

EcuaSistencia

Sistema de Gestión de Asistencia Escolar

Documentación Completa del Proyecto PHP

Estructura de Carpetas

ArchImp/
backups/
config/
.htaccess
config.php
database.php
env.php
controllers/
AcademicController.php
AssignmentController.php
AttendanceController.php
AuthController.php
BackupController.php
DashboardController.php
ExportController.php
InstitutionController.php
JustificationController.php
NotificationController.php
ProfileController.php
ReportController.php
ReportController2.php
RepresentativeController.php
ScheduleController.php
SearchController.php
StatsController.php
TutorController.php
UserController.php
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
public/

.htaccess
img.php
index.php
css/
global.css
js/
views/
academic/
course_edit.php
course_enroll.php
course_students.php
course_subject_edit.php
course_subjects.php
create.php
edit.php
enroll.php
index.php
school_year_create.php
school_year_edit.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
reviewed.php
submit.php
notifications/
index.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

tutor/

course_attendance.php

dashboard.php

no_tutor.php

search_students.php

users/

create.php

edit.php

index.php

Índice de Archivos

CONFIG

1. config/.htaccess
2. config/config.php
3. config/database.php
4. config/env.php

CONTROLLERS

5. controllers/AcademicController.php
6. controllers/AssignmentController.php
7. controllers/AttendanceController.php
8. controllers/AuthController.php
9. controllers/BackupController.php
10. controllers/DashboardController.php
11. controllers/ExportController.php
12. controllers/InstitutionController.php
13. controllers/JustificationController.php
14. controllers/NotificationController.php
15. controllers/ProfileController.php
16. controllers/ReportController.php
17. controllers/ReportController2.php
18. controllers/RepresentativeController.php
19. controllers/ScheduleController.php
20. controllers/SearchController.php
21. controllers/StatsController.php
22. controllers/TutorController.php
23. controllers/UserController.php

MODELS

24. models/Attendance.php
25. models/ClassSchedule.php
26. models/Course.php
27. models/Institution.php
28. models/InstitutionShift.php
29. models/Justification.php
30. models/Notification.php
31. models/Representative.php
32. models/Role.php
33. models/SchoolYear.php
34. models/Shift.php
35. models/Subject.php
36. models/TeacherAssignment.php
37. models/User.php

HELPERS

38. helpers/Backup.php
39. helpers/Logger.php
40. helpers/Mailer.php
41. helpers/Security.php

VIEWS

42. views/attendance/calendar.php
43. views/attendance/course_view.php
44. views/attendance/my_attendance.php
45. views/attendance/register.php

- 46. views/attendance/view.php
- 47. views/academic/course_edit.php
- 48. views/academic/course_enroll.php
- 49. views/academic/course_students.php
- 50. views/academic/course_subject_edit.php
- 51. views/academic/course_subjects.php
- 52. views/academic/create.php
- 53. views/academic/edit.php
- 54. views/academic/enroll.php
- 55. views/academic/index.php
- 56. views/academic/school_year_create.php
- 57. views/academic/school_year_edit.php
- 58. views/academic/subject_edit.php
- 59. views/users/create.php
- 60. views/users/edit.php
- 61. views/users/index.php
- 62. views/representatives/child_attendance.php
- 63. views/representatives/manage.php
- 64. views/representatives/my_children.php
- 65. views/tutor/course_attendance.php
- 66. views/tutor/dashboard.php
- 67. views/tutor/no_tutor.php
- 68. views/tutor/search_students.php
- 69. views/justifications/my_list.php
- 70. views/justifications/pending.php
- 71. views/justifications/reviewed.php
- 72. views/justifications/submit.php
- 73. views/schedule/index.php
- 74. views/schedule/manage.php
- 75. views/institution/index.php
- 76. views/stats/index.php
- 77. views/dashboard/index.php
- 78. views/partials/navbar.php
- 79. views/backup/index.php
- 80. views/auth/forgot.php
- 81. views/auth/login.php
- 82. views/auth/register.php
- 83. views/auth/reset.php
- 84. views/profile/change_password.php
- 85. views/profile/edit.php
- 86. views/profile/view.php
- 87. views/assignments/index.php
- 88. views/assignments/tutor.php
- 89. views/assignments/view_course.php
- 90. views/reports/index.php
- 91. views/notifications/index.php

PUBLIC

- 92. public/.htaccess
- 93. public/img.php
- 94. public/index.php
- 95. public/css/global.css

CONFIG

config/.htaccess

```
# Proteger directorio uploads
Options -Indexes
<FilesMatch "\.(pdf|jpg|jpeg|png)$">
    Order Allow,Deny
    Allow from all
</FilesMatch>
```

config/config.php

```
<?php
// config/config.php
// BASE_PATH y BASE_URL vienen de env.php
require_once __DIR__ . '/env.php';

date_default_timezone_set('America/Guayaquil');

define('EDIT_ATTENDANCE_HOURS', 48);

// Sesión
ini_set('session.gc_maxlifetime', 86400);
session_set_cookie_params([
    'lifetime' => 86400,
    'path' => '/',
    'domain' => '',
    'secure' => (!empty($_SERVER['HTTPS']) && $_SERVER['HTTPS'] !== 'off'),
    'httponly' => true,
    'samesite' => 'Lax',
]);
session_start();

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

// SMTP
define('SMTP_HOST', 'smtp.gmail.com');
define('SMTP_PORT', 587);
define('SMTP_USER', 'tu-email@gmail.com');
define('SMTP_PASS', 'tu-password-app');
define('SMTP_FROM', 'noreply@ecuasist.edu.ec');
define('SMTP_NAME', 'EcuAsistencia2026');

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

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;
}

```

config/env.php

```

<?php
// Detecta automáticamente si estás en local o producción
if ($_SERVER['SERVER_NAME'] === 'localhost' || $_SERVER['SERVER_NAME'] === '127.0.0.1') {
    // LOCAL
    define('BASE_PATH', __DIR__ . '/../');
    define('BASE_URL', 'http://localhost/ecuasistencia2026/public');
} else {
    // PRODUCCIÓN
    define('BASE_PATH', '/home/ecuasys/ecuasistencia');
    define('BASE_URL', 'https://www.ecuasys.com');
}

```


CONTROLLERS

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;
    private $db;

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

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

    public function index() {
        $courses = $this->courseModel->getAllWithTutor();
        $subjects = $this->subjectModel->getAll();
        $schoolYears = $this->schoolYearModel->getAll();
        $shifts = $this->shiftModel->getAll();
        $teachers = $this->userModel->getByRole('docente');

        // Asignaciones docente-materia
        require_once BASE_PATH . '/models/TeacherAssignment.php';
        $assignmentModel = new TeacherAssignment($this->db);

        $assignments = $assignmentModel->getAll();
        // Solo asignaciones de docente-materia (no tutores)
        $assignments = array_filter($assignments, fn($a) => !$a['is_tutor']);

        // Datos para modal de estudiantes
        $activeYear = $this->schoolYearModel->getActive();
        $availableStudents = $activeYear ? $this->userModel->getStudentsNotEnrolled($activeYear['id']) : [];
        $enrollmentsByCourse = [];
        $subjectsByCourse = [];
        foreach ($courses as $c) {
            $enrollmentsByCourse[$c['id']] = $this->courseModel->getEnrolledStudents($c['id']);
            $subjectsByCourse[$c['id']] = $this->getSubjectsByCourse($c['id']);
        }

        // Representantes por estudiante
        require_once BASE_PATH . '/models/Representative.php';
        $repModel = new Representative($this->db);
        $representatives = $this->userModel->getByRole('representante');
        $repsByStudent = [];
        foreach ($enrollmentsByCourse as $courseId => $students) {
            foreach ($students as $st) {
                $repsByStudent[$st['id']] = $repModel->getRepresentativesByStudent($st['id']);
            }
        }
    }
}
```

```

}
}

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

```

```

public function createCourse() {
if ($_SERVER['REQUEST_METHOD'] !== 'POST') return;

// Guardar datos del form en sesión para repoblar si hay error
$_SESSION['course_form'] = [
'education_type' => $_POST['education_type'] ?? '',
'grade_level' => $_POST['grade_level'] ?? '',
'specialty' => $_POST['specialty'] ?? '',
'carrera' => $_POST['carrera'] ?? '',
'parallel' => $_POST['parallel'] ?? '',
'shift_id' => $_POST['shift_id'] ?? '',
'name' => $_POST['name'] ?? '',

```

```

];

```

```

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

```

```

$gradeLevel = html_entity_decode(Security::sanitize($_POST['grade_level']), ENT_QUOTES, 'UTF-8');
$parallel = Security::sanitize($_POST['parallel']);
$shiftId = (int)$_POST['shift_id'];

```

