    private $db;  
  
    public function __construct($database) {  
        $this->db = $database->connect();  
    }  
  
    public function create($data) {  
        // Verificar si ya existe registro  
        $checkSql = "SELECT id FROM attendances  
        WHERE student_id = :student_id  
        AND course_id = :course_id  
        AND subject_id = :subject_id  
        AND date = :date  
        AND hour_period = :hour_period";  
  
        $checkStmt = $this->db->prepare($checkSql);  
        $checkStmt->execute([  
            ':student_id' => $data[':student_id'],  
            ':course_id' => $data[':course_id'],  
            ':subject_id' => $data[':subject_id'],  
            ':date' => $data[':date'],  
            ':hour_period' => $data[':hour_period']  
        ]);  
  
        $existing = $checkStmt->fetch();  
  
        if ($existing) {  
            // Ya existe, actualizar en lugar de insertar  
            $updateSql = "UPDATE attendances  
            SET status = :status,  
            observation = :observation,  
            updated_at = NOW()  
            WHERE id = :id";  
  
            $updateStmt = $this->db->prepare($updateSql);  
            return $updateStmt->execute([  
                ':status' => $data[':status'],  
                ':observation' => $data[':observation'],  
                ':id' => $existing['id']  
            ]);  
        }  
  
        // No existe, insertar nuevo  
        $sql = "INSERT INTO attendances  
        (student_id, course_id, subject_id, teacher_id, school_year_id, shift_id, date, hour_period, status,  
        observation)  
        VALUES (:student_id, :course_id, :subject_id, :teacher_id, :school_year_id, :shift_id, :date, :hour_period,  
        :status, :observation)";  
  
        $stmt = $this->db->prepare($sql);  
        return $stmt->execute($data);  
    }  
  
    public function canEdit($attendanceId) {  
        $sql = "SELECT created_at FROM attendances WHERE id = :id";  
        $stmt = $this->db->prepare($sql);
```



```

$stmt->execute([':id' => $attendanceId]);
$record = $stmt->fetch();

if (!$record) return false;

$hoursElapsed = (time() - strtotime($record['created_at'])) / 3600;
return $hoursElapsed <= EDIT_ATTENDANCE_HOURS;
}

public function update($id, $data) {
    if (!$this->canEdit($id)) return false;

    $sql = "UPDATE attendances SET status = :status, observation = :observation WHERE id = :id";
    $stmt = $this->db->prepare($sql);
    return $stmt->execute([
        ':status' => $data['status'],
        ':observation' => $data['observation'],
        ':id' => $id
    ]);
}

public function getByCourse($courseId, $date) {
    $sql = "SELECT a.*, u.first_name, u.last_name, s.name as subject_name,
    CONCAT(t.first_name, ' ', t.last_name) as teacher_name
    FROM attendances a
    INNER JOIN users u ON a.student_id = u.id
    INNER JOIN subjects s ON a.subject_id = s.id
    INNER JOIN users t ON a.teacher_id = t.id
    WHERE a.course_id = :course_id AND a.date = :date
    ORDER BY u.last_name, u.first_name";

    $stmt = $this->db->prepare($sql);
    $stmt->execute([':course_id' => $courseId, ':date' => $date]);
    return $stmt->fetchAll();
}

public function getByStudent($studentId) {
    $sql = "SELECT a.*, s.name as subject_name, c.name as course_name,
    sh.name as shift_name
    FROM attendances a
    INNER JOIN subjects s ON a.subject_id = s.id
    INNER JOIN courses c ON a.course_id = c.id
    INNER JOIN shifts sh ON a.shift_id = sh.id
    WHERE a.student_id = :student_id
    ORDER BY a.date DESC, a.hour_period
    LIMIT 100";

    $stmt = $this->db->prepare($sql);
    $stmt->execute([':student_id' => $studentId]);
    return $stmt->fetchAll();
}

// Agregar al final de la clase Attendance

public function getReportData($courseId, $startDate, $endDate) {
    $sql = "SELECT a.*,
    CONCAT(u.last_name, ' ', u.first_name) as student_name,
    s.name as subject_name
    FROM attendances a
    INNER JOIN users u ON a.student_id = u.id
    INNER JOIN subjects s ON a.subject_id = s.id
    WHERE a.course_id = :course_id
    AND a.date BETWEEN :start_date AND :end_date
    ORDER BY a.date, u.last_name, u.first_name";

```

```

$stmt = $this->db->prepare($sql);
$stmt->execute([
    ':course_id' => $courseId,
    ':start_date' => $startDate,
    ':end_date' => $endDate
]);
return $stmt->fetchAll();
}

public function getStatsByCourse($courseId, $startDate, $endDate) {
    $sql = "SELECT
COUNT(*) as total,
SUM(CASE WHEN status = 'presente' THEN 1 ELSE 0 END) as presente,
SUM(CASE WHEN status = 'ausente' THEN 1 ELSE 0 END) as ausente,
SUM(CASE WHEN status = 'tardanza' THEN 1 ELSE 0 END) as tardanza,
SUM(CASE WHEN status = 'justificado' THEN 1 ELSE 0 END) as justificado
FROM attendances
WHERE course_id = :course_id
AND date BETWEEN :start_date AND :end_date";

    $stmt = $this->db->prepare($sql);
    $stmt->execute([
        ':course_id' => $courseId,
        ':start_date' => $startDate,
        ':end_date' => $endDate
    ]);
    return $stmt->fetch();
}

public function getFiltered($filters) {
    $sql = "SELECT a.*,
CONCAT(u.last_name, ' ', u.first_name) as student_name,
c.name as course_name,
s.name as subject_name,
CONCAT(t.last_name, ' ', t.first_name) as teacher_name
FROM attendances a
INNER JOIN users u ON a.student_id = u.id
INNER JOIN courses c ON a.course_id = c.id
INNER JOIN subjects s ON a.subject_id = s.id
INNER JOIN users t ON a.teacher_id = t.id
WHERE 1=1";

    $params = [];

    if (!empty($filters['course_id'])) {
        $sql .= " AND a.course_id = :course_id";
        $params[':course_id'] = $filters['course_id'];
    }

    if (!empty($filters['student_id'])) {
        $sql .= " AND a.student_id = :student_id";
        $params[':student_id'] = $filters['student_id'];
    }

    if (!empty($filters['subject_id'])) {
        $sql .= " AND a.subject_id = :subject_id";
        $params[':subject_id'] = $filters['subject_id'];
    }

    if (!empty($filters['status'])) {
        $sql .= " AND a.status = :status";
        $params[':status'] = $filters['status'];
    }

    if (!empty($filters['start_date'])) {

```

```

$sql .= " AND a.date >= :start_date";
$params[':start_date'] = $filters['start_date'];
}

if (!empty($filters['end_date'])) {
    $sql .= " AND a.date <= :end_date";
    $params[':end_date'] = $filters['end_date'];
}

$sql .= " ORDER BY a.date DESC, u.last_name, u.first_name";

$stmt = $this->db->prepare($sql);
$stmt->execute($params);
return $stmt->fetchAll();
}

public function getDayStats($courseId, $date) {
    $sql = "SELECT
COUNT(*) as total,
SUM(CASE WHEN status = 'presente' THEN 1 ELSE 0 END) as presente,
SUM(CASE WHEN status = 'ausente' THEN 1 ELSE 0 END) as ausente,
SUM(CASE WHEN status = 'tardanza' THEN 1 ELSE 0 END) as tardanza
FROM attendances
WHERE course_id = :course_id AND date = :date";

    $stmt = $this->db->prepare($sql);
    $stmt->execute([
        ':course_id' => $courseId,
        ':date' => $date
    ]);

    $result = $stmt->fetch();
    return $result['total'] > 0 ? $result : null;
}
}

```

## 29. ArchImp/models/ClassSchedule.php

```

<?php
class ClassSchedule {
    private $db;

    public function __construct($database) {
        $this->db = $database->connect();
    }

    public function create($data) {
        // Verificar si ya existe una clase en ese horario
        $checkSql = "SELECT s.name as subject_name,
CONCAT(u.last_name, ' ', u.first_name) as teacher_name
FROM class_schedule cs
INNER JOIN subjects s ON cs.subject_id = s.id
INNER JOIN users u ON cs.teacher_id = u.id
WHERE cs.course_id = :course_id
AND cs.day_of_week = :day_of_week
AND cs.period_number = :period_number
AND cs.school_year_id = :school_year_id";

        $checkStmt = $this->db->prepare($checkSql);
        $checkStmt->execute([
            ':course_id' => $data[':course_id'],
            ':day_of_week' => $data[':day_of_week'],
            ':period_number' => $data[':period_number'],

```

```

':school_year_id' => $data[':school_year_id']
]);

$existing = $checkStmt->fetch();

if ($existing) {
    return [
        'success' => false,
        'message' => 'Ya existe ' . $existing['subject_name'] . ' con ' . $existing['teacher_name'] . ' en este
horario'
    ];
}

$sql = "INSERT INTO class_schedule
(course_id, subject_id, teacher_id, school_year_id, day_of_week, period_number)
VALUES (:course_id, :subject_id, :teacher_id, :school_year_id, :day_of_week, :period_number)";

$stmt = $this->db->prepare($sql);
$result = $stmt->execute($data);

return [
    'success' => $result,
    'message' => 'Clase agregada correctamente'
];
}

public function getByCourse($courseId, $schoolYearId) {
    $sql = "SELECT cs.*,
s.name as subject_name,
CONCAT(u.last_name, ' ', u.first_name) as teacher_name
FROM class_schedule cs
INNER JOIN subjects s ON cs.subject_id = s.id
INNER JOIN users u ON cs.teacher_id = u.id
WHERE cs.course_id = :course_id
AND cs.school_year_id = :school_year_id
ORDER BY cs.day_of_week, cs.period_number";

    $stmt = $this->db->prepare($sql);
    $stmt->execute([
        ':course_id' => $courseId,
        ':school_year_id' => $schoolYearId
    ]);
    return $stmt->fetchAll();
}

public function getTeacherScheduleToday($teacherId, $schoolYearId) {
    $dayName = $this->getCurrentDayName();

    $sql = "SELECT cs.*,
c.name as course_name,
c.grade_level,
s.name as subject_name
FROM class_schedule cs
INNER JOIN courses c ON cs.course_id = c.id
INNER JOIN subjects s ON cs.subject_id = s.id
WHERE cs.teacher_id = :teacher_id
AND cs.school_year_id = :school_year_id
AND cs.day_of_week = :day_of_week
ORDER BY cs.period_number";

    $stmt = $this->db->prepare($sql);
    $stmt->execute([
        ':teacher_id' => $teacherId,
        ':school_year_id' => $schoolYearId,
        ':day_of_week' => $dayName
    ]);
}

```

```

});
return $stmt->fetchAll();
}

public function delete($id) {
    $sql = "DELETE FROM class_schedule WHERE id = :id";
    $stmt = $this->db->prepare($sql);
    return $stmt->execute([':id' => $id]);
}

private function getCurrentDayName() {
    $days = [
        1 => 'lunes',
        2 => 'martes',
        3 => 'miercoles',
        4 => 'jueves',
        5 => 'viernes',
        6 => 'sabado'
    ];

    $dayNumber = date('N'); // 1=Lunes, 7=Domingo
    return $days[$dayNumber] ?? 'lunes';
}
}

```

### 30. ArchImp/models/Course.php

```

<?php
class Course {
    private $db;

    public function __construct($database) {
        $this->db = $database->connect();
    }

    public function getAll() {
        $sql = "SELECT c.*, s.name as shift_name, sy.name as year_name
        FROM courses c
        INNER JOIN shifts s ON c.shift_id = s.id
        INNER JOIN school_years sy ON c.school_year_id = sy.id
        WHERE c.institution_id = :institution_id
        ORDER BY c.name";

        $stmt = $this->db->prepare($sql);
        $stmt->execute([':institution_id' => $_SESSION['institution_id']]);
        return $stmt->fetchAll();
    }

    public function create($data) {
        $sql = "INSERT INTO courses (institution_id, school_year_id, name, grade_level, parallel, shift_id)
        VALUES (:institution_id, :school_year_id, :name, :grade_level, :parallel, :shift_id)";

        $stmt = $this->db->prepare($sql);
        return $stmt->execute($data);
    }

    public function getStudents($courseId) {
        $sql = "SELECT u.id, u.first_name, u.last_name, u.dni
        FROM users u
        INNER JOIN course_students cs ON u.id = cs.student_id
        WHERE cs.course_id = :course_id
        ORDER BY u.last_name, u.first_name";
    }
}

```

```

$stmt = $this->db->prepare($sql);
$stmt->execute([':course_id' => $courseId]);
return $stmt->fetchAll();
}

public function enrollStudent($courseId, $studentId, $schoolYearId) {
    // Primero verificar si ya está matriculado en otro curso del mismo año
    $checkSql = "SELECT COUNT(*) as count FROM course_students
    WHERE student_id = :student_id AND school_year_id = :school_year_id";
    $checkStmt = $this->db->prepare($checkSql);
    $checkStmt->execute([
        ':student_id' => $studentId,
        ':school_year_id' => $schoolYearId
    ]);
    $result = $checkStmt->fetch();

    if ($result['count'] > 0) {
        return false; // Ya está matriculado
    }

    $sql = "INSERT INTO course_students (student_id, course_id, school_year_id, enrollment_date)
    VALUES (:student_id, :course_id, :school_year_id, CURDATE())";

    $stmt = $this->db->prepare($sql);
    return $stmt->execute([
        ':student_id' => $studentId,
        ':course_id' => $courseId,
        ':school_year_id' => $schoolYearId
    ]);
}

public function getEnrolledStudents($courseId) {
    $sql = "SELECT u.id, u.first_name, u.last_name, u.dni, cs.enrollment_date
    FROM users u
    INNER JOIN course_students cs ON u.id = cs.student_id
    WHERE cs.course_id = :course_id
    ORDER BY u.last_name, u.first_name";

    $stmt = $this->db->prepare($sql);
    $stmt->execute([':course_id' => $courseId]);
    return $stmt->fetchAll();
}

public function findById($id) {
    $sql = "SELECT c.*, s.name as shift_name
    FROM courses c
    INNER JOIN shifts s ON c.shift_id = s.id
    WHERE c.id = :id";

    $stmt = $this->db->prepare($sql);
    $stmt->execute([':id' => $id]);
    return $stmt->fetch();
}

public function update($data) {
    $sql = "UPDATE courses SET
    name = :name,
    grade_level = :grade_level,
    parallel = :parallel,
    shift_id = :shift_id,
    updated_at = NOW()
    WHERE id = :id";

    $stmt = $this->db->prepare($sql);
    return $stmt->execute($data);
}

```

```

}

public function delete($id) {
    // Eliminar relaciones primero
    $sql = "DELETE FROM course_students WHERE course_id = :id";
    $stmt = $this->db->prepare($sql);
    $stmt->execute([':id' => $id]);

    // Eliminar horarios
    $sql = "DELETE FROM class_schedule WHERE course_id = :id";
    $stmt = $this->db->prepare($sql);
    $stmt->execute([':id' => $id]);

    // Eliminar curso
    $sql = "DELETE FROM courses WHERE id = :id";
    $stmt = $this->db->prepare($sql);
    return $stmt->execute([':id' => $id]);
}

public function unenrollStudent($studentId, $schoolYearId) {
    $sql = "DELETE FROM course_students
    WHERE student_id = :student_id AND school_year_id = :school_year_id";

    $stmt = $this->db->prepare($sql);
    return $stmt->execute([
        ':student_id' => $studentId,
        ':school_year_id' => $schoolYearId
    ]);
}
}

```

### 31. ArchImp/models/Institution.php

```

<?php
class Institution {
    private $db;

    public function __construct($database) {
        $this->db = $database->connect();
    }

    public function getById($id) {
        $sql = "SELECT * FROM institutions WHERE id = :id";
        $stmt = $this->db->prepare($sql);
        $stmt->execute([':id' => $id]);
        return $stmt->fetch();
    }

    public function getAll() {
        $sql = "SELECT * FROM institutions ORDER BY name";
        $stmt = $this->db->query($sql);
        return $stmt->fetchAll();
    }

    public function update($id, $data) {
        $sql = "UPDATE institutions SET
        name = :name,
        address = :address,
        province = :province,
        city = :city,
        phone = :phone,
        email = :email,

```

```

director_name = :director_name,
amie_code = :amie_code,
website = :website,
logo_path = :logo_path,
updated_at = NOW()
WHERE id = :id";

// Obtener logo actual si no se proporciona uno nuevo
$currentInstitution = $this->getById($id);
$logoPath = $data['logo_path'] ?? $currentInstitution['logo_path'];

$stmt = $this->db->prepare($sql);
return $stmt->execute([
    ':id' => $id,
    ':name' => $data['name'],
    ':address' => $data['address'],
    ':province' => $data['province'],
    ':city' => $data['city'],
    ':phone' => $data['phone'],
    ':email' => $data['email'],
    ':director_name' => $data['director_name'],
    ':amie_code' => $data['amie_code'],
    ':website' => $data['website'],
    ':logo_path' => $logoPath
]);
}

public function getShifts($institutionId) {
    $sql = "SELECT s.id, s.name,
    GROUP_CONCAT(s.name ORDER BY s.name) as shift_names
    FROM shifts s

    INNER JOIN institution_shifts ins ON s.id = ins.shift_id
    WHERE ins.institution_id = :institution_id
    GROUP BY ins.institution_id";

    $stmt = $this->db->prepare($sql);
    $stmt->execute([':institution_id' => $institutionId]);
    return $stmt->fetch();
}
}

```

## 32. ArchImp/models/InstitutionShift.php

```

<?php
class InstitutionShift {
    private $db;

    public function __construct($database) {
        $this->db = $database->connect();
    }

    public function assign($data) {
        $sql = "INSERT INTO institution_shifts (institution_id, shift_id)
        VALUES (:institution_id, :shift_id)
        ON DUPLICATE KEY UPDATE institution_id = institution_id";

        $stmt = $this->db->prepare($sql);
        return $stmt->execute($data);
    }

    public function remove($institutionId, $shiftId) {
        $sql = "DELETE FROM institution_shifts

```



```

WHERE institution_id = :institution_id AND shift_id = :shift_id";

$stmt = $this->db->prepare($sql);
return $stmt->execute([
    ':institution_id' => $institutionId,
    ':shift_id' => $shiftId
]);
}

public function getByInstitution($institutionId) {
    $sql = "SELECT s.id, s.name
FROM shifts s
INNER JOIN institution_shifts ins ON s.id = ins.shift_id
WHERE ins.institution_id = :institution_id
ORDER BY s.name";

    $stmt = $this->db->prepare($sql);
    $stmt->execute([':institution_id' => $institutionId]);
    return $stmt->fetchAll();
}

public function getInstitutionShiftIds($institutionId) {
    $sql = "SELECT shift_id FROM institution_shifts WHERE institution_id = :institution_id";
    $stmt = $this->db->prepare($sql);
    $stmt->execute([':institution_id' => $institutionId]);
    return $stmt->fetchAll(PDO::FETCH_COLUMN);
}
}

```

### 33. ArchImp/models/Justification.php

```

<?php
class Justification {
    private $db;

    public function __construct($database) {
        $this->db = $database->connect();
    }

    public function create($data) {
        $sql = "INSERT INTO justifications
(attendance_id, student_id, submitted_by, reason, document_path)
VALUES (:attendance_id, :student_id, :submitted_by, :reason, :document_path)";

        $stmt = $this->db->prepare($sql);
        return $stmt->execute($data);
    }

    public function getPending() {
        $sql = "SELECT j.*,
CONCAT(u.last_name, ' ', u.first_name) as student_name,
CONCAT(s.last_name, ' ', s.first_name) as submitted_by_name,
a.date as attendance_date,
sub.name as subject_name,
c.name as course_name
FROM justifications j
INNER JOIN users u ON j.student_id = u.id
INNER JOIN users s ON j.submitted_by = s.id
INNER JOIN attendances a ON j.attendance_id = a.id
INNER JOIN subjects sub ON a.subject_id = sub.id
LEFT JOIN course_students cs ON u.id = cs.student_id
LEFT JOIN courses c ON cs.course_id = c.id
WHERE j.status = 'pendiente'";
    }
}

```

```

ORDER BY j.created_at DESC";

$stmt = $this->db->query($sql);
return $stmt->fetchAll();
}

public function getByStudent($studentId) {
    $sql = "SELECT j.*,
    a.date as attendance_date,
    sub.name as subject_name,
    j.created_at as submitted_at,
    j.updated_at as reviewed_at,
    CONCAT(r.last_name, ' ', r.first_name) as reviewer_name
    FROM justifications j
    INNER JOIN attendances a ON j.attendance_id = a.id
    INNER JOIN subjects sub ON a.subject_id = sub.id
    LEFT JOIN users r ON j.reviewed_by = r.id
    WHERE j.student_id = :student_id
    ORDER BY j.created_at DESC";

    $stmt = $this->db->prepare($sql);
    $stmt->execute([':student_id' => $studentId]);
    return $stmt->fetchAll();
}

public function approve($justificationId, $reviewerId, $notes = '') {
    $sql = "UPDATE justifications
    SET status = 'aprobado',
    reviewed_by = :reviewed_by,
    review_notes = :notes,
    updated_at = NOW()
    WHERE id = :id";

    $stmt = $this->db->prepare($sql);
    $result = $stmt->execute([
    ':reviewed_by' => $reviewerId,
    ':notes' => $notes,
    ':id' => $justificationId
    ]);

    if ($result) {
        // Actualizar asistencia a justificado
        $sql2 = "UPDATE attendances a
        INNER JOIN justifications j ON a.id = j.attendance_id
        SET a.status = 'justificado'
        WHERE j.id = :id";
        $stmt2 = $this->db->prepare($sql2);
        $stmt2->execute([':id' => $justificationId]);
    }

    return $result;
}

public function reject($justificationId, $reviewerId, $notes) {
    $sql = "UPDATE justifications
    SET status = 'rechazado',
    reviewed_by = :reviewed_by,
    review_notes = :notes,
    updated_at = NOW()
    WHERE id = :id";

    $stmt = $this->db->prepare($sql);
    return $stmt->execute([
    ':reviewed_by' => $reviewerId,
    ':notes' => $notes,

```

```

':id' => $justificationId
]);
}

public function existsByAttendance($attendanceId) {
$sql = "SELECT COUNT(*) as count FROM justifications WHERE attendance_id = :attendance_id";
$stmt = $this->db->prepare($sql);
$stmt->execute([':attendance_id' => $attendanceId]);
$result = $stmt->fetch();
return $result['count'] > 0;
}
}

```

## 34. ArchImp/models/Notification.php

```

<?php
class Notification {
private $db;

public function __construct($database) {
$this->db = $database->connect();
}

public function create($userId, $title, $message, $type = 'info') {
$sql = "INSERT INTO notifications (user_id, title, message, type, created_at)
VALUES (:user_id, :title, :message, :type, NOW())";

$stmt = $this->db->prepare($sql);
return $stmt->execute([
':user_id' => $userId,
':title' => $title,
':message' => $message,
':type' => $type
]);
}

public function getByUser($userId, $limit = 10) {
$sql = "SELECT * FROM notifications
WHERE user_id = :user_id
ORDER BY created_at DESC
LIMIT :limit";

$stmt = $this->db->prepare($sql);
$stmt->bindValue(':user_id', $userId, PDO::PARAM_INT);
$stmt->bindValue(':limit', $limit, PDO::PARAM_INT);
$stmt->execute();
return $stmt->fetchAll();
}

public function markAsRead($notificationId) {
$sql = "UPDATE notifications SET is_read = 1 WHERE id = :id";
$stmt = $this->db->prepare($sql);
return $stmt->execute([':id' => $notificationId]);
}

public function getUnreadCount($userId) {
$sql = "SELECT COUNT(*) as count FROM notifications
WHERE user_id = :user_id AND is_read = 0";

$stmt = $this->db->prepare($sql);
$stmt->execute([':user_id' => $userId]);
return $stmt->fetch()['count'];
}
}

```

```
}
```

### 35. ArchImp/models/Representative.php

```
<?php
class Representative {
    private $db;

    public function __construct($database) {
        $this->db = $database->connect();
    }

    public function assignStudent($representativeId, $studentId, $relationship, $isPrimary = 0) {
        $sql = "INSERT INTO representatives
        (representative_id, student_id, relationship, is_primary)
        VALUES
        (:rep_id, :stu_id, :rel_ins, :prim_ins)
        ON DUPLICATE KEY UPDATE
        relationship = :rel_upd,
        is_primary = :prim_upd";

        $stmt = $this->db->prepare($sql);

        return $stmt->execute([
            ':rep_id' => $representativeId,
            ':stu_id' => $studentId,
            ':rel_ins' => $relationship,
            ':prim_ins' => $isPrimary,
            ':rel_upd' => $relationship,
            ':prim_upd' => $isPrimary
        ]);
    }

    public function getStudentsByRepresentative($representativeId) {
        $sql = "SELECT u.*, r.relationship, r.is_primary, c.name as course_name, sh.name as shift_name
        FROM users u
        INNER JOIN representatives r ON u.id = r.student_id
        LEFT JOIN course_students cs ON u.id = cs.student_id
        LEFT JOIN courses c ON cs.course_id = c.id
        LEFT JOIN shifts sh ON c.shift_id = sh.id
        WHERE r.representative_id = :representative_id
        ORDER BY r.is_primary DESC, u.last_name, u.first_name";

        $stmt = $this->db->prepare($sql);
        $stmt->execute([':representative_id' => $representativeId]);
        return $stmt->fetchAll();
    }

    public function getRepresentativesByStudent($studentId) {
        $sql = "SELECT u.*, r.relationship, r.is_primary
        FROM users u
        INNER JOIN representatives r ON u.id = r.representative_id
        WHERE r.student_id = :student_id
        ORDER BY r.is_primary DESC";

        $stmt = $this->db->prepare($sql);
        $stmt->execute([':student_id' => $studentId]);
        return $stmt->fetchAll();
    }

    public function removeStudent($representativeId, $studentId) {
        $sql = "DELETE FROM representatives
        WHERE representative_id = :representative_id AND student_id = :student_id";
    }
}
```

```

$stmt = $this->db->prepare($sql);
return $stmt->execute([
    ':representative_id' => $representativeId,
    ':student_id' => $studentId
]);
}
}

```

### 36. ArchImp/models/Role.php

```

<?php
class Role {
    private $db;

    public function __construct($database) {
        $this->db = $database->connect();
    }

    public function getAll() {
        $sql = "SELECT * FROM roles ORDER BY name";
        $stmt = $this->db->query($sql);
        return $stmt->fetchAll();
    }

    public function findById($id) {
        $sql = "SELECT * FROM roles WHERE id = :id";
        $stmt = $this->db->prepare($sql);
        $stmt->execute([':id' => $id]);
        return $stmt->fetch();
    }
}

```

### 37. ArchImp/models/SchoolYear.php

```

<?php
class SchoolYear {
    private $db;

    public function __construct($database) {
        $this->db = $database->connect();
    }

    public function getActive() {
        $sql = "SELECT * FROM school_years
        WHERE institution_id = :institution_id AND is_active = 1
        LIMIT 1";

        $stmt = $this->db->prepare($sql);
        $stmt->execute([':institution_id' => $_SESSION['institution_id']]);
        return $stmt->fetch();
    }

    public function getAll() {
        $sql = "SELECT * FROM school_years
        WHERE institution_id = :institution_id
        ORDER BY start_date DESC";

        $stmt = $this->db->prepare($sql);
        $stmt->execute([':institution_id' => $_SESSION['institution_id']]);
    }
}

```

```

return $stmt->fetchAll();
}

public function findById($id) {
$sql = "SELECT * FROM school_years
WHERE id = :id AND institution_id = :institution_id";

$stmt = $this->db->prepare($sql);
$stmt->execute([
':id' => $id,
':institution_id' => $_SESSION['institution_id']
]);

return $stmt->fetch();
}

public function create($data) {
$sql = "INSERT INTO school_years
(institution_id, name, start_date, end_date, is_active)
VALUES
(:institution_id, :name, :start_date, :end_date, :is_active)";

$stmt = $this->db->prepare($sql);
return $stmt->execute([
':institution_id' => $data['institution_id'],
':name' => $data['name'],
':start_date' => $data['start_date'],
':end_date' => $data['end_date'],
':is_active' => $data['is_active']
]);
}

public function update($data) {
$sql = "UPDATE school_years
SET name=:name,
start_date=:start_date,
end_date=:end_date
WHERE id=:id
AND institution_id=:institution_id";

$stmt = $this->db->prepare($sql);

return $stmt->execute([
':id'=>$data['id'],
':name'=>$data['name'],
':start_date'=>$data['start_date'],
':end_date'=>$data['end_date'],
':institution_id'=>$_SESSION['institution_id']
]);
}

public function delete($id) {
$stmt = $this->db->prepare(
"DELETE FROM school_years
WHERE id=:id AND institution_id=:institution_id"
);

return $stmt->execute([
':id'=>$id,
':institution_id'=>$_SESSION['institution_id']
]);
}

public function activate($id) {

```

```

// desactiva todos
$this->db->prepare(
"UPDATE school_years
SET is_active=0
WHERE institution_id=:institution_id"
)->execute([':institution_id'=>$_SESSION['institution_id']]);

if ($id == 0) return true;

// activa uno
$stmt = $this->db->prepare(
"UPDATE school_years
SET is_active=1
WHERE id=:id AND institution_id=:institution_id"
);

return $stmt->execute([
':id'=>$id,
':institution_id'=>$_SESSION['institution_id']
]);
}

public function deactivate($id) {
$stmt = $this->db->prepare(
"UPDATE school_years
SET is_active=0
WHERE id=:id AND institution_id=:institution_id"
);

return $stmt->execute([
':id'=>$id,
':institution_id'=>$_SESSION['institution_id']
]);
}

public function checkOverlap($start, $end, $excludeId = null) {

$sql = "SELECT COUNT(*)
FROM school_years
WHERE institution_id = :institution_id
AND (start_date <= :end AND end_date >= :start)";

if ($excludeId) {
$sql .= " AND id != :excludeId";
}

$stmt = $this->db->prepare($sql);

$params = [
':institution_id' => $_SESSION['institution_id'],
':start' => $start,
':end' => $end
];

if ($excludeId) {
$params[':excludeId'] = $excludeId;
}

$stmt->execute($params);
return $stmt->fetchColumn() > 0;
}
}

```

### 38. ArchImp/models/Shift.php

```
<?php
class Shift {
private $db;

public function __construct($database) {
    $this->db = $database->connect();
}

public function getAll() {
    $sql = "SELECT * FROM shifts ORDER BY id";
    $stmt = $this->db->query($sql);
    return $stmt->fetchAll();
}
}
```

### 39. ArchImp/models/Subject.php

```
<?php
class Subject {
private $db;

public function __construct($database) {
    $this->db = $database->connect();
}

public function getAll() {
    $sql = "SELECT * FROM subjects
    WHERE institution_id = :institution_id
    ORDER BY name";

    $stmt = $this->db->prepare($sql);
    $stmt->execute([':institution_id' => $_SESSION['institution_id']]);
    return $stmt->fetchAll();
}

public function create($data) {
    $sql = "INSERT INTO subjects (institution_id, name, code)
    VALUES (:institution_id, :name, :code)";

    $stmt = $this->db->prepare($sql);
    return $stmt->execute($data);
}

public function findById($id) {
    $sql = "SELECT * FROM subjects WHERE id = :id";
    $stmt = $this->db->prepare($sql);
    $stmt->execute([':id' => $id]);
    return $stmt->fetch();
}

public function update($data) {
    $sql = "UPDATE subjects SET
    name = :name,
    code = :code,
    updated_at = NOW()
    WHERE id = :id";

    $stmt = $this->db->prepare($sql);
    return $stmt->execute($data);
}
```



```

}

public function delete($id) {
    $sql = "DELETE FROM subjects WHERE id = :id";
    $stmt = $this->db->prepare($sql);
    return $stmt->execute([':id' => $id]);
}
}

```

## 40. ArchImp/models/TeacherAssignment.php

```

<?php
class TeacherAssignment {
    private $db;

    public function __construct($database) {
        $this->db = $database->connect();
    }

    public function assign($data) {
        // Verificar si ya existe asignación curso-materia
        $checkSql = "SELECT ta.id, CONCAT(u.last_name, ' ', u.first_name) as teacher_name
        FROM teacher_assignments ta
        INNER JOIN users u ON ta.teacher_id = u.id
        WHERE ta.course_id = :course_id
        AND ta.subject_id = :subject_id
        AND ta.school_year_id = :school_year_id";

        $checkStmt = $this->db->prepare($checkSql);
        $checkStmt->execute([
            ':course_id' => $data[':course_id'],
            ':subject_id' => $data[':subject_id'],
            ':school_year_id' => $data[':school_year_id']
        ]);

        $existing = $checkStmt->fetch();

        if ($existing) {
            return [
                'success' => false,
                'message' => 'La materia ya está asignada a: ' . $existing['teacher_name']
            ];
        }

        $sql = "INSERT INTO teacher_assignments
        (teacher_id, course_id, subject_id, school_year_id, is_tutor)
        VALUES (:teacher_id, :course_id, :subject_id, :school_year_id, :is_tutor)";

        $stmt = $this->db->prepare($sql);
        $result = $stmt->execute($data);

        return [
            'success' => $result,
            'message' => 'Asignación creada correctamente'
        ];
    }

    public function getByTeacher($teacherId) {
        $sql = "SELECT ta.*, c.name as course_name, s.name as subject_name,
        sh.name as shift_name, sy.name as year_name
        FROM teacher_assignments ta
        INNER JOIN courses c ON ta.course_id = c.id
        INNER JOIN subjects s ON ta.subject_id = s.id

```

```

INNER JOIN shifts sh ON c.shift_id = sh.id
INNER JOIN school_years sy ON ta.school_year_id = sy.id
WHERE ta.teacher_id = :teacher_id
ORDER BY c.name, s.name";

$stmt = $this->db->prepare($sql);
$stmt->execute([':teacher_id' => $teacherId]);
return $stmt->fetchAll();
}

public function getByCourse($courseId) {
    $sql = "SELECT ta.*,
    CONCAT(u.last_name, ' ', u.first_name) as teacher_name,
    s.name as subject_name
    FROM teacher_assignments ta
    INNER JOIN users u ON ta.teacher_id = u.id
    INNER JOIN subjects s ON ta.subject_id = s.id
    WHERE ta.course_id = :course_id
    ORDER BY s.name";

    $stmt = $this->db->prepare($sql);
    $stmt->execute([':course_id' => $courseId]);
    return $stmt->fetchAll();
}

public function getTutorByCourse($courseId, $schoolYearId) {
    $sql = "SELECT u.*, ta.id as assignment_id
    FROM teacher_assignments ta
    INNER JOIN users u ON ta.teacher_id = u.id
    WHERE ta.course_id = :course_id
    AND ta.school_year_id = :school_year_id
    AND ta.is_tutor = 1
    LIMIT 1";

    $stmt = $this->db->prepare($sql);
    $stmt->execute([
        ':course_id' => $courseId,
        ':school_year_id' => $schoolYearId
    ]);
    return $stmt->fetch();
}

public function setTutor($courseId, $teacherId, $schoolYearId) {
    // Verificar si el docente ya es tutor de otro curso en el mismo año lectivo
    $checkSql = "SELECT c.name
    FROM teacher_assignments ta
    INNER JOIN courses c ON ta.course_id = c.id
    WHERE ta.teacher_id = :teacher_id
    AND ta.school_year_id = :school_year_id
    AND ta.is_tutor = 1
    AND ta.course_id != :course_id";

    $checkStmt = $this->db->prepare($checkSql);
    $checkStmt->execute([
        ':teacher_id' => $teacherId,
        ':school_year_id' => $schoolYearId,
        ':course_id' => $courseId
    ]);

    $existingTutor = $checkStmt->fetch();

    if ($existingTutor) {
        return [
            'success' => false,
            'message' => 'Este docente ya es tutor del curso: ' . $existingTutor['name']
        ];
    }
}

```

```

};
}

// Quitar tutor actual del curso
$sql1 = "UPDATE teacher_assignments
SET is_tutor = 0
WHERE course_id = :course_id AND school_year_id = :school_year_id";
$stmt1 = $this->db->prepare($sql1);
$stmt1->execute([
':course_id' => $courseId,
':school_year_id' => $schoolYearId
]);