```

// Validar curso duplicado
$pdo = $this->db->connect();
$stmtCheck = $pdo->prepare(
"SELECT COUNT(*) as cnt FROM courses
WHERE institution_id=:iid AND school_year_id=:syid
AND grade_level=:gl AND parallel=:par AND shift_id=:sid"
);
$stmtCheck->execute([':iid'=>$_SESSION['institution_id'], ':syid'=>$activeYear['id'],
':gl'=>$gradeLevel, ':par'=>$parallel, ':sid'=>$shiftId]);
if ($stmtCheck->fetch()['cnt'] > 0) {
header('Location: ?action=academic&error=course_duplicate'); exit;
}

```

```

$data = [
':institution_id' => $_SESSION['institution_id'],
':school_year_id' => $activeYear['id'],
':name' => html_entity_decode(Security::sanitize($_POST['name']), ENT_QUOTES, 'UTF-8'),
':grade_level' => $gradeLevel,
':parallel' => $parallel,
':shift_id' => $shiftId,
];
$this->courseModel->create($data);
$courseId = (int)$this->db->connect()->lastInsertId();

```

```

// Auto-carga malla curricular
$malla = $this->getMallaCurricular();
$nuevas = 0;
if (isset($malla[$gradeLevel]) && $courseId > 0) {
foreach ($malla[$gradeLevel] as $nombre) {

```

```

$stmtB = $pdo->prepare("SELECT id FROM subjects WHERE institution_id=:iid AND
LOWER(TRIM(name))=LOWER(TRIM(:name)) LIMIT 1");
$stmtB->execute([':iid'=>$_SESSION['institution_id'], ':name'=>$nombre]);
$existe = $stmtB->fetch();
if ($existe) {
$sid = (int)$existe['id'];
} else {
$codigo = strtoupper(substr(preg_replace('/[^\a-zA-Z]/',' ',iconv('UTF-8','ASCII//TRANSLIT',$nombre)),0,4));
$this->subjectModel->create([':institution_id'=>$_SESSION['institution_id'], ':name'=>$nombre, ':code'=>$codigo]);
$sid = (int)$pdo->lastInsertId();
$nuevas++;

```

```

}
$this->linkSubjectToCourse($courseId, $sid);
}
}

// Éxito: limpiar datos guardados
unset($_SESSION['course_form']);
header('Location: ?action=academic&course_success=1&subjects_loaded='.$nuevas); 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' => $courseId,
'name' => html_entity_decode(Security::sanitize($_POST['name']), ENT_QUOTES, 'UTF-8'),
'grade_level' => html_entity_decode(Security::sanitize($_POST['grade_level']), ENT_QUOTES, 'UTF-8'),
'parallel' => Security::sanitize($_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');

```

[illegible]

[illegible]

```
header('Location: ?action=academic&open_subjects='.$courseId.'&error=hours_full&available='.$validation['available'].'&assigned='.$validation['currentTotal']);
exit;
}
$this->linkSubjectToCourse($courseId, $sid);
}
header('Location: ?action=academic&open_subjects='.$courseId.'&added=1'); exit;
}
```

[illegible][illegible]

```
$this->subjectModel->update(['id'=>$subjectId,'name'=>Security::sanitize($_POST['name'],'code'=>Security::sanitize($_POST['code']))]);
header('Location: ?action=course_subjects&course_id='.$courseId.'&updated=1'); exit;
}
include BASE_PATH . '/views/academic/course_subject_edit.php';
}
```

```
// Eliminar de course_subjects
$pdo->prepare("DELETE FROM course_subjects WHERE course_id=:cid AND subject_id=:sid")
->execute([':cid'=>$courseId, ':sid'=>$subjectId]);
```

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

```
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';
    }
}
```

```

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();

if ($_SERVER['REQUEST_METHOD'] === 'POST') {
$courseId = (int)$_POST['course_id'];
$redirectTo = $_POST['redirect_to'] ?? 'enroll_students';

$studentIds = $_POST['student_ids'] ?? [];

$enrolled = 0;
$errors = 0;

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

$openParam = ($redirectTo === 'academic') ? '&open_students=' . $courseId : '&course_id=' . $courseId;
header('Location: ?action=' . $redirectTo . $openParam . '&enrolled=' . $enrolled . '&errors=' . $errors);
exit;
}

$courseId = (int)($_GET['course_id'] ?? 0);
if (!$courseId) {
header('Location: ?action=academic');
exit;
}

$course = $this->courseModel->findById($courseId);
if (!$course) {
header('Location: ?action=academic');
exit;
}

$availableStudents = $this->userModel->getStudentsNotEnrolled($activeYear['id']);
$allStudents = $this->userModel->getByRole('estudiante');

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

public function unenrollStudent() {
if ($_SERVER['REQUEST_METHOD'] === 'POST') {
$studentId = (int)$_POST['student_id'];
$courseId = (int)($_POST['course_id'] ?? 0);

```



```

$redirectTo = $_POST['redirect_to'] ?? 'enroll_students';
$activeYear = $this->schoolYearModel->getActive();

$courseParam = ($redirectTo === 'academic') ? 'open_students=' . $courseId : 'course_id=' . $courseId;
$base = '?action=' . $redirectTo . '&' . $courseParam;

if (!$activeYear) {
    header('Location: ' . $base . '&error=no_active_year');
    exit;
}

$course = $this->userModel->getStudentCourse($studentId, $activeYear['id']);

if (!$course) {
    header('Location: ' . $base . '&error=not_enrolled');
    exit;
}

if ($this->courseModel->unenrollStudent($studentId, $activeYear['id'])) {
    header('Location: ' . $base . '&unenrolled=1');
    exit;
} else {
    header('Location: ' . $base . '&error=unenroll_failed');
    exit;
}
}

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

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

    $course = $this->courseModel->findById($courseId);
    if (!$course) {
        header('Location: ?action=academic');
        exit;
    }
}

```

```

}

$activeYear = $this->schoolYearModel->getActive();
$students = $this->courseModel->getEnrolledStudents($courseId);
$availableStudents = $activeYear ? $this->userModel->getStudentsNotEnrolled($activeYear['id']) : [];

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 ANTES de insertar
if ($data['is_active']) {
$this->schoolYearModel->activate(0);
}

```

```

// create() ahora devuelve el ID insertado (o 0 si falla)
$newId = $this->schoolYearModel->create($data);

```

```

if ($newId > 0) {
// Activar el nuevo año lectivo con el ID real
if ($data['is_active']) {
$this->schoolYearModel->activate($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;
        }
    }
}

public function setSubjectHours() {
    if ($_SERVER['REQUEST_METHOD'] !== 'POST') {
        header('Location: ?action=academic'); exit;
    }
    $courseId = (int)($_GET['course_id'] ?? $_POST['course_id'] ?? 0);
    $subjectId = (int)$_POST['subject_id'];
    $hours = max(1, min(20, (int)$_POST['hours_per_week']));
    $openPanel = $_POST['open_panel'] ?? 'subjects'; // subjects | students

    $pdo = $this->db->connect();

    // Cuantas horas ya asignadas en el horario

```

```

$stmtCount = $pdo->prepare(
    "SELECT COUNT(*) FROM class_schedule

```

```

WHERE course_id = :cid AND subject_id = :sid"
);
$stmtCount->execute([':cid' => $courseId, ':sid' => $subjectId]);
$assigned = (int)$stmtCount->fetchColumn();

// Si bajaron las horas por debajo de lo asignado, eliminar las sobrantes
if ($assigned > $hours) {
    $toDelete = $assigned - $hours;
    // Eliminar las ultimas entradas (ORDER BY id DESC para quitar las mas recientes)
    $stmtIds = $pdo->prepare(
        "SELECT id FROM class_schedule
        WHERE course_id = :cid AND subject_id = :sid
        ORDER BY id DESC LIMIT :lim"
    );
    $stmtIds->bindValue(':cid', $courseId, PDO::PARAM_INT);
    $stmtIds->bindValue(':sid', $subjectId, PDO::PARAM_INT);
    $stmtIds->bindValue(':lim', $toDelete, PDO::PARAM_INT);
    $stmtIds->execute();
    $ids = $stmtIds->fetchAll(PDO::FETCH_COLUMN);
    if ($ids) {
        $placeholders = implode(',', array_fill(0, count($ids), '?'));
        $pdo->prepare("DELETE FROM class_schedule WHERE id IN ($placeholders)")
        ->execute($ids);
    }
}