// Verificar si docente tiene asignación en el curso
$checkAssignment = "SELECT id FROM teacher_assignments
WHERE course_id = :course_id
AND teacher_id = :teacher_id
AND school_year_id = :school_year_id
LIMIT 1";
$stmt = $this->db->prepare($checkAssignment);
$stmt->execute([
':course_id' => $courseId,
':teacher_id' => $teacherId,
':school_year_id' => $schoolYearId
]);
$existing = $stmt->fetch();

if ($existing) {
// Actualizar asignación existente
$sql2 = "UPDATE teacher_assignments
SET is_tutor = 1
WHERE id = :id";
$stmt2 = $this->db->prepare($sql2);
$result = $stmt2->execute([':id' => $existing['id']]);
} else {
return [
'success' => false,
'message' => 'El docente debe tener al menos una asignatura en el curso antes de ser tutor'
];
}

return [
'success' => $result,
'message' => 'Tutor asignado correctamente'
];
}

public function remove($assignmentId) {
// Verificar si es tutor antes de eliminar
$checkSql = "SELECT is_tutor FROM teacher_assignments WHERE id = :id";
$checkStmt = $this->db->prepare($checkSql);
$checkStmt->execute([':id' => $assignmentId]);
$assignment = $checkStmt->fetch();

if ($assignment && $assignment['is_tutor'] == 1) {
return [
'success' => false,
'message' => 'No se puede eliminar: este docente es tutor del curso. Quite primero la tutoría.'
];
}

$sql = "DELETE FROM teacher_assignments WHERE id = :id";
$stmt = $this->db->prepare($sql);

```

```

$result = $stmt->execute([':id' => $assignmentId]);

return [
'success' => $result,
'message' => 'Asignación eliminada correctamente'
];
}

public function getAll() {
$sql = "SELECT ta.*,
CONCAT(u.last_name, ' ', u.first_name) as teacher_name,
c.name as course_name,
s.name as subject_name,
sy.name as year_name
FROM teacher_assignments ta
INNER JOIN users u ON ta.teacher_id = u.id
INNER JOIN courses c ON ta.course_id = c.id
INNER JOIN subjects s ON ta.subject_id = s.id
INNER JOIN school_years sy ON ta.school_year_id = sy.id
ORDER BY c.name, s.name";

$stmt = $this->db->query($sql);
return $stmt->fetchAll();
}
}

```

## 41. ArchImp/models/User.php

```

<?php
class User {
public $db;

public function __construct($database) {
$this->db = $database->connect();
}

public function create($data) {
$sql = "INSERT INTO users (institution_id, username, email, password, first_name, last_name, dni, phone)
VALUES (:institution_id, :username, :email, :password, :first_name, :last_name, :dni, :phone)";

$stmt = $this->db->prepare($sql);
return $stmt->execute([
':institution_id' => $data['institution_id'],
':username' => $data['username'],
':email' => $data['email'],
':password' => Security::hashPassword($data['password']),
':first_name' => $data['first_name'],
':last_name' => $data['last_name'],
':dni' => $data['dni'] ?? null,
':phone' => $data['phone'] ?? null
]);
}

public function findByEmail($email) {
$sql = "SELECT * FROM users WHERE email = :email AND is_active = 1";
$stmt = $this->db->prepare($sql);
$stmt->execute([':email' => $email]);
return $stmt->fetch();
}

public function findById($id) {
$sql = "SELECT * FROM users WHERE id = :id";
$stmt = $this->db->prepare($sql);
$stmt->execute([':id' => $id]);
}
}

```

```

return $stmt->fetch();
}

public function getUserRoles($userId) {
    $sql = "SELECT r.name FROM roles r
    INNER JOIN user_roles ur ON r.id = ur.role_id
    WHERE ur.user_id = :user_id";
    $stmt = $this->db->prepare($sql);
    $stmt->execute([':user_id' => $userId]);
    return $stmt->fetchAll(PDO::FETCH_COLUMN);
}

public function assignRole($userId, $roleId) {
    $sql = "INSERT IGNORE INTO user_roles (user_id, role_id) VALUES (:user_id, :role_id)";
    $stmt = $this->db->prepare($sql);
    return $stmt->execute([':user_id' => $userId, ':role_id' => $roleId]);
}

public function setResetToken($email, $token) {
    $sql = "UPDATE users SET reset_token = :token, reset_expires = DATE_ADD(NOW(), INTERVAL 1 HOUR)
    WHERE email = :email";
    $stmt = $this->db->prepare($sql);
    return $stmt->execute([':token' => $token, ':email' => $email]);
}

public function validateResetToken($token) {
    $sql = "SELECT * FROM users WHERE reset_token = :token AND reset_expires > NOW()";
    $stmt = $this->db->prepare($sql);
    $stmt->execute([':token' => $token]);
    return $stmt->fetch();
}

public function resetPassword($userId, $newPassword) {
    $sql = "UPDATE users SET password = :password, reset_token = NULL, reset_expires = NULL
    WHERE id = :id";
    $stmt = $this->db->prepare($sql);
    return $stmt->execute([
        ':password' => Security::hashPassword($newPassword),
        ':id' => $userId
    ]);
}

public function getAll() {
    $sql = "SELECT u.*, GROUP_CONCAT(r.name) as roles
    FROM users u
    LEFT JOIN user_roles ur ON u.id = ur.user_id
    LEFT JOIN roles r ON ur.role_id = r.id
    WHERE u.institution_id = :institution_id
    GROUP BY u.id
    ORDER BY u.last_name, u.first_name";

    $stmt = $this->db->prepare($sql);
    $stmt->execute([':institution_id' => $_SESSION['institution_id']]);
    return $stmt->fetchAll();
}

public function removeRole($userId, $roleId) {
    $sql = "DELETE FROM user_roles WHERE user_id = :user_id AND role_id = :role_id";
    $stmt = $this->db->prepare($sql);
    return $stmt->execute([':user_id' => $userId, ':role_id' => $roleId]);
}

public function getUserWithRoles($userId) {
    $user = $this->findById($userId);
    if ($user) {

```

```

$user['roles'] = $this->getUserRoles($userId);
}
return $user;
}

/* public function getByRole($roleName) {
$sql = "SELECT u.* FROM users u
INNER JOIN user_roles ur ON u.id = ur.user_id
INNER JOIN roles r ON ur.role_id = r.id
WHERE r.name = :role_name AND u.institution_id = :institution_id
ORDER BY u.last_name, u.first_name";

$stmt = $this->db->prepare($sql);
$stmt->execute([
':role_name' => $roleName,
':institution_id' => $_SESSION['institution_id']
]);
return $stmt->fetchAll();
} */

// Agregar al final de la clase User

public function getByRole($roleName) {
$sql = "SELECT u.* FROM users u
INNER JOIN user_roles ur ON u.id = ur.user_id
INNER JOIN roles r ON ur.role_id = r.id
WHERE r.name = :role_name AND u.institution_id = :institution_id
ORDER BY u.last_name, u.first_name";

$stmt = $this->db->prepare($sql);
$stmt->execute([
':role_name' => $roleName,
':institution_id' => $_SESSION['institution_id']
]);
return $stmt->fetchAll();
}

public function getStudentsNotEnrolled($schoolYearId) {
$sql = "SELECT u.* FROM users u
INNER JOIN user_roles ur ON u.id = ur.user_id
INNER JOIN roles r ON ur.role_id = r.id
WHERE r.name = 'estudiante'
AND u.institution_id = :institution_id
AND u.id NOT IN (
SELECT student_id FROM course_students
WHERE school_year_id = :school_year_id
)
ORDER BY u.last_name, u.first_name";

$stmt = $this->db->prepare($sql);
$stmt->execute([
':institution_id' => $_SESSION['institution_id'],
':school_year_id' => $schoolYearId
]);
return $stmt->fetchAll();
}

public function getStudentCourse($studentId, $schoolYearId) {
$sql = "SELECT c.*, s.name as shift_name
FROM courses c
INNER JOIN course_students cs ON c.id = cs.course_id
INNER JOIN shifts s ON c.shift_id = s.id
WHERE cs.student_id = :student_id
AND cs.school_year_id = :school_year_id";

```

```

$stmt = $this->db->prepare($sql);
$stmt->execute([
    ':student_id' => $studentId,
    ':school_year_id' => $schoolYearId
]);
return $stmt->fetch();
}

public function updateProfile($userId, $data) {
    $sql = "UPDATE users
    SET first_name = :first_name,
    last_name = :last_name,
    phone = :phone,
    email = :email,
    updated_at = NOW()
    WHERE id = :id";

    $stmt = $this->db->prepare($sql);
    return $stmt->execute([
        ':first_name' => $data['first_name'],
        ':last_name' => $data['last_name'],
        ':phone' => $data['phone'],
        ':email' => $data['email'],
        ':id' => $userId
    ]);
}

public function updatePassword($userId, $newPassword) {
    $sql = "UPDATE users SET password = :password WHERE id = :id";
    $stmt = $this->db->prepare($sql);
    return $stmt->execute([
        ':password' => Security::hashPassword($newPassword),
        ':id' => $userId
    ]);
}

public function findByEmailOrUsername($emailOrUsername) {
    $sql = "SELECT * FROM users
    WHERE (email = :email OR username = :username)
    AND is_active = 1";

    $stmt = $this->db->prepare($sql);
    $stmt->execute([
        ':email' => $emailOrUsername,
        ':username' => $emailOrUsername
    ]);
    return $stmt->fetch();
}

public function getUserRoleIds($userId) {
    $sql = "SELECT r.id FROM roles r
    INNER JOIN user_roles ur ON r.id = ur.role_id
    WHERE ur.user_id = :user_id
    ORDER BY r.name";
    $stmt = $this->db->prepare($sql);
    $stmt->execute([':user_id' => $userId]);
    return $stmt->fetchAll(PDO::FETCH_COLUMN);
}

public function findByUsername($username) {
    $sql = "SELECT * FROM users WHERE username = :username";
    $stmt = $this->db->prepare($sql);
    $stmt->execute([':username' => $username]);
    return $stmt->fetch();
}

```

```

public function update($data) {
    // Si se incluye password, actualizarla
    if (isset($data['password'])) {
        $sql = "UPDATE users SET
        username = :username,
        email = :email,
        password = :password,
        first_name = :first_name,

        last_name = :last_name,
        dni = :dni,
        phone = :phone,
        updated_at = NOW()
        WHERE id = :id";

        $params = [
            ':id' => $data['id'],
            ':username' => $data['username'],
            ':email' => $data['email'],
            ':password' => Security::hashPassword($data['password']),
            ':first_name' => $data['first_name'],
            ':last_name' => $data['last_name'],
            ':dni' => $data['dni'],
            ':phone' => $data['phone']
        ];
    } else {
        // No actualizar password
        $sql = "UPDATE users SET
        username = :username,
        email = :email,
        first_name = :first_name,
        last_name = :last_name,
        dni = :dni,
        phone = :phone,
        updated_at = NOW()
        WHERE id = :id";

        $params = [
            ':id' => $data['id'],
            ':username' => $data['username'],
            ':email' => $data['email'],
            ':first_name' => $data['first_name'],
            ':last_name' => $data['last_name'],
            ':dni' => $data['dni'],
            ':phone' => $data['phone']
        ];
    }

    $stmt = $this->db->prepare($sql);
    return $stmt->execute($params);
}

public function delete($userId) {
    // Primero eliminar roles
    $sql = "DELETE FROM user_roles WHERE user_id = :user_id";
    $stmt = $this->db->prepare($sql);
    $stmt->execute([':user_id' => $userId]);

    // Eliminar representaciones (tanto como representante como estudiante)
    $sql = "DELETE FROM representatives
    WHERE representative_id = :rep_id OR student_id = :stu_id";
    $stmt = $this->db->prepare($sql);
    $stmt->execute([
        ':rep_id' => $userId,
        ':stu_id' => $userId
    ]);
}

```



```

});

// Eliminar de course_students
$sql = "DELETE FROM course_students WHERE student_id = :user_id";
$stmt = $this->db->prepare($sql);
$stmt->execute([':user_id' => $userId]);

// Eliminar asignaciones docentes
$sql = "DELETE FROM teacher_assignments WHERE teacher_id = :user_id";
$stmt = $this->db->prepare($sql);
$stmt->execute([':user_id' => $userId]);

// Finalmente eliminar usuario
$sql = "DELETE FROM users WHERE id = :id";
$stmt = $this->db->prepare($sql);
return $stmt->execute([':id' => $userId]);
}

public function deactivate($userId) {
    $sql = "UPDATE users SET is_active = 0, updated_at = NOW() WHERE id = :id";
    $stmt = $this->db->prepare($sql);
    return $stmt->execute([':id' => $userId]);
}
}

```

## 42. ArchImp/views/academic/course\_edit.php

```

<?php
require_once BASE_PATH . '/helpers/Security.php';
Security::requireLogin();
if (!Security::hasRole('autoridad')) {
    die('Acceso denegado');
}
?>
<!DOCTYPE html>
<html lang="es">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Editar Curso - EcuAsist</title>
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet">
<style>
body { padding-top: 70px; }
</style>
</head>
<body>
<?php include BASE_PATH . '/views/partials/navbar.php'; ?>

<div class="container">
<div class="row justify-content-center">
<div class="col-md-8">
<div class="card shadow">
<div class="card-header bg-primary text-white">
<h4 class="mb-0">■ Editar Curso: <?= htmlspecialchars($course['name']) ?></h4>
</div>
<div class="card-body">
<?php if (!empty($errors)): ?>
<div class="alert alert-danger">
<?php foreach ($errors as $error): ?>
<div><?= htmlspecialchars($error) ?></div>
<?php endforeach; ?>
</div>

```

```

<?php endif; ?>

<form method="POST" id="courseForm">
<div class="mb-3">
<label class="form-label">Nivel *</label>
<select name="grade_level" id="grade_level" class="form-control" required>
<option value="">Seleccionar...</option>
<?php
$levels = ['8vo EGB', '9no EGB', '10mo EGB', '1ro BGU', '2do BGU', '3ro BGU', '1ro Técnico', '2do Técnico',
'3ro Técnico'];
foreach($levels as $level):
?>
<option value="<?=$level ?>" <?=$course['grade_level'] == $level ? 'selected' : '' ?><?=$level
?></option>
<?php endforeach; ?>
</select>
</div>

<div class="mb-3">
<label class="form-label">Paralelo *</label>
<select name="parallel" id="parallel" class="form-control" required>
<option value="">Seleccionar...</option>
<?php foreach(range('A', 'Z') as $letter): ?>
<option value="<?=$letter ?>" <?=$course['parallel'] == $letter ? 'selected' : '' ?><?=$letter
?></option>
<?php endforeach; ?>
</select>
</div>

<div class="mb-3">
<label class="form-label">Jornada *</label>
<select name="shift_id" id="shift_id" class="form-control" required>
<option value="">Seleccionar...</option>
<?php foreach($shifts as $shift): ?>
<option value="<?=$shift['id'] ?>"
data-shift="<?=$shift['name'] ?>"
<?=$course['shift_id'] == $shift['id'] ? 'selected' : '' ?>>
<?=$ucfirst($shift['name']) ?>
</option>
<?php endforeach; ?>
</select>
</div>

<div class="mb-3">
<label class="form-label">Nombre del Curso (generado automáticamente)</label>
<input type="text" name="name" id="course_name" class="form-control" readonly style="background: #f0f0f0;"
value="<?=$htmlspecialchars($course['name']) ?>">
</div>

<div class="d-flex justify-content-between mt-4">
<a href="?action=academic" class="btn btn-secondary">< Cancelar</a>
<button type="submit" class="btn btn-success">✓ Guardar Cambios</button>
</div>
</form>
</div>
</div>
</div>
</div>
</div>
</div>

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"></script>
<script>
function generateCourseName() {
const level = document.getElementById('grade_level').value;
const parallel = document.getElementById('parallel').value;

```

```

const shiftSelect = document.getElementById('shift_id');
const shiftOption = shiftSelect.options[shiftSelect.selectedIndex];
const shiftName = shiftOption ? shiftOption.getAttribute('data-shift') : '';

if (level && parallel && shiftName) {
const courseName = `${level} "${parallel}" - ${shiftName.charAt(0).toUpperCase() + shiftName.slice(1)}`;
document.getElementById('course_name').value = courseName;
}
}

document.getElementById('grade_level').addEventListener('change', generateCourseName);
document.getElementById('parallel').addEventListener('change', generateCourseName);
document.getElementById('shift_id').addEventListener('change', generateCourseName);
</script>
</body>
</html>

```

### 43. ArchImp/views/academic/course\_students.php

```

<?php Security::requireLogin(); ?>
<!DOCTYPE html>
<html lang="es">
<head>
<meta charset="UTF-8">
<title>Estudiantes del Curso - EcuAsist</title>
<style>
* { margin: 0; padding: 0; box-sizing: border-box; }
body { font-family: Arial, sans-serif; background: #f4f4f4; }
.navbar { background: #007bff; color: white; padding: 15px 30px; display: flex; justify-content:
space-between; align-items: center; }
.navbar h1 { font-size: 24px; }
.navbar a { color: white; text-decoration: none; margin-left: 20px; }
.container { max-width: 1000px; margin: 30px auto; padding: 0 20px; }
.card { background: white; padding: 30px; border-radius: 8px; box-shadow: 0 2px 10px rgba(0,0,0,0.1); }
table { width: 100%; border-collapse: collapse; margin-top: 20px; }
th, td { padding: 12px; text-align: left; border-bottom: 1px solid #ddd; }
th { background: #f8f9fa; font-weight: bold; }
h2 { margin-bottom: 10px; color: #333; }
.course-info { background: #e7f3ff; padding: 15px; border-radius: 4px; margin-bottom: 20px; }
</style>
</head>
<body>
<div class="navbar">
<h1>Estudiantes del Curso</h1>
<div>
<a href="?action=academic">← Configuración Académica</a>
<a href="?action=logout">Cerrar sesión</a>
</div>
</div>

<div class="container">
<div class="card">
<div class="course-info">
<h2><?= $course['name'] ?></h2>
<p><strong>Nivel:</strong> <?= $course['grade_level'] ?> | <strong>Paralelo:</strong> <?=
$course['parallel'] ?></p>
<p><strong>Jornada:</strong> <?= ucfirst($course['shift_name']) ?> | <strong>Año Lectivo:</strong> <?=
$course['year_name'] ?></p>
</div>

<h3>Estudiantes Matriculados (<?= count($students) ?>)</h3>

<?php if(count($students) > 0): ?>

```

```

<table>
<thead>
<tr>
<th>#</th>
<th>Apellidos y Nombres</th>
<th>Cédula</th>
<th>Fecha Matrícula</th>
</tr>
</thead>
<tbody>
<?php foreach($students as $index => $student): ?>
<tr>
<td><?= $index + 1 ?></td>
<td><?= $student['last_name'] . ' ' . $student['first_name'] ?></td>
<td><?= $student['dni'] ?? '-' ?></td>
<td><?= date('d/m/Y', strtotime($student['enrollment_date'])) ?></td>
</tr>
<?php endforeach; ?>
</tbody>
</table>
<?php else: ?>
<p style="color: #666; margin-top: 20px;">No hay estudiantes matriculados en este curso.</p>
<?php endif; ?>
</div>
</div>
</body>
</html>

```

## 44. ArchImp/views/academic/create.php

```

<?php
require_once BASE_PATH . '/helpers/Security.php';
Security::requireLogin();
if (!Security::hasRole('autoridad')) {
die('Acceso denegado');
}
?>
<!DOCTYPE html>
<html lang="es">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Crear Año Lectivo - EcuAsist</title>
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet">
<style>
body { padding-top: 70px; }
.form-section {
background: #f8f9fa;
padding: 20px;
border-radius: 8px;
margin-bottom: 20px;
}
.form-section h5 {
color: #007bff;
border-bottom: 2px solid #007bff;
padding-bottom: 10px;
margin-bottom: 20px;
}
.required::after {
content: " *";
color: red;
}

```

```

.info-box {
background: #d1ecf1;
border-left: 4px solid #0c5460;
padding: 15px;
margin-bottom: 20px;
}
</style>
</head>
<body>
<?php include BASE_PATH . '/views/partials/navbar.php'; ?>

<div class="container">
<div class="row justify-content-center">
<div class="col-md-8">
<div class="card shadow">
<div class="card-header bg-primary text-white">
<h4 class="mb-0">■ Crear Nuevo Año Lectivo</h4>
</div>
<div class="card-body">
<?php if (!empty($errors)): ?>
<div class="alert alert-danger">
<strong>Errores:</strong>
<ul class="mb-0">
<?php foreach ($errors as $error): ?>
<li><?= htmlspecialchars($error) ?></li>
<?php endforeach; ?>
</ul>
</div>
<?php endif; ?>

<div class="info-box">
<strong>■■ Información importante:</strong>
<ul class="mb-0 mt-2">
<li>Solo puede haber un año lectivo activo a la vez</li>
<li>Las fechas no pueden solaparse con otros años lectivos</li>
<li>Si marcas como activo, los demás se desactivarán automáticamente</li>
</ul>
</div>

<form method="POST" action="?action=create_school_year">
<div class="form-section">
<h5>■ Información General</h5>

<div class="mb-3">
<label class="form-label required">Nombre del Año Lectivo</label>
<input type="text"
name="name"
class="form-control"
placeholder="Ej: 2025-2026"
value="<?= isset($_POST['name']) ? htmlspecialchars($_POST['name']) : '' ?>"
required>
<small class="text-muted">Formato recomendado: YYYY-YYYY</small>
</div>

<div class="row">
<div class="col-md-6 mb-3">
<label class="form-label required">Fecha de Inicio</label>
<input type="date"
name="start_date"
class="form-control"
value="<?= isset($_POST['start_date']) ? $_POST['start_date'] : '' ?>"
required>
</div>
<div class="col-md-6 mb-3">
<label class="form-label required">Fecha de Fin</label>

```

```

<input type="date"
name="end_date"
class="form-control"
value="<?= isset($_POST['end_date']) ? $_POST['end_date'] : '' ?>"
required>
</div>
</div>

<div class="form-check">
<input class="form-check-input"
type="checkbox"
name="is_active"
id="is_active"
<?= isset($_POST['is_active']) ? 'checked' : '' ?>>
<label class="form-check-label" for="is_active">
<strong>Activar este año lectivo</strong>
<br>
<small class="text-muted">
Si se activa, todos los demás años lectivos se desactivarán automáticamente
</small>
</label>
</div>
</div>

<div class="d-flex justify-content-between mt-4">
<a href="?action=academic" class="btn btn-secondary">
← Cancelar
</a>
<button type="submit" class="btn btn-primary">
✓ Crear Año Lectivo
</button>
</div>
</form>
</div>
</div>
</div>
</div>
</div>
</div>

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"></script>
</body>
</html>

```

## 45. ArchImp/views/academic/edit.php

```

<?php
require_once BASE_PATH . '/helpers/Security.php';
Security::requireLogin();
if (!Security::hasRole('autoridad')) {
die('Acceso denegado');
}
?>
<!DOCTYPE html>
<html lang="es">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Editar Año Lectivo - EcuAsist</title>
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet">
<style>
body { padding-top: 70px; }
.form-section {

```

```

background: #f8f9fa;
padding: 20px;
border-radius: 8px;
margin-bottom: 20px;
}
.form-section h5 {
color: #007bff;
border-bottom: 2px solid #007bff;
padding-bottom: 10px;
margin-bottom: 20px;
}
.required::after {
content: " *";
color: red;
}
.info-box {
background: #d1ecf1;
border-left: 4px solid #0c5460;
padding: 15px;
margin-bottom: 20px;
}
.status-badge {
display: inline-block;
padding: 8px 15px;
border-radius: 20px;
font-weight: bold;
font-size: 14px;
}
.status-active {
background: #28a745;
color: white;
}
.status-inactive {
background: #6c757d;
color: white;
}
</style>
</head>
<body>
<?php include BASE_PATH . '/views/partials/navbar.php'; ?>

<div class="container">
<div class="row justify-content-center">
<div class="col-md-8">
<div class="card shadow">
<div class="card-header bg-primary text-white">
<div class="d-flex justify-content-between align-items-center">
<h4 class="mb-0">■ Editar Año Lectivo: <?= htmlspecialchars($year['name']) ?></h4>
<span class="status-badge <?= $year['is_active'] ? 'status-active' : 'status-inactive' ?>">
<?= $year['is_active'] ? '✓ ACTIVO' : 'Inactivo' ?>
</span>
</div>
</div>
<div class="card-body">
<?php if (!empty($errors)): ?>
<div class="alert alert-danger">
<strong>Errores:</strong>
<ul class="mb-0">
<?php foreach ($errors as $error): ?>
<li><?= htmlspecialchars($error) ?></li>
<?php endforeach; ?>
</ul>
</div>
<?php endif; ?>

```

```

<div class="info-box">
<strong>■■ Información importante:</strong>
<ul class="mb-0 mt-2">
<li>Las fechas no pueden solaparse con otros años lectivos</li>
<li>El estado (activo/inactivo) se gestiona desde la tabla principal</li>
<li>No se puede eliminar un año lectivo activo</li>
</ul>
</div>

<form method="POST" action="?action=edit_school_year&id=<?= $year['id'] ?>">
<div class="form-section">
<h5>■ Información General</h5>

<div class="mb-3">
<label class="form-label required">Nombre del Año Lectivo</label>
<input type="text"
name="name"
class="form-control"
placeholder="Ej: 2025-2026"
value="<?= htmlspecialchars($year['name']) ?>"
required>
<small class="text-muted">Formato recomendado: YYYY-YYYY</small>
</div>

<div class="row">
<div class="col-md-6 mb-3">
<label class="form-label required">Fecha de Inicio</label>
<input type="date"
name="start_date"
class="form-control"
value="<?= $year['start_date'] ?>"
required>
</div>
<div class="col-md-6 mb-3">
<label class="form-label required">Fecha de Fin</label>
<input type="date"
name="end_date"
class="form-control"
value="<?= $year['end_date'] ?>"
required>
</div>
</div>

<div class="alert alert-info">
<strong>■■ Estado del año lectivo:</strong>
Para activar o desactivar este año lectivo, usa los botones
"Activar" o "Desactivar" en la tabla principal de años lectivos.
</div>

<div class="d-flex justify-content-between mt-4">
<a href="?action=academic" class="btn btn-secondary">
← Cancelar
</a>
<button type="submit" class="btn btn-success">
✓ Guardar Cambios
</button>
</div>
</form>
</div>
</div>
</div>
</div>
</div>

```



```
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"></script>
</body>
</html>
```

## 46. ArchImp/views/academic/enroll.php

```
<?php Security::requireLogin(); ?>
<!DOCTYPE html>
<html lang="es">
<head>
<meta charset="UTF-8">
<title>Matricular Estudiantes - EcuAsist</title>
<style>
* { margin: 0; padding: 0; box-sizing: border-box; }
body { font-family: Arial, sans-serif; background: #f4f4f4; }
.navbar { background: #007bff; color: white; padding: 15px 30px; display: flex; justify-content:
space-between; align-items: center; }
.navbar h1 { font-size: 24px; }
.navbar a { color: white; text-decoration: none; margin-left: 20px; }
.container { max-width: 1200px; margin: 30px auto; padding: 0 20px; }
.card { background: white; padding: 30px; border-radius: 8px; box-shadow: 0 2px 10px rgba(0,0,0,0.1);
margin-bottom: 20px; }
.form-group { margin-bottom: 20px; }
label { display: block; margin-bottom: 5px; font-weight: bold; color: #333; }
select { width: 100%; padding: 10px; border: 1px solid #ddd; border-radius: 4px; }
.student-list { margin-top: 20px; }
.student-item { padding: 10px; border: 1px solid #ddd; border-radius: 4px; margin-bottom: 10px; display:
flex; align-items: center; }
.student-item input[type="checkbox"] { margin-right: 10px; }
button { padding: 12px 30px; background: #28a745; color: white; border: none; border-radius: 4px; cursor:
pointer; font-size: 16px; margin-top: 20px; }
button:hover { background: #218838; }
.success { background: #d4edda; color: #155724; padding: 15px; border-radius: 4px; margin-bottom: 20px; }
.warning { background: #fff3cd; color: #856404; padding: 15px; border-radius: 4px; margin-bottom: 20px; }
.info { background: #d1ecf1; color: #0c5460; padding: 10px; border-radius: 4px; margin-bottom: 15px; }
table { width: 100%; border-collapse: collapse; margin-top: 15px; }
th, td { padding: 10px; text-align: left; border-bottom: 1px solid #ddd; }
th { background: #f8f9fa; font-weight: bold; }
.badge { display: inline-block; padding: 4px 8px; border-radius: 3px; font-size: 12px; background: #007bff;
color: white; }
</style>
</head>
<body>
<div class="navbar">
<h1>Matricular Estudiantes</h1>
<div>
<a href="?action=academic">← Configuración Académica</a>
<a href="?action=logout">Cerrar sesión</a>
</div>
</div>

<div class="container">
<?php if(isset($_GET['enrolled'])): ?>
<div class="success">✓ <?= $_GET['enrolled'] ?> estudiante(s) matriculado(s) correctamente</div>
<?php endif; ?>

<?php if(isset($_GET['errors']) && $_GET['errors'] > 0): ?>
<div class="warning">■ <?= $_GET['errors'] ?> estudiante(s) no pudieron ser matriculados (ya están en otro
curso)</div>
<?php endif; ?>

<?php if(isset($_GET['unenrolled'])): ?>
<div class="success">✓ Estudiante retirado del curso correctamente</div>
```

```

<?php endif; ?>

<?php if(isset($_GET['error']) && $_GET['error'] === 'not_enrolled'): ?>
<div class="warning">■ El estudiante no está matriculado en ningún curso</div>
<?php endif; ?>

<?php if(isset($_GET['error']) && $_GET['error'] === 'unenroll_failed'): ?>
<div class="warning">✗ Error al retirar el estudiante del curso</div>
<?php endif; ?>

<!-- Resumen de estudiantes -->
<div class="card">
<h2>Resumen</h2>
<div class="info">
<strong>Total estudiantes registrados:</strong> <?= count($allStudents) ?><br>
<strong>Ya matriculados:</strong> <?= count($allStudents) - count($availableStudents) ?><br>
<strong>Disponibles para matricular:</strong> <?= count($availableStudents) ?>
</div>
</div>

<!-- Formulario de matrícula -->
<?php if(count($availableStudents) > 0): ?>
<div class="card">
<h2>Matricular Estudiantes Disponibles</h2>
<p style="color: #666; margin-bottom: 20px;">
Solo se muestran estudiantes que aún NO están matriculados en ningún curso del año lectivo actual.
</p>

<form method="POST">
<div class="form-group">
<label>Curso</label>
<select name="course_id" required>
<option value="">Seleccionar curso...</option>
<?php foreach($courses as $course): ?>
<option value="<?= $course['id'] ?>">
<?= $course['name'] ?> - <?= $course['shift_name'] ?>
</option>
<?php endforeach; ?>
</select>
</div>

<div class="student-list">
<label>Seleccionar Estudiantes</label>
<?php foreach($availableStudents as $student): ?>
<div class="student-item">
<input type="checkbox" name="student_ids[]" value="<?= $student['id'] ?>" id="student_<?= $student['id'] ?>">
<label for="student_<?= $student['id'] ?>" style="margin: 0; font-weight: normal;">
<?= $student['last_name'] . ' ' . $student['first_name'] ?>
(<?= $student['dni'] ?> 'Sin cédula' ?>)
</label>
</div>
<?php endforeach; ?>
</div>

<button type="submit">Matricular Seleccionados</button>
</form>
</div>
<?php else: ?>
<div class="card">
<div class="warning">
■ No hay estudiantes disponibles para matricular. Todos los estudiantes ya están asignados a cursos.
</div>
</div>
<?php endif; ?>

```

```

<!-- Lista de todos los estudiantes con su estado -->
<div class="card">
<h2>Estado de Todos los Estudiantes</h2>
<table>

<thead>
<tr>
<th>#</th>
<th>Apellidos y Nombres</th>
<th>Cédula</th>
<th>Curso</th>
<th>Estado</th>
<th>Acciones</th>
</tr>
</thead>
<tbody>
<?php
$counter = 1;
foreach($allStudents as $student):
$course = $this->userModel->getStudentCourse($student['id'], $activeYear['id']);
?>
<tr>
<td><?= $counter++ ?></td>
<td><?= $student['last_name'] . ' ' . $student['first_name'] ?></td>
<td><?= $student['dni'] ?? '-' ?></td>
<td>
<?php if($course): ?>
<strong><?= $course['name'] ?></strong>
<br>
<small style="color: #666;"><?= $course['shift_name'] ?></small>
<?php else: ?>
<span style="color: #999;">-</span>
<?php endif; ?>
</td>
<td>
<?php if($course): ?>
<span class="badge" style="background: #28a745;">
✓ Matriculado
</span>
<?php else: ?>
<span class="badge" style="background: #6c757d;">Sin matricular</span>
<?php endif; ?>
</td>
<td>
<?php if($course): ?>
<form method="POST" action="?action=unenroll_student" style="display: inline;"
onsubmit="return confirmUnenroll(event, '<?= addslashes($student['last_name']) . ' ' .
$student['first_name']) ?>', '<?= addslashes($course['name']) ?>')">
<input type="hidden" name="student_id" value="<?= $student['id'] ?>">
<button type="submit"
style="padding: 6px 12px; background: #dc3545; font-size: 13px;">
✗ Retirar
</button>
</form>
<?php else: ?>
<span style="color: #999;">-</span>
<?php endif; ?>
</td>
</tr>
<?php endforeach; ?>
</tbody>
</table>
</div>
</div>

```

```

<script>

function confirmUnenroll(event, studentName, courseName) {
event.preventDefault();

const modal = document.createElement('div');
modal.style.cssText = 'position: fixed; top: 0; left: 0; width: 100%; height: 100%; background:
rgba(0,0,0,0.5); display: flex; align-items: center; justify-content: center; z-index: 9999;';

const modalContent = document.createElement('div');
modalContent.style.cssText = 'background: white; padding: 30px; border-radius: 8px; max-width: 500px;
box-shadow: 0 4px 20px rgba(0,0,0,0.3);';

modalContent.innerHTML = `
<h3 style="margin: 0 0 15px 0; color: #dc3545;">■ Retirar Estudiante del Curso</h3>
<p style="margin: 0 0 20px 0; color: #666;">
¿Está seguro de retirar a <strong>${studentName}</strong> del curso <strong>${courseName}</strong>?
</p>
<p style="margin: 0 0 20px 0; color: #666; font-size: 14px; background: #fff3cd; padding: 10px;
border-radius: 4px;">
<strong>■ Advertencia:</strong> Esta acción quitará al estudiante del curso.
Los registros de asistencia se conservarán pero el estudiante no aparecerá como matriculado.
</p>
<div style="display: flex; gap: 10px; justify-content: flex-end;">
<button type="button" id="cancelUnenrollBtn" style="padding: 10px 20px; background: #6c757d; color: white;
border: none; border-radius: 4px; cursor: pointer;">
Cancelar
</button>
<button type="button" id="confirmUnenrollBtn" style="padding: 10px 20px; background: #dc3545; color: white;
border: none; border-radius: 4px; cursor: pointer;">
Sí, Retirar
</button>
</div>
`;

modal.appendChild(modalContent);
document.body.appendChild(modal);

const form = event.target;

document.getElementById('confirmUnenrollBtn').onclick = function() {
document.body.removeChild(modal);
form.submit();
};

document.getElementById('cancelUnenrollBtn').onclick = function() {
document.body.removeChild(modal);
};

return false;
}
</script>
</body>
</html>

```

## 47. ArchImp/views/academic/index.php

```

<?php Security::requireLogin(); ?>
<!DOCTYPE html>
<html lang="es">
<head>
<meta charset="UTF-8">
<title>Configuración Académica - EcuAsist</title>

```

```

<style>
* { margin: 0; padding: 0; box-sizing: border-box; }
body { font-family: Arial, sans-serif; background: #f4f4f4; }
.navbar { background: #007bff; color: white; padding: 15px 30px; display: flex; justify-content: space-between; align-items: center; }
.navbar h1 { font-size: 24px; }
.navbar a { color: white; text-decoration: none; margin-left: 20px; }
.container { max-width: 1400px; margin: 30px auto; padding: 0 20px; }
.grid { display: grid; grid-template-columns: repeat(auto-fit, minmax(450px, 1fr)); gap: 20px; margin-bottom: 20px; }
.card { background: white; padding: 30px; border-radius: 8px; box-shadow: 0 2px 10px rgba(0,0,0,0.1); }
.form-group { margin-bottom: 15px; }
.error { background: #f8d7da; color: #721c24; padding: 10px; border-radius: 4px; margin-bottom: 20px; }
label { display: block; margin-bottom: 5px; font-weight: bold; color: #333; }
input, select { width: 100%; padding: 10px; border: 1px solid #ddd; border-radius: 4px; }
button { padding: 10px 20px; background: #28a745; color: white; border: none; border-radius: 4px; cursor: pointer; margin-top: 10px; }
button:hover { background: #218838; }
.btn-primary { background: #007bff; }
.btn-primary:hover { background: #0056b3; }
table { width: 100%; border-collapse: collapse; margin-top: 15px; }
th, td { padding: 10px; text-align: left; border-bottom: 1px solid #ddd; font-size: 14px; }
th { background: #f8f9fa; font-weight: bold; }
.success { background: #d4edda; color: #155724; padding: 10px; border-radius: 4px; margin-bottom: 20px; }
h2 { margin-bottom: 20px; color: #333; }
</style>
</head>
<body>
<?php include BASE_PATH . '/views/partials/navbar.php'; ?>