// Validar que el nuevo total no supere la capacidad semanal
$validation = $this->courseModel->canAssignHours($courseId, $subjectId, $hours);
if (!$validation['allowed']) {
    $maxDisp = $validation['available'];
    $yaUsadas = $validation['currentTotal'] - ($validation['currentTotal'] - $maxDisp + ($hours -
($validation['remaining'] - 0)));
    $msg = "No se pueden asignar {$hours} horas: el total sería {$validation['newTotal']} pero el límite
semanal es {$validation['available']} horas. "
    . "Horas disponibles para esta asignatura: {$validation['remaining']}.";
    header('Location: ?action=academic&open_subjects=' . $courseId . '&error=hours_exceeded&msg=' .
urlencode($msg));
    exit;
}

// Actualizar horas en course_subjects

$stmtCount->prepare("UPDATE course_subjects SET hours_per_week = :h WHERE course_id = :cid AND subject_id = :sid")
->execute([':h' => $hours, ':cid' => $courseId, ':sid' => $subjectId]);

header('Location: ?action=academic&open_subjects=' . $courseId . '&hours_updated=1');
exit;
}
}

```

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

```

controllers/AttendanceController.php

```

<?php
// ARCHIVO COMPLETO - Reemplaza TODO el archivo
// controllers/AttendanceController.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';
require_once BASE_PATH . '/models/Notification.php';

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

    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);
        $this->notificationModel = new Notification($db);
    }
}

```

```

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

    if ($targetDate->format('Y-m-d') === $today->format('Y-m-d')) {
        return true;
    }
}

```

```

if ($targetDate > $today) {
    return false;
}

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

while ($current < $today) {
    $current->modify('+1 day');
    $dayOfWeek = (int)$current->format('N');
    if ($dayOfWeek < 6) {
        $businessHours += 24;
    }
}

return $businessHours <= EDIT_ATTENDANCE_HOURS;
}

```

```

private function getMaxEditDate() {
    $today = new DateTime();
    $businessHoursNeeded = EDIT_ATTENDANCE_HOURS;
    $current = clone $today;

```

```

while ($businessHoursNeeded > 0) {
    $current->modify('+1 day');
    $dayOfWeek = (int)$current->format('N');
    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 incluyendo nombre de asignatura
$db = new Database();
$pdo = $db->connect();
$sql = "SELECT cs.*, s.name as subject_name
FROM class_schedule cs
INNER JOIN subjects s ON cs.subject_id = s.id

```

```

WHERE cs.id = :id";
$stmt = $pdo->prepare($sql);
$stmt->execute([':id' => $scheduleId]);
$scheduleData = $stmt->fetch();

$subjectName = $scheduleData['subject_name'] ?? 'clase';
$dateFormatted = date('d/m/Y', strtotime($date));

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

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

        // Notificar al estudiante si fue marcado ausente
        // El enlace lleva directo a su asistencia para que pueda justificar
        if ($status === 'ausente') {
            $this->notificationModel->create(
                $studentId,
                '■ Ausencia registrada',
                "Se registr  una ausencia el {$dateFormatted} en {$subjectName}. Puedes justificarla.",
                'ausente',
                '?action=my_attendance'
            );
        }

        // Notificar al estudiante si fue marcado con tardanza
        if ($status === 'tardanza') {
            $this->notificationModel->create(
                $studentId,
                '■ Tardanza registrada',
                "Se registr  una tardanza el {$dateFormatted} en {$subjectName}.",
                'warning',
                '?action=my_attendance'
            );
        }
    }
}

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();
$maxEditDate = $this->getMaxEditDate();

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 = $date->format('N');
        if ($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];
    $prevMonth = date('Y-m', strtotime($month . '-01 -1 month'));
    $nextMonth = date('Y-m', strtotime($month . '-01 +1 month'));
    $firstDay = date('w', strtotime($year . '-' . $monthNum . '-01'));
    $daysInMonth = date('t', strtotime($year . '-' . $monthNum . '-01'));

    $calendarDays = [];

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

    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';
    }
}

```

```

$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;
}

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;
}

public function getExistingAttendance() {
header('Content-Type: application/json');

if (!isset($_SESSION['user_id'])) {
echo json_encode([]);
exit;
}

$scheduleId = (int)($_GET['schedule_id'] ?? 0);
$date = $_GET['date'] ?? date('Y-m-d');

if (!$scheduleId) {
echo json_encode([]);
exit;
}

$db = new Database();
$pdo = $db->connect();
$sql = "SELECT * FROM class_schedule WHERE id = :id";
$stmt = $pdo->prepare($sql);
$stmt->execute([':id' => $scheduleId]);
$schedule = $stmt->fetch();

if (!$schedule) {
echo json_encode([]);
exit;
}

$sql2 = "SELECT student_id, status, observation
FROM attendances
WHERE course_id = :course_id
AND subject_id = :subject_id
AND date = :date
AND hour_period = :hour_period";
$stmt2 = $pdo->prepare($sql2);
$stmt2->execute([
':course_id' => $schedule['course_id'],
':subject_id' => $schedule['subject_id'],
':date' => $date,
':hour_period' => $schedule['period_number'] . 'ra hora'
]);

echo json_encode($stmt2->fetchAll(PDO::FETCH_ASSOC));
exit;
}
}

```

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 . '/?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']);
                $_SESSION['is_superadmin'] = !empty($user['is_superadmin']);

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

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

    public function logout() {
        session_destroy();
        header('Location: ' . BASE_URL . '/?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);

            $resetLink = BASE_URL . "?action=reset&token={$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 . '/?action=login&reset=1');
            exit;
        }
    }

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

```

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;
}
}

```

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']);
        }
    }
}

```

```
$stats['tutor_course'] = $this->getTutorCourse($_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'];
}

private function getTutorCourse($teacherId) {
    $sql = "SELECT c.name as course_name
FROM teacher_assignments ta
INNER JOIN courses c ON ta.course_id = c.id
INNER JOIN school_years sy ON ta.school_year_id = sy.id
WHERE ta.teacher_id = :teacher_id
AND ta.is_tutor = 1
AND sy.is_active = 1
LIMIT 1";

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

    return $row ? $row['course_name'] : null;
}
}

```

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;
}
}

```

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);
        $allShifts = $this->shiftModel->getAll();
        $assignedShiftIds = $this->institutionShiftModel->getInstitutionShiftIds(1);

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

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

        $current = $this->institutionModel->getById(1);
        // Mantener logo actual - triple respaldo: BD → hidden → null
        $logoPath = !empty($current['logo_path'])
            ? $current['logo_path']
            : (!empty($_POST['current_logo_path']) ? $_POST['current_logo_path'] : null);

        // Solo procesar si realmente viene un archivo
        $hasFile = isset($_FILES['logo'])
            && is_array($_FILES['logo'])
            && $_FILES['logo']['error'] === UPLOAD_ERR_OK
            && $_FILES['logo']['size'] > 0;

        if ($hasFile) {
            $uploadDir = BASE_PATH . '/uploads/institution/';

            // Crear carpeta si no existe (permisos 0775 para hosting compartido)
            if (!is_dir($uploadDir)) {

```

```

mkdir($uploadDir, 0775, true);
}

$ext = strtolower(pathinfo($_FILES['logo']['name'], PATHINFO_EXTENSION));
$allowed = ['jpg', 'jpeg', 'png', 'gif', 'webp'];
$maxSize = 2 * 1024 * 1024;

if (in_array($ext, $allowed) && $_FILES['logo']['size'] <= $maxSize) {
    $filename = 'logo_' . $_SESSION['institution_id'] . '_' . time() . '.' . $ext;
    $uploadPath = $uploadDir . $filename;

    if (move_uploaded_file($_FILES['logo']['tmp_name'], $uploadPath)) {
        // Eliminar logo anterior solo si existe físicamente
        if (!empty($current['logo_path'])) {
            $oldFile = BASE_PATH . '/' . ltrim($current['logo_path'], '/');
            if (file_exists($oldFile)) {
                @unlink($oldFile);
            }
        }
        $logoPath = 'uploads/institution/' . $filename;
    }
    // Si move_uploaded_file falla: $logoPath mantiene el valor anterior
}
}

```

```

$data = [
    'name' => Security::sanitize($_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' => $logoPath,
    'working_days_list' => json_encode(array_values($_POST['working_days'] ??
['lunes', 'martes', 'miercoles', 'jueves', 'viernes'])),
];

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

// Toggle: si está asignada la quita, si no la asigna - responde JSON para AJAX
public function toggleShift() {
    header('Content-Type: application/json');

    if ($_SERVER['REQUEST_METHOD'] !== 'POST') {
        echo json_encode(['error' => 'Método no permitido']); exit;
    }

    $shiftId = (int)$_POST['shift_id'];
    $assignedShiftIds = $this->institutionShiftModel->getInstitutionShiftIds(1);

    if (in_array($shiftId, $assignedShiftIds)) {
        $this->institutionShiftModel->remove(1, $shiftId);
        echo json_encode(['action' => 'removed', 'shift_id' => $shiftId]);
    } else {
        $this->institutionShiftModel->assign([
            ':institution_id' => 1,
            ':shift_id' => $shiftId,
        ]);
        echo json_encode(['action' => 'assigned', 'shift_id' => $shiftId]);
    }
    exit;
}

```

```

// Mantener compatibilidad con rutas antiguas
public function assignShift() {

```

```

if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    $this->institutionShiftModel->assign([
        ':institution_id' => 1,
        ':shift_id' => (int)$_POST['shift_id'],
    ]);
    header('Location: ?action=institution&success=1');
    exit;
}

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

```

controllers/JustificationController.php

```

<?php
// ARCHIVO COMPLETO - Reemplaza TODO el archivo
// controllers/JustificationController.php

require_once BASE_PATH . '/config/config.php';
require_once BASE_PATH . '/models/Justification.php';
require_once BASE_PATH . '/models/Attendance.php';
require_once BASE_PATH . '/models/Attendance.php';
require_once BASE_PATH . '/models/Notification.php';

class JustificationController {
    private $justificationModel;
    private $attendanceModel;
    private $notificationModel;

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

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

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

        $studentId = Security::hasRole('estudiante')
            ? $_SESSION['user_id']
            : (isset($_GET['student_id']) ? (int)$_GET['student_id'] : null);

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

            $studentId = Security::hasRole('estudiante')
                ? $_SESSION['user_id']
                : (int)$_POST['student_id'];

            // Ausencias seleccionadas

            $attendanceIds = $_POST['attendance_ids'] ?? [];
            if (empty($attendanceIds)) {
                header('Location: ?action=submit_justification&error=no_absences');
                exit;
            }

            // Obtener fechas de las ausencias seleccionadas para calcular días laborables

```



```

$db = new Database();
$pdo = $db->connect();
$in = implode(',', array_map('intval', $attendanceIds));
$rows = $pdo->query(
    "SELECT date FROM attendances WHERE id IN ($in) AND student_id = $studentId ORDER BY date"
)->fetchAll();

if (empty($rows)) {
    header('Location: ?action=submit_justification&error=no_absences');
    exit;
}

$dates = array_column($rows, 'date');
$dateFrom = min($dates);
$dateTo = max($dates);
$workingDays = count($attendanceIds); // cada ausencia = 1 día (pueden ser no consecutivos)
$canApprove = Justification::resolveApprover($workingDays);

// Subir documento
$documentPath = null;
if (isset($_FILES['document']) && $_FILES['document']['error'] == 0) {
    $uploadDir = BASE_PATH . '/uploads/justifications/';
    if (!file_exists($uploadDir)) mkdir($uploadDir, 0755, true);
    $ext = strtolower(pathinfo($_FILES['document']['name'], PATHINFO_EXTENSION));
    $allowed = ['pdf', 'jpg', 'jpeg', 'png'];
    if (in_array($ext, $allowed) && $_FILES['document']['size'] <= 5 * 1024 * 1024) {
        $filename = uniqid() . '.' . $ext;
        if (move_uploaded_file($_FILES['document']['tmp_name'], $uploadDir . $filename)) {
            $documentPath = 'uploads/justifications/' . $filename;
        }
    }
}

```

```

// Motivo
$reasonType = Security::sanitize($_POST['reason_type'] ?? '');
$reasonText = Security::sanitize($_POST['reason'] ?? '');
$reasonFinal = ($reasonType === 'Otro')
    ? $reasonText
    : ($reasonType . ($reasonText ? ': ' . $reasonText : ''));

$data = [
    ':student_id' => $studentId,
    ':submitted_by' => $_SESSION['user_id'],
    ':date_from' => $dateFrom,
    ':date_to' => $dateTo,
    ':working_days' => $workingDays,
    ':reason_type' => $reasonType,
    ':reason' => $reasonFinal,
    ':document_path' => $documentPath,
    ':can_approve' => $canApprove,
];

$this->justificationModel->createForAttendances($attendanceIds, $data);

// Notificar
if ($canApprove === 'tutor') {
    $stmt = $pdo->prepare(
        "SELECT ta.teacher_id FROM teacher_assignments ta
        INNER JOIN course_students cs ON ta.course_id = cs.course_id
        WHERE cs.student_id = :sid AND ta.is_tutor = 1 LIMIT 1"
    );
    $stmt->execute([':sid' => $studentId]);
    $tutorId = $stmt->fetchColumn();
    if ($tutorId) {
        $this->notificationModel->create(
            $tutorId,
            '■ Justificación pendiente (tutor)',
            "Un estudiante de tu curso necesita justificación por $workingDays día(s).",
            'info',
            '?action=tutor_pending_justifications'
        );
    }
}

```

```

} else {

$this->_notifyReviewers(
    '■ Nueva justificación pendiente',
    "Justificación de $workingDays día(s) requiere revisión.",
    'info',
    '?action=pending_justifications'
);
}

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

// GET: cargar ausencias no justificadas
$absences = $this->attendanceModel->getUnjustifiedAbsences($studentId ?? $_SESSION['user_id']);

// Si representante, cargar sus hijos
$myChildren = [];
if (Security::hasRole('representante')) {
    $db = new Database();
    $stmt = $db->connect()->prepare(
        "SELECT u.id, CONCAT(u.last_name, ' ', u.first_name) as full_name
        FROM users u INNER JOIN representatives r ON u.id = r.student_id
        WHERE r.representative_id = :rid ORDER BY u.last_name"
    );
    $stmt->execute([':rid' => $_SESSION['user_id']]);
    $myChildren = $stmt->fetchAll();
}

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

public function pendingForTutor() {
    if (!Security::hasRole('docente')) die('Acceso denegado');

    $db = new Database();
    $attModel = new Attendance($db);
    $courseId = $attModel->getTutorCourseId($_SESSION['user_id']);

    if (!$courseId) {
header('Location: ?action=dashboard&error=not_tutor'); exit;
    }

    $justifications = $this->justificationModel->getPendingForTutor($courseId);
    include BASE_PATH . '/views/justifications/pending.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'] ?? '');

    // Obtener datos ANTES de actualizar para poder notificar
    $justification = $this->justificationModel->getById($justificationId);

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

```

```

if ($justification) {
    // Notificar al estudiante - enlace a sus justificaciones
    $this->notificationModel->create(
        $justification['student_id'],
        '■ Justificación aprobada',
        'Tu justificación fue aprobada.' . ($notes ? " Nota: $notes" : ''),
        'success',
        '?action=my_justifications'
    );
    // Notificar al representante si fue distinto del estudiante
    if ($justification['submitted_by'] != $justification['student_id']) {
        $this->notificationModel->create(
            $justification['submitted_by'],
            '■ Justificación aprobada',
            'La justificación que enviaste fue aprobada.' . ($notes ? " Nota: $notes" : ''),
            'success',
            '?action=my_justifications'
        );
    }
} else {
    $this->justificationModel->reject($justificationId, $_SESSION['user_id'], $notes);

    if ($justification) {
        // Notificar al estudiante - puede enviar una nueva justificación
        $this->notificationModel->create(
            $justification['student_id'],
            '■ Justificación rechazada',
            'Tu justificación fue rechazada.' . ($notes ? " Motivo: $notes" : ''),
            'danger',
            '?action=my_justifications'
        );
        if ($justification['submitted_by'] != $justification['student_id']) {
            $this->notificationModel->create(
                $justification['submitted_by'],
                '■ Justificación rechazada',
                'La justificación que enviaste fue rechazada.' . ($notes ? " Motivo: $notes" : ''),
                'danger',
                '?action=my_justifications'
            );
        }
    }
}

```

```

}
}
}

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

public function reviewed() {
    if (!Security::hasRole(['autoridad', 'inspector'])) {
        die('Acceso denegado');
    }
    $filter = $_GET['filter'] ?? 'all';
    $justifications = $this->justificationModel->getReviewed();

    if ($filter !== 'all') {
        $justifications = array_values(array_filter($justifications, function($j) use ($filter) {
            return $j['status'] === $filter;
        }));
    }
}

```

```

}

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

// Privado: notificar a todos los revisores
private function _notifyReviewers($title, $message, $type = 'info', $link = null) {
$db = new Database();
$pdo = $db->connect();

$sql = "SELECT DISTINCT u.id 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 IN ('autoridad','inspector') AND u.id != 'sender'";
$stmt = $pdo->prepare($sql);
$stmt->execute([':sender' => $_SESSION['user_id']]);
$reviewers = $stmt->fetchAll(PDO::FETCH_COLUMN);

foreach ($reviewers as $reviewerId) {
$this->notificationModel->create($reviewerId, $title, $message, $type, $link);
}
}
}

```

controllers/NotificationController.php