<div class="container">
<?php if(isset($_GET['course_success'])): ?>
<div class="success">✓ Curso creado correctamente</div>
<?php endif; ?>
<?php if(isset($_GET['course_updated'])): ?>
<div class="success">✓ Curso actualizado correctamente</div>
<?php endif; ?>
<?php if(isset($_GET['course_deleted'])): ?>
<div class="success">✓ Curso eliminado correctamente</div>
<?php endif; ?>

<?php if(isset($_GET['subject_success'])): ?>
<div class="success">✓ Asignatura creada correctamente</div>
<?php endif; ?>
<?php if(isset($_GET['subject_updated'])): ?>
<div class="success">✓ Asignatura actualizada correctamente</div>
<?php endif; ?>
<?php if(isset($_GET['subject_deleted'])): ?>
<div class="success">✓ Asignatura eliminada correctamente</div>
<?php endif; ?>

<?php if(isset($_GET['sy_created'])): ?>
<div class="success">✓ Año lectivo creado correctamente</div>
<?php endif; ?>
<?php if(isset($_GET['sy_updated'])): ?>
<div class="success">✓ Año lectivo actualizado correctamente</div>
<?php endif; ?>
<?php if(isset($_GET['sy_deleted'])): ?>
<div class="success">✓ Año lectivo eliminado correctamente</div>
<?php endif; ?>
<?php if(isset($_GET['sy_activated'])): ?>
<div class="success">✓ Año lectivo activado correctamente</div>
<?php endif; ?>
<?php if(isset($_GET['sy_deactivated'])): ?>
<div class="success">✓ Año lectivo desactivado correctamente</div>

```

```

<?php endif; ?>

<?php if(isset($_GET['error']) && $_GET['error'] === 'no_active_year'): ?>
<div class="error">✗ No hay un año lectivo activo</div>
<?php endif; ?>
<?php if(isset($_GET['error']) && $_GET['error'] === 'course_not_found'): ?>
<div class="error">✗ Curso no encontrado</div>
<?php endif; ?>
<?php if(isset($_GET['error']) && $_GET['error'] === 'course_has_students'): ?>
<div class="error">✗ No se puede eliminar el curso porque tiene estudiantes matriculados</div>
<?php endif; ?>
<?php if(isset($_GET['error']) && $_GET['error'] === 'course_has_assignments'): ?>
<div class="error">✗ No se puede eliminar el curso porque tiene asignaciones docentes</div>
<?php endif; ?>
<?php if(isset($_GET['error']) && $_GET['error'] === 'subject_not_found'): ?>
<div class="error">✗ Asignatura no encontrada</div>
<?php endif; ?>
<?php if(isset($_GET['error']) && $_GET['error'] === 'subject_has_assignments'): ?>
<div class="error">✗ No se puede eliminar la asignatura porque tiene asignaciones docentes</div>
<?php endif; ?>
<?php if(isset($_GET['error']) && $_GET['error'] === 'cannot_delete_active'): ?>
<div class="error">✗ No se puede eliminar el año lectivo activo</div>
<?php endif; ?>
<?php if(isset($_GET['error']) && $_GET['error'] === 'has_courses'): ?>
<div class="error">✗ No se puede eliminar el año lectivo porque tiene cursos asociados</div>
<?php endif; ?>
<?php if(isset($_GET['error']) && $_GET['error'] === 'year_not_found'): ?>
<div class="error">✗ Año lectivo no encontrado</div>
<?php endif; ?>

<!-- Años Lectivos -->
<div class="card" style="margin-bottom: 20px;">
<div style="display: flex; justify-content: space-between; align-items: center; margin-bottom: 20px;">
<h2 style="margin: 0;">■ Años Lectivos</h2>
<button onclick="location.href='?action=create_school_year'" class="btn-primary" style="margin: 0;">
+ Crear Año Lectivo
</button>
</div>

<table>
<thead>
<tr>
<th>#</th>
<th>Nombre</th>
<th>Fecha Inicio</th>
<th>Fecha Fin</th>
<th>Estado</th>
<th>Acciones</th>
</tr>
</thead>
<tbody>
<?php if(empty($schoolYears)): ?>

<tr>
<td colspan="6" style="text-align:center; color: #999;">No hay años lectivos registrados</td>
</tr>
<?php else: ?>
<?php $counter = 1; foreach($schoolYears as $year): ?>
<tr style="background: #e7f3ff; : '' ?>">
<td>#<?php $counter++ ?></td>
<td><strong><?php htmlspecialchars($year['name']) ?></strong></td>
<td><?php date('d/m/Y', strtotime($year['start_date'])) ?></td>
<td><?php date('d/m/Y', strtotime($year['end_date'])) ?></td>
<td>
<?php if($year['is_active']): ?>
<span style="background: #28a745; color: white; padding: 4px 10px; border-radius: 12px; font-size: 12px;

```

```

font-weight: bold;">
✓ ACTIVO
</span>
<?php else: ?>
<span style="background: #6c757d; color: white; padding: 4px 10px; border-radius: 12px; font-size: 12px;">
Inactivo
</span>
<?php endif: ?>
</td>

<td style="white-space: nowrap;">
<button onclick="location.href='?action=edit_school_year&id=<?= $year['id'] ?>'"
style="padding: 5px 10px; font-size: 12px; background: #ffc107; color: #212529;">
—■ Editar
</button>

<?php if($year['is_active']): ?>
<form method="POST" action="?action=deactivate_school_year" style="display: inline;"
onsubmit="return confirm('¿Desactivar este año lectivo?')">
<input type="hidden" name="year_id" value="<?= $year['id'] ?>">
<button type="submit" style="padding: 5px 10px; font-size: 12px; background: #6c757d;">
■ Desactivar
</button>
</form>
<?php else: ?>
<form method="POST" action="?action=activate_school_year" style="display: inline;"
onsubmit="return confirm('¿Activar este año lectivo? Se desactivarán los demás.')">
<input type="hidden" name="year_id" value="<?= $year['id'] ?>">
<button type="submit" style="padding: 5px 10px; font-size: 12px; background: #28a745;">
✓ Activar
</button>
</form>

<form method="POST" action="?action=delete_school_year" style="display: inline;"
onsubmit="return confirmDelete(event, '¿?>= htmlspecialchars($year['name']) ?>')">
<input type="hidden" name="year_id" value="<?= $year['id'] ?>">
<button type="submit" style="padding: 5px 10px; font-size: 12px; background: #dc3545;">
■■ Eliminar
</button>
</form>
<?php endif: ?>
</td>
</tr>
<?php endforeach: ?>
<?php endif: ?>
</tbody>
</table>
</div>

<div class="grid">
<!-- Crear Curso -->
<div class="card">
<h2>Crear Curso</h2>
<form method="POST" action="?action=create_course" id="courseForm">
<div class="form-group">
<label>Nivel</label>
<select name="grade_level" id="grade_level" required>
<option value="">Seleccionar...</option>
<option value="8vo EGB">8vo EGB</option>
<option value="9no EGB">9no EGB</option>
<option value="10mo EGB">10mo EGB</option>
<option value="1ro BGU">1ro BGU</option>
<option value="2do BGU">2do BGU</option>
<option value="3ro BGU">3ro BGU</option>
<option value="1ro Técnico">1ro Técnico</option>
<option value="2do Técnico">2do Técnico</option>
<option value="3ro Técnico">3ro Técnico</option>

```

```

</select>
</div>
<div class="form-group">
<label>Paralelo</label>
<select name="parallel" id="parallel" required>
<option value="">Seleccionar...</option>
<?php foreach(range('A', 'Z') as $letter): ?>
<option value="<?= $letter ?>"><?= $letter ?></option>
<?php endforeach; ?>
</select>
</div>
<div class="form-group">
<label>Jornada</label>
<select name="shift_id" id="shift_id" required>
<option value="">Seleccionar...</option>
<?php foreach($shifts as $shift): ?>
<option value="<?= $shift['id'] ?>" data-shift="<?= $shift['name'] ?>">
<?= ucfirst($shift['name']) ?>
</option>
<?php endforeach; ?>
</select>
</div>
<div class="form-group">
<label>Nombre del Curso (generado automáticamente)</label>
<input type="text" name="name" id="course_name" readonly style="background: #f0f0f0;">
</div>
<button type="submit">Crear Curso</button>
</form>
</div>

<script>
function generateCourseName() {
const level = document.getElementById('grade_level').value;
const parallel = document.getElementById('parallel').value;
const shiftSelect = document.getElementById('shift_id');
const shiftOption = shiftSelect.options[shiftSelect.selectedIndex];
const shiftName = shiftOption ? shiftOption.getAttribute('data-shift') : '';

if (level && parallel && shiftName) {
const courseName = `${level} ${parallel} - ${shiftName.charAt(0).toUpperCase() + shiftName.slice(1)}`;
document.getElementById('course_name').value = courseName;
}
}

document.getElementById('grade_level').addEventListener('change', generateCourseName);
document.getElementById('parallel').addEventListener('change', generateCourseName);
document.getElementById('shift_id').addEventListener('change', generateCourseName);
</script>

<!-- Crear Asignatura -->
<div class="card">
<h2>Crear Asignatura</h2>
<form method="POST" action="?action=create_subject">
<div class="form-group">
<label>Nombre</label>
<input type="text" name="name" placeholder="Ej: Matemáticas" required>
</div>
<div class="form-group">
<label>Código</label>
<input type="text" name="code" placeholder="Ej: MAT" required>
</div>
<button type="submit">Crear Asignatura</button>
</form>
</div>
</div>

```



```

<!-- Lista de Cursos -->
<div class="card">
<h2>Cursos Registrados</h2>
<table>
<thead>
<tr>
<th>ID</th>
<th>Nombre</th>
<th>Nivel</th>
<th>Paralelo</th>
<th>Jornada</th>
<th>Acciones</th>
</tr>
</thead>
<tbody>
<?php foreach($courses as $course): ?>
<tr>
<td><?= $course['id'] ?></td>
<td><?= $course['name'] ?></td>
<td><?= $course['grade_level'] ?></td>
<td><?= $course['parallel'] ?></td>
<td><?= ucfirst($course['shift_name']) ?></td>
<td style="white-space: nowrap;">
<button onclick="location.href='?action=view_course_students&course_id=<?= $course['id'] ?>'"
style="padding: 5px 10px; font-size: 12px; background: #007bff;">
■ Estudiantes
</button>
<button onclick="location.href='?action=edit_course&id=<?= $course['id'] ?>'"
style="padding: 5px 10px; font-size: 12px; background: #ffc107; color: #000;">
=■ Editar
</button>
<form method="POST" action="?action=delete_course" style="display: inline;"
onsubmit="return confirmDeleteCourse(event, '<?= addslashes($course['name']) ?>')">
<input type="hidden" name="course_id" value="<?= $course['id'] ?>">
<button type="submit" style="padding: 5px 10px; font-size: 12px; background: #dc3545;">
■■ Eliminar
</button>
</form>
</td>
</tr>
<?php endforeach; ?>
</tbody>
</table>
<button onclick="location.href='?action=enroll_students'" class="btn-primary" style="margin-top: 15px;">
Matricular Estudiantes
</button>
</div>

<!-- Lista de Asignaturas -->
<div class="card">
<h2>Asignaturas Registradas</h2>
<table>
<thead>
<tr>
<th>ID</th>
<th>Código</th>
<th>Nombre</th>
<th>Acciones</th>
</tr>
</thead>
<tbody>
<?php foreach($subjects as $subject): ?>
<tr>
<td><?= $subject['id'] ?></td>

```

```

<td><?= $subject['code'] ?></td>
<td><?= $subject['name'] ?></td>
<td style="white-space: nowrap;">
<button onclick="location.href='?action=edit_subject&id=<?= $subject['id'] ?>'"
style="padding: 5px 10px; font-size: 12px; background: #ffc107; color: #000;">
=■ Editar
</button>
<form method="POST" action="?action=delete_subject" style="display: inline;"
onsubmit="return confirmDeleteSubject(event, '<?= addslashes($subject['name']) ?>')">
<input type="hidden" name="subject_id" value="<?= $subject['id'] ?>">
<button type="submit" style="padding: 5px 10px; font-size: 12px; background: #dc3545;">
■■ Eliminar
</button>
</form>
</td>
</tr>
<?php endforeach; ?>
</tbody>
</table>
</div>
</div>

<script>
function confirmDeleteCourse(event, courseName) {
event.preventDefault();

const modal = document.createElement('div');
modal.style.cssText = 'position: fixed; top: 0; left: 0; width: 100%; height: 100%; background:
rgba(0,0,0,0.5); display: flex; align-items: center; justify-content: center; z-index: 9999;';

const modalContent = document.createElement('div');
modalContent.style.cssText = 'background: white; padding: 30px; border-radius: 8px; max-width: 500px;
box-shadow: 0 4px 20px rgba(0,0,0,0.3);';

modalContent.innerHTML = `
<h3 style="margin: 0 0 15px 0; color: #dc3545;">■■ Eliminar Curso</h3>
<p style="margin: 0 0 20px 0; color: #666;">
¿Está seguro de eliminar el curso <strong>${courseName}</strong>?
</p>
<p style="margin: 0 0 20px 0; color: #666; font-size: 14px;">
<strong>Nota:</strong> No se puede eliminar si tiene estudiantes o asignaciones docentes.
</p>
<div style="display: flex; gap: 10px; justify-content: flex-end;">
<button type="button" id="cancelDeleteBtn" style="padding: 10px 20px; background: #6c757d; color: white;
border: none; border-radius: 4px; cursor: pointer;">
Cancelar
</button>
<button type="button" id="confirmDeleteBtn" style="padding: 10px 20px; background: #dc3545; color: white;
border: none; border-radius: 4px; cursor: pointer;">
Sí, Eliminar
</button>
</div>
`;

modal.appendChild(modalContent);
document.body.appendChild(modal);

const form = event.target;

document.getElementById('confirmDeleteBtn').onclick = function() {
document.body.removeChild(modal);
form.submit();
};

document.getElementById('cancelDeleteBtn').onclick = function() {

```

```

document.body.removeChild(modal);
};

return false;
}

function confirmDeleteSubject(event, subjectName) {
event.preventDefault();

const modal = document.createElement('div');
modal.style.cssText = 'position: fixed; top: 0; left: 0; width: 100%; height: 100%; background:
rgba(0,0,0,0.5); display: flex; align-items: center; justify-content: center; z-index: 9999;';

const modalContent = document.createElement('div');
modalContent.style.cssText = 'background: white; padding: 30px; border-radius: 8px; max-width: 500px;
box-shadow: 0 4px 20px rgba(0,0,0,0.3);';

modalContent.innerHTML = `
<h3 style="margin: 0 0 15px 0; color: #dc3545;">■ ■ ■ Eliminar Asignatura</h3>
<p style="margin: 0 0 20px 0; color: #666;">
¿Está seguro de eliminar la asignatura <strong>${subjectName}</strong>?
</p>
<p style="margin: 0 0 20px 0; color: #666; font-size: 14px;">
<strong>Nota:</strong> No se puede eliminar si tiene asignaciones docentes.
</p>
<div style="display: flex; gap: 10px; justify-content: flex-end;">
<button type="button" id="cancelDeleteSubBtn" style="padding: 10px 20px; background: #6c757d; color: white;
border: none; border-radius: 4px; cursor: pointer;">
Cancelar
</button>
<button type="button" id="confirmDeleteSubBtn" style="padding: 10px 20px; background: #dc3545; color: white;
border: none; border-radius: 4px; cursor: pointer;">
Sí, Eliminar
</button>
</div>
`;

modal.appendChild(modalContent);
document.body.appendChild(modal);

const form = event.target;

document.getElementById('confirmDeleteSubBtn').onclick = function() {
document.body.removeChild(modal);
form.submit();
};

document.getElementById('cancelDeleteSubBtn').onclick = function() {
document.body.removeChild(modal);
};

return false;
}

function confirmDelete(event, yearName) {
event.preventDefault();

const modal = document.createElement('div');
modal.style.cssText = 'position: fixed; top: 0; left: 0; width: 100%; height: 100%; background:
rgba(0,0,0,0.5); display: flex; align-items: center; justify-content: center; z-index: 9999;';

const modalContent = document.createElement('div');
modalContent.style.cssText = 'background: white; padding: 30px; border-radius: 8px; max-width: 500px;
box-shadow: 0 4px 20px rgba(0,0,0,0.3);';

```

```

modalContent.innerHTML = `
<h3 style="margin: 0 0 15px 0; color: #dc3545;">■ ■ ■ Eliminar Año Lectivo</h3>
<p style="margin: 0 0 20px 0; color: #666;">
¿Está seguro de eliminar el año lectivo <strong>${yearName}</strong>?
</p>
<p style="margin: 0 0 20px 0; color: #666; font-size: 14px;">
<strong>Nota:</strong> No se puede eliminar si tiene cursos asociados.
</p>
<div style="display: flex; gap: 10px; justify-content: flex-end;">
<button type="button" id="cancelDeleteBtn" style="padding: 10px 20px; background: #6c757d; color: white;
border: none; border-radius: 4px; cursor: pointer;">
Cancelar
</button>
<button type="button" id="confirmDeleteBtn" style="padding: 10px 20px; background: #dc3545; color: white;
border: none; border-radius: 4px; cursor: pointer;">
Sí, Eliminar
</button>
</div>
`;

modal.appendChild(modalContent);
document.body.appendChild(modal);

const form = event.target;

document.getElementById('confirmDeleteBtn').onclick = function() {
document.body.removeChild(modal);
form.submit();
};

document.getElementById('cancelDeleteBtn').onclick = function() {
document.body.removeChild(modal);
};

return false;
}
</script>
</body>
</html>

```

## 48. ArchImp/views/academic/subject\_edit.php

```

<?php
require_once BASE_PATH . '/helpers/Security.php';
Security::requireLogin();
if (!Security::hasRole('autoridad')) {
die('Acceso denegado');
}
?>
<!DOCTYPE html>
<html lang="es">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Editar Asignatura - EcuAsist</title>
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet">
<style>
body { padding-top: 70px; }
</style>
</head>
<body>
<?php include BASE_PATH . '/views/partials/navbar.php'; ?>

```

```

<div class="container">
<div class="row justify-content-center">
<div class="col-md-8">
<div class="card shadow">
<div class="card-header bg-primary text-white">
<h4 class="mb-0">■ Editor Asignatura: <?= htmlspecialchars($subject['name']) ?></h4>
</div>
<div class="card-body">
<?php if (!empty($errors)): ?>
<div class="alert alert-danger">
<?php foreach ($errors as $error): ?>
<div><?= htmlspecialchars($error) ?></div>
<?php endforeach; ?>
</div>
<?php endif; ?>

<form method="POST">
<div class="mb-3">
<label class="form-label">Nombre *</label>
<input type="text"
name="name"
class="form-control"
placeholder="Ej: Matemáticas"
value="<?= htmlspecialchars($subject['name']) ?>"
required>
</div>

<div class="mb-3">
<label class="form-label">Código *</label>
<input type="text"
name="code"
class="form-control"
placeholder="Ej: MAT"
value="<?= htmlspecialchars($subject['code']) ?>"
required>
<small class="text-muted">Código corto para identificar la asignatura</small>
</div>

<div class="d-flex justify-content-between mt-4">
<a href="?action=academic" class="btn btn-secondary">← Cancelar</a>
<button type="submit" class="btn btn-success">✓ Guardar Cambios</button>
</div>
</form>
</div>
</div>
</div>
</div>
</div>

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"></script>
</body>
</html>

```

## 49. ArchImp/views/assignments/index.php

```

<?php Security::requireLogin(); ?>
<!DOCTYPE html>
<html lang="es">
<head>
<meta charset="UTF-8">
<title>Asignaciones Docentes - EcuAsist</title>

```

```

<style>
* { margin: 0; padding: 0; box-sizing: border-box; }
body { font-family: Arial, sans-serif; background: #f4f4f4; }
.container { max-width: 1400px; margin: 30px auto; padding: 0 20px; }
.card { background: white; padding: 30px; border-radius: 8px; box-shadow: 0 2px 10px rgba(0,0,0,0.1);
margin-bottom: 20px; }
.form-group { margin-bottom: 15px; }
.error { background: #f8d7da; color: #721c24; padding: 10px; border-radius: 4px; margin-bottom: 20px; }
label { display: block; margin-bottom: 5px; font-weight: bold; color: #333; }
select { width: 100%; padding: 10px; border: 1px solid #ddd; border-radius: 4px; }
button { padding: 10px 20px; background: #28a745; color: white; border: none; border-radius: 4px; cursor:
pointer; }
button:hover { background: #218838; }
.btn-danger { background: #dc3545; padding: 5px 10px; font-size: 12px; }
.btn-danger:hover { background: #c82333; }
table { width: 100%; border-collapse: collapse; margin-top: 15px; }
th, td { padding: 10px; text-align: left; border-bottom: 1px solid #ddd; font-size: 14px; }
th { background: #f8f9fa; font-weight: bold; }
.success { background: #d4edda; color: #155724; padding: 10px; border-radius: 4px; margin-bottom: 20px; }
.badge { padding: 4px 8px; border-radius: 3px; font-size: 11px; background: #ffc107; color: #333; }
h2 { margin-bottom: 20px; color: #333; }
</style>
</head>
<body>
<?php include BASE_PATH . '/views/partials/navbar.php'; ?>

<div class="container">
<?php if(isset($_GET['success'])): ?>
<div class="success">✓ Asignación creada correctamente</div>
<?php endif; ?>
<?php if(isset($_GET['error'])): ?>
<div class="error">✗ <?= htmlspecialchars($_GET['error']) ?></div>
<?php endif; ?>
<?php if(isset($_GET['removed'])): ?>
<div class="success">✓ Asignación eliminada</div>
<?php endif; ?>

<div class="card">
<h2>Asignar Docente a Materia</h2>
<form method="POST" action="?action=create_assignment">
<div class="form-group">
<label>Docente</label>
<select name="teacher_id" required>
<option value="">Seleccionar...</option>
<?php foreach($teachers as $teacher): ?>
<option value="<?= $teacher['id'] ?>">
<?= $teacher['last_name'] . ' ' . $teacher['first_name'] ?>
</option>
<?php endforeach; ?>
</select>
</div>

<div class="form-group">
<label>Curso</label>
<select name="course_id" required>
<option value="">Seleccionar...</option>
<?php foreach($courses as $course): ?>
<option value="<?= $course['id'] ?>">
<?= $course['name'] ?> - <?= $course['shift_name'] ?>
</option>
<?php endforeach; ?>
</select>
</div>

<div class="form-group">

```

```

<label>Asignatura</label>
<select name="subject_id" required>
<option value="">Seleccionar...</option>
<?php foreach($subjects as $subject): ?>
<option value="<?= $subject['id'] ?>"><?= $subject['name'] ?></option>
<?php endforeach; ?>
</select>
</div>

<button type="submit">Asignar</button>
</form>
</div>

<div class="card">
<h2>Asignaciones Docente-Materia</h2>

<!-- Filtros -->
<form method="GET" action="" style="margin-bottom: 20px;">
<input type="hidden" name="action" value="assignments">
<div style="display: grid; grid-template-columns: repeat(3, 1fr) auto; gap: 15px; align-items: flex-end;">
<div class="form-group" style="margin-bottom: 0;">
<label>Filtrar por Curso</label>
<select name="filter_course" onchange="this.form.submit()">
<option value="">Todos los cursos</option>
<?php foreach($courses as $course): ?>
<option value="<?= $course['id'] ?>" <?= (isset($_GET['filter_course']) && $_GET['filter_course'] ==
$course['id']) ? 'selected' : '' ?>>
<?= $course['name'] ?>
</option>
<?php endforeach; ?>
</select>
</div>

<div class="form-group" style="margin-bottom: 0;">
<label>Filtrar por Asignatura</label>
<select name="filter_subject" onchange="this.form.submit()">
<option value="">Todas las asignaturas</option>
<?php foreach($subjects as $subject): ?>
<option value="<?= $subject['id'] ?>" <?= (isset($_GET['filter_subject']) && $_GET['filter_subject'] ==
$subject['id']) ? 'selected' : '' ?>>
<?= $subject['name'] ?>
</option>
<?php endforeach; ?>
</select>
</div>

<div class="form-group" style="margin-bottom: 0;">
<label>Filtrar por Docente</label>
<select name="filter_teacher" onchange="this.form.submit()">
<option value="">Todos los docentes</option>
<?php foreach($teachers as $teacher): ?>
<option value="<?= $teacher['id'] ?>" <?= (isset($_GET['filter_teacher']) && $_GET['filter_teacher'] ==
$teacher['id']) ? 'selected' : '' ?>>
<?= $teacher['last_name'] . ' ' . $teacher['first_name'] ?>
</option>
<?php endforeach; ?>
</select>
</div>

<?php if(isset($_GET['filter_course']) || isset($_GET['filter_subject']) || isset($_GET['filter_teacher'])):
?>
<div>
<a href="?action=assignments" style="padding: 10px 15px; background: #6c757d; color: white; text-decoration:
none; border-radius: 4px; display: inline-block;">
Limpiar

```

```

</a>
</div>
<?php endif; ?>
</div>
</form>

<table>
<thead>
<tr>
<th>Curso</th>
<th>Asignatura</th>
<th>Docente</th>
<th>Acción</th>
</tr>
</thead>
<tbody>
<?php
// Aplicar filtros
$filteredAssignments = $assignments;

if(isset($_GET['filter_course']) && $_GET['filter_course'] != '') {
$filterCourseId = (int)$_GET['filter_course'];
$courseName = '';
foreach($courses as $c) {
if($c['id'] == $filterCourseId) {
$courseName = $c['name'];
break;
}
}
$filteredAssignments = array_filter($filteredAssignments, function($a) use ($courseName) {
return $a['course_name'] == $courseName;
});
}

if(isset($_GET['filter_subject']) && $_GET['filter_subject'] != '') {
$filterSubjectId = (int)$_GET['filter_subject'];
$subjectName = '';
foreach($subjects as $s) {
if($s['id'] == $filterSubjectId) {
$subjectName = $s['name'];
break;
}
}
$filteredAssignments = array_filter($filteredAssignments, function($a) use ($subjectName) {
return $a['subject_name'] == $subjectName;
});
}

if(isset($_GET['filter_teacher']) && $_GET['filter_teacher'] != '') {
$filterTeacherId = (int)$_GET['filter_teacher'];
$teacherName = '';

foreach($teachers as $t) {
if($t['id'] == $filterTeacherId) {
$teacherName = $t['last_name'] . ' ' . $t['first_name'];
break;
}
}
$filteredAssignments = array_filter($filteredAssignments, function($a) use ($teacherName) {
return $a['teacher_name'] == $teacherName;
});
}
?>

<?php if(empty($filteredAssignments)): ?>
<tr>

```



```

<td colspan="4" style="text-align:center;">No hay asignaciones que coincidan con los filtros</td>
</tr>
<?php else: ?>
<?php foreach($filteredAssignments as $assignment): ?>
<tr>
<td><?= $assignment['course_name'] ?></td>
<td><?= $assignment['subject_name'] ?></td>
<td><?= $assignment['teacher_name'] ?></td>
<td>
<form method="POST" action="?action=remove_assignment" style="display: inline;">
<input type="hidden" name="assignment_id" value="<?= $assignment['id'] ?>">
<button type="submit" class="btn-danger" onclick="confirmDeleteAssignment(event, '<?=
$assignment['teacher_name'] ?>', '<?= $assignment['course_name'] ?>', '<?= $assignment['subject_name']
?>')">
Eliminar
</button>
</form>
</td>
</tr>
<?php endforeach; ?>
<?php endif; ?>
</tbody>
</table>
</div>
</div>

<script>
function confirmDeleteAssignment(event, teacherName, courseName, subjectName) {
event.preventDefault();

const modal = document.createElement('div');
modal.style.cssText = 'position: fixed; top: 0; left: 0; width: 100%; height: 100%; background:
rgba(0,0,0,0.5); display: flex; align-items: center; justify-content: center; z-index: 9999;';

const modalContent = document.createElement('div');
modalContent.style.cssText = 'background: white; padding: 30px; border-radius: 8px; max-width: 500px;
box-shadow: 0 4px 20px rgba(0,0,0,0.3);';

modalContent.innerHTML = `
<h3 style="margin: 0 0 15px 0; color: #dc3545;">■ ■ ■ Eliminar Asignación</h3>
<p style="margin: 0 0 20px 0; color: #666;">
¿Está seguro de eliminar esta asignación?<br><br>
<strong>Docente:</strong> ${teacherName}<br>
<strong>Curso:</strong> ${courseName}<br>
<strong>Asignatura:</strong> ${subjectName}
</p>
<div style="display: flex; gap: 10px; justify-content: flex-end;">
<button type="button" id="cancelDeleteBtn" style="padding: 10px 20px; background: #6c757d; color: white;
border: none; border-radius: 4px; cursor: pointer;">
Cancelar
</button>

<button type="button" id="confirmDeleteBtn" style="padding: 10px 20px; background: #dc3545; color: white;
border: none; border-radius: 4px; cursor: pointer;">
Sí, Eliminar
</button>
</div>
`;

modal.appendChild(modalContent);
document.body.appendChild(modal);

const form = event.target.closest('form');

document.getElementById('confirmDeleteBtn').onclick = function() {
document.body.removeChild(modal);

```

```

form.submit();
};

document.getElementById('cancelDeleteBtn').onclick = function() {
document.body.removeChild(modal);
};

return false;
}
</script>
</body>
</html>

```

## 50. ArchImp/views/assignments/tutor.php

```

<?php Security::requireLogin(); ?>
<!DOCTYPE html>
<html lang="es">
<head>
<meta charset="UTF-8">
<title>Asignar Docente Tutor - EcuAsist</title>
<style>
* { margin: 0; padding: 0; box-sizing: border-box; }
body { font-family: Arial, sans-serif; background: #f4f4f4; }
.container { max-width: 1200px; margin: 30px auto; padding: 0 20px; }
.card { background: white; padding: 30px; border-radius: 8px; box-shadow: 0 2px 10px rgba(0,0,0,0.1);
margin-bottom: 20px; }
.form-group { margin-bottom: 15px; }
.error { background: #f8d7da; color: #721c24; padding: 10px; border-radius: 4px; margin-bottom: 20px; }
label { display: block; margin-bottom: 5px; font-weight: bold; color: #333; }
select { width: 100%; padding: 10px; border: 1px solid #ddd; border-radius: 4px; }
button { padding: 10px 20px; background: #007bff; color: white; border: none; border-radius: 4px; cursor:
pointer; }
button:hover { background: #0056b3; }
.btn-danger { background: #dc3545; }
.btn-danger:hover { background: #c82333; }
.success { background: #d4edda; color: #155724; padding: 10px; border-radius: 4px; margin-bottom: 20px; }
table { width: 100%; border-collapse: collapse; margin-top: 15px; }
th, td { padding: 10px; text-align: left; border-bottom: 1px solid #ddd; font-size: 14px; }
th { background: #f8f9fa; font-weight: bold; }
h2 { margin-bottom: 20px; color: #333; }
</style>
</head>
<body>
<?php include BASE_PATH . '/views/partials/navbar.php'; ?>
<div class="container">
<?php if(isset($_GET['tutor_success'])): ?>
<div class="success">✓ Tutor asignado correctamente</div>
<?php endif; ?>
<?php if(isset($_GET['tutor_error'])): ?>
<div class="error">✗ <?= htmlspecialchars($_GET['tutor_error']) ?></div>
<?php endif; ?>
<?php if(isset($_GET['tutor_removed'])): ?>
<div class="success">✓ Tutor eliminado correctamente</div>
<?php endif; ?>

<div class="card">
<h2>Asignar Docente Tutor</h2>
<form method="POST" action="?action=set_tutor" id="tutorForm">
<div class="form-group">
<label>Curso</label>
<select name="course_id" id="course_tutor" required>
<option value="">Seleccionar...</option>

```

```

<?php foreach($courses as $course): ?>
<option value="<? $course['id'] ?>">
<? $course['name'] ?> - <? $course['shift_name'] ?>
</option>
<?php endforeach; ?>
</select>
</div>

<div class="form-group">
<label>Docente Tutor</label>
<select name="teacher_id" id="teacher_tutor" required>
<option value="">Primero seleccione un curso...</option>
</select>
</div>

<button type="submit">Asignar Tutor</button>
</form>

<hr style="margin: 30px 0; border: none; border-top: 1px solid #ddd;">

</div>

<div class="card">
<h2>Gestión de Tutores por Curso</h2>
<table>
<thead>
<tr>
<th>Curso</th>
<th>Docente Tutor</th>
<th>Acción</th>
</tr>
</thead>
<tbody>
<?php foreach($courses as $course): ?>
<?php
// Buscar si el curso tiene tutor
$courseTutor = null;
foreach($assignments as $assignment) {
if ($assignment['course_name'] == $course['name'] && $assignment['is_tutor']) {
$courseTutor = $assignment;
break;
}
}
?>
<tr>
<td><? $course['name'] ?></td>
<td>
<?php if($courseTutor): ?>
<? $courseTutor['teacher_name'] ?>
<?php else: ?>
<em style="color: #999;">Sin tutor asignado</em>
<?php endif; ?>
</td>
<td>
<?php if($courseTutor): ?>
<form method="POST" action="?action=remove_tutor" style="display: inline;" onsubmit="return
confirmRemoveTutor(event, '<? $courseTutor['teacher_name'] ?>', '<? $course['name'] ?>')">
<input type="hidden" name="course_id" value="<? $course['id'] ?>">
<button type="submit" class="btn-danger" style="padding: 5px 10px; font-size: 12px;">
× Quitar
</button>
</form>
<?php else: ?>
-
<?php endif; ?>

```

```

</td>
</tr>
<?php endforeach; ?>
</tbody>
</table>
</div>
</div>
</div>

<script>
let currentTutor = null;

document.getElementById('course_tutor').addEventListener('change', function() {
const courseId = this.value;
const teacherSelect = document.getElementById('teacher_tutor');
currentTutor = null;

teacherSelect.innerHTML = '<option value="">Cargando...</option>';

if (!courseId) {
teacherSelect.innerHTML = '<option value="">Primero seleccione un curso...</option>';
return;
}

fetch('?action=check_course_tutor&course_id=' + courseId)
.then(response => response.json())
.then(tutor => {
if (tutor && tutor.tutor_name) {
currentTutor = tutor.tutor_name;
}
});

fetch('?action=get_course_teachers&course_id=' + courseId)
.then(response => response.json())
.then(data => {
if (data.length === 0) {
teacherSelect.innerHTML = '<option value="">No hay docentes asignados a este curso</option>';
} else {
teacherSelect.innerHTML = '<option value="">Seleccionar...</option>';
data.forEach(teacher => {
const option = document.createElement('option');
option.value = teacher.teacher_id;
option.textContent = teacher.teacher_name;
teacherSelect.appendChild(option);
});
}
})
.catch(error => {
teacherSelect.innerHTML = '<option value="">Error al cargar docentes</option>';
console.error('Error:', error);
});

document.getElementById('tutorForm').addEventListener('submit', function(e) {
if (currentTutor) {
e.preventDefault();

const modal = document.createElement('div');
modal.style.cssText = 'position: fixed; top: 0; left: 0; width: 100%; height: 100%; background:
rgba(0,0,0,0.5); display: flex; align-items: center; justify-content: center; z-index: 9999;';

const modalContent = document.createElement('div');
modalContent.style.cssText = 'background: white; padding: 30px; border-radius: 8px; max-width: 500px;
box-shadow: 0 4px 20px rgba(0,0,0,0.3);';

```

```

modalContent.innerHTML = `
<h3 style="margin: 0 0 15px 0; color: #333;">■ Confirmar Cambio de Tutor</h3>
<p style="margin: 0 0 20px 0; color: #666;">
El curso ya tiene como tutor a: <strong>${currentTutor}</strong>.<br>
¿Desea reemplazarlo por el nuevo docente?
</p>
<div style="display: flex; gap: 10px; justify-content: flex-end;">
<button type="button" id="cancelBtn" style="padding: 10px 20px; background: #6c757d; color: white; border:
none; border-radius: 4px; cursor: pointer;">
Cancelar
</button>
<button type="button" id="confirmBtn" style="padding: 10px 20px; background: #007bff; color: white; border:
none; border-radius: 4px; cursor: pointer;">
Sí, Reemplazar
</button>
</div>
`;

modal.appendChild(modalContent);
document.body.appendChild(modal);

const form = this;

document.getElementById('confirmBtn').onclick = function() {
document.body.removeChild(modal);
currentTutor = null;
form.submit();
};

document.getElementById('cancelBtn').onclick = function() {
document.body.removeChild(modal);
};
});

function confirmRemoveTutor(event, tutorName, courseName) {
event.preventDefault();

const modal = document.createElement('div');
modal.style.cssText = 'position: fixed; top: 0; left: 0; width: 100%; height: 100%; background:
rgba(0,0,0,0.5); display: flex; align-items: center; justify-content: center; z-index: 9999;';

const modalContent = document.createElement('div');
modalContent.style.cssText = 'background: white; padding: 30px; border-radius: 8px; max-width: 500px;
box-shadow: 0 4px 20px rgba(0,0,0,0.3);';

modalContent.innerHTML = `
<h3 style="margin: 0 0 15px 0; color: #dc3545;">■ Quitar Tutor</h3>
<p style="margin: 0 0 20px 0; color: #666;">
¿Está seguro de quitar a <strong>${tutorName}</strong> como tutor del curso <strong>${courseName}</strong>?
</p>
<div style="display: flex; gap: 10px; justify-content: flex-end;">
<button type="button" id="cancelRemoveBtn" style="padding: 10px 20px; background: #6c757d; color: white;
border: none; border-radius: 4px; cursor: pointer;">
Cancelar
</button>
<button type="button" id="confirmRemoveBtn" style="padding: 10px 20px; background: #dc3545; color: white;
border: none; border-radius: 4px; cursor: pointer;">
Sí, Quitar
</button>
</div>
`;

modal.appendChild(modalContent);
document.body.appendChild(modal);

```

```

const form = event.target;

document.getElementById('confirmRemoveBtn').onclick = function() {
document.body.removeChild(modal);
form.submit();
};

document.getElementById('cancelRemoveBtn').onclick = function() {
document.body.removeChild(modal);
};

return false;
}
</script>
</body>
</html>

```

## 51. ArchImp/views/assignments/view\_course.php

```

<?php Security::requireLogin(); ?>
<!DOCTYPE html>
<html lang="es">
<head>
<meta charset="UTF-8">
<title>Asignaciones del Curso - EcuAsist</title>
<style>
* { margin: 0; padding: 0; box-sizing: border-box; }
body { font-family: Arial, sans-serif; background: #f4f4f4; }
.navbar { background: #007bff; color: white; padding: 15px 30px; display: flex; justify-content:
space-between; align-items: center; }
.navbar h1 { font-size: 24px; }
.navbar a { color: white; text-decoration: none; margin-left: 20px; }
.container { max-width: 1000px; margin: 30px auto; padding: 0 20px; }
.card { background: white; padding: 30px; border-radius: 8px; box-shadow: 0 2px 10px rgba(0,0,0,0.1);
margin-bottom: 20px; }
table { width: 100%; border-collapse: collapse; margin-top: 15px; }
th, td { padding: 12px; text-align: left; border-bottom: 1px solid #ddd; }
th { background: #f8f9fa; font-weight: bold; }
.course-info { background: #e7f3ff; padding: 15px; border-radius: 4px; margin-bottom: 20px; }
.tutor-box { background: #fff3cd; padding: 15px; border-radius: 4px; border-left: 4px solid #ffc107;
margin-bottom: 20px; }
h2, h3 { color: #333; margin-bottom: 15px; }
</style>
</head>
<body>
<div class="navbar">
<h1>Asignaciones del Curso</h1>
<div>
<a href="?action=assignments"><- Asignaciones</a>
<a href="?action=logout">Cerrar sesión</a>
</div>
</div>

<div class="container">
<div class="card">
<div class="course-info">
<h2><?= $course['name'] ?></h2>
<p><strong>Nivel:</strong> <?= $course['grade_level'] ?> | <strong>Paralelo:</strong> <?=
$course['parallel'] ?></p>
<p><strong>Jornada:</strong> <?= ucfirst($course['shift_name']) ?> | <strong>Año:</strong> <?=
$course['year_name'] ?></p>

```

```

</div>

<?php if($tutor): ?>
<div class="tutor-box">
<h3>■■■■ Docente Tutor</h3>
<p><?= $tutor['last_name'] . ' ' . $tutor['first_name'] ?></p>
<p style="font-size: 14px; color: #666;">Email: <?= $tutor['email'] ?></p>
</div>
<?php else: ?>
<div class="tutor-box">
<p><strong>■ Este curso no tiene docente tutor asignado</strong></p>
</div>
<?php endif; ?>

<h3>Asignaturas y Docentes</h3>
<?php if(count($assignments) > 0): ?>
<table>
<thead>
<tr>
<th>Asignatura</th>
<th>Docente</th>
</tr>
</thead>

<tbody>
<?php foreach($assignments as $assignment): ?>
<tr>
<td><?= $assignment['subject_name'] ?></td>
<td>
<?= $assignment['teacher_name'] ?>
<?php if($assignment['is_tutor']): ?>
<span style="color: #ffc107; font-weight: bold;">(Tutor)</span>
<?php endif; ?>
</td>
</tr>
<?php endforeach; ?>
</tbody>
</table>
<?php else: ?>
<p style="color: #666; margin-top: 15px;">No hay docentes asignados a este curso.</p>
<?php endif; ?>
</div>
</div>
</body>
</html>

```

## 52. ArchImp/views/attendance/calendar.php

```

<?php Security::requireLogin(); ?>
<!DOCTYPE html>
<html lang="es">
<head>
<meta charset="UTF-8">
<title>Calendario de Asistencia - EcuAsist</title>
<style>
* { margin: 0; padding: 0; box-sizing: border-box; }
body { font-family: Arial, sans-serif; background: #f4f4f4; }
.navbar { background: #007bff; color: white; padding: 15px 30px; display: flex; justify-content: space-between; align-items: center; }
.navbar h1 { font-size: 24px; }
.navbar a { color: white; text-decoration: none; margin-left: 20px; }
.container { max-width: 1200px; margin: 30px auto; padding: 0 20px; }
.card { background: white; padding: 30px; border-radius: 8px; box-shadow: 0 2px 10px rgba(0,0,0,0.1); }

```

```

margin-bottom: 20px; }
.calendar { display: grid; grid-template-columns: repeat(7, 1fr); gap: 10px; margin-top: 20px; }
.calendar-header { text-align: center; font-weight: bold; padding: 10px; background: #f8f9fa; }
.calendar-day { border: 1px solid #ddd; padding: 10px; min-height: 80px; border-radius: 4px; position:
relative; }
.calendar-day.today { background: #fff3cd; border-color: #ffc107; }
.calendar-day.has-attendance { background: #d4edda; }
.calendar-day.weekend { background: #f8f9fa; color: #999; }
.day-number { font-weight: bold; margin-bottom: 5px; }
.attendance-summary { font-size: 11px; color: #666; }
.month-nav { display: flex; justify-content: space-between; align-items: center; margin-bottom: 20px; }
.btn { padding: 8px 16px; background: #007bff; color: white; border: none; border-radius: 4px; cursor:
pointer; text-decoration: none; }
.btn:hover { background: #0056b3; }
.form-group { margin-bottom: 15px; }
select { padding: 10px; border: 1px solid #ddd; border-radius: 4px; }
</style>
</head>
<body>
<?php include BASE_PATH . '/views/partials/navbar.php'; ?>

<div class="container">
<div class="card">
<div class="form-group">
<label>Seleccionar Curso</label>
<select id="course-select" onchange="loadCalendar()">
<option value="">Seleccionar curso...</option>
<?php foreach($courses as $course): ?>
<option value="<?=$course['id'] ?>" <?=$course['id'] ?> <?=$course['id'] ?> <?=$course['id'] ?> ?
'selected' : '' ?>>
<?=$course['name'] ?> - <?=$course['shift_name'] ?>
</option>
<?php endforeach; ?>
</select>
</div>

<?php if(isset($_GET['course_id'])): ?>
<div class="month-nav">
<a href="?action=attendance_calendar&course_id=<?=$_GET['course_id'] ?>&month=<?=$prevMonth ?>"
class="btn">← Anterior</a>
<h2><?=$monthName ?> <?=$year ?></h2>
<a href="?action=attendance_calendar&course_id=<?=$_GET['course_id'] ?>&month=<?=$nextMonth ?>"
class="btn">Siguiente →</a>
</div>

<div class="calendar">
<?php foreach(['Dom', 'Lun', 'Mar', 'Mié', 'Jue', 'Vie', 'Sáb'] as $day): ?>
<div class="calendar-header"><?=$day ?></div>
<?php endforeach; ?>

<?php foreach($calendarDays as $day): ?>
<div class="calendar-day <?=$day['classes'] ?>">

<?php if($day['day']): ?>
<div class="day-number"><?=$day['day'] ?></div>
<?php if(isset($day['attendance'])): ?>
<div class="attendance-summary">
✓ <?=$day['attendance']['presente'] ?>
✗ <?=$day['attendance']['ausente'] ?>
</div>
<?php endif; ?>
<?php endif; ?>
</div>
<?php endforeach; ?>
</div>
<?php endif; ?>
</div>

```



```

</div>

<script>
function loadCalendar() {
const courseId = document.getElementById('course-select').value;
if (courseId) {
window.location.href = '?action=attendance_calendar&course_id=' + courseId;
}
}
</script>
</body>
</html>

```

### 53. ArchImp/views/attendance/course\_view.php

```

<?php Security::requireLogin(); ?>
<!DOCTYPE html>
<html lang="es">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Asistencia del Curso - EcuAsist</title>
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet">
<style>
body { padding-top: 70px; background: #f4f4f4; }
.badge-presente { background-color: #28a745; }
.badge-ausente { background-color: #dc3545; }
.badge-tardanza { background-color: #ffc107; color: #000; }
.badge-justificado { background-color: #17a2b8; }
</style>
</head>
<body>
<?php include BASE_PATH . '/views/partials/navbar.php'; ?>

<div class="container mt-4">

<?php if(empty($courses)): ?>
<div class="alert alert-warning">
<h5>■■ Sin Cursos Asignados</h5>
<p class="mb-0">
<?php if(Security::hasRole('docente')): ?>
No tiene cursos asignados como tutor ni asignaturas asignadas.
Contacte con la autoridad para que le asignen cursos.
<?php else: ?>
No hay cursos disponibles en el sistema.
<?php endif; ?>
</p>
</div>
<?php else: ?>

<div class="card mb-4">
<div class="card-header bg-primary text-white">
<h5 class="mb-0">■ Asistencia General del Curso</h5>
<?php if(Security::hasRole('docente')): ?>
<small>Mostrando solo cursos donde es tutor o tiene asignaturas</small>
<?php endif; ?>
</div>
<div class="card-body">
<form method="GET" id="filterForm">
<input type="hidden" name="action" value="view_course_attendance">
<div class="row">
<div class="col-md-4 mb-3">
<label class="form-label">Curso *</label>

```

```

<select name="course_id" class="form-select" required>
<option value="">Seleccionar curso...</option>
<?php foreach($courses as $c):
$sisTutor = isset($c['is_tutor']) && $c['is_tutor'];
$label = html_entity_decode($c['name'], ENT_QUOTES, 'UTF-8');
if ($sisTutor) $label .= ' ■ (Mi curso / Tutor)';
?>
<option value="<?=$c['id'] ?>"
<?=$sisTutor && $courseId == $c['id'] ? 'selected' : '' ?>
<?=$label ?>
</option>
<?php endforeach; ?>
</select>
</div>

<div class="col-md-3 mb-3">
<label class="form-label">Fecha Inicio</label>
<input type="date" name="start_date" class="form-control"
value="<?=$startDate ?? date('Y-m-01') ?>">
</div>

<div class="col-md-3 mb-3">
<label class="form-label">Fecha Fin</label>
<input type="date" name="end_date" class="form-control"
value="<?=$endDate ?? date('Y-m-d') ?>">
</div>

<div class="col-md-2 mb-3">
<label class="form-label">&nbsp;</label>
<button type="submit" class="btn btn-primary w-100">
■ Consultar
</button>
</div>
</div>
</form>
</div>
</div>

<?php if(isset($courseId) && !empty($attendanceData)): ?>

<!-- Resumen estadístico -->
<div class="row mb-4">
<?php
$totalPresente = array_sum(array_column($attendanceData, 'presente'));
$totalAusente = array_sum(array_column($attendanceData, 'ausente'));
$totalTardanza = array_sum(array_column($attendanceData, 'tardanza'));
$totalJustificado = array_sum(array_column($attendanceData, 'justificado'));
$totalRegistros = array_sum(array_column($attendanceData, 'total_registros'));
?>

<div class="col-md-3">
<div class="card text-center border-success">
<div class="card-body">
<h3 class="text-success"><?=$totalPresente ?></h3>
<p class="mb-0">Presentes</p>
</div>
</div>
</div>

<div class="col-md-3">
<div class="card text-center border-danger">
<div class="card-body">
<h3 class="text-danger"><?=$totalAusente ?></h3>
<p class="mb-0">Ausentes</p>
</div>
</div>

```

```

</div>
</div>

<div class="col-md-3">
<div class="card text-center border-warning">
<div class="card-body">
<h3 class="text-warning"><?= $totalTardanza ?></h3>
<p class="mb-0">Tardanzas</p>
</div>
</div>
</div>

<div class="col-md-3">
<div class="card text-center border-info">
<div class="card-body">
<h3 class="text-info"><?= $totalJustificado ?></h3>
<p class="mb-0">Justificados</p>
</div>
</div>
</div>
</div>

<!-- Tabla de estudiantes -->
<div class="card">
<div class="card-header bg-light d-flex justify-content-between align-items-center">
<h5 class="mb-0">Estudiantes del Curso</h5>
<span class="badge bg-primary"><?= count($attendanceData) ?> estudiantes</span>
</div>
<div class="card-body">
<div class="alert alert-info">
<strong>■ Período:</strong>
<?= date('d/m/Y', strtotime($startDate)) ?> al <?= date('d/m/Y', strtotime($endDate)) ?>
</div>

<div class="table-responsive">
<table class="table table-hover table-bordered">
<thead class="table-light">
<tr>
<th>#</th>
<th>Estudiante</th>
<th class="text-center">Total Registros</th>
<th class="text-center">Presentes</th>
<th class="text-center">Ausentes</th>
<th class="text-center">Tardanzas</th>
<th class="text-center">Justificados</th>
<th class="text-center">% Asistencia</th>
</tr>
</thead>
<tbody>
<?php
$counter = 1;
foreach($attendanceData as $data):
$percentage = $data['total_registros'] > 0
? round(($data['presente'] / $data['total_registros']) * 100, 1)
: 0;

$percentageClass = '';
if ($percentage >= 90) $percentageClass = 'text-success fw-bold';
elseif ($percentage >= 75) $percentageClass = 'text-warning fw-bold';
else $percentageClass = 'text-danger fw-bold';
?>
<tr>
<td><?= $counter++ ?></td>
<td><strong><?= $data['student_name'] ?></strong></td>
<td class="text-center"><?= $data['total_registros'] ?></td>

```

```

<td class="text-center">
<span class="badge badge-presente"><?= $data['presente'] ?></span>
</td>
<td class="text-center">
<span class="badge badge-ausente"><?= $data['ausente'] ?></span>
</td>
<td class="text-center">
<span class="badge badge-tardanza"><?= $data['tardanza'] ?></span>
</td>
<td class="text-center">
<span class="badge badge-justificado"><?= $data['justificado'] ?></span>
</td>
<td class="text-center <?= $percentageClass ?>">
<?= $percentage ?>%
</td>
</tr>
<?php endforeach; ?>
</tbody>
</table>
</div>

<div class="mt-3">
<small class="text-muted">
<strong>Nota:</strong> El porcentaje de asistencia se calcula sobre el total de registros (Presentes / Total
Registros x 100).
</small>
</div>
</div>
</div>

<?php elseif(isset($courseId) && empty($attendanceData)): ?>
<div class="alert alert-warning text-center">
■■ No hay registros de asistencia para el curso seleccionado en el período indicado.
</div>

<?php else: ?>
<div class="alert alert-info text-center">
■ Seleccione un curso y período para ver la asistencia.
</div>
<?php endif; ?>

<?php endif; // Fin de validación de cursos ?>
</div>

<script src="https://cdn.jsdelivrivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"></script>
</body>
</html>

```

## 54. ArchImp/views/attendance/my\_attendance.php

```

<?php Security::requireLogin(); ?>
<!DOCTYPE html>
<html lang="es">
<head>
<meta charset="UTF-8">
<title>Mi Asistencia - EcuAsist</title>
<style>
* { margin: 0; padding: 0; box-sizing: border-box; }
body { font-family: Arial, sans-serif; background: #f4f4f4; }
.navbar { background: #007bff; color: white; padding: 15px 30px; display: flex; justify-content:
space-between; align-items: center; }
.navbar h1 { font-size: 24px; }

```

```

.navbar a { color: white; text-decoration: none; margin-left: 20px; }
.container { max-width: 1200px; margin: 30px auto; padding: 0 20px; }
.card { background: white; padding: 30px; border-radius: 8px; box-shadow: 0 2px 10px rgba(0,0,0,0.1); }
table { width: 100%; border-collapse: collapse; margin-top: 20px; }
th, td { padding: 12px; text-align: left; border-bottom: 1px solid #ddd; }
th { background: #f8f9fa; font-weight: bold; }
.badge { padding: 5px 10px; border-radius: 3px; color: white; font-size: 12px; }
.badge-presente { background: #28a745; }
.badge-ausente { background: #dc3545; }
.badge-tardanza { background: #ffc107; color: #333; }
.badge-justificado { background: #17a2b8; }
.stats { display: grid; grid-template-columns: repeat(auto-fit, minmax(200px, 1fr)); gap: 20px;
margin-bottom: 30px; }
.stat-card { color: white; padding: 20px; border-radius: 8px; text-align: center; }
.stat-number { font-size: 36px; font-weight: bold; margin-bottom: 5px; }
.stat-label { font-size: 14px; opacity: 0.9; }
.btn-justify {
color: #007bff;
text-decoration: none;
font-size: 12px;
font-weight: bold;
}
.btn-justify:hover {
text-decoration: underline;
}
</style>
</head>
<body>
<?php include BASE_PATH . '/views/partials/navbar.php'; ?>

<div class="container">
<?php
$total = count($attendances);
$presente = count(array_filter($attendances, fn($a) => $a['status'] === 'presente'));
$ausente = count(array_filter($attendances, fn($a) => $a['status'] === 'ausente'));
$tardanza = count(array_filter($attendances, fn($a) => $a['status'] === 'tardanza'));
$asistencia = $total > 0 ? round(($presente / $total) * 100, 1) : 0;
?>

<div class="stats">
<div class="stat-card" style="background: linear-gradient(135deg, #667eea 0%, #764ba2 100%);">
<div class="stat-number"><?= $total ?></div>
<div class="stat-label">Total Registros</div>
</div>
<div class="stat-card" style="background: linear-gradient(135deg, #28a745 0%, #20c997 100%);">
<div class="stat-number"><?= $asistencia ?>%</div>
<div class="stat-label">Asistencia</div>
</div>
<div class="stat-card" style="background: linear-gradient(135deg, #dc3545 0%, #fd7e14 100%);">
<div class="stat-number"><?= $ausente ?></div>
<div class="stat-label">Ausencias</div>
</div>
<div class="stat-card" style="background: linear-gradient(135deg, #ffc107 0%, #fd7e14 100%);">
<div class="stat-number"><?= $tardanza ?></div>
<div class="stat-label">Tardanzas</div>
</div>
</div>

<div class="card">
<h2>Historial de Asistencia</h2>
<table>
<thead>
<tr>
<th>Fecha</th>
<th>Curso</th>

```

```

<th>Asignatura</th>
<th>Jornada</th>
<th>Hora</th>
<th>Estado</th>
<th>Observación</th>
<th>Acciones</th>
</tr>
</thead>
<tbody>
<?php foreach($attendances as $att): ?>
<tr>
<td><?= date('d/m/Y', strtotime($att['date'])) ?></td>
<td><?= $att['course_name'] ?></td>
<td><?= $att['subject_name'] ?></td>
<td><?= ucfirst($att['shift_name']) ?></td>
<td><?= $att['hour_period'] ?></td>
<td>
<span class="badge badge-<?= $att['status'] ?>">
<?= ucfirst($att['status']) ?>
</span>
</td>
<td><?= $att['observation'] ?: '-' ?></td>
<td>
<?php if($att['status'] === 'ausente'): ?>
<a href="?action=submit_justification&attendance_id=<?= $att['id'] ?>"
class="btn-justify">
Justificar
</a>
<?php else: ?>
-
<?php endif; ?>
</td>
</tr>
<?php endforeach; ?>

<?php if(empty($attendances)): ?>
<tr>
<td colspan="8" style="text-align:center;">No hay registros de asistencia.</td>
</tr>
<?php endif; ?>
</tbody>
</table>
</div>
</div>
</body>
</html>

```

## 55. ArchImp/views/attendance/register.php

```

<?php Security::requireLogin(); ?>
<!DOCTYPE html>
<html lang="es">
<head>
<meta charset="UTF-8">
<title>Registrar Asistencia - EcuAsist</title>
<style>
* { margin: 0; padding: 0; box-sizing: border-box; }
body { font-family: Arial, sans-serif; background: #f4f4f4; }
.container { max-width: 1200px; margin: 30px auto; padding: 0 20px; }
.card { background: white; padding: 30px; border-radius: 8px; box-shadow: 0 2px 10px rgba(0,0,0,0.1);
margin-bottom: 20px; }

```

```

.form-group { margin-bottom: 20px; }
label { display: block; margin-bottom: 5px; font-weight: bold; color: #333; }
input, select { width: 100%; padding: 10px; border: 1px solid #ddd; border-radius: 4px; }
button { padding: 12px 30px; background: #007bff; color: white; border: none; border-radius: 4px; cursor: pointer; font-size: 16px; }
button:hover { background: #0056b3; }
table { width: 100%; border-collapse: collapse; margin-top: 20px; }
th, td { padding: 12px; text-align: left; border-bottom: 1px solid #ddd; }
th { background: #f8f9fa; }
.success { background: #d4edda; color: #155724; padding: 15px; border-radius: 4px; margin-bottom: 20px; }
.error { background: #f8d7da; color: #721c24; padding: 15px; border-radius: 4px; margin-bottom: 20px; }
#student-list { display: none; }
select.status-select { width: auto; padding: 5px; }
.warning { background: #fff3cd; color: #856404; padding: 10px; border-radius: 4px; margin-bottom: 15px; font-size: 14px; }
.class-card { border: 1px solid #ddd; padding: 15px; border-radius: 4px; margin-bottom: 10px; cursor: pointer; transition: all 0.3s; }
.class-card:hover { background: #f8f9fa; border-color: #007bff; }
.class-card.selected { background: #e7f3ff; border-color: #007bff; }
.class-info { display: flex; justify-content: space-between; align-items: center; }
.class-details h3 { color: #007bff; margin-bottom: 5px; }
.class-details p { color: #666; font-size: 14px; }
</style>
</head>
<body>
<?php include BASE_PATH . '/views/partials/navbar.php'; ?>

<div class="container">
<?php if(isset($_GET['success'])): ?>
<div class="success">✓ Asistencia registrada correctamente</div>
<?php endif; ?>

<?php if(isset($_GET['error'])): ?>
<?php if($_GET['error'] == 'future'): ?>
<div class="error">✗ No se puede registrar asistencia de fechas futuras</div>
<?php elseif($_GET['error'] == 'toolate'): ?>
<div class="error">✗ Solo se puede registrar asistencia de hasta 48 horas atrás</div>
<?php endif; ?>
<?php endif; ?>

<div class="card">
<h2>Mis Clases de Hoy</h2>
<div class="warning">
■ <strong>Importante:</strong> Solo puede registrar asistencia del día actual o hasta 48 horas atrás.
</div>

<?php if(empty($todayClasses)): ?>
<p style="color: #666; text-align: center; padding: 30px;">
No tienes clases programadas para hoy según tu horario.
</p>
<?php else: ?>
<div id="classes-list">
<?php foreach($todayClasses as $class): ?>
<div class="class-card" onclick="selectClass(<?= $class['id'] ?>, '<?= $class['course_name'] ?>', '<?= $class['subject_name'] ?>', '<?= $class['period_number'] ?>)">
<div class="class-info">
<div class="class-details">
<h3><?= $class['period_number'] ?>ra hora - <?= $class['subject_name'] ?></h3>
<p>Curso: <?= $class['course_name'] ?></p>
</div>
<div>
<button type="button" style="padding: 8px 15px; font-size: 14px;">
Tomar Asistencia
</button>
</div>

```

```

</div>
</div>
<?php endforeach; ?>
</div>
<?php endif; ?>
</div>

<div class="card" id="date-selector" style="display: none;">
<h2>Seleccionar Fecha</h2>
<form id="config-form">
<input type="hidden" id="schedule_id" name="schedule_id">
<div class="form-group">
<label>Clase Seleccionada</label>
<input type="text" id="selected_class_info" readonly style="background: #f0f0f0;">
</div>
<div class="form-group">
<label>Fecha</label>
<input type="date" id="date" name="date" value="<?= date('Y-m-d') ?>"
max="<?= date('Y-m-d') ?>"
min="<?= $minDate ?>"
required>
<small style="color: #666; display: block; margin-top: 5px;">
Permitido: Desde el <?= date('d/m/Y', strtotime($minDate)) ?> hasta hoy
</small>
</div>
<button type="button" onclick="loadStudents()">Cargar Estudiantes</button>
</form>
</div>

<form method="POST" id="student-list">
<input type="hidden" name="schedule_id" id="schedule_id_hidden">
<input type="hidden" name="date" id="date_hidden">

<div class="card">
<h2>Lista de Estudiantes</h2>
<table id="students-table">
<thead>
<tr>
<th>#</th>
<th>Apellidos y Nombres</th>
<th>Cédula</th>
<th>Estado</th>
<th>Observación</th>
</tr>
</thead>
<tbody></tbody>
</table>
<button type="submit" style="margin-top: 20px;">Guardar Asistencia</button>
</div>
</form>
</div>

<script>
const minDateAllowed = '<?= $minDate ?>';
let selectedScheduleId = null;

function selectClass(scheduleId, courseName, subjectName, periodNumber) {
selectedScheduleId = scheduleId;

// Remover selección anterior
document.querySelectorAll('.class-card').forEach(card => {
card.classList.remove('selected');
});

// Seleccionar actual

```



```

event.currentTarget.classList.add('selected');

// Mostrar selector de fecha
document.getElementById('date-selector').style.display = 'block';
document.getElementById('schedule_id').value = scheduleId;
document.getElementById('selected_class_info').value = periodNumber + 'ra hora - ' + subjectName + ' (' +
courseName + ')';

// Ocultar lista de estudiantes
document.getElementById('student-list').style.display = 'none';
}

document.getElementById('date').addEventListener('change', function() {
const selectedDate = this.value;
const today = new Date().toISOString().split('T')[0];

if (selectedDate > today) {
alert('No se puede seleccionar una fecha futura');
this.value = today;
return;
}

if (selectedDate < minDateAllowed) {
alert('Solo puede registrar asistencia de hasta 48 horas hábiles atrás');
this.value = today;
}
});

function loadStudents() {
const scheduleId = document.getElementById('schedule_id').value;
const date = document.getElementById('date').value;

if (!scheduleId || !date) {
alert('Complete todos los campos');
return;
}

document.getElementById('schedule_id_hidden').value = scheduleId;
document.getElementById('date_hidden').value = date;

// Obtener course_id del horario
fetch('?action=get_schedule_info&schedule_id=' + scheduleId)
.then(response => response.json())
.then(scheduleData => {
const formData = new FormData();
formData.append('course_id', scheduleData.course_id);

return fetch('?action=get_students', {
method: 'POST',
body: formData
});
})
.then(response => response.json())
.then(students => {
if (students.length === 0) {
alert('No hay estudiantes matriculados en este curso');
return;
}

const tbody = document.querySelector('#students-table tbody');
tbody.innerHTML = '';

students.forEach((student, index) => {
const row = `
<tr>

```

```

<td>${index + 1}</td>
<td>${student.last_name} ${student.first_name}</td>
<td>${student.dni || '-'}</td>
<td>
<select name="status_${student.id}" class="status-select">
<option value="presente" selected>Presente</option>
<option value="ausente">Ausente</option>
<option value="tardanza">Tardanza</option>
<option value="justificado">Justificado</option>
</select>
</td>
<td>
<input type="text" name="obs_${student.id}" placeholder="Opcional" style="width: 100%;">
</td>
</tr>
`;
tbody.innerHTML += row;
});

document.getElementById('student-list').style.display = 'block';
document.getElementById('student-list').scrollIntoView({ behavior: 'smooth' });
})
.catch(error => {
console.error('Error:', error);
alert('Error al cargar estudiantes');
});
}
</script>
</body>
</html>

```

## 56. ArchImp/views/attendance/view.php

```

<?php Security::requireLogin(); ?>
<!DOCTYPE html>
<html lang="es">
<head>
<meta charset="UTF-8">
<title>Ver Asistencias - EcuAsist</title>
<style>
* { margin: 0; padding: 0; box-sizing: border-box; }
body { font-family: Arial, sans-serif; background: #f4f4f4; }
.navbar { background: #007bff; color: white; padding: 15px 30px; display: flex; justify-content: space-between; align-items: center; }
.navbar h1 { font-size: 24px; }
.navbar a { color: white; text-decoration: none; margin-left: 20px; }
.container { max-width: 1200px; margin: 30px auto; padding: 0 20px; }
.card { background: white; padding: 30px; border-radius: 8px; box-shadow: 0 2px 10px rgba(0,0,0,0.1); margin-bottom: 20px; }
.form-group { margin-bottom: 15px; }
label { display: block; margin-bottom: 5px; font-weight: bold; }
input, select { width: 100%; padding: 10px; border: 1px solid #ddd; border-radius: 4px; }
button { padding: 12px 30px; background: #007bff; color: white; border: none; border-radius: 4px; cursor: pointer; }
button:hover { background: #0056b3; }
table { width: 100%; border-collapse: collapse; margin-top: 20px; }
th, td { padding: 12px; text-align: left; border-bottom: 1px solid #ddd; }
th { background: #f8f9fa; }
.badge { padding: 5px 10px; border-radius: 3px; color: white; font-size: 12px; }
.badge-presente { background: #28a745; }
.badge-ausente { background: #dc3545; }
.badge-tardanza { background: #ffc107; color: #333; }
.badge-justificado { background: #17a2b8; }

```

```

</style>
</head>
<body>
<?php include BASE_PATH . '/views/partials/navbar.php'; ?>

<div class="container">
<div class="card">
<h2>Consultar</h2>
<form method="POST">
<div class="form-group">
<label>Curso</label>
<select name="course_id" required>
<option value="">Seleccionar...</option>
<?php foreach($courses as $course): ?>
<option value="<?= $course['id'] ?>"><?= $course['name'] ?> - <?= $course['shift_name'] ?></option>
<?php endforeach; ?>
</select>
</div>
<div class="form-group">
<label>Fecha</label>
<input type="date" name="date" value="<?= date('Y-m-d') ?>" required>
</div>
<button type="submit">Buscar</button>
</form>
</div>

<?php if(!empty($attendances)): ?>
<div class="card">
<h2>Resultados</h2>
<table>
<thead>
<tr>
<th>Estudiante</th>
<th>Asignatura</th>
<th>Docente</th>
<th>Hora</th>
<th>Estado</th>
<th>Observación</th>
</tr>
</thead>
<tbody>
<?php foreach($attendances as $att): ?>
<tr>
<td><?= $att['last_name'] . ' ' . $att['first_name'] ?></td>
<td><?= $att['subject_name'] ?></td>
<td><?= $att['teacher_name'] ?></td>
<td><?= $att['hour_period'] ?></td>
<td>
<span class="badge badge-<?= $att['status'] ?>">
<?= ucfirst($att['status']) ?>
</span>
</td>
<td><?= $att['observation'] ?>: '-' ?></td>
</tr>
<?php endforeach; ?>
</tbody>
</table>
</div>
<?php endif; ?>
</div>
</body>
</html>

```

## 57. ArchImp/views/auth/forgot.php

```
<!DOCTYPE html>
<html lang="es">
<head>
<meta charset="UTF-8">
<title>Recuperar Contraseña - EcuAsist</title>
<style>
* { margin: 0; padding: 0; box-sizing: border-box; }
body { font-family: Arial, sans-serif; background: #f4f4f4; display: flex; justify-content: center;
align-items: center; height: 100vh; }
.container { background: white; padding: 40px; border-radius: 8px; box-shadow: 0 2px 10px rgba(0,0,0,0.1);
width: 100%; max-width: 400px; }
h2 { margin-bottom: 20px; color: #333; }
input { width: 100%; padding: 12px; margin: 10px 0; border: 1px solid #ddd; border-radius: 4px; }
button { width: 100%; padding: 12px; background: #007bff; color: white; border: none; border-radius: 4px;
cursor: pointer; font-size: 16px; }
button:hover { background: #0056b3; }
.success { color: green; margin-bottom: 10px; }
a { color: #007bff; }
</style>
</head>
<body>
<div class="container">
<h2>Recuperar Contraseña</h2>
<?php if(isset($message)): ?>
<p class="success"><?= $message ?></p>
<?php endif; ?>

<form method="POST">
<input type="hidden" name="csrf_token" value="<?= Security::generateToken() ?>">
<input type="email" name="email" placeholder="Correo electrónico" required>
<button type="submit">Enviar enlace</button>
</form>

<p style="margin-top: 15px;"><a href="?action=login">Volver al login</a></p>
</div>
</body>
</html>
```

## 58. ArchImp/views/auth/login.php

```
<?php
if(isset($_SESSION['user_id'])) {
header('Location: ?action=dashboard');
exit;
}
?>
<!DOCTYPE html>
<html lang="es">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Iniciar Sesión - EcuAsist 2026</title>
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css" rel="stylesheet">
<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/font-awesome@6.4.0/css/all.min.css">
<style>
body {
background: linear-gradient(135deg, #667eea 0%, #764ba2 100%);
min-height: 100vh;
display: flex;
align-items: center;
justify-content: center;
}
```

```

}
.login-card {
background: white;
border-radius: 15px;
box-shadow: 0 10px 40px rgba(0,0,0,0.2);
max-width: 420px;
width: 100%;
}
</style>
</head>
<body>
<div class="login-card p-5">
<div class="text-center mb-4">
<i class="fas fa-school fa-4x text-primary mb-3"></i>
<h2>EcuAsist 2026</h2>
<p class="text-muted">Sistema de Asistencia Escolar</p>
</div>

<?php if(isset($error)): ?>
<div class="alert alert-danger alert-dismissible fade show" role="alert">
<i class="fas fa-exclamation-circle"></i> <?= htmlspecialchars($error) ?>
<button type="button" class="btn-close" data-bs-dismiss="alert"></button>
</div>
<?php endif; ?>

<?php if(isset($_GET['registered'])): ?>
<div class="alert alert-success alert-dismissible fade show" role="alert">
<i class="fas fa-check-circle"></i> Registro exitoso. Ahora puede iniciar sesión.
<button type="button" class="btn-close" data-bs-dismiss="alert"></button>
</div>
<?php endif; ?>

<?php if(isset($_GET['reset'])): ?>
<div class="alert alert-success alert-dismissible fade show" role="alert">
<i class="fas fa-check-circle"></i> Contraseña restablecida correctamente.
<button type="button" class="btn-close" data-bs-dismiss="alert"></button>
</div>
<?php endif; ?>

<form method="POST">
<?php Security::generateToken(); ?>
<input type="hidden" name="csrf_token" value="<?= $_SESSION['csrf_token'] ?>">

<div class="mb-3">
<label for="email" class="form-label">
<i class="fas fa-user"></i> Usuario o Correo Electrónico
</label>
<input type="text" class="form-control" id="email" name="email" required autofocus placeholder="Ingrese usuario o email">
</div>

<div class="mb-3">
<label for="password" class="form-label">
<i class="fas fa-lock"></i> Contraseña
</label>
<input type="password" class="form-control" id="password" name="password" required placeholder="Ingrese contraseña">
</div>

<button type="submit" class="btn btn-primary w-100 py-2 mb-3">
<i class="fas fa-sign-in-alt"></i> Iniciar Sesión
</button>
</form>

<div class="text-center">

```

```

<a href="?action=forgot" class="text-decoration-none small">
<i class="fas fa-key"></i> ¿Olvidó su contraseña?
</a>
</div>

<hr class="my-4">

<div class="text-center">
<small class="text-muted">
<i class="fas fa-info-circle"></i> Usuario de prueba: <strong>admin</strong> / <strong>password</strong>
</small>
</div>
</div>

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js"></script>
</body>
</html>

```

## 59. ArchImp/views/auth/register.php

```

<!DOCTYPE html>
<html lang="es">
<head>
<meta charset="UTF-8">
<title>Registro - EcuAsist</title>
<style>
* { margin: 0; padding: 0; box-sizing: border-box; }
body { font-family: Arial, sans-serif; background: #f4f4f4; display: flex; justify-content: center;
align-items: center; min-height: 100vh; padding: 20px; }
.container { background: white; padding: 40px; border-radius: 8px; box-shadow: 0 2px 10px rgba(0,0,0,0.1);
width: 100%; max-width: 500px; }
h2 { margin-bottom: 20px; color: #333; }
input { width: 100%; padding: 12px; margin: 10px 0; border: 1px solid #ddd; border-radius: 4px; }
button { width: 100%; padding: 12px; background: #28a745; color: white; border: none; border-radius: 4px;
cursor: pointer; font-size: 16px; margin-top: 10px; }
button:hover { background: #218838; }
a { color: #007bff; }
</style>
</head>
<body>
<div class="container">
<h2>Registro de Usuario</h2>
<form method="POST">
<input type="hidden" name="csrf_token" value="<?= Security::generateToken() ?>">
<input type="text" name="username" placeholder="Usuario" required>
<input type="email" name="email" placeholder="Correo electrónico" required>
<input type="password" name="password" placeholder="Contraseña" required>
<input type="text" name="first_name" placeholder="Nombres" required>
<input type="text" name="last_name" placeholder="Apellidos" required>
<input type="text" name="dni" placeholder="Cédula (opcional)">
<input type="text" name="phone" placeholder="Teléfono (opcional)">
<button type="submit">Registrarse</button>
</form>
<p style="margin-top: 15px;"><a href="?action=login">Ya tengo cuenta</a></p>
</div>
</body>
</html>

```

## 60. ArchImp/views/auth/reset.php