```

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

class NotificationController {
    private $notificationModel;

    public function __construct() {
        Security::requireLogin();
        $db = new Database();
        $this->notificationModel = new Notification($db);
    }

    // Vista principal: todas las notificaciones del usuario
    public function index() {
        $userId = $_SESSION['user_id'];
        $perPage = 15;
        $page = max(1, (int)($_GET['page'] ?? 1));
        $offset = ($page - 1) * $perPage;

        $notifications = $this->notificationModel->getByUser($userId, $perPage, $offset);
        $total = $this->notificationModel->countByUser($userId);
        $totalPages = (int)ceil($total / $perPage);
        $unreadCount = $this->notificationModel->getUnreadCount($userId);

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

    // AJAX: marcar una como leída y devolver nuevo conteo
    public function markRead() {
        $userId = $_SESSION['user_id'];
        $id = (int)($_GET['id'] ?? 0);

        if ($id) {
            $this->notificationModel->markAsRead($id, $userId);
        }

        // Responder JSON para el dropdown del navbar
        header('Content-Type: application/json');
        echo json_encode([

```

```

'success' => true,
'unread' => $this->notificationModel->getUnreadCount($userId)
]);
exit;
}

// Marcar todas como leídas (POST desde vista)
public function markAllRead() {
$userId = $_SESSION['user_id'];
$this->notificationModel->markAllAsRead($userId);
header('Location: ?action=notifications&success=read');
exit;
}

// Eliminar una notificación (POST desde vista)
public function delete() {
$userId = $_SESSION['user_id'];
$id = (int)($_POST['notification_id'] ?? 0);

if ($id) {
$this->notificationModel->delete($id, $userId);
}

header('Location: ?action=notifications&success=deleted');
exit;
}

// Eliminar todas las leídas
public function deleteRead() {
$userId = $_SESSION['user_id'];
$this->notificationModel->deleteRead($userId);
header('Location: ?action=notifications&success=cleaned');
exit;
}

// AJAX: obtener no leídas para el dropdown del navbar
public function getUnread() {
$userId = $_SESSION['user_id'];
$notifications = $this->notificationModel->getUnread($userId, 8);
$count = $this->notificationModel->getUnreadCount($userId);

header('Content-Type: application/json');
echo json_encode([
'notifications' => $notifications,
'count' => $count
]);
exit;
}
}

```

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';
}
}

```

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="16.66666%">Fecha</th>
<th width="16.66666%">Estudiante</th>
<th width="16.66666%">Asignatura</th>
<th width="16.66666%">Hora</th>
<th width="16.66666%">Estado</th>
<th width="16.66666%">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);

// Bordas
$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;
}

```

```

<?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);

// Bordas
$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';
    }
}
}

```

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');
        $errorMsg = null;

        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;

$result = $this->representativeModel->assignStudent($repId, $studentId, $relationship, $isPrimary);

if ($result === true) {
    header('Location: ?action=manage_representatives&success=1');
    exit;
}

```

```

} else {
    // Devolver error sin redirigir para mostrar mensaje
    $errorMsg = $result['error'];
}

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

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

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

    if ($repId && $studentId) {
        $this->representativeModel->togglePrimary($repId, $studentId);
    }

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

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);

$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';

```

[illegible]

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;
    private $db; // <-- AÑADE ESTA LÍNEA (propiedad para la conexión)

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

        $this->db = new Database(); // <-- GUARDAR LA CONEXIÓN EN LA PROPIEDAD
        $this->scheduleModel = new ClassSchedule($this->db);
        $this->courseModel = new Course($this->db);
        $this->subjectModel = new Subject($this->db);
        $this->userModel = new User($this->db);
        $this->schoolYearModel = new SchoolYear($this->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=schedule');
        }
    }
}

exit;
}

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

    // Verificar si la materia tiene un docente asignado en teacher_assignments
    $teacherCheck = $this->db->connect()->prepare(
        'SELECT teacher_id FROM teacher_assignments
        WHERE course_id = :course_id
        AND subject_id = :subject_id
        AND school_year_id = :school_year_id
        AND is_tutor = 0
        LIMIT 1'
    );
    $teacherCheck->execute([
        'course_id' => $courseId,
        'subject_id' => (int)$_POST['subject_id'],
        'school_year_id' => $activeYear['id']
    ]);
    $teacherData = $teacherCheck->fetch();

    // Priorizar: 1. POST teacher_id, 2. teacher_assignments, 3. null
    $teacherId = !empty($_POST['teacher_id']) ? (int)$_POST['teacher_id'] :
        ($teacherData ? $teacherData['teacher_id'] : null);

    $data = [
        'course_id' => $courseId,
        'subject_id' => (int)$_POST['subject_id'],
        'teacher_id' => $teacherId,
        '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 del curso (ahora incluye TODAS, incluso sin docente)
$sql = 'SELECT DISTINCT
s.id as subject_id,
s.name as subject_name,
ta.teacher_id,
COALESCE(CONCAT(u.last_name, " ", u.first_name), "Sin docente") as teacher_name,
COALESCE(cs.hours_per_week, 1) as hours_per_week
FROM course_subjects cs
INNER JOIN subjects s ON cs.subject_id = s.id
LEFT JOIN teacher_assignments ta ON ta.subject_id = s.id
AND ta.course_id = :course_id
AND ta.school_year_id = (SELECT id FROM school_years WHERE is_active = 1 LIMIT 1)
AND ta.is_tutor = 0
LEFT JOIN users u ON u.id = ta.teacher_id
WHERE cs.course_id = :course_id2
ORDER BY s.name';

```

```

$stmt = $this->db->connect()->prepare($sql); // <-- AHORA SÍ FUNCIONA PORQUE $this->db EXISTE
$stmt->execute(['course_id' => $courseId, 'course_id2' => $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;
}
```

```
$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
LEFT 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 = $this->db->connect()->prepare($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;
}
```

```
public function swapClass() {
```

```
header('Content-Type: application/json');
if ($_SERVER['REQUEST_METHOD'] !== 'POST') { echo json_encode(['success'=>false]); exit; }
```

```
$sidA = (int)($_POST['sid_a'] ?? 0);
$sidB = (int)($_POST['sid_b'] ?? 0);
$courseId = (int)($_POST['course_id'] ?? 0);
```

```
if (!$sidA || !$sidB || !$courseId) { echo json_encode(['success'=>false, 'message'=>'Datos incompletos']);
exit; }
```

```
try {
    $pdo = (new Database())->connect();
    $stmt = $pdo->prepare("SELECT id, day_of_week, period_number FROM class_schedule WHERE id IN (:a,:b) AND
course_id = :cid");
    $stmt->execute([':a'=>$sidA, ':b'=>$sidB, ':cid'=>$courseId]);
    $rows = $stmt->fetchAll(PDO::FETCH_ASSOC);
```

```

if (count($rows) !== 2) { echo json_encode(['success'=>false,'message'=>'Registros no encontrados']); exit;
}

$a = $rows[0]['id'] == $sidA ? $rows[0] : $rows[1];
$b = $rows[0]['id'] == $sidB ? $rows[0] : $rows[1];

$upd = $pdo->prepare("UPDATE class_schedule SET day_of_week=:day, period_number=:per WHERE id=:id");
$upd->execute([':day'=>$b['day_of_week'],':per'=>$b['period_number'],':id'=>$sidA]);
$upd->execute([':day'=>$a['day_of_week'],':per'=>$a['period_number'],':id'=>$sidB]);

echo json_encode(['success'=>true]);
} catch (Exception $e) {
echo json_encode(['success'=>false,'message'=>$e->getMessage()]);
}
exit;
}

// Mover una clase a una celda vacía
public function moveClass() {
header('Content-Type: application/json');
if ($_SERVER['REQUEST_METHOD'] !== 'POST') { echo json_encode(['success'=>false]); exit; }

$sid = (int)($_POST['schedule_id'] ?? 0);
$day = $_POST['day_of_week'] ?? '';
$period = (int)($_POST['period_number'] ?? 0);
$courseId = (int)($_POST['course_id'] ?? 0);

if (!$sid || !$day || !$period || !$courseId) { echo json_encode(['success'=>false,'message'=>'Datos
incompletos']); exit; }

try {
$pdo = (new Database())->connect();

// Verificar que la celda destino esté vacía
$check = $pdo->prepare("SELECT id FROM class_schedule WHERE course_id=:cid AND day_of_week=:day AND
period_number=:per");
$check->execute([':cid'=>$courseId,':day'=>$day,':per'=>$period]);
if ($check->fetch()) { echo json_encode(['success'=>false,'message'=>'La celda destino ya está ocupada']);
exit; }