```

<!DOCTYPE html>
<html lang="es">
<head>
<meta charset="UTF-8">
<title>Nueva Contraseña - EcuAsist</title>
<style>
* { margin: 0; padding: 0; box-sizing: border-box; }
body { font-family: Arial, sans-serif; background: #f4f4f4; display: flex; justify-content: center;
align-items: center; height: 100vh; }
.container { background: white; padding: 40px; border-radius: 8px; box-shadow: 0 2px 10px rgba(0,0,0,0.1);
width: 100%; max-width: 400px; }
h2 { margin-bottom: 20px; color: #333; }
input { width: 100%; padding: 12px; margin: 10px 0; border: 1px solid #ddd; border-radius: 4px; }
button { width: 100%; padding: 12px; background: #28a745; color: white; border: none; border-radius: 4px;
cursor: pointer; font-size: 16px; }
button:hover { background: #218838; }
</style>
</head>
<body>
<div class="container">
<h2>Nueva Contraseña</h2>
<form method="POST">
<input type="hidden" name="csrf_token" value="{?= Security::generateToken() ?}">
<input type="password" name="password" placeholder="Nueva contraseña" required>
<input type="password" name="password_confirm" placeholder="Confirmar contraseña" required>
<button type="submit">Cambiar contraseña</button>
</form>
</div>
</body>
</html>

```

## 61. ArchImp/views/backup/index.php

```

<?php Security::requireLogin(); ?>
<!DOCTYPE html>
<html lang="es">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>RespalDOS - EcuAsist</title>
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet">
<style>
body { padding-top: 70px; background: #f4f4f4; }
.container { max-width: 1200px; margin: 30px auto; }
.card { background: white; padding: 30px; border-radius: 8px; box-shadow: 0 2px 10px rgba(0,0,0,0.1);
margin-bottom: 20px; }
</style>
</head>
<body>
<?php include BASE_PATH . '/views/partials/navbar.php'; ?>

<div class="container">
<div class="card">
<div class="d-flex justify-content-between align-items-center mb-4">
<h2 class="mb-0">■ RespalDOS de Base de Datos</h2>
<button onclick="createBackup()" class="btn btn-primary">
■ Crear RespalDO Ahora
</button>
</div>

<?php if(isset($_GET['success'])): ?>
<div class="alert alert-success alert-dismissible fade show">
✓ RespalDO creado correctamente: <strong>{?= htmlspecialchars($_GET['file']) ?}</strong>
<button type="button" class="btn-close" data-bs-dismiss="alert"></button>

```

```

</div>
<?php endif; ?>

<?php if(isset($_GET['error'])): ?>
<div class="alert alert-danger alert-dismissible fade show">
✗ Error al crear el respaldo. Verifique la configuración de MySQL.
<button type="button" class="btn-close" data-bs-dismiss="alert"></button>
</div>
<?php endif; ?>

<?php if(isset($_GET['cleanup'])): ?>
<div class="alert alert-info alert-dismissible fade show">
■■ <?= (int)$_GET['cleanup'] ?> respaldo(s) antiguo(s) eliminado(s).
<button type="button" class="btn-close" data-bs-dismiss="alert"></button>
</div>
<?php endif; ?>

<?php if(isset($_GET['deleted'])): ?>
<div class="alert alert-success alert-dismissible fade show">
✓ Respaldo eliminado correctamente.
<button type="button" class="btn-close" data-bs-dismiss="alert"></button>
</div>
<?php endif; ?>

<div class="alert alert-info">
<strong>■■ Información:</strong><br>


- Los respaldos se crean automáticamente en formato SQL<br>
- Se recomienda descargar y guardar en ubicación segura<br>
- Los respaldos antiguos (>30 días) pueden eliminarse para liberar espacio


</div>

<div class="d-flex justify-content-between mb-3">
<h4>Respaldos Disponibles (<?= count($backups) ?>)</h4>
<?php if(count($backups) > 0): ?>
<button onclick="confirmCleanup()" class="btn btn-warning btn-sm">
■■ Limpiar Antiguos (>30 días)
</button>
<?php endif; ?>
</div>

<?php if(empty($backups)): ?>
<div class="alert alert-secondary text-center">
■ No hay respaldos disponibles. Crea uno usando el botón superior.
</div>
<?php else: ?>
<div class="table-responsive">
<table class="table table-hover">
<thead class="table-light">
<tr>
<th>#</th>
<th>Nombre del Archivo</th>
<th>Fecha de Creación</th>
<th>Tamaño</th>
<th>Acciones</th>
</tr>
</thead>
<tbody>
<?php
$counter = 1;
foreach($backups as $backup):
?>
<tr>
<td><?= $counter++ ?></td>
<td>
<strong><?= htmlspecialchars($backup['filename']) ?></strong>

```



```

</td>
<td><?=$backup['date'] ?></td>
<td>
<span class="badge bg-secondary"><?=$backup['size'] ?></span>
</td>
<td>
<a href="?action=download_backup&file=<?=$urlencode($backup['filename']) ?>"
class="btn btn-sm btn-success">
■■ Descargar
</a>
<button onclick="confirmDelete('<?=$addslashes($backup['filename']) ?>')"
class="btn btn-sm btn-danger">
■■ Eliminar
</button>
</td>
</tr>
<?php endforeach; ?>
</tbody>
</table>
</div>
<?php endif; ?>
</div>
</div>

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"></script>
<script>
function createBackup() {
const modal = document.createElement('div');
modal.style.cssText = 'position: fixed; top: 0; left: 0; width: 100%; height: 100%; background:
rgba(0,0,0,0.5); display: flex; align-items: center; justify-content: center; z-index: 9999;';

const modalContent = document.createElement('div');
modalContent.style.cssText = 'background: white; padding: 30px; border-radius: 8px; max-width: 500px;
box-shadow: 0 4px 20px rgba(0,0,0,0.3);';

modalContent.innerHTML = `
<h3 style="margin: 0 0 15px 0; color: #007bff;">■■ Crear Respaldo</h3>
<p style="margin: 0 0 20px 0; color: #666;">
¿Crear un nuevo respaldo de la base de datos?
</p>
<p style="margin: 0 0 20px 0; color: #666; font-size: 14px; background: #e7f3ff; padding: 10px;
border-radius: 4px;">
<strong>■■ Nota:</strong> Este proceso puede tardar unos segundos dependiendo del tamaño de la base de
datos.
</p>
<div style="display: flex; gap: 10px; justify-content: flex-end;">
<button type="button" id="cancelCreateBtn" style="padding: 10px 20px; background: #6c757d; color: white;
border: none; border-radius: 4px; cursor: pointer;">
Cancelar
</button>
<button type="button" id="confirmCreateBtn" style="padding: 10px 20px; background: #007bff; color: white;
border: none; border-radius: 4px; cursor: pointer;">
Sí, Crear
</button>
</div>
`;

modal.appendChild(modalContent);
document.body.appendChild(modal);

document.getElementById('confirmCreateBtn').onclick = function() {
document.body.removeChild(modal);
window.location.href = '?action=create_backup';
};

```

```

document.getElementById('cancelCreateBtn').onclick = function() {
document.body.removeChild(modal);
};
}

function confirmCleanup() {
const modal = document.createElement('div');
modal.style.cssText = 'position: fixed; top: 0; left: 0; width: 100%; height: 100%; background:
rgba(0,0,0,0.5); display: flex; align-items: center; justify-content: center; z-index: 9999;';

const modalContent = document.createElement('div');
modalContent.style.cssText = 'background: white; padding: 30px; border-radius: 8px; max-width: 500px;
box-shadow: 0 4px 20px rgba(0,0,0,0.3);';

modalContent.innerHTML = `
<h3 style="margin: 0 0 15px 0; color: #ffc107;">■■■ Limpiar Respaldos Antiguos</h3>
<p style="margin: 0 0 20px 0; color: #666;">
¿Está seguro de eliminar todos los respaldos con más de <strong>30 días</strong> de antigüedad?
</p>
<p style="margin: 0 0 20px 0; color: #666; font-size: 14px; background: #fff3cd; padding: 10px;
border-radius: 4px;">
<strong>■■■ Advertencia:</strong> Esta acción no se puede deshacer.
Se recomienda descargar los respaldos importantes antes de eliminarlos.
</p>
<div style="display: flex; gap: 10px; justify-content: flex-end;">
<button type="button" id="cancelCleanupBtn" style="padding: 10px 20px; background: #6c757d; color: white;
border: none; border-radius: 4px; cursor: pointer;">
Cancelar
</button>
<button type="button" id="confirmCleanupBtn" style="padding: 10px 20px; background: #ffc107; color: #000;
border: none; border-radius: 4px; cursor: pointer; font-weight: bold;">
Sí, Limpiar
</button>
</div>
`;

modal.appendChild(modalContent);
document.body.appendChild(modal);

document.getElementById('confirmCleanupBtn').onclick = function() {
document.body.removeChild(modal);
window.location.href = '?action=cleanup_backups';
};

document.getElementById('cancelCleanupBtn').onclick = function() {
document.body.removeChild(modal);
};
}

function confirmDelete(filename) {
const modal = document.createElement('div');
modal.style.cssText = 'position: fixed; top: 0; left: 0; width: 100%; height: 100%; background:
rgba(0,0,0,0.5); display: flex; align-items: center; justify-content: center; z-index: 9999;';

const modalContent = document.createElement('div');
modalContent.style.cssText = 'background: white; padding: 30px; border-radius: 8px; max-width: 500px;
box-shadow: 0 4px 20px rgba(0,0,0,0.3);';

modalContent.innerHTML = `
<h3 style="margin: 0 0 15px 0; color: #dc3545;">■■■ Eliminar Respaldo</h3>
<p style="margin: 0 0 20px 0; color: #666;">
¿Está seguro de eliminar el respaldo <strong>${filename}</strong>?
</p>
<p style="margin: 0 0 20px 0; color: #666; font-size: 14px; background: #f8d7da; padding: 10px;
border-radius: 4px;">

```

```

<strong>■■ Advertencia:</strong> Esta acción no se puede deshacer.
</p>
<div style="display: flex; gap: 10px; justify-content: flex-end;">
<button type="button" id="cancelDeleteBtn" style="padding: 10px 20px; background: #6c757d; color: white;
border: none; border-radius: 4px; cursor: pointer;">
Cancelar
</button>
<button type="button" id="confirmDeleteBtn" style="padding: 10px 20px; background: #dc3545; color: white;
border: none; border-radius: 4px; cursor: pointer;">
Sí, Eliminar
</button>
</div>
`;

modal.appendChild(modalContent);
document.body.appendChild(modal);

document.getElementById('confirmDeleteBtn').onclick = function() {
document.body.removeChild(modal);
window.location.href = '?action=delete_backup&file=' + encodeURIComponent(filename);
};

document.getElementById('cancelDeleteBtn').onclick = function() {
document.body.removeChild(modal);
};
}
</script>
</body>
</html>

```

## 62. ArchImp/views/dashboard/index.php

```

<?php Security::requireLogin(); ?>
<!DOCTYPE html>
<html lang="es">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Dashboard - EcuAsist</title>
<style>
* { margin: 0; padding: 0; box-sizing: border-box; }
body { font-family: Arial, sans-serif; background: #f4f4f4; }
.navbar { background: #007bff; color: white; padding: 15px 30px; display: flex; justify-content:
space-between; align-items: center; box-shadow: 0 2px 5px rgba(0,0,0,0.1); }
.navbar h1 { font-size: 24px; }
.navbar .user-info { display: flex; align-items: center; gap: 15px; }
.navbar a { color: white; text-decoration: none; padding: 8px 15px; border-radius: 4px; transition:
background 0.3s; }
.navbar a:hover { background: rgba(255,255,255,0.2); }
.container { max-width: 1400px; margin: 30px auto; padding: 0 20px; }
.welcome-card { background: linear-gradient(135deg, #667eea 0%, #764ba2 100%); color: white; padding: 30px;
border-radius: 8px; margin-bottom: 30px; box-shadow: 0 4px 15px rgba(0,0,0,0.1); }
.welcome-card h2 { margin-bottom: 10px; font-size: 28px; }
.badge { display: inline-block; background: rgba(255,255,255,0.3); color: white; padding: 5px 10px;
border-radius: 4px; font-size: 12px; margin: 2px; }

.stats-grid { display: grid; grid-template-columns: repeat(auto-fit, minmax(200px, 1fr)); gap: 20px;
margin-bottom: 30px; }
.stat-card { background: white; padding: 25px; border-radius: 8px; box-shadow: 0 2px 10px rgba(0,0,0,0.1);
text-align: center; transition: transform 0.2s; }
.stat-card:hover { transform: translateY(-5px); box-shadow: 0 4px 20px rgba(0,0,0,0.15); }
.stat-number { font-size: 36px; font-weight: bold; color: #007bff; margin-bottom: 5px; }
.stat-label { color: #666; font-size: 14px; }

```

```

.stat-card.green .stat-number { color: #28a745; }
.stat-card.red .stat-number { color: #dc3545; }
.stat-card.orange .stat-number { color: #fd7e14; }
.stat-card.purple .stat-number { color: #6f42c1; }

.menu-grid { display: grid; grid-template-columns: repeat(auto-fit, minmax(280px, 1fr)); gap: 20px; }
.menu-item { background: white; padding: 30px; border-radius: 8px; box-shadow: 0 2px 10px rgba(0,0,0,0.1);
cursor: pointer; transition: all 0.3s; border-left: 4px solid #007bff; }
.menu-item:hover { transform: translateY(-5px); box-shadow: 0 4px 15px rgba(0,0,0,0.15); }
.menu-item h3 { color: #007bff; margin-bottom: 10px; font-size: 18px; }
.menu-item p { color: #666; font-size: 14px; }
.menu-item.green { border-left-color: #28a745; }
.menu-item.green h3 { color: #28a745; }
.menu-item.orange { border-left-color: #fd7e14; }
.menu-item.orange h3 { color: #fd7e14; }
.menu-item.purple { border-left-color: #6f42c1; }
.menu-item.purple h3 { color: #6f42c1; }
</style>
</head>
<body>
<?php include BASE_PATH . '/views/partials/navbar.php'; ?>

<div class="container">
<div class="welcome-card">
<h2>Bienvenido, <?= $_SESSION['first_name'] . ' ' . $_SESSION['last_name'] ?></h2>
<p>Roles asignados:
<?php foreach($_SESSION['roles'] as $role): ?>
<span class="badge"><?= ucfirst($role) ?></span>
<?php endforeach; ?>
</p>
</div>

<?php if(Security::hasRole('autoridad') && isset($stats['total_students'])): ?>
<div class="stats-grid">
<div class="stat-card">
<div class="stat-number"><?= $stats['total_students'] ?></div>
<div class="stat-label">Estudiantes</div>
</div>
<div class="stat-card green">
<div class="stat-number"><?= $stats['total_teachers'] ?></div>
<div class="stat-label">Docentes</div>
</div>
<div class="stat-card orange">
<div class="stat-number"><?= $stats['total_courses'] ?></div>
<div class="stat-label">Cursos</div>
</div>
<?php if($stats['today_attendance']['total'] > 0): ?>
<div class="stat-card purple">
<div class="stat-number">
<?= round(($stats['today_attendance']['presente'] / $stats['today_attendance']['total']) * 100, 1) ?>%
</div>
<div class="stat-label">Asistencia Hoy</div>
</div>
<?php endif; ?>
</div>
<?php endif; ?>

<?php if(Security::hasRole('docente') && isset($stats['my_courses'])): ?>
<div class="stats-grid">
<div class="stat-card">
<div class="stat-number"><?= $stats['my_courses'] ?></div>
<div class="stat-label">Mis Cursos</div>
</div>
<div class="stat-card green">
<div class="stat-number"><?= $stats['my_students'] ?></div>

```

```

<div class="stat-label">Mis Estudiantes</div>
</div>
</div>
<?php endif; ?>

<?php if(Security::hasRole('estudiante') && isset($stats['my_attendance'])): ?>
<div class="stats-grid">
<div class="stat-card green">
<div class="stat-number">
<?= $stats['my_attendance']['total'] > 0 ? round(($stats['my_attendance']['presente'] /
$stats['my_attendance']['total']) * 100, 1) : 0 ?>%
</div>
<div class="stat-label">Mi Asistencia (30 días)</div>
</div>
<div class="stat-card red">
<div class="stat-number"><?= $stats['my_attendance']['ausente'] ?></div>
<div class="stat-label">Ausencias</div>
</div>
<div class="stat-card orange">
<div class="stat-number"><?= $stats['my_attendance']['tardanza'] ?></div>
<div class="stat-label">Tardanzas</div>
</div>
</div>
<?php endif; ?>

<?php if(Security::hasRole('representante') && isset($stats['my_children'])): ?>
<div class="stats-grid">
<div class="stat-card">
<div class="stat-number"><?= $stats['my_children'] ?></div>
<div class="stat-label">Representados</div>
</div>
</div>
<?php endif; ?>

<div class="menu-grid">
<?php if(Security::hasRole(['docente', 'autoridad'])): ?>
<div class="menu-item green" onclick="location.href='?action=attendance_register'">
<h3>■ Registrar Asistencia</h3>
<p>Tomar asistencia de estudiantes</p>
</div>
<?php endif; ?>

<?php if(Security::hasRole(['docente', 'inspector', 'autoridad'])): ?>
<div class="menu-item" onclick="location.href='?action=attendance_view'">
<h3>■ Ver Asistencias</h3>
<p>Consultar registros de asistencia</p>
</div>
<?php endif; ?>

<?php if(Security::hasRole('autoridad')): ?>
<div class="menu-item purple" onclick="location.href='?action=stats'">
<h3>■ Estadísticas</h3>
<p>Análisis y métricas de asistencia</p>
</div>
<?php endif; ?>

<?php if(Security::hasRole(['docente', 'autoridad'])): ?>
<div class="menu-item orange" onclick="location.href='?action=attendance_calendar'">
<h3>■ Calendario</h3>
<p>Vista mensual de asistencias</p>
</div>
<?php endif; ?>

<?php if(Security::hasRole('estudiante')): ?>
<div class="menu-item" onclick="location.href='?action=my_attendance'">

```

```

<h3>■ Mi Asistencia</h3>
<p>Ver mi historial de asistencia</p>
</div>
<?php endif; ?>

<?php if(Security::hasRole('estudiante')): ?>
<div class="menu-item green" onclick="location.href='?action=my_justifications'">
<h3>■ Mis Justificaciones</h3>
<p>Justificar ausencias</p>
</div>
<?php endif; ?>

<?php if(Security::hasRole(['autoridad', 'inspector'])): ?>
<div class="menu-item orange" onclick="location.href='?action=pending_justifications'">
<h3>■ Revisar Justificaciones</h3>
<p>Aprobar o rechazar justificaciones</p>
</div>
<?php endif; ?>

<?php if(Security::hasRole('autoridad')): ?>
<div class="menu-item orange" onclick="location.href='?action=users'">
<h3>■ Gestión de Usuarios</h3>
<p>Administrar usuarios y roles</p>
</div>

<?php if(Security::hasRole('autoridad')): ?>
<div class="menu-item purple" onclick="location.href='?action=backups'">
<h3>■ RespalDOS</h3>
<p>Gestionar respaldos de base de datos</p>
</div>
<?php endif; ?>

<div class="menu-item purple" onclick="location.href='?action=reports'">
<h3>■ Reportes</h3>
<p>Generar informes PDF y Excel</p>
</div>

<div class="menu-item" onclick="location.href='?action=academic'">
<h3>■ Configuración Académica</h3>
<p>Cursos, materias, matrículas</p>
</div>

<div class="menu-item green" onclick="location.href='?action=assignments'">
<h3>■ Asignaciones Docentes</h3>
<p>Gestionar docentes y tutores</p>
</div>

<div class="menu-item orange" onclick="location.href='?action=manage_representatives'">
<h3>■■■■■■■ Gestión Representantes</h3>
<p>Asignar representantes a estudiantes</p>
</div>
<?php endif; ?>

<?php if(Security::hasRole('representante')): ?>
<div class="menu-item purple" onclick="location.href='?action=my_children'">
<h3>■■■■■■■ Mis Representados</h3>
<p>Ver estudiantes y su asistencia</p>
</div>
<?php endif; ?>
</div>
</div>
</body>
</html>

```

## 63. ArchImp/views/institution/index.php

```
<?php Security::requireLogin(); ?>
<!DOCTYPE html>
<html lang="es">
<head>
<meta charset="UTF-8">
<title>Configuración de Institución - EcuAsist</title>
<style>
* { margin: 0; padding: 0; box-sizing: border-box; }
body { font-family: Arial, sans-serif; background: #f4f4f4; }
.container { max-width: 1200px; margin: 30px auto; padding: 0 20px; }
.card { background: white; padding: 30px; border-radius: 8px; box-shadow: 0 2px 10px rgba(0,0,0,0.1);
margin-bottom: 20px; }
.form-group { margin-bottom: 15px; }
label { display: block; margin-bottom: 5px; font-weight: bold; color: #333; }
input, select, textarea { width: 100%; padding: 10px; border: 1px solid #ddd; border-radius: 4px; }
button { padding: 10px 20px; background: #007bff; color: white; border: none; border-radius: 4px; cursor:
pointer; }
button:hover { background: #0056b3; }
.success { background: #d4edda; color: #155724; padding: 10px; border-radius: 4px; margin-bottom: 20px; }
.grid { display: grid; grid-template-columns: 1fr 1fr; gap: 15px; }
.grid-3 { display: grid; grid-template-columns: 2fr 1fr; gap: 15px; }
h2 { margin-bottom: 20px; color: #333; }
.badge {
display: inline-block;
background: #28a745;
color: white;
padding: 5px 10px;
border-radius: 3px;
font-size: 12px;
margin: 3px;
}
.btn-remove {
background: none;
border: none;
color: white;
font-size: 16px;
font-weight: bold;
cursor: pointer;
padding: 0;
margin-left: 5px;
}
.btn-remove:hover { color: #ff0000; }
</style>
</head>
<body>
<?php include BASE_PATH . '/views/partials/navbar.php'; ?>

<div class="container">
<?php if(isset($_GET['success'])): ?>
<div class="success">✓ Información actualizada correctamente</div>
<?php endif; ?>
<?php if(isset($_GET['shift_assigned'])): ?>
<div class="success">✓ Jornada asignada correctamente</div>
<?php endif; ?>
<?php if(isset($_GET['shift_removed'])): ?>
<div class="success">✓ Jornada eliminada correctamente</div>
<?php endif; ?>

<div class="card">
<h2>■ Información de la Institución</h2>
<form method="POST" action="?action=update_institution" enctype="multipart/form-data">
<div class="grid-3">
```

```

<div class="form-group">
<label>Nombre de la Institución *</label>
<input type="text" name="name" value="<?= $institution['name'] ?>" required>
</div>

<div class="form-group">
<label>Logo de la Institución</label>
<?php if(isset($institution['logo_path']) && $institution['logo_path']): ?>

<?php endif; ?>
<input type="file" name="logo" accept="image/png,image/jpeg,image/jpg">
<small style="color: #666; display: block; margin-top: 5px;">PNG, JPG - 300x300px recomendado</small>
</div>
</div>

<div class="grid">
<div class="form-group">
<label>Código AMIE</label>
<input type="text" name="amie_code" value="<?= $institution['amie_code'] ?? ' ' ?>" placeholder="Ej:
01H00001">
</div>

<div class="form-group">
<label>Nombre del Director/Rector</label>
<input type="text" name="director_name" value="<?= $institution['director_name'] ?? ' ' ?>">
</div>

<div class="form-group">
<label>Provincia *</label>
<select name="province" id="province" required>
<option value="">Seleccionar provincia...</option>
<option value="Pichincha" <?= ($institution['province'] ?? ' ') == 'Pichincha' ? 'selected' : ' '
?>>Pichincha</option>
<option value="Guayas" <?= ($institution['province'] ?? ' ') == 'Guayas' ? 'selected' : ' ' ?>>Guayas</option>
<option value="Azuay" <?= ($institution['province'] ?? ' ') == 'Azuay' ? 'selected' : ' ' ?>>Azuay</option>
<option value="Manabí" <?= ($institution['province'] ?? ' ') == 'Manabí' ? 'selected' : ' ' ?>>Manabí</option>
<option value="El Oro" <?= ($institution['province'] ?? ' ') == 'El Oro' ? 'selected' : ' ' ?>>El Oro</option>
<option value="Los Ríos" <?= ($institution['province'] ?? ' ') == 'Los Ríos' ? 'selected' : ' ' ?>>Los
Ríos</option>
<option value="Tungurahua" <?= ($institution['province'] ?? ' ') == 'Tungurahua' ? 'selected' : ' '
?>>Tungurahua</option>
<option value="Esmeraldas" <?= ($institution['province'] ?? ' ') == 'Esmeraldas' ? 'selected' : ' '
?>>Esmeraldas</option>
<option value="Chimborazo" <?= ($institution['province'] ?? ' ') == 'Chimborazo' ? 'selected' : ' '
?>>Chimborazo</option>
<option value="Imbabura" <?= ($institution['province'] ?? ' ') == 'Imbabura' ? 'selected' : ' '
?>>Imbabura</option>
<option value="Cotopaxi" <?= ($institution['province'] ?? ' ') == 'Cotopaxi' ? 'selected' : ' '
?>>Cotopaxi</option>
<option value="Santo Domingo de los Tsáchilas" <?= ($institution['province'] ?? ' ') == 'Santo Domingo de los
Tsáchilas' ? 'selected' : ' ' ?>>Santo Domingo de los Tsáchilas</option>
<option value="Santa Elena" <?= ($institution['province'] ?? ' ') == 'Santa Elena' ? 'selected' : ' ' ?>>Santa
Elena</option>
<option value="Loja" <?= ($institution['province'] ?? ' ') == 'Loja' ? 'selected' : ' ' ?>>Loja</option>
<option value="Cañar" <?= ($institution['province'] ?? ' ') == 'Cañar' ? 'selected' : ' ' ?>>Cañar</option>
<option value="Sucumbíos" <?= ($institution['province'] ?? ' ') == 'Sucumbíos' ? 'selected' : ' '
?>>Sucumbíos</option>
<option value="Orellana" <?= ($institution['province'] ?? ' ') == 'Orellana' ? 'selected' : ' '
?>>Orellana</option>
<option value="Carchi" <?= ($institution['province'] ?? ' ') == 'Carchi' ? 'selected' : ' ' ?>>Carchi</option>
<option value="Bolívar" <?= ($institution['province'] ?? ' ') == 'Bolívar' ? 'selected' : ' '
?>>Bolívar</option>
<option value="Napo" <?= ($institution['province'] ?? ' ') == 'Napo' ? 'selected' : ' ' ?>>Napo</option>
<option value="Pastaza" <?= ($institution['province'] ?? ' ') == 'Pastaza' ? 'selected' : ' '
?>>Pastaza</option>

```



```

<option value="Morona Santiago" <?= ($institution['province'] ?? '') == 'Morona Santiago' ? 'selected' : ''
?>>Morona Santiago</option>
<option value="Zamora Chinchipe" <?= ($institution['province'] ?? '') == 'Zamora Chinchipe' ? 'selected' : ''
?>>Zamora Chinchipe</option>
<option value="Galápagos" <?= ($institution['province'] ?? '') == 'Galápagos' ? 'selected' : ''
?>>Galápagos</option>
</select>
</div>

<div class="form-group">
<label>Ciudad *</label>
<select name="city" id="city" required>
<option value="">Primero seleccione provincia...</option>
</select>
</div>
</div>

<div class="form-group">
<label>Dirección *</label>
<input type="text" name="address" value="<?= $institution['address'] ?>" required>
</div>

<div class="grid">
<div class="form-group">
<label>Teléfono</label>
<input type="tel" name="phone" value="<?= $institution['phone'] ?? '' ?>">
</div>

<div class="form-group">
<label>Email *</label>
<input type="email" name="email" value="<?= $institution['email'] ?>" required>
</div>

<div class="form-group">
<label>Sitio Web</label>
<input type="text" name="website" id="website" value="<?= $institution['website'] ?? '' ?>" placeholder="Ej:
www.iepomasqui.com">
</div>
</div>

<button type="submit">Guardar Cambios</button>
</form>
</div>

<div class="card">
<h2>■ Jornadas de la Institución</h2>

<div style="margin-bottom: 20px;">
<strong>Jornadas Activas:</strong><br>
<?php
$assignedShifts = $this->institutionShiftModel->getByInstitution(1);
if(empty($assignedShifts)):
?>
<em style="color: #999;">No hay jornadas asignadas</em>
<?php else: ?>
<?php foreach($assignedShifts as $shift): ?>
<span class="badge">
<?= ucfirst($shift['name']) ?>
<form method="POST" action="?action=remove_institution_shift" style="display: inline;" onsubmit="return
confirmRemoveShift(event, ' <?= ucfirst($shift['name']) ?>')">
<input type="hidden" name="shift_id" value="<?= $shift['id'] ?>">
<button type="submit" class="btn-remove">x</button>
</form>
</span>
<?php endforeach; ?>

```

```

<?php endif; ?>
</div>

<form method="POST" action="?action=assign_institution_shift" style="display: flex; gap: 10px; align-items: flex-end;">
<div style="flex: 1;">
<label>Asignar Nueva Jornada</label>
<select name="shift_id" required>
<option value="">Seleccionar...</option>
<?php foreach($allShifts as $shift): ?>

<?php if(!in_array($shift['id'], $assignedShiftIds)): ?>
<option value="<?=$shift['id'] ?>"><?=$shift['name'] ?></option>
<?php endif; ?>
<?php endforeach; ?>
</select>
</div>
<button type="submit">Asignar Jornada</button>
</form>
</div>
</div>

<script>
const cities = {
'Pichincha': ['Quito', 'Cayambe', 'Mejía', 'Pedro Moncayo', 'Rumiñahui', 'San Miguel de los Bancos', 'Pedro Vicente Maldonado', 'Puerto Quito'],
'Guayas': ['Guayaquil', 'Durán', 'Milagro', 'Daule', 'Samborondón', 'Balao', 'Balzar', 'Colimes', 'El Empalme', 'El Triunfo', 'Naranjal', 'Naranjito', 'Palestina', 'Pedro Carbo', 'Santa Lucía', 'Yaguachi', 'Playas', 'Simón Bolívar', 'Coronel Marcelino Maridueña', 'Lomas de Sargentillo', 'Nobol', 'General Antonio Elizalde', 'Isidro Ayora'],
'Azuay': ['Cuenca', 'Gualaceo', 'Paute', 'Sigüig', 'Girón', 'Santa Isabel', 'Nabón', 'Oña', 'Pucará', 'San Fernando', 'Chordeleg', 'El Pan', 'Guachapala', 'Camilo Ponce Enríquez', 'Sevilla de Oro'],
'Manabí': ['Portoviejo', 'Manta', 'Chone', 'El Carmen', 'Flavio Alfaro', 'Jipijapa', 'Junín', 'Montecristi', 'Paján', 'Pichincha', 'Rocafuerte', 'Santa Ana', 'Sucre', 'Tosagua', 'Veinticuatro de Mayo', 'Pedernales', 'Bolívar', 'Olmedo', 'Puerto López', 'Jama', 'Jaramijó', 'San Vicente'],
'El Oro': ['Machala', 'Pasaje', 'Santa Rosa', 'Huaquillas', 'Arenillas', 'Atahualpa', 'Balsas', 'Chilla', 'El Guabo', 'Marcabelí', 'Piñas', 'Portovelo', 'Zaruma', 'Las Lajas'],
'Los Ríos': ['Babahoyo', 'Quevedo', 'Baba', 'Montalvo', 'Puebloviejo', 'Urdaneta', 'Ventanas', 'Vinces', 'Palenque', 'Buena Fe', 'Valencia', 'Mocache', 'Quinsaloma'],
'Tungurahua': ['Ambato', 'Baños de Agua Santa', 'Pelileo', 'Píllaro', 'Cevallos', 'Mocha', 'Patate', 'Quero', 'Tisaleo'],
'Esmeraldas': ['Esmeraldas', 'Quinindé', 'Atacames', 'Eloy Alfaro', 'Muisne', 'San Lorenzo', 'Rioverde', 'La Concordia'],
'Chimborazo': ['Riobamba', 'Alausí', 'Guano', 'Colta', 'Chambo', 'Chunchi', 'Guamote', 'Pallatanga', 'Penipe', 'Cumandá'],
'Imbabura': ['Ibarra', 'Otavalo', 'Cotacachi', 'Antonio Ante', 'Pimampiro', 'San Miguel de Urcuquí'],
'Cotopaxi': ['Latacunga', 'La Maná', 'Pujilí', 'Salcedo', 'Saquisilí', 'Pangua', 'Sigchos'],
'Santo Domingo de los Tsáchilas': ['Santo Domingo'],
'Santa Elena': ['La Libertad', 'Salinas', 'Santa Elena'],
'Loja': ['Loja', 'Catamayo', 'Macará', 'Saraguro', 'Calvas', 'Celica', 'Chaguarpamba', 'Espíndola', 'Gonzanamá', 'Paltas', 'Puyango', 'Sozoranga', 'Zapotillo', 'Pindal', 'Quilanga', 'Olmedo'],
'Cañar': ['Azogues', 'Cañar', 'La Troncal', 'Biblián', 'El Tambo', 'Déleg', 'Suscal'],
'Sucumbíos': ['Nueva Loja (Lago Agrio)', 'Shushufindi', 'Gonzalo Pizarro', 'Putumayo', 'Sucumbíos', 'Cascales', 'Cuyabeno'],
'Orellana': ['Francisco de Orellana (Coca)', 'La Joya de los Sachas', 'Aguarico', 'Loreto'],
'Carchi': ['Tulcán', 'Montúfar', 'Bolívar', 'Espejo', 'Mira', 'San Pedro de Huaca'],
'Bolívar': ['Guaranda', 'San Miguel', 'Chillanes', 'Chimbo', 'Echeandía', 'Caluma', 'Las Naves'],
'Napo': ['Tena', 'Archidona', 'El Chaco', 'Quijos', 'Carlos Julio Arosemena Tola'],
'Pastaza': ['Puyo', 'Mera', 'Arajuno', 'Santa Clara'],
'Morona Santiago': ['Macas', 'Gualaquiza', 'Sucúa', 'Limón Indanza', 'Palora', 'Santiago', 'Huamboya', 'San Juan Bosco', 'Taisha', 'Logroño', 'Pablo Sexto', 'Tiwiwintza'],
'Zamora Chinchipe': ['Zamora', 'Yantzaza', 'Chinchipe', 'Nangaritzta', 'Yacuambi', 'El Pangui', 'Centinela del Cóndor', 'Palanda', 'Paquisha'],
'Galápagos': ['Puerto Baquerizo Moreno', 'Puerto Ayora', 'Puerto Villamil']
};

```

```

const currentCity = '<?=$institution['city'] ?? ' ' ?>';
const currentProvince = '<?=$institution['province'] ?? ' ' ?>';

document.getElementById('province').addEventListener('change', function() {
const province = this.value;
const citySelect = document.getElementById('city');

citySelect.innerHTML = '<option value="">Seleccionar ciudad...</option>';

if (province && cities[province]) {
cities[province].forEach(city => {
const option = document.createElement('option');
option.value = city;
option.textContent = city;
citySelect.appendChild(option);
});
}

// Cargar ciudades al inicio si hay provincia seleccionada
if (currentProvince && cities[currentProvince]) {
const citySelect = document.getElementById('city');
citySelect.innerHTML = '<option value="">Seleccionar ciudad...</option>';
cities[currentProvince].forEach(city => {
const option = document.createElement('option');
option.value = city;
option.textContent = city;
if (city === currentCity) {
option.selected = true;
}
citySelect.appendChild(option);
});
}

// Autocompletar URL
document.getElementById('website').addEventListener('blur', function() {
let url = this.value.trim();

if (url && !url.match(/^https?:\/\//)) {
if (!url.startsWith('www.')) {
url = 'www.' + url;
}
this.value = 'https://' + url;
}
});

function confirmRemoveShift(event, shiftName) {
event.preventDefault();

const modal = document.createElement('div');
modal.style.cssText = 'position: fixed; top: 0; left: 0; width: 100%; height: 100%; background:
rgba(0,0,0,0.5); display: flex; align-items: center; justify-content: center; z-index: 9999;';

const modalContent = document.createElement('div');
modalContent.style.cssText = 'background: white; padding: 30px; border-radius: 8px; max-width: 500px;
box-shadow: 0 4px 20px rgba(0,0,0,0.3);';

modalContent.innerHTML = `
<h3 style="margin: 0 0 15px 0; color: #dc3545;">■ ■ Eliminar Jornada</h3>
<p style="margin: 0 0 20px 0; color: #666;">
¿Está seguro de eliminar la jornada <strong>${shiftName}</strong> de esta institución?
</p>
<div style="display: flex; gap: 10px; justify-content: flex-end;">
<button type="button" id="cancelBtn" style="padding: 10px 20px; background: #6c757d; color: white; border:
none; border-radius: 4px; cursor: pointer;">

```

```

Cancelar
</button>
<button type="button" id="confirmBtn" style="padding: 10px 20px; background: #dc3545; color: white; border:
none; border-radius: 4px; cursor: pointer;">
Sí, Eliminar
</button>
</div>
`;

modal.appendChild(modalContent);
document.body.appendChild(modal);

const form = event.target;

document.getElementById('confirmBtn').onclick = function() {
document.body.removeChild(modal);
form.submit();
};

document.getElementById('cancelBtn').onclick = function() {
document.body.removeChild(modal);
};

return false;
}
</script>
</body>
</html>

```

## 64. ArchImp/views/justifications/my\_list.php

```

<?php Security::requireLogin(); ?>
<!DOCTYPE html>
<html lang="es">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Mis Justificaciones - EcuAsist</title>
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet">
<style>
body { padding-top: 70px; background: #f4f4f4; }
.container { max-width: 1200px; margin: 30px auto; }
.card { background: white; padding: 30px; border-radius: 8px; box-shadow: 0 2px 10px rgba(0,0,0,0.1);
margin-bottom: 20px; }
.badge-pending { background: #ffc107; color: #000; padding: 6px 12px; border-radius: 12px; font-size: 12px;
font-weight: bold; }
.badge-approved { background: #28a745; color: white; padding: 6px 12px; border-radius: 12px; font-size:
12px; font-weight: bold; }
.badge-rejected { background: #dc3545; color: white; padding: 6px 12px; border-radius: 12px; font-size:
12px; font-weight: bold; }
.table th { background: #f8f9fa; font-weight: bold; }
.empty-state {
text-align: center;
padding: 60px 20px;
color: #999;
}
.empty-state .icon {
font-size: 64px;
margin-bottom: 20px;
opacity: 0.5;
}
</style>

```

```

</head>
<body>
<?php include BASE_PATH . '/views/partials/navbar.php'; ?>

<div class="container">
<div class="card">
<div class="d-flex justify-content-between align-items-center mb-4">
<h2 class="mb-0">■ Mis Justificaciones</h2>
</div>

<?php if(isset($_GET['success'])): ?>
<div class="alert alert-success alert-dismissible fade show">
✓ Justificación enviada correctamente. Está pendiente de revisión.
<button type="button" class="btn-close" data-bs-dismiss="alert"></button>
</div>
<?php endif; ?>

<?php if(empty($justifications)): ?>
<div class="empty-state">
<div class="icon">■</div>
<h4>No tienes justificaciones enviadas</h4>
<p class="text-muted">
Las justificaciones que envíes aparecerán aquí.<br>
Puedes justificar ausencias desde "Mi Asistencia".
</p>
</div>
<?php else: ?>
<div class="table-responsive">
<table class="table table-hover">
<thead>
<tr>
<th>#</th>
<th>Fecha</th>
<th>Motivo</th>
<th>Documento</th>
<th>Estado</th>
<th>Fecha Envío</th>
<th>Revisado Por</th>
<th>Observaciones</th>
</tr>
</thead>
<tbody>
<?php
$counter = 1;
foreach($justifications as $just):
?>
<tr>
<td><?=$counter++ ?></td>
<td><?=$date('d/m/Y', strtotime($just['attendance_date'])) ?></td>
<td><?=$htmlspecialchars($just['reason']) ?></td>
<td>
<?php if($just['document_path']): ?>
<a href="<?=$BASE_URL ?>/<?=$just['document_path'] ?>"
target="_blank"
class="btn btn-sm btn-outline-primary">
■ Ver Documento
</a>
<?php else: ?>
<span class="text-muted">Sin documento</span>
<?php endif; ?>
</td>
<td>
<?php if($just['status'] === 'pendiente'): ?>
<span class="badge-pending">■ Pendiente</span>
<?php elseif($just['status'] === 'aprobado'): ?>

```

```

<span class="badge-approved">✓ Aprobado</span>
<?php else: ?>
<span class="badge-rejected">✗ Rechazado</span>
<?php endif; ?>
</td>
<td><?= date('d/m/Y H:i', strtotime($just['submitted_at'])) ?></td>
<td>
<?php if($just['reviewed_by']): ?>
<?= htmlspecialchars($just['reviewer_name']) ?>
<br>
<small class="text-muted">
<?= date('d/m/Y H:i', strtotime($just['reviewed_at'])) ?>
</small>
<?php else: ?>
<span class="text-muted">--</span>
<?php endif; ?>
</td>
<td>
<?php if($just['review_notes']): ?>
<small><?= htmlspecialchars($just['review_notes']) ?></small>
<?php else: ?>
<span class="text-muted">--</span>
<?php endif; ?>
</td>
</tr>
<?php endforeach; ?>
</tbody>
</table>
</div>

<!-- Resumen estadístico -->
<div class="row mt-4">
<div class="col-md-4">
<div class="card bg-warning bg-opacity-10 border-warning">
<div class="card-body text-center">
<h3 class="mb-0">
<?= count(array_filter($justifications, fn($j) => $j['status'] === 'pendiente')) ?>
</h3>
<small class="text-muted">Pendientes</small>
</div>
</div>
</div>
<div class="col-md-4">
<div class="card bg-success bg-opacity-10 border-success">
<div class="card-body text-center">
<h3 class="mb-0">
<?= count(array_filter($justifications, fn($j) => $j['status'] === 'aprobado')) ?>
</h3>
<small class="text-muted">Aprobadas</small>
</div>
</div>
</div>
<div class="col-md-4">
<div class="card bg-danger bg-opacity-10 border-danger">
<div class="card-body text-center">
<h3 class="mb-0">
<?= count(array_filter($justifications, fn($j) => $j['status'] === 'rechazado')) ?>
</h3>
<small class="text-muted">Rechazadas</small>
</div>
</div>
</div>
</div>
<?php endif; ?>
</div>

```

```

<div class="text-center">
<a href="?action=my_attendance" class="btn btn-secondary">
← Volver a Mi Asistencia
</a>
</div>
</div>

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"></script>
</body>
</html>

```

## 65. ArchImp/views/justifications/pending.php

```

<?php Security::requireLogin(); ?>
<!DOCTYPE html>
<html lang="es">
<head>
<meta charset="UTF-8">
<title>Justificaciones Pendientes - EcuAsist</title>
<style>
* { margin: 0; padding: 0; box-sizing: border-box; }
body { font-family: Arial, sans-serif; background: #f4f4f4; }
.navbar { background: #007bff; color: white; padding: 15px 30px; display: flex; justify-content:
space-between; align-items: center; }
.navbar h1 { font-size: 24px; }
.navbar a { color: white; text-decoration: none; margin-left: 20px; }
.container { max-width: 1200px; margin: 30px auto; padding: 0 20px; }
.card { background: white; padding: 30px; border-radius: 8px; box-shadow: 0 2px 10px rgba(0,0,0,0.1);
margin-bottom: 20px; }
table { width: 100%; border-collapse: collapse; margin-top: 15px; }
th, td { padding: 12px; text-align: left; border-bottom: 1px solid #ddd; }
th { background: #f8f9fa; font-weight: bold; }
.btn { padding: 6px 12px; border: none; border-radius: 4px; cursor: pointer; font-size: 13px; margin-right:
5px; }
.btn-approve { background: #28a745; color: white; }
.btn-reject { background: #dc3545; color: white; }
.btn-view { background: #007bff; color: white; }
.modal { display: none; position: fixed; top: 0; left: 0; width: 100%; height: 100%; background:
rgba(0,0,0,0.5); z-index: 1000; }
.modal-content { background: white; margin: 50px auto; padding: 30px; border-radius: 8px; max-width: 600px;
}
.modal-close { float: right; font-size: 28px; cursor: pointer; }
textarea { width: 100%; padding: 10px; border: 1px solid #ddd; border-radius: 4px; margin: 10px 0; }
.success { background: #d4edda; color: #155724; padding: 15px; border-radius: 4px; margin-bottom: 20px; }
</style>
</head>
<body>
<?php include BASE_PATH . '/views/partials/navbar.php'; ?>

<div class="container">
<?php if(isset($_GET['success'])): ?>
<div class="success">✓ Justificación procesada correctamente</div>
<?php endif; ?>

<div class="card">
<h2 style="margin-bottom: 20px;">Justificaciones por Revisar (<?= count($justifications) ?>)</h2>

<?php if(count($justifications) > 0): ?>
<table>
<thead>
<tr>
<th>Estudiante</th>

```

```

<th>Curso</th>
<th>Fecha Ausencia</th>
<th>Asignatura</th>
<th>Presentado por</th>
<th>Fecha Solicitud</th>
<th>Acciones</th>
</tr>
</thead>
<tbody>
<?php foreach($justifications as $just): ?>
<tr>
<td><?= $just['student_name'] ?></td>
<td><?= $just['course_name'] ?? '-' ?></td>
<td><?= date('d/m/Y', strtotime($just['attendance_date'])) ?></td>
<td><?= $just['subject_name'] ?></td>
<td><?= $just['submitted_by_name'] ?></td>
<td><?= date('d/m/Y H:i', strtotime($just['created_at'])) ?></td>
<td>
<button class="btn btn-view" onclick="viewJustification(<?= $just['id'] ?>, '<?= addslashes($just['reason'])
?>', '<?= $just['document_path'] ?? '' ?>')">Ver</button>
</td>
</tr>
<?php endforeach; ?>
</tbody>
</table>
<?php else: ?>
<p style="color: #666; text-align: center; padding: 40px;">No hay justificaciones pendientes</p>
<?php endif; ?>
</div>
</div>

<!-- Modal -->
<div id="modal" class="modal">
<div class="modal-content">
<span class="modal-close" onclick="closeModal()">&times;</span>
<h3>Revisar Justificación</h3>
<div id="modal-body"></div>

<form method="POST" action="?action=review_justification">
<input type="hidden" name="justification_id" id="justification_id">
<textarea name="notes" placeholder="Notas de revisión (opcional)"></textarea>

<div style="margin-top: 15px;">
<button type="submit" name="review_action" value="approve" class="btn btn-approve">✓ Aprobar</button>
<button type="submit" name="review_action" value="reject" class="btn btn-reject">✗ Rechazar</button>
</div>
</form>
</div>
</div>

<script>
function viewJustification(id, reason, docPath) {
const justificationIdInput = window.document.querySelector('#justification_id');
const modalBody = window.document.querySelector('#modal-body');
const modal = window.document.querySelector('#modal');

if (!justificationIdInput || !modalBody || !modal) {
console.error('Elementos del modal no encontrados');
return;
}

justificationIdInput.value = id;

let html = '<p><strong>Motivo:</strong></p><p>' + reason + '</p>';

```



```

if (docPath && docPath.length > 0) {
html += '<p style="margin-top: 15px;"><strong>Documento:</strong> <a href="<?= BASE_URL ?>/'+ docPath + '"
target="_blank" style="color: #007bff; text-decoration: underline;">■ Ver archivo adjunto</a></p>';
} else {
html += '<p style="margin-top: 15px; color: #999;"><em>Sin documento adjunto</em></p>';
}

modalBody.innerHTML = html;
modal.style.display = 'block';
}

function closeModal() {
const modal = window.document.querySelector('#modal');
if (modal) {
modal.style.display = 'none';
}
}

window.onclick = function(event) {
const modal = window.document.querySelector('#modal');
if (modal && event.target == modal) {
closeModal();
}
}
</script>
</body>
</html>

```

## 66. ArchImp/views/justifications/submit.php

```

<?php Security::requireLogin(); ?>
<!DOCTYPE html>
<html lang="es">
<head>
<meta charset="UTF-8">
<title>Justificar Ausencia - EcuAsist</title>
<style>
* { margin: 0; padding: 0; box-sizing: border-box; }
body { font-family: Arial, sans-serif; background: #f4f4f4; }
.navbar { background: #007bff; color: white; padding: 15px 30px; display: flex; justify-content:
space-between; align-items: center; }
.navbar h1 { font-size: 24px; }
.navbar a { color: white; text-decoration: none; margin-left: 20px; }
.container { max-width: 700px; margin: 30px auto; padding: 0 20px; }
.card { background: white; padding: 30px; border-radius: 8px; box-shadow: 0 2px 10px rgba(0,0,0,0.1); }
.form-group { margin-bottom: 20px; }
label { display: block; margin-bottom: 5px; font-weight: bold; color: #333; }
input, textarea { width: 100%; padding: 10px; border: 1px solid #ddd; border-radius: 4px; }
textarea { min-height: 100px; resize: vertical; }
button { padding: 12px 30px; background: #28a745; color: white; border: none; border-radius: 4px; cursor:
pointer; }
button:hover { background: #218838; }
.info { background: #d1ecf1; color: #0c5460; padding: 15px; border-radius: 4px; margin-bottom: 20px; }
</style>
</head>
<body>
<?php include BASE_PATH . '/views/partials/navbar.php'; ?>

<div class="container">
<div class="card">
<div class="info">
■ Complete el formulario para justificar una ausencia. Puede adjuntar documentos de respaldo (certificados

```

```

médicos, etc.)
</div>

<form method="POST" enctype="multipart/form-data">
<div class="form-group">
<label>Motivo de la Ausencia *</label>
<textarea name="reason" required placeholder="Describe el motivo de la ausencia..."></textarea>
</div>

<div class="form-group">
<label>Documento de Respaldo (opcional)</label>
<input type="file" name="document" accept=".pdf,.jpg,.jpeg,.png">
<small style="color: #666; display: block; margin-top: 5px;">
Formatos permitidos: PDF, JPG, PNG - Máximo 5MB
</small>
</div>

<button type="submit">Enviar Justificación</button>
</form>
</div>
</div>
</body>
</html>

```

## 67. ArchImp/views/partials/navbar.php

```

<?php
$current_page = $_GET['action'] ?? 'dashboard';
?>
<style>
.main-navbar { background: #007bff; color: white; box-shadow: 0 2px 5px rgba(0,0,0,0.1); position: sticky;
top: 0; z-index: 1000; }
.navbar-container { max-width: 1400px; margin: 0 auto; padding: 0 20px; display: flex; justify-content:
space-between; align-items: center; }
.navbar-brand { font-size: 24px; font-weight: bold; padding: 15px 0; }
.navbar-menu { display: flex; gap: 20px; align-items: center; }
.navbar-section { position: relative; }
.navbar-section > a { color: white; text-decoration: none; padding: 15px 12px; display: block; cursor:
pointer; font-size: 14px; font-weight: 500; }
.navbar-section:hover > a { background: rgba(255,255,255,0.2); }
.dropdown { display: none; position: absolute; top: 100%; left: 0; background: white; min-width: 220px;
box-shadow: 0 4px 15px rgba(0,0,0,0.2); border-radius: 4px; z-index: 1000; }
.navbar-section:hover .dropdown { display: block; }
.dropdown a { color: #333; text-decoration: none; padding: 12px 20px; display: block; transition: background
0.2s; border-left: 3px solid transparent; }
.dropdown a:hover { background: #f8f9fa; border-left-color: #007bff; }
.dropdown a.active { background: #e7f3ff; border-left-color: #007bff; font-weight: 600; }
.dropdown-divider { height: 1px; background: #e0e0e0; margin: 5px 0; }
.navbar-user { display: flex; align-items: center; gap: 15px; }
.navbar-user a { color: white; text-decoration: none; padding: 8px 15px; border-radius: 4px; transition:
background 0.3s; }
.navbar-user a:hover { background: rgba(255,255,255,0.2); }
@media (max-width: 992px) {
.navbar-container { flex-direction: column; align-items: flex-start; }
.navbar-menu { width: 100%; flex-direction: column; gap: 0; align-items: stretch; }
.navbar-section { width: 100%; }
.navbar-section > a { width: 100%; }
.dropdown { position: static; width: 100%; box-shadow: none; background: rgba(255,255,255,0.1); }
.dropdown a { color: white; }
.dropdown a:hover { background: rgba(255,255,255,0.1); }
}
</style>

<nav class="main-navbar">

```

```

<div class="navbar-container">
<div class="navbar-brand">EcuAsist 2026</div>

<div class="navbar-menu">
<div class="navbar-section">
<a href="?action=dashboard">■ Inicio</a>
</div>

<!-- ASISTENCIA -->
<?php if(Security::hasRole(['docente', 'autoridad', 'inspector', 'estudiante'])): ?>
<div class="navbar-section">
<a>■ Asistencia ■</a>
<div class="dropdown">
<?php if(Security::hasRole(['docente', 'autoridad'])): ?>
<a href="?action=attendance_register" class="<?= $current_page == 'attendance_register' ? 'active' : '' ?>">
■ Registrar Asistencia
</a>
<?php endif; ?>

<?php if(Security::hasRole(['docente', 'inspector', 'autoridad'])): ?>
<a href="?action=attendance_view" class="<?= $current_page == 'attendance_view' ? 'active' : '' ?>">
■ Ver Asistencias
</a>
<a href="?action=attendance_calendar" class="<?= $current_page == 'attendance_calendar' ? 'active' : '' ?>">
■ Calendario
</a>
<?php endif; ?>

<?php if(Security::hasRole('estudiante')): ?>
<a href="?action=my_attendance" class="<?= $current_page == 'my_attendance' ? 'active' : '' ?>">
■ Mi Asistencia
</a>
<?php endif; ?>
</div>
</div>
<?php endif; ?>

<!-- JUSTIFICACIONES -->
<?php if(Security::hasRole(['estudiante', 'autoridad', 'inspector'])): ?>
<div class="navbar-section">
<a>■ Justificaciones ■</a>
<div class="dropdown">
<?php if(Security::hasRole('estudiante')): ?>
<a href="?action=my_justifications" class="<?= $current_page == 'my_justifications' ? 'active' : '' ?>">
■ Mis Justificaciones
</a>
<?php endif; ?>

<?php if(Security::hasRole(['autoridad', 'inspector'])): ?>
<a href="?action=pending_justifications" class="<?= $current_page == 'pending_justifications' ? 'active' : '' ?>">
■ Revisar Justificaciones
</a>
<?php endif; ?>
</div>
</div>
<?php endif; ?>

<!-- ADMINISTRACIÓN -->
<?php if(Security::hasRole('autoridad')): ?>
<div class="navbar-section">
<a>■■ Administración ■</a>
<div class="dropdown">
<a href="?action=users" class="<?= $current_page == 'users' ? 'active' : '' ?>">
■ Gestión de Usuarios

```

```

</a>
<a href="?action=academic" class="<?= $current_page == 'academic' ? 'active' : '' ?>">
■ Configuración Académica
</a>
<a href="?action=assignments" class="<?= $current_page == 'assignments' ? 'active' : '' ?>">
■ Asignar Docente-Materia
</a>
<a href="?action=tutor_management" class="<?= $current_page == 'tutor_management' ? 'active' : '' ?>">
■■■ Asignar Docente Tutor
</a>
<a href="?action=manage_representatives" class="<?= $current_page == 'manage_representatives' ? 'active' : '' ?>">
' ' ?>">
■■■■■■■ Gestión Representantes
</a>
<div class="dropdown-divider"></div>
<a href="?action=backups" class="<?= $current_page == 'backups' ? 'active' : '' ?>">
■ RespalDOS
</a>
<a href="?action=schedules" class="<?= $current_page == 'schedules' ? 'active' : '' ?>">
■ Horarios de Clases
</a>
<a href="?action=institution" class="<?= $current_page == 'institution' ? 'active' : '' ?>">
■ Configuración de Institución
</a>
</div>
</div>
<?php endif; ?>

<!-- REPORTE Y ESTADÍSTICAS -->
<?php if(Security::hasRole('autoridad')): ?>
<div class="navbar-section">
<a>■ Reportes ■</a>
<div class="dropdown">
<a href="?action=reports" class="<?= $current_page == 'reports' ? 'active' : '' ?>">
■ Generar Reportes
</a>
<a href="?action=stats" class="<?= $current_page == 'stats' ? 'active' : '' ?>">
■ Estadísticas
</a>
</div>
</div>
</div>
<?php endif; ?>

<!-- REPRESENTANTES -->
<?php if(Security::hasRole('representante')): ?>
<div class="navbar-section">
<a href="?action=my_children">■■■■■■■ Mis Representados</a>
</div>
<?php endif; ?>
</div>

<div class="navbar-user">
<a href="?action=profile">
■ <?= $_SESSION['first_name'] . ' ' . $_SESSION['last_name'] ?>
</a>
<a href="?action=logout">Cerrar sesión</a>
</div>
</div>
</nav>

```

## 68. ArchImp/views/profile/change\_password.php

```

<?php Security::requireLogin(); ?>
<!DOCTYPE html>
<html lang="es">
<head>
<meta charset="UTF-8">
<title>Cambiar Contraseña - EcuAsist</title>
<style>
* { margin: 0; padding: 0; box-sizing: border-box; }
body { font-family: Arial, sans-serif; background: #f4f4f4; }
.navbar { background: #007bff; color: white; padding: 15px 30px; display: flex; justify-content:
space-between; align-items: center; }
.navbar h1 { font-size: 24px; }
.navbar a { color: white; text-decoration: none; margin-left: 20px; }
.container { max-width: 600px; margin: 30px auto; padding: 0 20px; }
.card { background: white; padding: 30px; border-radius: 8px; box-shadow: 0 2px 10px rgba(0,0,0,0.1); }
.form-group { margin-bottom: 20px; }
label { display: block; margin-bottom: 5px; font-weight: bold; color: #333; }
input { width: 100%; padding: 10px; border: 1px solid #ddd; border-radius: 4px; }
button { padding: 12px 30px; background: #007bff; color: white; border: none; border-radius: 4px; cursor:
pointer; }
button:hover { background: #0056b3; }
.btn-cancel { background: #6c757d; margin-left: 10px; }
.btn-cancel:hover { background: #5a6268; }
.error { background: #f8d7da; color: #721c24; padding: 15px; border-radius: 4px; margin-bottom: 20px; }
</style>
</head>
<body>
<?php include BASE_PATH . '/views/partials/navbar.php'; ?>

<div class="container">
<div class="card">
<?php if(isset($error)): ?>
<div class="error"><?= $error ?></div>
<?php endif; ?>

<h2 style="margin-bottom: 20px;">Nueva Contraseña</h2>
<form method="POST">
<div class="form-group">
<label>Contraseña Actual</label>
<input type="password" name="current_password" required>
</div>

<div class="form-group">
<label>Nueva Contraseña</label>
<input type="password" name="new_password" minlength="6" required>
</div>

<div class="form-group">
<label>Confirmar Nueva Contraseña</label>
<input type="password" name="confirm_password" minlength="6" required>
</div>

<button type="submit">Cambiar Contraseña</button>
<button type="button" class="btn-cancel" onclick="location.href='?action=profile'">Cancelar</button>
</form>
</div>
</div>
</body>
</html>

```

## 69. ArchImp/views/profile/edit.php

```

<?php Security::requireLogin(); ?>
<!DOCTYPE html>

```

```

<html lang="es">
<head>
<meta charset="UTF-8">
<title>Editor Perfil - EcuAsist</title>
<style>
* { margin: 0; padding: 0; box-sizing: border-box; }
body { font-family: Arial, sans-serif; background: #f4f4f4; }
.navbar { background: #007bff; color: white; padding: 15px 30px; display: flex; justify-content:
space-between; align-items: center; }
.navbar h1 { font-size: 24px; }
.navbar a { color: white; text-decoration: none; margin-left: 20px; }
.container { max-width: 600px; margin: 30px auto; padding: 0 20px; }
.card { background: white; padding: 30px; border-radius: 8px; box-shadow: 0 2px 10px rgba(0,0,0,0.1); }
.form-group { margin-bottom: 20px; }
label { display: block; margin-bottom: 5px; font-weight: bold; color: #333; }
input { width: 100%; padding: 10px; border: 1px solid #ddd; border-radius: 4px; }
button { padding: 12px 30px; background: #28a745; color: white; border: none; border-radius: 4px; cursor:
pointer; }
button:hover { background: #218838; }
.btn-cancel { background: #6c757d; margin-left: 10px; }
.btn-cancel:hover { background: #5a6268; }
</style>
</head>
<body>
<?php include BASE_PATH . '/views/partials/navbar.php'; ?>

<div class="container">
<div class="card">
<h2 style="margin-bottom: 20px;">Actualizar Información</h2>
<form method="POST">
<div class="form-group">
<label>Nombres</label>
<input type="text" name="first_name" value="<?= $user['first_name'] ?>" required>
</div>

<div class="form-group">
<label>Apellidos</label>
<input type="text" name="last_name" value="<?= $user['last_name'] ?>" required>
</div>

<div class="form-group">
<label>Email</label>
<input type="email" name="email" value="<?= $user['email'] ?>" required>
</div>

<div class="form-group">
<label>Teléfono</label>
<input type="text" name="phone" value="<?= $user['phone'] ?>">
</div>

<button type="submit">Guardar Cambios</button>
<button type="button" class="btn-cancel" onclick="location.href='?action=profile'">Cancelar</button>
</form>
</div>
</div>
</body>
</html>

```

## 70. ArchImp/views/profile/view.php

```

<?php Security::requireLogin(); ?>
<!DOCTYPE html>
<html lang="es">

```

```

<head>
<meta charset="UTF-8">
<title>Mi Perfil - EcuAsist</title>
<style>
* { margin: 0; padding: 0; box-sizing: border-box; }
body { font-family: Arial, sans-serif; background: #f4f4f4; }
.navbar { background: #007bff; color: white; padding: 15px 30px; display: flex; justify-content:
space-between; align-items: center; }
.navbar h1 { font-size: 24px; }
.navbar a { color: white; text-decoration: none; margin-left: 20px; }
.container { max-width: 800px; margin: 30px auto; padding: 0 20px; }
.card { background: white; padding: 30px; border-radius: 8px; box-shadow: 0 2px 10px rgba(0,0,0,0.1);
margin-bottom: 20px; }
.profile-header { display: flex; align-items: center; gap: 20px; margin-bottom: 30px; }
.avatar { width: 100px; height: 100px; border-radius: 50%; background: linear-gradient(135deg, #667eea 0%,
#764ba2 100%); display: flex; align-items: center; justify-content: center; color: white; font-size: 40px;
font-weight: bold; }
.info-row { display: flex; padding: 15px 0; border-bottom: 1px solid #f0f0f0; }
.info-row:last-child { border-bottom: none; }
.label { font-weight: bold; color: #666; width: 150px; }
.value { color: #333; flex: 1; }
.btn { display: inline-block; padding: 10px 20px; background: #007bff; color: white; text-decoration: none;
border-radius: 4px; margin-right: 10px; margin-top: 10px; }
.btn:hover { background: #0056b3; }
.btn-secondary { background: #6c757d; }
.btn-secondary:hover { background: #5a6268; }
.badge { display: inline-block; padding: 5px 10px; border-radius: 3px; background: #28a745; color: white;
font-size: 12px; margin: 2px; }
.success { background: #d4edda; color: #155724; padding: 15px; border-radius: 4px; margin-bottom: 20px; }
</style>
</head>
<body>
<?php include BASE_PATH . '/views/partials/navbar.php'; ?>

<div class="container">
<?php if(isset($_GET['success'])): ?>
<div class="success">✓ Perfil actualizado correctamente</div>
<?php endif; ?>

<?php if(isset($_GET['password_changed'])): ?>
<div class="success">✓ Contraseña cambiada correctamente</div>
<?php endif; ?>

<div class="card">
<div class="profile-header">
<div class="avatar">
<?= strtoupper(substr($user['first_name'], 0, 1) . substr($user['last_name'], 0, 1)) ?>
</div>
<div>
<h2><?= $user['first_name'] . ' ' . $user['last_name'] ?></h2>
<p style="color: #666;">@<?= $user['username'] ?></p>
</div>
</div>

<div class="info-row">
<span class="label">Email:</span>
<span class="value"><?= $user['email'] ?></span>
</div>

<div class="info-row">
<span class="label">Cédula:</span>
<span class="value"><?= $user['dni'] ?? 'No registrada' ?></span>
</div>

<div class="info-row">

```

```

<span class="label">Teléfono:</span>
<span class="value"><?= $user['phone'] ?? 'No registrado' ?></span>
</div>

<div class="info-row">
<span class="label">Roles:</span>
<span class="value">
<?php foreach($roles as $role): ?>
<span class="badge"><?= ucfirst($role) ?></span>
<?php endforeach; ?>
</span>
</div>

<div class="info-row">
<span class="label">Miembro desde:</span>
<span class="value"><?= date('d/m/Y', strtotime($user['created_at'])) ?></span>
</div>

<div style="margin-top: 20px;">
<a href="?action=edit_profile" class="btn">Editar Perfil</a>
<a href="?action=change_password" class="btn btn-secondary">Cambiar Contraseña</a>
</div>
</div>
</div>
</body>
</html>

```

## 71. ArchImp/views/reports/index.php

```

<?php Security::requireLogin(); ?>
<!DOCTYPE html>
<html lang="es">
<head>
<meta charset="UTF-8">
<title>Reportes - EcuAsist</title>
<style>
* { margin: 0; padding: 0; box-sizing: border-box; }
body { font-family: Arial, sans-serif; background: #f4f4f4; }
.navbar { background: #007bff; color: white; padding: 15px 30px; display: flex; justify-content: space-between; align-items: center; }
.navbar h1 { font-size: 24px; }
.navbar a { color: white; text-decoration: none; margin-left: 20px; }
.container { max-width: 1400px; margin: 30px auto; padding: 0 20px; }
.card { background: white; padding: 30px; border-radius: 8px; box-shadow: 0 2px 10px rgba(0,0,0,0.1); margin-bottom: 20px; }
.form-group { margin-bottom: 15px; }
label { display: block; margin-bottom: 5px; font-weight: bold; color: #333; }
input, select { width: 100%; padding: 10px; border: 1px solid #ddd; border-radius: 4px; }
.btn-group { display: flex; gap: 10px; margin-top: 20px; }
button { flex: 1; padding: 12px; border: none; border-radius: 4px; cursor: pointer; font-size: 16px; font-weight: bold; }
.btn-preview { background: #007bff; color: white; }
.btn-preview:hover { background: #0056b3; }
.btn-pdf { background: #dc3545; color: white; }
.btn-pdf:hover { background: #c82333; }
.btn-excel { background: #28a745; color: white; }
.btn-excel:hover { background: #218838; }
.info { background: #e7f3ff; padding: 15px; border-radius: 4px; border-left: 4px solid #007bff; margin-bottom: 20px; }
h2 { margin-bottom: 20px; color: #333; }
table { width: 100%; border-collapse: collapse; margin-top: 15px; }
th, td { padding: 10px; text-align: left; border-bottom: 1px solid #ddd; font-size: 14px; }
th { background: #f8f9fa; font-weight: bold; }

```



```

.badge { padding: 5px 10px; border-radius: 3px; color: white; font-size: 12px; }
.badge-presente { background: #28a745; }
.badge-ausente { background: #dc3545; }
.badge-tardanza { background: #ffc107; color: #333; }
.badge-justificado { background: #17a2b8; }
.stats { background: #f8f9fa; padding: 15px; border-radius: 4px; margin-bottom: 20px; display: grid;
grid-template-columns: repeat(auto-fit, minmax(150px, 1fr)); gap: 15px; }
.stat-item { text-align: center; }
.stat-number { font-size: 24px; font-weight: bold; color: #007bff; }
.stat-label { font-size: 12px; color: #666; margin-top: 5px; }
</style>
</head>
<body>
<?php include BASE_PATH . '/views/partials/navbar.php'; ?>

<div class="container">
<div class="info">
<strong>■ Tipos de reporte disponibles:</strong><br>
• <strong>Vista Previa:</strong> Ver datos en pantalla antes de exportar<br>
• <strong>PDF:</strong> Reporte formal con formato institucional<br>
• <strong>Excel:</strong> Datos estructurados para análisis
</div>

<div class="card">
<h2>Configuración del Reporte</h2>

<form method="POST" id="report-form">
<div class="form-group">
<label>Curso</label>
<select name="course_id" required>
<option value="">Seleccionar curso...</option>
<?php foreach($courses as $c): ?>
<option value="<?=$c['id'] ?>" <?=$c['id'] == $c['id'] ? 'selected' : '' ?>>
<?=$c['name'] ?> - <?=$c['shift_name'] ?>
</option>
<?php endforeach; ?>
</select>
</div>

<div class="form-group">
<label>Fecha Inicio</label>
<input type="date" name="start_date" value="<?=$startDate ?? date('Y-m-01') ?>" required>
</div>

<div class="form-group">
<label>Fecha Fin</label>
<input type="date" name="end_date" value="<?=$endDate ?? date('Y-m-d') ?>" required>
</div>

<div class="btn-group">
<button type="submit" name="preview" class="btn-preview">
■ Vista Previa
</button>
<button type="button" class="btn-pdf" onclick="generateReport('pdf')" <?=$empty($data) ? 'disabled' : '' ?>>
■ Generar PDF
</button>
<button type="button" class="btn-excel" onclick="generateReport('excel')" <?=$empty($data) ? 'disabled' : '' ?>>
■ Generar Excel
</button>
</div>
</form>
</div>

<?php if(!empty($data)): ?>

```

```

<div class="card">
<h2>Vista Previa del Reporte</h2>

<div class="info">
<strong>Curso:</strong> <?= $course['name'] ?> - <?= $course['shift_name'] ?><br>
<strong>Período:</strong> <?= date('d/m/Y', strtotime($startDate)) ?> al <?= date('d/m/Y',
strtotime($endDate)) ?><br>
<strong>Total Registros:</strong> <?= count($data) ?>
</div>

<?php
$totalPresente = count(array_filter($data, fn($d) => $d['status'] === 'presente'));
$totalAusente = count(array_filter($data, fn($d) => $d['status'] === 'ausente'));
$totalTardanza = count(array_filter($data, fn($d) => $d['status'] === 'tardanza'));
$totalJustificado = count(array_filter($data, fn($d) => $d['status'] === 'justificado'));
$porcentaje = count($data) > 0 ? round(($totalPresente / count($data)) * 100, 1) : 0;
?>

<div class="stats">
<div class="stat-item">
<div class="stat-number" style="color: #28a745;"><?= $totalPresente ?></div>
<div class="stat-label">Presentes</div>
</div>
<div class="stat-item">
<div class="stat-number" style="color: #dc3545;"><?= $totalAusente ?></div>
<div class="stat-label">Ausentes</div>
</div>
<div class="stat-item">
<div class="stat-number" style="color: #ffc107;"><?= $totalTardanza ?></div>
<div class="stat-label">Tardanzas</div>
</div>
<div class="stat-item">
<div class="stat-number" style="color: #17a2b8;"><?= $totalJustificado ?></div>
<div class="stat-label">Justificados</div>
</div>
<div class="stat-item">
<div class="stat-number" style="color: #6f42c1;"><?= $porcentaje ?>%</div>
<div class="stat-label">Asistencia</div>
</div>
</div>

<table>
<thead>
<tr>
<th>Fecha</th>
<th>Estudiante</th>
<th>Asignatura</th>
<th>Hora</th>
<th>Estado</th>
<th>Observación</th>
</tr>
</thead>
<tbody>
<?php foreach($data as $row): ?>
<tr>
<td><?= date('d/m/Y', strtotime($row['date'])) ?></td>
<td><?= $row['student_name'] ?></td>
<td><?= $row['subject_name'] ?></td>
<td><?= $row['hour_period'] ?></td>
<td>
<span class="badge badge-<?= $row['status'] ?>">
<?= ucfirst($row['status']) ?>
</span>
</td>
<td><?= $row['observation'] ?> : '-' ?></td>
</tr>

```

```

</tr>
<?php endforeach; ?>
</tbody>
</table>
</div>
<?php elseif($_SERVER['REQUEST_METHOD'] === 'POST'): ?>
<div class="card">
<p style="text-align: center; color: #666; padding: 40px;">
No hay registros de asistencia para el período seleccionado
</p>
</div>
<?php endif; ?>
</div>

<script>
function generateReport(type) {
const form = document.getElementById('report-form');

if (!form.checkValidity()) {
form.reportValidity();
return;
}

// Crear un formulario temporal
const tempForm = document.createElement('form');
tempForm.method = 'POST';
tempForm.style.display = 'none';

if (type === 'pdf') {
tempForm.action = '?action=generate_pdf';
} else {
tempForm.action = '?action=generate_excel';
}