// Verificar que la escuela activa coincida
$upd = $pdo->prepare("UPDATE class_schedule SET day_of_week=:day, period_number=:per WHERE id=:id AND
course_id=:cid");
$upd->execute([':day'=>$day,':per'=>$period,':id'=>$sid,':cid'=>$courseId]);

echo json_encode(['success'=>true]);
} catch (Exception $e) {
echo json_encode(['success'=>false,'message'=>$e->getMessage()]);
}
exit;
}
}

```

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';
}
}

```

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();
}
}

```

controllers/TutorController.php

```

<?php
// ARCHIVO COMPLETO - Reemplaza TODO el archivo
// controllers/TutorController.php

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

class TutorController {
    private $attendanceModel;
    private $courseModel;

    public function __construct() {
        Security::requireLogin();
        if (!Security::hasRole('docente')) {
            die('Acceso denegado');
        }
        $db = new Database();
        $this->attendanceModel = new Attendance($db);
        $this->courseModel = new Course($db);
    }

    public function courseAttendance() {
        $teacherId = $_SESSION['user_id'];
        $courseId = $this->attendanceModel->getTutorCourseId($teacherId);

        if (!$courseId) {
            include BASE_PATH . '/views/tutor/no_tutor.php';
        }
    }
}

```

```

return;
}

$course = $this->courseModel->findById($courseId);
$subjects = $this->attendanceModel->getSubjectsByCourse($courseId);
$students = $this->attendanceModel->getStudentsByCourse($courseId);
$stopAbsences = $this->attendanceModel->getTutorTopAbsences($courseId, 5);

// Leer filtros del GET
$filters = [];
if (!empty($_GET['subject_id'])) $filters['subject_id'] = (int)$_GET['subject_id'];
if (!empty($_GET['student_id'])) $filters['student_id'] = (int)$_GET['student_id'];
if (!empty($_GET['status'])) $filters['status'] = $_GET['status'];
if (!empty($_GET['start_date'])) $filters['start_date'] = $_GET['start_date'];
if (!empty($_GET['end_date'])) $filters['end_date'] = $_GET['end_date'];

$attendances = $this->attendanceModel->getTutorCourseAttendance($courseId, $filters);
$stats = $this->attendanceModel->getTutorCourseStats($courseId, $filters);

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

// Endpoint AJAX
public function ajax() {
    header('Content-Type: application/json');

    $teacherId = $_SESSION['user_id'];
    $courseId = $this->attendanceModel->getTutorCourseId($teacherId);

    if (!$courseId) {
        echo json_encode(['error' => 'No es tutor']);
        exit;
    }

    $filters = [];
    if (!empty($_GET['subject_id'])) $filters['subject_id'] = (int)$_GET['subject_id'];
    if (!empty($_GET['student_id'])) $filters['student_id'] = (int)$_GET['student_id'];
    if (!empty($_GET['status'])) $filters['status'] = $_GET['status'];
    if (!empty($_GET['start_date'])) $filters['start_date'] = $_GET['start_date'];
    if (!empty($_GET['end_date'])) $filters['end_date'] = $_GET['end_date'];

    $attendances = $this->attendanceModel->getTutorCourseAttendance($courseId, $filters);
    $stats = $this->attendanceModel->getTutorCourseStats($courseId, $filters);

    echo json_encode([
        'stats' => $stats,
        'attendances' => $attendances,
        'total' => count($attendances),
    ]);
    exit;
}

```

```
// Dashboard del tutor
public function dashboard() {
    $teacherId = $_SESSION['user_id'];
    $courseId = $this->attendanceModel->getTutorCourseId($teacherId);

    if (!$courseId) {
        include BASE_PATH . '/views/tutor/no_tutor.php';
        return;
    }

    $course = $this->courseModel->findById($courseId);
    $students = $this->attendanceModel->getStudentsByCourse($courseId);
    $topAbsences = $this->attendanceModel->getTutorTopAbsences($courseId, 10);
    $stats = $this->attendanceModel->getTutorCourseStats($courseId);

    // Asistencia de hoy
    $db = new Database();
    $pdo = $db->connect();
    $stmt = $pdo->prepare(
```



```

$stmt = $pdo->prepare($sql);
$stmt->execute($params);
$students = $stmt->fetchAll();

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

```

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";
        }

        // Validar cédula ecuatoriana (si no es extranjero y hay valor)
        $esExtranjero = !empty($_POST['es_extranjero']);
        $dniValue = null;
        if (!$esExtranjero && !empty($_POST['dni'])) {
            $ced = preg_replace('/\D/', '', $_POST['dni']);
            if (!$this->validarCedulaEcuador($ced)) {
                $errors[] = "Cédula ecuatoriana inválida";
            } else {
                $dniValue = $ced;
            }
        } elseif ($esExtranjero && !empty($_POST['passport'])) {
            $dniValue = Security::sanitize($_POST['passport']);
        }

        // Validar teléfono Ecuador
        $phoneValue = null;
        if (!empty($_POST['phone'])) {
            $tel = preg_replace('/[\s-]/', '', $_POST['phone']);

```

```

        if (!preg_match('/^09\d{8}$/', $tel) && !preg_match('/^0[2-7]\d{7}$/', $tel)) {
            $errors[] = "Teléfono inválido. Celular: 09XXXXXXXXX · Fijo: 0XXXXXXXXX";
        } else {
            $phoneValue = $tel;
        }
    }
}

```

```

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' => $dniValue,
    'phone' => $phoneValue,
];

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');
        }
    }
}
```


[illegible]


```

$_SESSION['modal_edit_errors'] = $errors;
$_SESSION['modal_edit_data'] = array_merge($_POST, ['user_id' => $userId]);
$_SESSION['open_modal'] = 'edit';
header('Location: ?action=users'); exit;
}

$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' => $dniValue,
'phone' => !empty($_POST['phone']) ? Security::sanitize($_POST['phone']) : null,
];
if (!empty($_POST['password'])) $updateData['password'] = $_POST['password'];

if ($this->userModel->update($updateData)) {
$warnParam = ($dniResult['warning'] === 'cedula_invalida') ? '&dni_warn=1' : '';
header('Location: ?action=users&updated=1' . $warnParam); exit;
}

$_SESSION['modal_edit_errors'] = ['Error al actualizar el usuario'];
$_SESSION['modal_edit_data'] = array_merge($_POST, ['user_id' => $userId]);
$_SESSION['open_modal'] = 'edit';
header('Location: ?action=users'); exit;
}
}

```

MODELS

models/Attendance.php

```
<?php
// ARCHIVO COMPLETO - Reemplaza TODO el archivo
// models/Attendance.php

class Attendance {
    private $db;

    public function __construct($database) {
        $this->db = $database->connect();
    }

    public function create($data) {
        $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) {
            $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']
            ]);
        }

        $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;
        return (time() - strtotime($record['created_at'])) <= (EDIT_ATTENDANCE_HOURS * 3600);
    }

    public function update($id, $data) {
        $sql = "UPDATE attendances
        SET status = :status, observation = :observation, updated_at = NOW()
        WHERE id = :id";
        $stmt = $this->db->prepare($sql);
```

```

return $stmt->execute([':id' => $id, ':status' => $data['status'], ':observation' => $data['observation']]);
}

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();
}

```

```

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

```



```

public function getTutorCourseAttendance($courseId, $filters = []) {
    $sql = "SELECT
    a.id, a.date, a.hour_period, a.status, a.observation,
    CONCAT(u.last_name, ' ', u.first_name) as student_name,

u.id as student_id,
s.name as subject_name,
s.id as subject_id,
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 subjects s ON a.subject_id = s.id
INNER JOIN users t ON a.teacher_id = t.id
WHERE a.course_id = :course_id";
    $params = [':course_id' => $courseId];

    if (!empty($filters['subject_id'])) { $sql .= " AND a.subject_id = :subject_id"; $params[':subject_id'] =
    $filters['subject_id']; }
    if (!empty($filters['student_id'])) { $sql .= " AND a.student_id = :student_id"; $params[':student_id'] =
    $filters['student_id']; }
    if (!empty($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, s.name";
    $stmt = $this->db->prepare($sql);
    $stmt->execute($params);
    return $stmt->fetchAll();
}

// Estadísticas globales del curso (vista del tutor)
public function getTutorCourseStats($courseId, $filters = []) {
    $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 a
    WHERE a.course_id = :course_id";
    $params = [':course_id' => $courseId];

    if (!empty($filters['subject_id'])) { $sql .= " AND a.subject_id = :subject_id"; $params[':subject_id'] =
    $filters['subject_id']; }
    if (!empty($filters['student_id'])) { $sql .= " AND a.student_id = :student_id"; $params[':student_id'] =
    $filters['student_id']; }
    if (!empty($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']; }

    $stmt = $this->db->prepare($sql);
    $stmt->execute($params);
    return $stmt->fetch();
}

// Top estudiantes con más ausencias (vista del tutor)
public function getTutorTopAbsences($courseId, $limit = 10) {
    $sql = "SELECT CONCAT(u.last_name, ' ', u.first_name) as student_name,
    COUNT(*) as total_ausencias
    FROM attendances a
    INNER JOIN users u ON a.student_id = u.id
    WHERE a.course_id = :course_id AND a.status = 'ausente'
    GROUP BY a.student_id, u.last_name, u.first_name
    ORDER BY total_ausencias DESC
    LIMIT :lim";
    $stmt = $this->db->prepare($sql);
    $stmt->bindValue(':course_id', $courseId, PDO::PARAM_INT);
    $stmt->bindValue(':lim', $limit, PDO::PARAM_INT);