// Copiar valores del formulario original
const courseId = form.querySelector('[name="course_id"]').value;
const startDate = form.querySelector('[name="start_date"]').value;
const endDate = form.querySelector('[name="end_date"]').value;

// Crear inputs ocultos
const inputs = [
{ name: 'course_id', value: courseId },
{ name: 'start_date', value: startDate },
{ name: 'end_date', value: endDate }
];

inputs.forEach(input => {
const hiddenInput = document.createElement('input');
hiddenInput.type = 'hidden';
hiddenInput.name = input.name;
hiddenInput.value = input.value;
tempForm.appendChild(hiddenInput);
});

// Agregar al body, enviar y eliminar
document.body.appendChild(tempForm);
tempForm.submit();
document.body.removeChild(tempForm);
}
</script>
</body>
</html>

```

## 72. ArchImp/views/representatives/child\_attendance.php

```
<?php Security::requireLogin(); ?>
<!DOCTYPE html>
<html lang="es">
<head>
<meta charset="UTF-8">
<title>Asistencia - EcuAsist</title>
<style>
* { margin: 0; padding: 0; box-sizing: border-box; }
body { font-family: Arial, sans-serif; background: #f4f4f4; }
.navbar { background: #007bff; color: white; padding: 15px 30px; display: flex; justify-content:
space-between; align-items: center; }
.navbar h1 { font-size: 24px; }
.navbar a { color: white; text-decoration: none; margin-left: 20px; }
.container { max-width: 1200px; margin: 30px auto; padding: 0 20px; }
.card { background: white; padding: 30px; border-radius: 8px; box-shadow: 0 2px 10px rgba(0,0,0,0.1);
margin-bottom: 20px; }
table { width: 100%; border-collapse: collapse; margin-top: 20px; }
th, td { padding: 12px; text-align: left; border-bottom: 1px solid #ddd; }
th { background: #f8f9fa; font-weight: bold; }
.badge { padding: 5px 10px; border-radius: 3px; color: white; font-size: 12px; }
.badge-presente { background: #28a745; }
.badge-ausente { background: #dc3545; }
.badge-tardanza { background: #ffc107; color: #333; }
.badge-justificado { background: #17a2b8; }
.stats { display: grid; grid-template-columns: repeat(auto-fit, minmax(150px, 1fr)); gap: 15px;
margin-bottom: 20px; }
.stat-card { background: linear-gradient(135deg, #667eea 0%, #764ba2 100%); color: white; padding: 20px;
border-radius: 8px; text-align: center; }
.stat-number { font-size: 32px; font-weight: bold; margin-bottom: 5px; }
.stat-label { font-size: 13px; opacity: 0.9; }
</style>
</head>
<body>
<?php include BASE_PATH . '/views/partials/navbar.php'; ?>

<div class="container">
<?php
$total = count($attendances);
$presente = count(array_filter($attendances, fn($a) => $a['status'] === 'presente'));
$ausente = count(array_filter($attendances, fn($a) => $a['status'] === 'ausente'));
$tardanza = count(array_filter($attendances, fn($a) => $a['status'] === 'tardanza'));
$asistencia = $total > 0 ? round(($presente / $total) * 100, 1) : 0;
?>

<div class="stats">
<div class="stat-card" style="background: linear-gradient(135deg, #667eea 0%, #764ba2 100%);">
<div class="stat-number"><?= $total ?></div>
<div class="stat-label">Registros</div>
</div>
<div class="stat-card" style="background: linear-gradient(135deg, #28a745 0%, #20c997 100%);">
<div class="stat-number"><?= $asistencia ?>%</div>
<div class="stat-label">Asistencia</div>
</div>
<div class="stat-card" style="background: linear-gradient(135deg, #dc3545 0%, #fd7e14 100%);">
<div class="stat-number"><?= $ausente ?></div>
<div class="stat-label">Ausencias</div>
</div>
<div class="stat-card" style="background: linear-gradient(135deg, #ffc107 0%, #fd7e14 100%);">
<div class="stat-number"><?= $tardanza ?></div>
<div class="stat-label">Tardanzas</div>
</div>
</div>
```

```

<div class="card">
<h2>Historial Completo</h2>
<table>
<thead>
<tr>
<th>Fecha</th>
<th>Asignatura</th>
<th>Hora</th>
<th>Estado</th>
<th>Observación</th>
</tr>
</thead>
<tbody>
<?php foreach($attendances as $att): ?>
<tr>
<td><?= date('d/m/Y', strtotime($att['date'])) ?></td>
<td><?= $att['subject_name'] ?></td>
<td><?= $att['hour_period'] ?></td>
<td>
<span class="badge badge-<?= $att['status'] ?>">
<?= ucfirst($att['status']) ?>
</span>
</td>
<td><?= $att['observation'] ?: '-' ?></td>
</tr>
<?php endforeach; ?>
</tbody>
</table>
</div>
</div>
</body>
</html>

```

## 73. ArchImp/views/representatives/manage.php

```

<?php Security::requireLogin(); ?>
<!DOCTYPE html>
<html lang="es">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Gestión de Representantes - EcuAsist</title>
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet">
<style>
body { padding-top: 70px; background: #f4f4f4; }
</style>
</head>
<body>
<?php include BASE_PATH . '/views/partials/navbar.php'; ?>

<div class="container mt-4">

<?php if(isset($_GET['success'])): ?>
<div class="alert alert-success alert-dismissible fade show">
✓ Relación asignada correctamente
<button type="button" class="btn-close" data-bs-dismiss="alert"></button>
</div>
<?php endif; ?>

<?php if(isset($_GET['removed'])): ?>
<div class="alert alert-success alert-dismissible fade show">
✓ Relación eliminada correctamente

```

```

<button type="button" class="btn-close" data-bs-dismiss="alert"></button>
</div>
<?php endif; ?>

<?php if(isset($_GET['error'])): ?>
<div class="alert alert-danger alert-dismissible fade show">
✖ Error al procesar la solicitud
<button type="button" class="btn-close" data-bs-dismiss="alert"></button>
</div>
<?php endif; ?>

<!-- Formulario de asignación -->
<div class="card mb-4">
<div class="card-header bg-primary text-white">
<h5 class="mb-0">■ Asignar Estudiante a Representante</h5>
</div>
<div class="card-body">
<form method="POST">
<div class="row">
<div class="col-md-6 mb-3">
<label class="form-label">Representante *</label>
<select name="representative_id" class="form-select" required>
<option value="">Seleccionar representante...</option>
<?php foreach($representatives as $rep): ?>
<option value="<?= $rep['id'] ?>">
<?= $rep['last_name'] . ' ' . $rep['first_name'] ?>
(<?= $rep['dni'] ?? 'Sin cédula' ?>)
</option>
<?php endforeach; ?>
</select>
</div>

<div class="col-md-6 mb-3">
<label class="form-label">Estudiante *</label>
<select name="student_id" class="form-select" required>
<option value="">Seleccionar estudiante...</option>
<?php foreach($students as $student): ?>
<option value="<?= $student['id'] ?>">
<?= $student['last_name'] . ' ' . $student['first_name'] ?>
</option>
<?php endforeach; ?>
</select>
</div>

<div class="col-md-6 mb-3">
<label class="form-label">Parentesco *</label>
<select name="relationship" class="form-select" required>
<option value="">Seleccionar parentesco...</option>
<option value="Padre">Padre</option>
<option value="Madre">Madre</option>
<option value="Tutor Legal">Tutor Legal</option>
<option value="Abuelo/a">Abuelo/a</option>
<option value="Tío/a">Tío/a</option>
<option value="Hermano/a">Hermano/a</option>
<option value="Otro">Otro</option>
</select>
</div>

<div class="col-md-6 mb-3">
<label class="form-label">&nbsp;</label>
<div class="form-check" style="padding-top: 8px;">
<input type="checkbox" name="is_primary" value="1" class="form-check-input" id="isPrimary">
<label class="form-check-label" for="isPrimary">
<strong>Representante Principal</strong>
</label>

```

```

</div>
</div>
</div>

<button type="submit" class="btn btn-success">
■ Asignar Relación
</button>
</form>
</div>
</div>

<!-- Filtros -->
<div class="card mb-4">
<div class="card-header bg-light">
<h5 class="mb-0">■ Filtros de Búsqueda</h5>
</div>
<div class="card-body">
<div class="row">
<div class="col-md-4 mb-3">
<label class="form-label">Buscar por Representante</label>
<input type="text" id="filterRepresentative" class="form-control" placeholder="Nombre del representante...">
</div>
<div class="col-md-4 mb-3">
<label class="form-label">Buscar por Estudiante</label>
<input type="text" id="filterStudent" class="form-control" placeholder="Nombre del estudiante...">
</div>
<div class="col-md-4 mb-3">
<label class="form-label">Buscar por Curso</label>
<input type="text" id="filterCourse" class="form-control" placeholder="Nombre del curso...">
</div>
</div>
<button onclick="clearFilters()" class="btn btn-secondary btn-sm">
■■ Limpiar Filtros
</button>
</div>
</div>

<!-- Lista de relaciones -->
<div class="card">
<div class="card-header bg-light d-flex justify-content-between align-items-center">
<h5 class="mb-0">■ Representantes y sus Estudiantes</h5>
<span class="badge bg-primary" id="totalCount">
<?php
$totalRelations = 0;
foreach($representatives as $rep) {
$totalRelations += count($this->representativeModel->getStudentsByRepresentative($rep['id']));
}
echo $totalRelations;
?> relaciones
</span>
</div>
<div class="card-body">
<?php
$hasRelations = false;
foreach($representatives as $rep):
$children = $this->representativeModel->getStudentsByRepresentative($rep['id']);
if(count($children) > 0):
$hasRelations = true;
?>
<div class="representative-block mb-4 p-3 border rounded"
data-representative="<?> strtolower($rep['last_name']) . ' ' . $rep['first_name']) ?>"

<div class="d-flex justify-content-between align-items-center mb-2">
<h5 class="mb-0">
■ <?> $rep['last_name'] . ' ' . $rep['first_name'] ?>

```

```

</h5>
<small class="text-muted"><?= $rep['email'] ?></small>
</div>

<div class="table-responsive">
<table class="table table-sm table-hover mb-0">
<thead class="table-light">
<tr>
<th>Estudiante</th>
<th>Parentesco</th>
<th>Curso</th>
<th>Tipo</th>
<th>Acciones</th>
</tr>
</thead>
<tbody>
<?php foreach($children as $child): ?>
<tr class="student-row"
data-student="<?= strtolower($child['last_name'] . ' ' . $child['first_name']) ?>"
data-course="<?= strtolower($child['course_name'] ?? '') ?>">
<td>
<strong><?= $child['last_name'] . ' ' . $child['first_name'] ?></strong>
</td>
<td><?= $child['relationship'] ?></td>
<td>
<?php if($child['course_name']): ?>
<span class="badge bg-info"><?= $child['course_name'] ?></span>
<?php else: ?>
<span class="text-muted">Sin curso</span>
<?php endif; ?>
</td>
<td>
<?php if($child['is_primary']): ?>
<span class="badge bg-warning text-dark">■ Principal</span>
<?php else: ?>
<span class="badge bg-secondary">Secundario</span>
<?php endif; ?>
</td>
<td>
<button onclick="confirmRemove(<?= $rep['id'] ?>, <?= $child['id'] ?>, '<?= addslashes($rep['last_name'] . ' ' . $rep['first_name']) ?>', '<?= addslashes($child['last_name'] . ' ' . $child['first_name']) ?>')"
class="btn btn-danger btn-sm">
✕ Eliminar
</button>
</td>
</tr>
<?php endforeach; ?>
</tbody>
</table>
</div>
</div>
<?php
endif;
endforeach;

if(!$hasRelations):
?>
<div class="alert alert-secondary text-center">
■ No hay relaciones representante-estudiante registradas.
<br>Use el formulario superior para crear una.
</div>
<?php endif; ?>
</div>
</div>
</div>

```



```

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"></script>
<script>
// Filtros en tiempo real
const filterRepresentative = document.getElementById('filterRepresentative');
const filterStudent = document.getElementById('filterStudent');
const filterCourse = document.getElementById('filterCourse');

function applyFilters() {
const repValue = filterRepresentative.value.toLowerCase();
const stuValue = filterStudent.value.toLowerCase();
const courValue = filterCourse.value.toLowerCase();

const blocks = document.querySelectorAll('.representative-block');
let visibleCount = 0;

blocks.forEach(block => {
const repName = block.dataset.representative;
const rows = block.querySelectorAll('.student-row');
let hasVisibleRows = false;

rows.forEach(row => {
const stuName = row.dataset.student;
const course = row.dataset.course;

const matchRep = !repValue || repName.includes(repValue);
const matchStu = !stuValue || stuName.includes(stuValue);
const matchCou = !courValue || course.includes(courValue);

if (matchRep && matchStu && matchCou) {
row.style.display = '';
hasVisibleRows = true;
visibleCount++;
} else {
row.style.display = 'none';
}
});

block.style.display = hasVisibleRows ? '' : 'none';
});

// Actualizar contador
document.getElementById('totalCount').textContent = visibleCount + ' relaciones';
}

function clearFilters() {
filterRepresentative.value = '';
filterStudent.value = '';
filterCourse.value = '';
applyFilters();
}

filterRepresentative.addEventListener('input', applyFilters);
filterStudent.addEventListener('input', applyFilters);
filterCourse.addEventListener('input', applyFilters);

// Modal de confirmación
function confirmRemove(repId, studentId, repName, studentName) {
const modal = document.createElement('div');
modal.style.cssText = 'position: fixed; top: 0; left: 0; width: 100%; height: 100%; background:
rgba(0,0,0,0.5); display: flex; align-items: center; justify-content: center; z-index: 9999;';

const modalContent = document.createElement('div');
modalContent.style.cssText = 'background: white; padding: 30px; border-radius: 8px; max-width: 500px;
box-shadow: 0 4px 20px rgba(0,0,0,0.3);';

```

```

modalContent.innerHTML = `
<h3 style="margin: 0 0 15px 0; color: #dc3545;">■ ■ Eliminar Relación</h3>
<p style="margin: 0 0 20px 0; color: #666;">
¿Está seguro de eliminar la relación entre:
</p>
<p style="margin: 0 0 10px 0; padding: 10px; background: #f8f9fa; border-radius: 4px;">
<strong>Representante:</strong> ${repName}<br>
<strong>Estudiante:</strong> ${studentName}
</p>
<p style="margin: 0 0 20px 0; color: #666; font-size: 14px; background: #f8d7da; padding: 10px;
border-radius: 4px;">
<strong>■ ■ Advertencia:</strong> El representante ya no podrá ver la información del estudiante.
</p>
<div style="display: flex; gap: 10px; justify-content: flex-end;">
<button type="button" id="cancelRemoveBtn" style="padding: 10px 20px; background: #6c757d; color: white;
border: none; border-radius: 4px; cursor: pointer;">
Cancelar
</button>
<button type="button" id="confirmRemoveBtn" style="padding: 10px 20px; background: #dc3545; color: white;
border: none; border-radius: 4px; cursor: pointer;">
Sí, Eliminar
</button>
</div>
`;

modal.appendChild(modalContent);
document.body.appendChild(modal);

document.getElementById('confirmRemoveBtn').onclick = function() {
document.body.removeChild(modal);
window.location.href = '?action=remove_representative&rep_id=' + repId + '&student_id=' + studentId;
};

document.getElementById('cancelRemoveBtn').onclick = function() {
document.body.removeChild(modal);
};
}
</script>
</body>
</html>

```

## 74. ArchImp/views/representatives/my\_children.php

```

<?php Security::requireLogin(); ?>
<!DOCTYPE html>
<html lang="es">
<head>
<meta charset="UTF-8">
<title>Mis Representados - EcuAsist</title>
<style>
* { margin: 0; padding: 0; box-sizing: border-box; }
body { font-family: Arial, sans-serif; background: #f4f4f4; }
.navbar { background: #007bff; color: white; padding: 15px 30px; display: flex; justify-content:
space-between; align-items: center; }
.navbar h1 { font-size: 24px; }
.navbar a { color: white; text-decoration: none; margin-left: 20px; }
.container { max-width: 1000px; margin: 30px auto; padding: 0 20px; }
.student-grid { display: grid; grid-template-columns: repeat(auto-fit, minmax(300px, 1fr)); gap: 20px; }
.student-card { background: white; padding: 25px; border-radius: 8px; box-shadow: 0 2px 10px
rgba(0,0,0,0.1); }
.student-card h3 { color: #007bff; margin-bottom: 15px; }

```

```

.info-row { padding: 8px 0; border-bottom: 1px solid #f0f0f0; }
.info-row:last-child { border-bottom: none; }
.label { font-weight: bold; color: #666; display: inline-block; width: 120px; }
.btn { display: inline-block; padding: 10px 20px; background: #007bff; color: white; text-decoration: none;
border-radius: 4px; margin-top: 15px; text-align: center; }
.btn:hover { background: #0056b3; }
</style>
</head>
<body>
<?php include BASE_PATH . '/views/partials/navbar.php'; ?>

<div class="container">
<?php if(count($children) > 0): ?>
<div class="student-grid">
<?php foreach($children as $child): ?>
<div class="student-card">
<h3><?= $child['last_name'] . ' ' . $child['first_name'] ?></h3>

<div class="info-row">
<span class="label">Cédula:</span>
<?= $child['dni'] ?? 'No registrada' ?>
</div>

<div class="info-row">
<span class="label">Parentesco:</span>
<?= $child['relationship'] ?>
<?php if($child['is_primary']): ?>
<strong style="color: #ffc107;">(Principal)</strong>
<?php endif; ?>
</div>

<div class="info-row">
<span class="label">Curso:</span>
<?= $child['course_name'] ?? 'Sin asignar' ?>
</div>

<div class="info-row">
<span class="label">Jornada:</span>
<?= $child['shift_name'] ? ucfirst($child['shift_name']) : '-' ?>
</div>

<a href="?action=child_attendance&student_id=<?= $child['id'] ?>" class="btn">
Ver Asistencia
</a>
</div>
<?php endforeach; ?>
</div>
<?php else: ?>
<div class="student-card" style="text-align: center; padding: 40px;">
<p style="color: #666;">No tiene estudiantes asignados como representante.</p>
</div>
<?php endif; ?>
</div>
</body>
</html>

```

## 75. ArchImp/views/schedule/index.php

```

<?php Security::requireLogin(); ?>
<!DOCTYPE html>
<html lang="es">
<head>

```

```

<meta charset="UTF-8">
<title>Gestión de Horarios - EcuAsist</title>
<style>
* { margin: 0; padding: 0; box-sizing: border-box; }
body { font-family: Arial, sans-serif; background: #f4f4f4; }
.container { max-width: 1200px; margin: 30px auto; padding: 0 20px; }
.card { background: white; padding: 30px; border-radius: 8px; box-shadow: 0 2px 10px rgba(0,0,0,0.1); }
table { width: 100%; border-collapse: collapse; margin-top: 20px; }
th, td { padding: 12px; text-align: left; border-bottom: 1px solid #ddd; }
th { background: #f8f9fa; font-weight: bold; }
.btn { padding: 8px 15px; background: #007bff; color: white; border: none; border-radius: 4px;
text-decoration: none; display: inline-block; cursor: pointer; }
.btn:hover { background: #0056b3; }
h2 { margin-bottom: 20px; color: #333; }
</style>
</head>
<body>
<?php include BASE_PATH . '/views/partials/navbar.php'; ?>

<div class="container">
<div class="card">
<h2>Gestión de Horarios por Curso</h2>
<p style="color: #666; margin-bottom: 20px;">
Seleccione un curso para configurar su horario de clases
</p>
<table>
<thead>
<tr>
<th>Curso</th>
<th>Nivel</th>
<th>Paralelo</th>
<th>Jornada</th>
<th>Acción</th>
</tr>
</thead>
<tbody>
<?php foreach($courses as $course): ?>
<tr>
<td><?= $course['name'] ?></td>
<td><?= $course['grade_level'] ?></td>
<td><?= $course['parallel'] ?></td>
<td><?= ucfirst($course['shift_name']) ?></td>
<td>
<a href="?action=manage_schedule&course_id=<?= $course['id'] ?>" class="btn">
Configurar Horario
</a>
</td>
</tr>
<?php endforeach; ?>
</tbody>
</table>
</div>
</div>
</body>
</html>

```

## 76. ArchImp/views/schedule/manage.php

```

<?php Security::requireLogin(); ?>
<!DOCTYPE html>
<html lang="es">
<head>
<meta charset="UTF-8">

```

```

<title>Horario de <?= $course['name'] ?> - EcuAsist</title>
<style>
* { margin: 0; padding: 0; box-sizing: border-box; }
body { font-family: Arial, sans-serif; background: #f4f4f4; }
.container { max-width: 1400px; margin: 30px auto; padding: 0 20px; }
.card { background: white; padding: 30px; border-radius: 8px; box-shadow: 0 2px 10px rgba(0,0,0,0.1);
margin-bottom: 20px; }
.form-group { margin-bottom: 15px; }
.error { background: #f8d7da; color: #721c24; padding: 10px; border-radius: 4px; margin-bottom: 20px; }
label { display: block; margin-bottom: 5px; font-weight: bold; color: #333; }
select, input { width: 100%; padding: 10px; border: 1px solid #ddd; border-radius: 4px; }
button { padding: 10px 20px; background: #28a745; color: white; border: none; border-radius: 4px; cursor:
pointer; }
button:hover { background: #218838; }
.btn-danger { background: #dc3545; padding: 5px 10px; font-size: 12px; }
.btn-danger:hover { background: #c82333; }
table { width: 100%; border-collapse: collapse; margin-top: 20px; }
th, td { padding: 12px; text-align: left; border-bottom: 1px solid #ddd; font-size: 14px; }
th { background: #f8f9fa; font-weight: bold; }
.success { background: #d4edda; color: #155724; padding: 10px; border-radius: 4px; margin-bottom: 20px; }
.grid { display: grid; grid-template-columns: 1fr 1fr; gap: 20px; }
h2 { margin-bottom: 20px; color: #333; }
.course-info { background: #e7f3ff; padding: 15px; border-radius: 4px; margin-bottom: 20px; }
</style>
</head>
<body>
<?php include BASE_PATH . '/views/partials/navbar.php'; ?>

<div class="container">
<?php if(isset($_GET['success'])): ?>
<div class="success">✓ Clase agregada al horario</div>
<?php endif; ?>
<?php if(isset($_GET['deleted'])): ?>
<div class="success">✓ Clase eliminada del horario</div>
<?php endif; ?>
<?php if(isset($_GET['error'])): ?>
<div class="error">✗ <?= htmlspecialchars($_GET['error']) ?></div>
<?php endif; ?>

<div class="course-info">
<h2><?= $course['name'] ?></h2>
<p><strong>Nivel:</strong> <?= $course['grade_level'] ?> | <strong>Paralelo:</strong> <?=
$course['parallel'] ?> | <strong>Jornada:</strong> <?= ucfirst($course['shift_name']) ?></p>
</div>

<div class="grid">
<div class="card">
<h2>Agregar Clase al Horario</h2>
<form method="POST" id="scheduleForm">
<div class="form-group">
<label>Día de la Semana</label>
<select name="day_of_week" id="day_of_week" required>
<option value="">Seleccionar...</option>
<option value="lunes">Lunes</option>
<option value="martes">Martes</option>
<option value="miercoles">Miércoles</option>
<option value="jueves">Jueves</option>
<option value="viernes">Viernes</option>
<option value="sabado">Sábado</option>
</select>
</div>

<div class="form-group">
<label>Número de Hora</label>
<select name="period_number" id="period_number" required>

```

```

<option value="">Seleccionar...</option>
<?php
$maxHours = strpos($course['grade_level'], 'Técnico') !== false ? 8 : 7;
for($i = 1; $i <= $maxHours; $i++):
?>
<option value="<?= $i ?>"><?= $i ?>ra hora</option>
<?php endfor; ?>
</select>
</div>

<div class="form-group">
<label>Asignatura</label>
<select name="subject_id" id="subject_id" required>
<option value="">Seleccionar...</option>
</select>
</div>

<div class="form-group">
<label>Docente Asignado</label>
<input type="text" id="teacher_name" readonly style="background: #f0f0f0;" placeholder="Se asignará automáticamente">
<input type="hidden" name="teacher_id" id="teacher_id">
</div>

<button type="submit" id="submitBtn" disabled style="background: #6c757d;">Agregar Clase</button>
</form>

<script>
const courseId = <?= $course['id'] ?>;

// Cargar asignaturas del curso al cargar la página
fetch('?action=get_course_subjects_schedule&course_id=' + courseId)
.then(response => response.json())
.then(data => {
const subjectSelect = document.getElementById('subject_id');
if (data.length === 0) {
subjectSelect.innerHTML = '<option value="">No hay asignaturas asignadas a este curso</option>';
} else {
subjectSelect.innerHTML = '<option value="">Seleccionar...</option>';
data.forEach(item => {
const option = document.createElement('option');
option.value = item.subject_id;
option.textContent = item.subject_name;
option.setAttribute('data-teacher-id', item.teacher_id);
option.setAttribute('data-teacher-name', item.teacher_name);
subjectSelect.appendChild(option);
});
}
});

// Al seleccionar asignatura, mostrar docente automáticamente
document.getElementById('subject_id').addEventListener('change', function() {
const selectedOption = this.options[this.selectedIndex];
const teacherId = selectedOption.getAttribute('data-teacher-id');
const teacherName = selectedOption.getAttribute('data-teacher-name');

if (teacherId && teacherName) {
document.getElementById('teacher_id').value = teacherId;
document.getElementById('teacher_name').value = teacherName;
checkFormComplete();
} else {
document.getElementById('teacher_id').value = '';
document.getElementById('teacher_name').value = '';
document.getElementById('submitBtn').disabled = true;
}
});

```

```

document.getElementById('submitBtn').style.background = '#6c757d';
}
});

document.getElementById('day_of_week').addEventListener('change', checkFormComplete);
document.getElementById('period_number').addEventListener('change', checkFormComplete);

function checkFormComplete() {
const day = document.getElementById('day_of_week').value;
const period = document.getElementById('period_number').value;
const subject = document.getElementById('subject_id').value;
const teacher = document.getElementById('teacher_id').value;

const btn = document.getElementById('submitBtn');
if (day && period && subject && teacher) {
btn.disabled = false;
btn.style.background = '#28a745';
} else {
btn.disabled = true;
btn.style.background = '#6c757d';
}
}
document.getElementById('day_of_week').addEventListener('change', checkExistingSchedule);
document.getElementById('period_number').addEventListener('change', checkExistingSchedule);

function checkExistingSchedule() {
const day = document.getElementById('day_of_week').value;
const period = document.getElementById('period_number').value;

if (!day || !period) return;

fetch('?action=check_schedule_conflict&course_id=' + courseId + '&day=' + day + '&period=' + period)
.then(response => response.json())
.then(data => {
if (data.exists) {
showWarning(`■ Esta hora ya está ocupada con ${data.subject_name} (${data.teacher_name}). Debe eliminarla
antes de agregar otra.`);
document.getElementById('submitBtn').disabled = true;
document.getElementById('submitBtn').style.background = '#6c757d';
}
});
}

function showWarning(message) {
// Remover advertencia anterior si existe
const existingWarning = document.getElementById('scheduleWarning');
if (existingWarning) {
existingWarning.remove();
}

const warning = document.createElement('div');
warning.id = 'scheduleWarning';
warning.style.cssText = 'background: #fff3cd; color: #856404; padding: 12px; border-radius: 4px; margin-top:
15px; border-left: 4px solid #ffc107;';
warning.innerHTML = message;

document.getElementById('scheduleForm').appendChild(warning);

setTimeout(() => {
warning.style.transition = 'opacity 0.5s';
warning.style.opacity = '0';
setTimeout(() => warning.remove(), 500);
}, 4000);
}
</script>

```

```

</div>

<div class="card">
<h2>Horario Actual</h2>
<table>
<thead>
<tr>
<th>Día</th>
<th>Hora</th>
<th>Asignatura</th>
<th>Docente</th>
<th>Acción</th>
</tr>
</thead>
<tbody>
<?php if(empty($schedule)): ?>
<tr>
<td colspan="5" style="text-align:center;">No hay clases en el horario</td>
</tr>
<?php else: ?>
<?php foreach($schedule as $class): ?>
<tr>
<td><?= ucfirst($class['day_of_week']) ?></td>
<td><?= $class['period_number'] ?>ra hora</td>
<td><?= $class['subject_name'] ?></td>
<td><?= $class['teacher_name'] ?></td>
<td>
<form method="POST" action="?action=delete_schedule_class" style="display: inline;">
<input type="hidden" name="schedule_id" value="<?= $class['id'] ?>">
<input type="hidden" name="course_id" value="<?= $course['id'] ?>">
<button type="submit" class="btn-danger" onclick="return confirm('¿Eliminar esta clase del horario?')">
Eliminar
</button>
</form>
</td>
</tr>
<?php endforeach; ?>
<?php endif; ?>
</tbody>
</table>
</div>
</div>
</div>
</body>
</html>

```

## 77. ArchImp/views/stats/index.php

```

<?php Security::requireLogin(); ?>
<!DOCTYPE html>
<html lang="es">
<head>
<meta charset="UTF-8">
<title>Estadísticas - EcuAsist</title>
<style>
* { margin: 0; padding: 0; box-sizing: border-box; }
body { font-family: Arial, sans-serif; background: #f4f4f4; }
.navbar { background: #007bff; color: white; padding: 15px 30px; display: flex; justify-content: space-between; align-items: center; }
.navbar h1 { font-size: 24px; }
.navbar a { color: white; text-decoration: none; margin-left: 20px; }
.container { max-width: 1400px; margin: 30px auto; padding: 0 20px; }
.card { background: white; padding: 30px; border-radius: 8px; box-shadow: 0 2px 10px rgba(0,0,0,0.1); }

```



```

margin-bottom: 20px; }
.stats-grid { display: grid; grid-template-columns: repeat(auto-fit, minmax(200px, 1fr)); gap: 20px;
margin-bottom: 30px; }
.stat-card { background: linear-gradient(135deg, #667eea 0%, #764ba2 100%); color: white; padding: 25px;
border-radius: 8px; text-align: center; }
.stat-number { font-size: 36px; font-weight: bold; margin-bottom: 5px; }
.stat-label { font-size: 14px; opacity: 0.9; }
table { width: 100%; border-collapse: collapse; margin-top: 15px; }
th, td { padding: 12px; text-align: left; border-bottom: 1px solid #ddd; }
th { background: #f8f9fa; font-weight: bold; }
.progress-bar { background: #e9ecef; border-radius: 4px; height: 20px; overflow: hidden; }
.progress-fill { background: #28a745; height: 100%; transition: width 0.3s; }
.form-group { margin-bottom: 15px; display: inline-block; margin-right: 15px; }
label { display: block; margin-bottom: 5px; font-weight: bold; }
input, button { padding: 10px; border: 1px solid #ddd; border-radius: 4px; }
button { background: #007bff; color: white; border: none; cursor: pointer; }
button:hover { background: #0056b3; }
h2 { margin-bottom: 20px; color: #333; }
</style>
</head>
<body>
<?php include BASE_PATH . '/views/partials/navbar.php'; ?>

<div class="container">
<div class="card">
<form method="GET">
<input type="hidden" name="action" value="stats">
<div class="form-group">
<label>Fecha Inicio</label>
<input type="date" name="start_date" value="<?=$_GET['start_date'] ?? date('Y-m-01') ?>">
</div>
<div class="form-group">
<label>Fecha Fin</label>
<input type="date" name="end_date" value="<?=$_GET['end_date'] ?? date('Y-m-d') ?>">
</div>
<button type="submit">Filtrar</button>
</form>
</div>

<div class="stats-grid">
<div class="stat-card" style="background: linear-gradient(135deg, #667eea 0%, #764ba2 100%);">
<div class="stat-number"><?=$_stats['overall']['total_records'] ?></div>
<div class="stat-label">Registros Totales</div>
</div>
<div class="stat-card" style="background: linear-gradient(135deg, #28a745 0%, #20c997 100%);">
<div class="stat-number"><?=$_stats['overall']['presente'] ?></div>
<div class="stat-label">Presentes</div>
</div>
<div class="stat-card" style="background: linear-gradient(135deg, #dc3545 0%, #fd7e14 100%);">
<div class="stat-number"><?=$_stats['overall']['ausente'] ?></div>
<div class="stat-label">Ausentes</div>
</div>
<div class="stat-card" style="background: linear-gradient(135deg, #ffc107 0%, #fd7e14 100%);">
<div class="stat-number"><?=$_stats['overall']['tardanza'] ?></div>
<div class="stat-label">Tardanzas</div>
</div>
</div>

<div class="card">
<h2>Asistencia por Curso</h2>
<table>
<thead>
<tr>
<th>Curso</th>
<th>Total</th>
<th>Presentes</th>

```

```

<th>Ausentes</th>
<th>% Asistencia</th>
<th>Indicador</th>
</tr>
</thead>
<tbody>
<?php foreach($stats['by_course'] as $course): ?>
<tr>
<td><?= $course['course_name'] ?></td>
<td><?= $course['total'] ?></td>
<td><?= $course['presente'] ?></td>
<td><?= $course['ausente'] ?></td>
<td><?= $course['percentage'] ?>%</td>
<td>
<div class="progress-bar">
<div class="progress-fill" style="width: <?= $course['percentage'] ?>%;"></div>
</div>
</td>
</tr>
<?php endforeach; ?>
</tbody>
</table>
</div>

<div class="card">
<h2>Top 10 - Estudiantes con Más Ausencias</h2>
<table>
<thead>
<tr>
<th>#</th>
<th>Estudiante</th>
<th>Curso</th>
<th>Total Ausencias</th>
</tr>
</thead>
<tbody>
<?php foreach($stats['by_student'] as $index => $student): ?>
<tr>
<td><?= $index + 1 ?></td>
<td><?= $student['student_name'] ?></td>
<td><?= $student['course_name'] ?></td>
<td><strong style="color: #dc3545;"><?= $student['total_absences'] ?></strong></td>
</tr>
<?php endforeach; ?>
</tbody>
</table>
</div>

<div class="card">
<h2>Tendencia Diaria</h2>
<table>
<thead>
<tr>
<th>Fecha</th>
<th>Registros</th>
<th>Presentes</th>
<th>% Asistencia</th>
</tr>
</thead>
<tbody>
<?php foreach($stats['by_day'] as $day): ?>
<tr>
<td><?= date('d/m/Y', strtotime($day['date'])) ?></td>
<td><?= $day['total'] ?></td>
<td><?= $day['presente'] ?></td>

```

```

<td><?= $day['percentage'] ?>%</td>
</tr>
<?php endforeach; ?>
</tbody>
</table>
</div>
</div>
</body>
</html>

```

## 78. ArchImp/views/users/create.php

```

<?php
require_once BASE_PATH . '/helpers/Security.php';
Security::requireLogin();
if (!Security::hasRole('autoridad')) {
die('Acceso denegado');
}
?>
<!DOCTYPE html>
<html lang="es">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Crear Usuario - EcuAsist</title>
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet">
<style>
body { padding-top: 70px; }
.form-section {
background: #f8f9fa;
padding: 20px;
border-radius: 8px;
margin-bottom: 20px;
}
.form-section h5 {
color: #007bff;
border-bottom: 2px solid #007bff;
padding-bottom: 10px;
margin-bottom: 20px;
}
.required::after {
content: " *";
color: red;
}
</style>
</head>
<body>
<?php include BASE_PATH . '/views/partials/navbar.php'; ?>

<div class="container">
<div class="row justify-content-center">
<div class="col-md-10">
<div class="card shadow">
<div class="card-header bg-primary text-white">
<h4 class="mb-0">■ Crear Nuevo Usuario</h4>
</div>
<div class="card-body">
<?php if (!empty($errors)): ?>
<div class="alert alert-danger">
<strong>Errores:</strong>
<ul class="mb-0">
<?php foreach ($errors as $error): ?>
<li><?= htmlspecialchars($error) ?></li>

```

```

<?php endforeach; ?>
</ul>
</div>
<?php endif; ?>

<form method="POST" action="?action=create_user">
<!-- Sección Datos de Acceso -->
<div class="form-section">
<h5>■ Datos de Acceso</h5>

<div class="row">
<div class="col-md-6 mb-3">
<label class="form-label required">Nombre de Usuario</label>
<input type="text"
name="username"
class="form-control"
value="<?= isset($_POST['username']) ? htmlspecialchars($_POST['username']) : '' ?>"
required>
<small class="text-muted">Sin espacios ni caracteres especiales</small>
</div>
<div class="col-md-6 mb-3">
<label class="form-label required">Email</label>
<input type="email"
name="email"
class="form-control"
value="<?= isset($_POST['email']) ? htmlspecialchars($_POST['email']) : '' ?>"
required>
</div>
<div class="col-md-6 mb-3">
<label class="form-label required">Contraseña</label>
<input type="password"
name="password"
class="form-control"
minlength="6"
required>
<small class="text-muted">Mínimo 6 caracteres</small>
</div>
<div class="col-md-6 mb-3">
<label class="form-label required">Confirmar Contraseña</label>
<input type="password"
name="confirm_password"
class="form-control"
minlength="6"
required>
</div>
</div>
</div>
</div>