```

```

$stmt->execute();
return $stmt->fetchAll();
}

// Ausencias sin justificación pendiente o aprobada (para el formulario de justificación)
public function getUnjustifiedAbsences($studentId) {
    $sql = "SELECT a.id, a.date, a.hour_period, a.status,
    s.name as subject_name,
    c.name as course_name
    FROM attendances a
    INNER JOIN subjects s ON a.subject_id = s.id
    INNER JOIN courses c ON a.course_id = c.id
    WHERE a.student_id = :student_id
    AND a.status = 'ausente'
    AND a.id NOT IN (
    SELECT attendance_id FROM justifications
    WHERE attendance_id IS NOT NULL
    AND status IN ('pendiente','aprobado')
    )
    ORDER BY a.date DESC, a.hour_period ASC";
    $stmt = $this->db->prepare($sql);
    $stmt->execute([':student_id' => $studentId]);

return $stmt->fetchAll();
}
}

```

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,
        COALESCE(CONCAT(u.last_name, " ", u.first_name), "Sin docente") as teacher_name
        FROM class_schedule cs
        INNER JOIN subjects s ON cs.subject_id = s.id
        LEFT 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'
            ];
        }

        // SOLUCIÓN: Si teacher_id está vacío, usar NULL
        $teacherId = !empty($data['teacher_id']) ? $data['teacher_id'] : null;

```

```

$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([
'course_id' => $data['course_id'],
'subject_id' => $data['subject_id'],
'teacher_id' => $teacherId,
'school_year_id' => $data['school_year_id'],
'day_of_week' => $data['day_of_week'],
'period_number' => $data['period_number']
]);

return [
'success' => $result,
'message' => 'Clase agregada correctamente'
];
}

public function getByCourse($courseId, $schoolYearId) {
$sql = 'SELECT cs.*,
s.name as subject_name,
-- Siempre obtener el docente actual de teacher_assignments
COALESCE(
(SELECT CONCAT(u2.last_name, " ", u2.first_name)
FROM teacher_assignments ta2
INNER JOIN users u2 ON ta2.teacher_id = u2.id
WHERE ta2.course_id = cs.course_id
AND ta2.subject_id = cs.subject_id
AND ta2.school_year_id = cs.school_year_id
AND ta2.is_tutor = 0
LIMIT 1),
"Sin docente"
) as teacher_name,
-- Traer también el ID del docente actual
(SELECT teacher_id
FROM teacher_assignments ta3
WHERE ta3.course_id = cs.course_id
AND ta3.subject_id = cs.subject_id
AND ta3.school_year_id = cs.school_year_id
AND ta3.is_tutor = 0
LIMIT 1) as current_teacher_id
FROM class_schedule cs
INNER JOIN subjects s ON cs.subject_id = s.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() {
    // Forzar zona horaria Ecuador
    $tz = new DateTimeZone('America/Guayaquil');
    $now = new DateTime('now', $tz);
    $dayNumber = (int)$now->format('N'); // 1=Lunes, 7=Domingo

    $days = [
        1 => 'lunes',
        2 => 'martes',
        3 => 'miercoles',
        4 => 'jueves',
        5 => 'viernes',
        6 => 'sabado'
    ];

    return $days[$dayNumber] ?? 'lunes';
}
}

```

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) {
// 1. Desmatricular estudiantes
$this->db->prepare("DELETE FROM course_students WHERE course_id = :id")->execute([':id' => $id]);
// 2. Eliminar asignaciones docentes (incluye tutor)
$this->db->prepare("DELETE FROM teacher_assignments WHERE course_id = :id")->execute([':id' => $id]);
// 3. Eliminar horarios
$this->db->prepare("DELETE FROM class_schedule WHERE course_id = :id")->execute([':id' => $id]);
// 4. Eliminar asignaturas del curso
$this->db->prepare("DELETE FROM course_subjects WHERE course_id = :id")->execute([':id' => $id]);
// 5. Eliminar curso
return $this->db->prepare("DELETE FROM courses WHERE id = :id")->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
]);
}

```

```

public function getAllWithTutor() {
$sql = "SELECT c.*, s.name as shift_name, sy.name as year_name,
u.first_name as tutor_first, u.last_name as tutor_last
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
LEFT JOIN teacher_assignments ta ON (ta.course_id = c.id AND ta.is_tutor = 1 AND ta.school_year_id =
c.school_year_id)
LEFT JOIN users u ON u.id = ta.teacher_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();
}

```

```

/**
 * Obtiene el máximo de horas por día según el tipo de curso
 */
public function getMaxHoursPerDay($courseId) {
$sql = "SELECT grade_level FROM courses WHERE id = :id";
$stmt = $this->db->prepare($sql);
$stmt->execute([':id' => $courseId]);
$course = $stmt->fetch();

if (!$course) return 7; // Valor por defecto

$gradeLevel = $course['grade_level'];

// Determinar si es técnico (BT) para 8 horas, si no 7 horas
if (strpos($gradeLevel, 'BT') !== false || strpos($gradeLevel, 'Técnico') !== false) {
return 8; // Bachillerato Técnico: 8 horas
}
}

```

```

return 7; // Educación regular: 7 horas
}

```

```

/**
 * Calcula las horas totales disponibles en la semana
 */
public function getTotalWeeklyHoursAvailable($courseId) {
$hoursPerDay = $this->getMaxHoursPerDay($courseId);
$workingDays = $this->getWorkingDaysCount(); // Necesitamos este método
return $hoursPerDay * $workingDays;
}

```

```

/**
 * Obtiene la cantidad de días laborables configurados
 */
public function getWorkingDaysCount() {
    // Por defecto 5 días (lunes a viernes)
    $defaultDays = 5;

    if (empty($_SESSION['institution_id'])) return $defaultDays;

    $sql = "SELECT working_days_list FROM institutions WHERE id = :id";
    $stmt = $this->db->prepare($sql);
    $stmt->execute(['id' => $_SESSION['institution_id']]);
    $institution = $stmt->fetch();

    if (empty($institution['working_days_list'])) return $defaultDays;

    $days = json_decode($institution['working_days_list'], true);
    return is_array($days) ? count($days) : $defaultDays;
}

/**
 * Calcula la suma total de horas asignadas a todas las materias del curso
 */
public function getTotalAssignedHours($courseId) {
    $sql = "SELECT SUM(cs.hours_per_week) as total
    FROM course_subjects cs
    WHERE cs.course_id = :course_id";

$stmt = $this->db->prepare($sql);
$stmt->execute(['course_id' => $courseId]);
$result = $stmt->fetch();

return (int)($result['total'] ?? 0);
}

/**
 * Verifica si se puede asignar una cantidad de horas a una materia
 */
public function canAssignHours($courseId, $subjectId, $newHours) {
    $totalAvailable = $this->getTotalWeeklyHoursAvailable($courseId);
    $currentTotal = $this->getTotalAssignedHours($courseId);

    // Obtener horas actuales de esta materia específica
    $sql = "SELECT hours_per_week FROM course_subjects
    WHERE course_id = :course_id AND subject_id = :subject_id";
    $stmt = $this->db->prepare($sql);
    $stmt->execute(['course_id' => $courseId, 'subject_id' => $subjectId]);
    $current = $stmt->fetch();

    $currentSubjectHours = $current ? (int)$current['hours_per_week'] : 0;

    // Calcular nuevo total restando lo que ya tenía esta materia
    $newTotal = ($currentTotal - $currentSubjectHours) + $newHours;

    return [
        'allowed' => $newTotal <= $totalAvailable,
        'currentTotal' => $currentTotal,
        'newTotal' => $newTotal,
        'available' => $totalAvailable,
        'remaining' => $totalAvailable - ($currentTotal - $currentSubjectHours)
    ];
}
}

```

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();
}
}

```

```

<?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);
    }
}

```

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;
}

public function getById($id) {
$sql = "SELECT j.*, j.student_id, j.submitted_by,
CONCAT(u.last_name, ' ', u.first_name) as student_name
FROM justifications j
INNER JOIN users u ON j.student_id = u.id
WHERE j.id = :id";
$stmt = $this->db->prepare($sql);
$stmt->execute([':id' => $id]);
return $stmt->fetch();
}