<!-- Sección Datos Personales -->
<div class="form-section">
<h5>■■■ Datos Personales</h5>
<div class="row">
<div class="col-md-6 mb-3">
<label class="form-label required">Nombres</label>
<input type="text"
name="first_name"
class="form-control"
value="<?= isset($_POST['first_name']) ? htmlspecialchars($_POST['first_name']) : '' ?>"
required>
</div>
<div class="col-md-6 mb-3">
<label class="form-label required">Apellidos</label>
<input type="text"
name="last_name"
class="form-control"
value="<?= isset($_POST['last_name']) ? htmlspecialchars($_POST['last_name']) : '' ?>"

```

```

required>
</div>
<div class="col-md-6 mb-3">
<label class="form-label">Cédula/DNI</label>

<input type="text"
name="dni"
class="form-control"
value="<?= isset($_POST['dni']) ? htmlspecialchars($_POST['dni']) : '' ?>">
</div>
<div class="col-md-6 mb-3">
<label class="form-label">Teléfono</label>
<input type="text"
name="phone"
class="form-control"
value="<?= isset($_POST['phone']) ? htmlspecialchars($_POST['phone']) : '' ?>">
</div>
</div>
</div>

<!-- Sección Roles -->
<div class="form-section">
<h5>■ Roles del Usuario</h5>
<p class="text-muted">Selecciona uno o varios roles (opcional, puedes asignarlos después)</p>
<div class="row">
<?php foreach ($roles as $role): ?>
<div class="col-md-4 mb-2">
<div class="form-check">
<input class="form-check-input"
type="checkbox"
name="roles[]"
value="<?= $role['id'] ?>"
id="role_<?= $role['id'] ?>"
<?= isset($_POST['roles']) && in_array($role['id'], $_POST['roles']) ? 'checked' : '' ?>
<label class="form-check-label" for="role_<?= $role['id'] ?>">
<?= ucfirst($role['name']) ?>
</label>
</div>
</div>
<?php endforeach; ?>
</div>
</div>

<!-- Botones -->
<div class="d-flex justify-content-between mt-4">
<a href="?action=users" class="btn btn-secondary">
← Cancelar
</a>
<button type="submit" class="btn btn-primary">
✓ Crear Usuario
</button>
</div>
</form>
</div>
</div>
</div>
</div>
</div>

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"></script>
</body>
</html>

```

## 79. ArchImp/views/users/edit.php

```
<?php
require_once BASE_PATH . '/helpers/Security.php';
Security::requireLogin();
if (!Security::hasRole('autoridad')) {
    die('Acceso denegado');
}
?>
<!DOCTYPE html>
<html lang="es">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Editar Usuario - EcuAsist</title>
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet">
<style>
body { padding-top: 70px; }
.form-section {
background: #f8f9fa;
padding: 20px;
border-radius: 8px;
margin-bottom: 20px;
}
.form-section h5 {
color: #007bff;
border-bottom: 2px solid #007bff;
padding-bottom: 10px;
margin-bottom: 20px;
}
.required::after {
content: " *";
color: red;
}
.password-info {
background: #fff3cd;
border-left: 4px solid #ffc107;
padding: 10px;
margin-bottom: 15px;
}
</style>
</head>
<body>
<?php include BASE_PATH . '/views/partials/navbar.php'; ?>

<div class="container">
<div class="row justify-content-center">
<div class="col-md-10">
<div class="card shadow">
<div class="card-header bg-primary text-white">
<h4 class="mb-0">■ Editor Usuario: <?= htmlspecialchars($user['first_name']) . ' ' . $user['last_name']>
?></h4>
</div>
<div class="card-body">
<?php if (!empty($errors)): ?>
<div class="alert alert-danger">
<strong>Errores:</strong>
<ul class="mb-0">
<?php foreach ($errors as $error): ?>
<li><?= htmlspecialchars($error) ?></li>
<?php endforeach; ?>
</ul>
</div>
<?php endif; ?>
```

```

<form method="POST" action="?action=edit_user&id=<?= $user['id'] ?>">
<!-- Sección Datos de Acceso -->
<div class="form-section">
<h5>■ Datos de Acceso</h5>
<div class="row">
<div class="col-md-6 mb-3">
<label class="form-label required">Nombre de Usuario</label>
<input type="text"
name="username"
class="form-control"
value="<?= htmlspecialchars($user['username']) ?>"
required>
<small class="text-muted">Sin espacios ni caracteres especiales</small>
</div>
<div class="col-md-6 mb-3">
<label class="form-label required">Email</label>
<input type="email"
name="email"
class="form-control"
value="<?= htmlspecialchars($user['email']) ?>"
required>
</div>
</div>

<div class="password-info">
<strong>■■ Contraseña:</strong> Deja en blanco si no deseas cambiarla
</div>

<div class="row">
<div class="col-md-6 mb-3">
<label class="form-label">Nueva Contraseña</label>
<input type="password"
name="password"
class="form-control"
minlength="6">
<small class="text-muted">Mínimo 6 caracteres</small>
</div>
<div class="col-md-6 mb-3">
<label class="form-label">Confirmar Nueva Contraseña</label>
<input type="password"
name="confirm_password"
class="form-control"
minlength="6">
</div>
</div>
</div>

<!-- Sección Datos Personales -->
<div class="form-section">
<h5>■■■ Datos Personales</h5>
<div class="row">
<div class="col-md-6 mb-3">
<label class="form-label required">Nombres</label>
<input type="text"
name="first_name"
class="form-control"
value="<?= htmlspecialchars($user['first_name']) ?>"
required>
</div>
<div class="col-md-6 mb-3">
<label class="form-label required">Apellidos</label>
<input type="text"
name="last_name"
class="form-control"
value="<?= htmlspecialchars($user['last_name']) ?>"

```

```

required>
</div>
<div class="col-md-6 mb-3">
<label class="form-label">Cédula/DNI</label>
<input type="text"
name="dni"
class="form-control"
value="<?= htmlspecialchars($user['dni'] ?? '') ?>">
</div>
<div class="col-md-6 mb-3">
<label class="form-label">Teléfono</label>
<input type="text"
name="phone"
class="form-control"
value="<?= htmlspecialchars($user['phone'] ?? '') ?>">
</div>
</div>
</div>

<!-- Información de Roles -->
<div class="alert alert-info">
<strong>■■ Roles:</strong> Los roles se gestionan desde la tabla de usuarios principal usando los botones de
asignar/quitar roles.
</div>

<!-- Botones -->
<div class="d-flex justify-content-between mt-4">
<a href="?action=users" class="btn btn-secondary">
← Cancelar
</a>
<button type="submit" class="btn btn-success">
✓ Guardar Cambios
</button>
</div>
</form>
</div>
</div>
</div>
</div>
</div>

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"></script>
</body>
</html>

```

## 80. ArchImp/views/users/index.php

```

<?php Security::requireLogin(); ?>
<!DOCTYPE html>
<html lang="es">
<head>
<meta charset="UTF-8">
<title>Gestión de Usuarios - EcuAsist</title>
<style>
* { margin: 0; padding: 0; box-sizing: border-box; }
body { font-family: Arial, sans-serif; background: #f4f4f4; }
.container { max-width: 1400px; margin: 30px auto; padding: 0 20px; }
.card { background: white; padding: 30px; border-radius: 8px; box-shadow: 0 2px 10px rgba(0,0,0,0.1); }
table { width: 100%; border-collapse: collapse; margin-top: 20px; }
th, td { padding: 12px; text-align: left; border-bottom: 1px solid #ddd; }
th { background: #f8f9fa; font-weight: bold; }
.badge {
display: inline-block;

```



```

background: #28a745;
color: white;
padding: 3px 8px;
border-radius: 3px;
font-size: 11px;
margin: 2px;
position: relative;
}
.btn-remove-role {
background: none;
border: none;
color: white;
font-size: 16px;
font-weight: bold;
cursor: pointer;
padding: 0;
margin-left: 5px;
line-height: 1;
}
.btn-remove-role:hover {
color: #ff0000;
}
select, button { padding: 8px 12px; margin: 5px; border: 1px solid #ddd; border-radius: 4px; }
button { background: #007bff; color: white; border: none; cursor: pointer; }
button:hover { background: #0056b3; }
.btn-danger { background: #dc3545; }
.btn-danger:hover { background: #c82333; }
.btn-success { background: #28a745; }
.btn-success:hover { background: #218838; }
.btn-warning { background: #ffc107; color: #212529; }
.btn-warning:hover { background: #e0a800; }
.success { background: #d4edda; color: #155724; padding: 10px; border-radius: 4px; margin-bottom: 20px; }
.error { background: #f8d7da; color: #721c24; padding: 10px; border-radius: 4px; margin-bottom: 20px; }
.info { background: #d1ecf1; color: #0c5460; padding: 10px; border-radius: 4px; margin-bottom: 20px; }
h2 { margin-bottom: 20px; color: #333; }
.header-actions {
display: flex;
justify-content: space-between;
align-items: center;
margin-bottom: 20px;
}
.btn-sm {
padding: 5px 10px;
font-size: 12px;
margin: 2px;
}
.action-buttons {
white-space: nowrap;
}
</style>
</head>
<body>
<?php include BASE_PATH . '/views/partials/navbar.php'; ?>

<div class="container">
<div class="card">
<?php if(isset($_GET['created'])): ?>
<div class="success">✓ Usuario creado correctamente</div>
<?php endif; ?>
<?php if(isset($_GET['updated'])): ?>
<div class="success">✓ Usuario actualizado correctamente</div>
<?php endif; ?>
<?php if(isset($_GET['deleted'])): ?>
<div class="success">✓ Usuario eliminado correctamente</div>
<?php endif; ?>

```

```

<?php if(isset($_GET['deactivated'])): ?>
<div class="info">■ Usuario desactivado (tiene registros de asistencia)</div>
<?php endif; ?>
<?php if(isset($_GET['success'])): ?>
<div class="success">✓ Rol asignado correctamente</div>
<?php endif; ?>
<?php if(isset($_GET['removed'])): ?>
<div class="success">✓ Rol eliminado correctamente</div>
<?php endif; ?>
<?php if(isset($_GET['error']) && $_GET['error'] === 'has_assignments'): ?>
<div class="error">✗ No se puede eliminar el rol docente porque tiene asignaciones activas. Elimine primero las asignaciones.</div>
<?php endif; ?>
<?php if(isset($_GET['error']) && $_GET['error'] === 'not_found'): ?>
<div class="error">✗ Usuario no encontrado</div>
<?php endif; ?>
<?php if(isset($_GET['error']) && $_GET['error'] === 'self_delete'): ?>
<div class="error">✗ No puedes eliminar tu propia cuenta</div>
<?php endif; ?>
<?php if(isset($_GET['error']) && $_GET['error'] === 'delete_failed'): ?>
<div class="error">✗ Error al eliminar el usuario</div>
<?php endif; ?>

<div class="header-actions">
<h2>■ Gestión de Usuarios</h2>
<a href="?action=create_user" style="padding: 10px 20px; background: #28a745; color: white; text-decoration: none; border-radius: 4px; display: inline-block;">
+ Crear Usuario
</a>
</div>

<!-- Filtro por Rol -->
<form method="GET" action="" style="margin-bottom: 20px;">
<input type="hidden" name="action" value="users">
<div style="display: flex; gap: 15px; align-items: flex-end;">
<div style="flex: 1;">
<label style="display: block; margin-bottom: 5px; font-weight: bold; color: #333;">Filtrar por Rol</label>
<select name="filter_role" onchange="this.form.submit()" style="width: 100%; padding: 10px; border: 1px solid #ddd; border-radius: 4px;">
<option value="">Todos los roles</option>
<?php foreach($roles as $role): ?>
<option value="<?= $role['name'] ?>" <?= (isset($_GET['filter_role']) && $_GET['filter_role'] == $role['name']) ? 'selected' : '' ?>>
<?= ucfirst($role['name']) ?>
</option>
<?php endforeach; ?>
</select>
</div>
<?php if(isset($_GET['filter_role']) && $_GET['filter_role'] != ''): ?>
<div>
<a href="?action=users" style="padding: 10px 15px; background: #6c757d; color: white; text-decoration: none; border-radius: 4px; display: inline-block;">
Limpiar
</a>
</div>
<?php endif; ?>
</div>
</form>

<table>
<thead>
<tr>
<th>#</th>
<th>Nombre Completo</th>
<th>Email</th>

```

```

<th>Roles Actuales</th>
<th>Asignar Rol</th>
<th>Acciones</th>
</tr>
</thead>
<tbody>
<?php
// Aplicar filtro
$filteredUsers = $users;

if(isset($_GET['filter_role']) && $_GET['filter_role'] != '') {
$filterRole = $_GET['filter_role'];
$filteredUsers = array_filter($users, function($user) use ($filterRole) {
if($user['roles']) {
$userRoles = explode(',', $user['roles']);
return in_array($filterRole, $userRoles);
}
return false;
});
}
?>

<?php if(empty($filteredUsers)): ?>
<tr>
<td colspan="7" style="text-align:center;">No hay usuarios con este rol</td>
</tr>
<?php else: ?>
<?php
$counter = 1;
foreach($filteredUsers as $user):
?>
<tr>
<td><?= $counter++ ?></td>
<td><?= $user['last_name'] . ' ' . $user['first_name'] ?></td>
<td><?= $user['email'] ?></td>
<td>
<?php if($user['roles']): ?>
<?php
// Obtener IDs de roles con nombres
$db = new Database();
$sqlRoles = "SELECT r.id, r.name
FROM roles r
INNER JOIN user_roles ur ON r.id = ur.role_id
WHERE ur.user_id = :user_id
ORDER BY r.name";
$stmtRoles = $db->connect()->prepare($sqlRoles);
$stmtRoles->execute([':user_id' => $user['id']]);
$userRolesData = $stmtRoles->fetchAll(PDO::FETCH_ASSOC);

foreach($userRolesData as $roleData):
?>
<span class="badge">
<?= ucfirst($roleData['name']) ?>
<form method="POST" action="?action=remove_role<?= isset($_GET['filter_role']) ? '&filter_role=' .
$_GET['filter_role'] : '' ?>" style="display: inline;" onsubmit="return confirmRemoveRole(event, '<?=
ucfirst($roleData['name']) ?>', '<?= $user['last_name'] . ' ' . $user['first_name'] ?>')">
<input type="hidden" name="user_id" value="<?= $user['id'] ?>">
<input type="hidden" name="role_id" value="<?= $roleData['id'] ?>">
<button type="submit" class="btn-remove-role">x</button>
</form>
</span>
<?php endforeach; ?>
<?php else: ?>
<em>Sin roles</em>
<?php endif; ?>

```

```

</td>
<td>
<form method="POST" action="?action=assign_role= isset($_GET['filter_role']) ? '&amp;filter_role=' .
$_GET['filter_role'] : '' ?" style="display: inline-flex; gap: 5px;">
<input type="hidden" name="user_id" value="= $user['id'] ?">
<select name="role_id" required style="padding: 5px;">
<option value="">Seleccionar...</option>
<?php foreach($roles as $role): ?>
<option value="= $role['id'] ?"><?php ucfirst($role['name']) ?></option>
<?php endforeach; ?>
</select>
<button type="submit" style="padding: 5px 10px;">Asignar</button>
</form>
</td>
<td class="action-buttons">
<a href="?action=edit_user&id== $user['id'] ?"
class="btn-warning btn-sm"
style="text-decoration: none; padding: 5px 10px; border-radius: 4px;">
=■ Editar
</a>
<form method="POST" action="?action=delete_user" style="display: inline;" onsubmit="return
confirmDelete(event, '= $user['last_name'] . ' ' . $user['first_name'] ?&gt;')"&gt;
&lt;input type="hidden" name="user_id" value="<?= $user['id'] ?">
<button type="submit" class="btn-danger btn-sm">
■■ Eliminar
</button>
</form>
</td>
</tr>
<?php endforeach; ?>
<?php endif; ?>
</tbody>
</table>
</div>
</div>

<script>
function confirmDelete(event, userName) {
event.preventDefault();

const modal = document.createElement('div');
modal.style.cssText = 'position: fixed; top: 0; left: 0; width: 100%; height: 100%; background:
rgba(0,0,0,0.5); display: flex; align-items: center; justify-content: center; z-index: 9999;';

const modalContent = document.createElement('div');
modalContent.style.cssText = 'background: white; padding: 30px; border-radius: 8px; max-width: 500px;
box-shadow: 0 4px 20px rgba(0,0,0,0.3);';

modalContent.innerHTML = `
<h3 style="margin: 0 0 15px 0; color: #dc3545;">■■ Eliminar Usuario</h3>
<p style="margin: 0 0 20px 0; color: #666;">
¿Está seguro de eliminar al usuario <strong>${userName}</strong>?
</p>
<p style="margin: 0 0 20px 0; color: #666; font-size: 14px;">
<strong>Nota:</strong> Si el usuario tiene asistencias registradas, será desactivado en lugar de eliminado.
</p>
<div style="display: flex; gap: 10px; justify-content: flex-end;">
<button type="button" id="cancelDeleteBtn" style="padding: 10px 20px; background: #6c757d; color: white;
border: none; border-radius: 4px; cursor: pointer;">
Cancelar
</button>
<button type="button" id="confirmDeleteBtn" style="padding: 10px 20px; background: #dc3545; color: white;
border: none; border-radius: 4px; cursor: pointer;">
Sí, Eliminar
</button>

```

```

</div>
`;

modal.appendChild(modalContent);
document.body.appendChild(modal);

const form = event.target;

document.getElementById('confirmDeleteBtn').onclick = function() {
document.body.removeChild(modal);
form.submit();
};

document.getElementById('cancelDeleteBtn').onclick = function() {
document.body.removeChild(modal);
};

return false;
}

function confirmRemoveRole(event, roleName, userName) {
event.preventDefault();

const modal = document.createElement('div');
modal.style.cssText = 'position: fixed; top: 0; left: 0; width: 100%; height: 100%; background:
rgba(0,0,0,0.5); display: flex; align-items: center; justify-content: center; z-index: 9999;';

const modalContent = document.createElement('div');
modalContent.style.cssText = 'background: white; padding: 30px; border-radius: 8px; max-width: 500px;
box-shadow: 0 4px 20px rgba(0,0,0,0.3);';

modalContent.innerHTML = `
<h3 style="margin: 0 0 15px 0; color: #dc3545;">■■■ Eliminar Rol</h3>
<p style="margin: 0 0 20px 0; color: #666;">
¿Está seguro de eliminar el rol <strong>${roleName}</strong> del usuario <strong>${userName}</strong>?
</p>
<div style="display: flex; gap: 10px; justify-content: flex-end;">
<button type="button" id="cancelRemoveRoleBtn" style="padding: 10px 20px; background: #6c757d; color: white;
border: none; border-radius: 4px; cursor: pointer;">
Cancelar
</button>
<button type="button" id="confirmRemoveRoleBtn" style="padding: 10px 20px; background: #dc3545; color:
white; border: none; border-radius: 4px; cursor: pointer;">
Sí, Eliminar
</button>
</div>
`;

modal.appendChild(modalContent);
document.body.appendChild(modal);

const form = event.target;

document.getElementById('confirmRemoveRoleBtn').onclick = function() {
document.body.removeChild(modal);
form.submit();
};

document.getElementById('cancelRemoveRoleBtn').onclick = function() {
document.body.removeChild(modal);
};

return false;
}
</script>

```

```
</body>
</html>
```

## 81. README.md

```
## EcuSistenc A s i s t e n c i a 2026
#
# EcuSistenc A s i s t 2026 - S i s t e m a d e A s i s t e n c i a E s c o l l a r
#
#
# S i s t e m a d e g e s t i o n d e a s i s t e n c i a e s c o l l a r
# d e s a r r o l l a d o e n P H P p u r o o r i e n t a d o a o b j e t o s ,
# M y S Q L y s i n f r a m e w o r k s .
#
#
# ## = C a r a c t e r s t i c a s
#
#
# ## M o d u l o s P r i n c i p a l e s
#
# - ' A u t e n t i c a c i o n y r o l l e s m u l t i - u s u a r i o
#
# - ' G e s t i o n d e u s u a r i o s ( e s t u d i a n t e s , d o c e n t e s ,
# a u t o r i d a d e s , i n s p e c t o r e s , r e p r e s e n t a n t e s )
#
# - ' R e g i s t r o d e a s i s t e n c i a c o n v a l i d a c i o n d e
# f e c h a s
#
# - ' S i s t e m a d e j u s t i f i c a c i o n e s c o n a p r o b a c i o n
#
# - ' R e p o r t e s e n P D F y E x c e l
#
# - ' C a l e n d a r i o d e a s i s t e n c i a s
#
# - ' S i s t e m a d e h o r a r i o s p o r c u r s o
# ( d a a / h o r a / m a t e r i a / d o c e n t e )
#
# - ' C o n f i g u r a c i o n d e i n s t i t u c i o n c o n
# p r o v i n c i a s / c i u d a d e s E c u a d o r
#
# - ' G e s t i o n d e j o r n a d a s p o r i n s t i t u c i o n
# ( m a t u t i n a / v e s p e r t i n a / n o c t u r n a )
#
# - ' G e s t i o n s e p a r a d a d e t u t o r e s
#
# - ' E s t a d s t i c a s a v a n z a d a s
#
# - ' A s i g n a c i o n e s d o c e n t e s y t u t o r e s
#
# - ' R e l a c i o n e s r e p r e s e n t a n t e - e s t u d i a n t e
#
# - ' S i s t e m a d e r e s p a l d o s
#
# - ' L o g s d e a c t i v i d a d
#
#
# ## T e c n o l o g a s
#
# - P H P 7 . 4 +
```

```

■
■-■ ■M■y■S■Q■L■ ■5■.■7■+■ ■/■ ■M■a■r■i■a■D■B■
■
■-■ ■P■D■O■ ■p■a■r■a■ ■b■a■s■e■ ■d■e■ ■d■a■t■o■s■
■
■-■ ■P■H■P■M■a■i■l■e■r■ ■p■a■r■a■ ■c■o■r■r■e■o■s■
■
■-■ ■T■C■P■D■F■ ■p■a■r■a■ ■P■D■F■
■
■-■ ■P■h■p■S■p■r■e■a■d■s■h■e■e■t■ ■p■a■r■a■ ■E■x■c■e■l■
■
■
■
■#■#■ ■=■ ■I■n■s■t■a■l■l■a■c■i■n■
■
■
■
■#■#■#■ ■R■e■q■u■i■s■i■t■o■s■
■
■-■ ■X■A■M■P■P■ ■/■ ■W■A■M■P■ ■/■ ■L■A■M■P■
■
■-■ ■P■H■P■ ■>■=■ ■7■.■4■
■
■-■ ■M■y■S■Q■L■ ■>■=■ ■5■.■7■
■
■-■ ■C■o■m■p■o■s■e■r■
■
■
■
■#■#■#■ ■P■a■s■o■s■
■
■
■
■1■.■ ■*■*■C■l■o■n■a■r■ ■r■e■p■o■s■i■t■o■r■i■o■*■*■
■
■`■`■`■b■a■s■h■
■
■g■i■t■ ■c■l■o■n■e■
■h■t■t■p■s■:■/■/■g■i■t■h■u■b■.■c■o■m■/■t■u■-■u■s■u■a■r■i■o■/■e■c■u■a■s■i■s■t■-■2■0■2■6■.■g■i■t■
■
■c■d■ ■e■c■u■a■s■i■s■t■-■2■0■2■6■
■
■`■`■`■
■
■
■2■.■ ■*■*■I■n■s■t■a■l■l■a■r■ ■d■e■p■e■n■d■e■n■c■i■a■s■*■*■
■
■`■`■`■b■a■s■h■
■
■c■o■m■p■o■s■e■r■ ■i■n■s■t■a■l■l■
■
■`■`■`■
■
■
■3■.■ ■*■*■C■r■e■a■r■ ■b■a■s■e■ ■d■e■ ■d■a■t■o■s■*■*■
■
■-■ ■I■m■p■o■r■t■a■r■ ■`■e■c■u■a■s■i■s■t■e■n■c■i■a■2■0■2■6■_■d■b■.■s■q■l■`■ ■e■n■
■p■h■p■M■y■A■d■m■i■n■
■
■
■
■4■.■ ■*■*■C■o■n■f■i■g■u■r■a■r■*■*■
■

```

```

-- E_d_i_t_a_r `c_o_n_f_i_g/d_a_t_a_b_a_s_e..p_h_p` c_o_n c_r_e_d_e_n_c_i_a_l_e_s
M_y_S_Q_L
-- E_d_i_t_a_r `c_o_n_f_i_g/c_o_n_f_i_g..p_h_p` c_o_n U_R_L_b_a_s_e_y
c_o_n_f_i_g_u_r_a_c_i_n S_M_T_P
6. ***C_r_e_a_r c_a_r_p_e_t_a_s d_e_e u_p_l_o_a_d_s***
` `` `b_a_s_h
m_k_d_i_r u_p_l_o_a_d_s/i_n_s_t_i_t_u_t_i_o_n
7. ***P_e_r_m_i_s_o_s*** (L_i_n_u_x/M_a_c)***
` `` `b_a_s_h
c_h_m_o_d -R 755 .
c_h_m_o_d -R 777 u_p_l_o_a_d_s
c_h_m_o_d -R 777 b_a_c_k_u_p_s
` ``
8. ***A_c_c_e_d_e_r***
` ``
h_t_p://l_o_c_a_l_h_o_s_t/e_c_u_a_s_i_s_t_e_n_c_i_a_2026/p_u_b_l_i_c/
` ``
## # =d C_r_e_d_e_n_c_i_a_l_e_s p_o_r D_e_f_e_c_t_o
U_s_u_a_r_i_o d_e_e p_r_u_e_b_a (d_e_s_p_u_s d_e_e i_m_p_o_r_t_a_r S_Q_L d_e_e
p_r_u_e_b_a):
-- ***U_s_u_a_r_i_o:*** p_r_o_f.g_a_r_c_i_a
-- ***C_o_n_t_r_a_s_e_a:*** p_a_s_s_w_o_r_d
## # = E_s_t_r_u_c_t_u_r_a d_e_e_l P_r_o_y_e_c_t_o
` ``
e_c_u_a_s_i_s_t/
% % % c_o_n_f_i_g/ # C_o_n_f_i_g_u_r_a_c_i_n
% % % c_o_n_t_r_o_l_l_e_r_s/ # C_o_n_t_r_o_l_a_d_o_r_e_s
% % % m_o_d_e_l_s/ # M_o_d_e_l_o_s d_e_e d_a_t_o_s

```



```

■ %%% vview/ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ # vistsas HTML/PHP
■
■ %%% vview/schedulle/ ■ ■ ■ ■ ■ ■ ■ # Getsitinn dee hooraario
■
■ %%% vview/sititutio/ ■ ■ ■ ■ ■ ■ ■ # Confing in sititucion
■
■ %%% uplloads/sititutio/ ■ ■ ■ ■ ■ ■ ■ # Logs
■ in sititucionales
■
■ %%% helpes/ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ # Utillidades
■
■ %%% public/ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ # Puntto dee enttrad
■
■ %%% uplloads/ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ # Archivos subidos
■
■ %%% backups/ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ # Respalldos autotomaticos
■
■ %%% vedor/ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ # Deependendcias Compoeer
■
■ `~`~`~
■
■
■
■
■ ## ■= ■ Rolles y Permisos
■
■
■
■
■ - ■*■*■ Autotridad: ■*■*■ Accesos ototall all sistema
■
■ - ■*■*■ Docentte: ■*■*■ Regisrtro dee asisstenencia, vee
■ cursos asilgnados
■
■ - ■*■*■ Insepecto: ■*■*■ Vee asisstenencias, aporobar
■ jutsitificacicones
■
■ - ■*■*■ Estudidiantte: ■*■*■ Vee su ppropia asisstenencia,
■ jutsitificacar ausenencias
■
■ - ■*■*■ Represenatante: ■*■*■ Vee asisstenencia dee
■ represenatados
■
■
■
■ ## ■= ■ Reporrtes
■
■
■
■
■ - ■*■*■ PDF: ■*■*■ Formatto in sititucionall con llogo y
■ encabezados
■
■ - ■*■*■ Excel: ■*■*■ Datos estrtuturados para anllisis
■
■ - ■*■*■ Calendario: ■*■*■ Vistsa mensusual dee asisstenencia
■
■ - ■*■*■ Estadsticass: ■*■*■ Mtricass poor cursos,
■ estudidiantte y pperoodo
■
■
■
■
■ ## ■= ■ Vallidaciones Imporrtantes
■
■
■
■
■ - ■ Regisrtro dee asisstenencia: ■ mxximo 48 hooras
■ hbbiles atres

```

```
-# NNo ssee ppermmittte rreggiissttroo dde ffeechhaas futtuuraas
-# Finnee s dde seemaanaa noo cuueenntaan een eel cllcullo dde
488 hooraas
-# Ediciinn dde asissteenncaa: haasta 488 hooraas
ddee spuuss
-#
-#
-#
-## = Sleguuriddad
-#
-#
-#
-# Cconttraaseeaa shaashheaddaa con `paasswoord_haash`
-# Pprotteecciiinn CSRF een foormullarrioss
-# Vallidaciiinn dde ennttraadaas
-# Saanitiizaciiinn dde daattoos
-# Cconttrooll dde accceesso poor roollees
-#
-#
-#
-## = Soopportee
-#
-#
-#
Paara soopportee oo conssulttaas:
sooportee@ecuaasistt,.edu.ec
-#
-#
-#
-## = Licencaa
-#
-#
-#
Proyyecto educaatiivo -# Uuso libree con atribucciinn
-#
-#
-#
-## =' Manteenimientoo
-#
-#
-#
-#### Respalldoos
-#
-# Aautoomtticcoos een `/backups`
-#
-# See recommienda ddeescargaar seemaanaalmentee
-#
-# Limpiiiza a autoomtticcaa dde >300 ddaas
-#
-#
-#
-##### Loggs
-#
-# Acctiividadd dde usuarioss een `acctiivit_y_lloggs`
-#
-# Revisaar peerriiddicamementee een paaneel dde aautooriiddad
```

```

■
■*■*■E■n■R■o■a■d■m■a■p■(■r■e■e■m■p■l■a■z■a■r■t■o■d■o■):■*■*■
■
■`■`■`■m■a■r■k■d■o■w■n■
■
■-■[■x■]■S■i■s■t■e■m■a■d■e■h■o■r■a■r■i■o■s■
■
■-■[■x■]■C■o■n■f■i■g■u■r■a■c■i■o■n■m■u■l■t■i■-■j■o■r■n■a■d■a■
■
■-■[■x■]■P■r■o■v■i■n■c■i■a■s■/■c■i■u■d■a■d■e■s■E■c■u■a■d■o■r■
■
■-■[■ ]■G■r■f■i■c■o■s■i■n■t■e■r■a■c■t■i■v■o■s■
■
■-■[■ ]■B■s■g■u■e■d■a■g■l■o■b■a■l■
■
■-■[■ ]■M■o■d■o■o■s■c■u■r■o■
■
■
■
■-■-■-■
■
■D■e■s■a■r■r■o■l■l■a■d■o■c■o■n■d'■p■a■r■a■i■n■s■t■i■t■u■c■i■o■n■e■s■
■e■d■u■c■a■t■i■v■a■s■e■c■u■a■t■o■r■i■a■n■a■s■

```

## 82. ResumenEjecutivo\_04C.md

■ RESUMEN EJECUTIVO – ECUASIST 2026 (Consolidado)

Fecha: 17 de Febrero de 2026

Versión: v1.6 – Sistema consolidado con mejoras UX/UI, infraestructura y módulos completos

■ ESTADO DEL PROYECTO

Sistema de asistencia escolar desarrollado en PHP OOP puro + MySQL, sin frameworks.

Módulos completados: 18/18

Bugs críticos: 0

Enfoque: Usabilidad, consistencia visual, automatización académica y estabilidad operativa

Listo para producción tras verificación final de rutas, uploads y timezone

■ MEJORAS IMPLEMENTADAS

1■ Sistema de Horarios

Tabla class\_schedule con estructura por día y período

Validación anti-duplicados por curso/día/hora

Auto-asignación docente desde teacher\_assignments

Horas diferenciadas por nivel académico

Vista de gestión por curso

Detección de conflictos en tiempo real (AJAX)

Flujo

Crear curso

Asignar docente-materia

Configurar horario

Docente visualiza clases automáticamente

2■■ Registro de Asistencia Inteligente

Eliminada selección manual de curso/materia

Docente ve clases programadas del día

Interfaz visual tipo tarjetas

Validación 48h hábiles

UPDATE automático si existe registro previo

Método clave

getScheduleInfo()

3■■ Asignaciones Docentes Reorganizadas

A – Docente-Materia

Filtros avanzados

Validación única por curso

B – Tutor de Curso

Vista independiente

Solo docentes elegibles

Confirmación modal

Restricción: 1 tutor por docente

Debe impartir materia en el curso

4■■ Configuración Institucional

Campos nuevos:

Provincia

Ciudad

Director

AMIE

Web

Logo

Tabla nueva:

institution\_shifts

Funciones:

Jornadas múltiples

Select cascada Ecuador

Autocompletar URL

Gestión visual de jornadas

5■■■ Reportes PDF y Excel

Institución dinámica desde BD

Vista previa estable

Nombre de archivo limpio y contextual

Eliminación duplicidad de jornada

Corrección entidades HTML

6■■■ Módulo de RespalDOS

Interfaz completa

Detección automática mysqldump

Validación tamaño archivo

Eliminación individual

Limpieza automática > 30 días

Ruta pendiente

delete\_backup

7■■■ Gestión de Representantes

Filtros en tiempo real

Eliminación de relaciones con confirmación

Método removeRelation()

Ruta pendiente

remove\_representative

8■■■ Diseño Unificado Bootstrap

Parciales globales:

head.php

footer.php

Migración iniciada (2/23 vistas)

Pendiente migración total

9■■■ UX/UI Global

Navbar sticky

Modales personalizados

Filtros persistentes

Ordenamiento lógico

Badges visuales

Warnings temporales

Colores semánticos consistentes

■ Validaciones Críticas

Asistencia sin duplicados

Tutor único

Materia única por curso

Horario sin conflicto

Roles protegidos

Estudiante único por año

Jornadas múltiples

Eliminaciones protegidas

■■ BASE DE DATOS

Tablas principales

institutions

users

roles

permissions

Académico

school\_years

courses

subjects

teacher\_assignments

course\_students

class\_schedule

institution\_shifts

Asistencia

attendances

justifications

Sistema

notifications

activity\_logs

representatives

■■ ESTRUCTURA DEL PROYECTO

Arquitectura MVC modular con:

Models académicos y sistema

Controllers especializados

Helpers seguridad, mail, backup, logging

Views Bootstrap unificadas progresivamente

Router central public/index.php

■■ CONFIGURACIÓN

Zona horaria Ecuador

Sesiones persistentes 24h

Timeout 30 min

BASE\_PATH definido

Cookies seguras SameSite Lax

Carpetas con permisos de escritura:

/uploads

/uploads/institution

/backups

#### ■ BUGS CORREGIDOS

Roles incorrectos al eliminar

Asistencia duplicada

Selectores sin filtrado

Validación URL estricta

Guardado de logo

Reportes con HTML entities

Sesiones mal inicializadas

Vista previa PDF/Excel

Backups vacíos

#### ■ NAVBAR ACTUALIZADO

Dashboard

Asistencia

Justificaciones

Administración

Usuarios

Académico

Asignaciones

Tutores

Representantes

Horarios

Institución

RespalDOS

Reportes

Representados

#### ■ HELPERS CLAVE

Security::requireLogin()  
Security::hasRole()  
  
Security::sanitize()  
Logger->log()  
html\_entity\_decode()

#### ■ DEPENDENCIAS

phpmailer/phpmailer  
phpoffice/phpspreadsheet  
tecnickcom/tcpdf

#### ■ PRÓXIMOS PASOS

Alta prioridad

Migrar vistas restantes a Bootstrap

Agregar rutas pendientes

Pruebas completas módulo backups

Media

Toast notifications

Breadcrumbs

Gráficos estadísticos

Búsqueda global

Baja

Modo oscuro

Calendario visual horarios

#### ■ ESTADO FINAL

Área Estado  
Arquitectura Estable  
Módulos Completos  
UX Mejorada  
Seguridad Sólida  
Base datos Normalizada  
Bootstrap En proceso

#### ■ CONCLUSIÓN

El sistema alcanzó madurez funcional con mejoras en:

Automatización académica

Estabilidad operativa

Consistencia visual

Escalabilidad modular

Actualmente el proyecto se encuentra en fase de optimización y refinamiento más que construcción base.

## 83. composer.json

```
{  
  "require": {
```



```
"phpmailer/phpmailer": "^7.0",  
"phpoffice/phpspreadsheet": "^5.4",  
"tecnickcom/tcpdf": "^6.10"  
}  
}
```