```

```
public function getReviewed($institutionId = null) {  
    $sql = "SELECT j.*,  
        CONCAT(u.last_name, ' ', u.first_name) as student_name,  
        CONCAT(s.last_name, ' ', s.first_name) as submitted_by_name,  
        CONCAT(r.last_name, ' ', r.first_name) as reviewer_name,  
        a.date as attendance_date,  
        a.hour_period,  
        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 users r ON j.reviewed_by = r.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 IN ('aprobado', 'rechazado')  
ORDER BY j.updated_at DESC";  
  
    $stmt = $this->db->query($sql);  
    return $stmt->fetchAll();  
}  
  
// ■■■ Crear justificación por rango de fechas ■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■  
public function createByRange($data) {  
    $sql = "INSERT INTO justifications  
        (student_id, submitted_by, date_from, date_to,  
         working_days, reason_type, reason, document_path, can_approve, status)  
VALUES  
(:student_id, :submitted_by, :date_from, :date_to,  
:working_days, :reason_type, :reason, :document_path, :can_approve, 'pendiente')";  
    $stmt = $this->db->prepare($sql);  
    return $stmt->execute($data);  
}  
  
// ■■■ Contar días laborables entre dos fechas (lun-vie) ■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■  
public static function countWorkingDays($from, $to) {  
    $count = 0;  
    $cur = strtotime($from);
```

[illegible][illegible]

```
$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
LEFT JOIN attendances a ON j.attendance_id = a.id
LEFT 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' AND $where
ORDER BY j.created_at DESC";
```

```
return $this->db->query($sql)->fetchAll();
}
```

```
public function getPendingForTutor($tutorCourseId) {
    $sql = "SELECT j.*,
    CONCAT(u.last_name, ' ', u.first_name) as student_name,
    CONCAT(s.last_name, ' ', s.first_name) as submitted_by_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 course_students cs ON u.id = cs.student_id
    INNER JOIN courses c ON cs.course_id = c.id
    WHERE j.status = 'pendiente'
    AND j.can_approve = 'tutor'
    AND cs.course_id = :course_id
    ORDER BY j.created_at DESC";
    $stmt = $this->db->prepare($sql);
    $stmt->execute([':course_id' => $tutorCourseId]);
    return $stmt->fetchAll();
}
```

[illegible]


```

$result = $stmt->execute([
    ':reviewed_by' => $reviewerId,
    ':notes' => $notes,
    ':id' => $justificationId,
]);

if ($result) {
    // Si tiene attendance_id único, actualizar también esa asistencia
    $j = $this->getById($justificationId);

    if (!empty($j['attendance_id'])) {
        $this->db->prepare(
            "UPDATE attendances SET status='justificado' WHERE id=:id"
        )->execute([':id' => $j['attendance_id']]);
    }
    // Si tiene rango de fechas, actualizar todas las ausencias del estudiante en ese rango
    if (!empty($j['date_from']) && !empty($j['date_to'])) {
        $this->db->prepare(
            "UPDATE attendances SET status='justificado'
            WHERE student_id=:sid AND date BETWEEN :df AND :dt AND status='ausente'"
        )->execute([
            ':sid' => $j['student_id'],
            ':df' => $j['date_from'],
            ':dt' => $j['date_to'],
        ]);
    }
}
return $result;
}

// Crear una justificación por cada asistencia seleccionada
public function createForAttendances($attendanceIds, $data) {
    $sql = "INSERT INTO justifications
    (attendance_id, student_id, submitted_by,
    date_from, date_to, working_days,
    reason_type, reason, document_path, can_approve, status)
    VALUES
    (:attendance_id, :student_id, :submitted_by,
    :date_from, :date_to, :working_days,
    :reason_type, :reason, :document_path, :can_approve, 'pendiente')";
    $stmt = $this->db->prepare($sql);
    $ok = true;
    foreach ($attendanceIds as $attId) {
        $row = $data;
        $row[':attendance_id'] = (int)$attId;
        if (!$stmt->execute($row)) $ok = false;
    }
    return $ok;
}

// Contar ausencias ya justificadas (pendiente o aprobada) para un estudiante en un rango
public function countPendingJustifications($studentId) {
    $sql = "SELECT COUNT(*) FROM justifications
    WHERE student_id = :sid AND status = 'pendiente'";
    $stmt = $this->db->prepare($sql);
    $stmt->execute([':sid' => $studentId]);
    return (int)$stmt->fetchColumn();
}
}

```

models/Notification.php

```

<?php
class Notification {
    private $db;

    public function __construct($database) {
        $this->db = $database->connect();
    }
}

```

```
// Crear notificación
// $link: URL relativa destino, ej: '?action=pending_justifications'
public function create($userId, $title, $message, $type = 'info', $link = null) {
    $sql = "INSERT INTO notifications (user_id, title, message, type, link, is_read, created_at)
VALUES (:user_id, :title, :message, :type, :link, 0, NOW())";
    $stmt = $this->db->prepare($sql);
    return $stmt->execute([
        ':user_id' => $userId,
        ':title' => $title,
        ':message' => $message,
        ':type' => $type,
        ':link' => $link
    ]);
}
```

```
// Obtener notificaciones de un usuario (con paginación)
public function getByUser($userId, $limit = 20, $offset = 0) {
    $sql = "SELECT * FROM notifications
WHERE user_id = :user_id
ORDER BY created_at DESC
LIMIT :limit OFFSET :offset";
    $stmt = $this->db->prepare($sql);
    $stmt->bindValue(':user_id', $userId, PDO::PARAM_INT);
    $stmt->bindValue(':limit', $limit, PDO::PARAM_INT);
    $stmt->bindValue(':offset', $offset, PDO::PARAM_INT);
    $stmt->execute();
    return $stmt->fetchAll();
}
```

```
// Solo no leídas (para el dropdown del navbar)
public function getUnread($userId, $limit = 10) {
    $sql = "SELECT * FROM notifications
```

```
WHERE user_id = :user_id AND is_read = 0
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();
}
```

```
// Contar no leídas
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' => $userId]);
    return (int)$stmt->fetch()['count'];
}
```

```
// Marcar una como leída
public function markAsRead($notificationId, $userId) {
    $sql = "UPDATE notifications SET is_read = 1
WHERE id = :id AND user_id = :user_id";
    $stmt = $this->db->prepare($sql);
    return $stmt->execute([':id' => $notificationId, ':user_id' => $userId]);
}
```

```
// Marcar TODAS como leídas
public function markAllAsRead($userId) {
    $sql = "UPDATE notifications SET is_read = 1
WHERE user_id = :user_id AND is_read = 0";
    $stmt = $this->db->prepare($sql);
    return $stmt->execute([':user_id' => $userId]);
}
```

```
// Eliminar una notificación
public function delete($notificationId, $userId) {
    $sql = "DELETE FROM notifications WHERE id = :id AND user_id = :user_id";
```

```

$stmt = $this->db->prepare($sql);
return $stmt->execute([':id' => $notificationId, ':user_id' => $userId]);
}

// Eliminar todas las leídas
public function deleteRead($userId) {
$sql = "DELETE FROM notifications WHERE user_id = :user_id AND is_read = 1";
$stmt = $this->db->prepare($sql);
return $stmt->execute([':user_id' => $userId]);
}

// Total de notificaciones del usuario (para paginación)
public function countByUser($userId) {
$sql = "SELECT COUNT(*) as count FROM notifications WHERE user_id = :user_id";
$stmt = $this->db->prepare($sql);
$stmt->execute([':user_id' => $userId]);
return (int)$stmt->fetch()['count'];
}
}

```

models/Representative.php

```

<?php
class Representative {
    private $db;

    public function __construct($database) {
        $this->db = $database->connect();
    }

    /**
     * Verifica si ya existe un representante con el mismo parentesco exclusivo (Padre/Madre)
     * para el estudiante dado. Excluye al propio representante si es una actualización.
     */
    public function hasExclusiveRelationship($studentId, $relationship, $excludeRepId = null) {
        $exclusivos = ['Padre', 'Madre'];
        if (!in_array($relationship, $exclusivos)) return false;

        $sql = "SELECT COUNT(*) FROM representatives
        WHERE student_id = :student_id
        AND relationship = :relationship";

        if ($excludeRepId) {
            $sql .= " AND representative_id != :exclude_rep";
        }

        $stmt = $this->db->prepare($sql);
        $params = [':student_id' => $studentId, ':relationship' => $relationship];
        if ($excludeRepId) $params[':exclude_rep'] = $excludeRepId;

        $stmt->execute($params);
        return $stmt->fetchColumn() > 0;
    }

    public function assignStudent($representativeId, $studentId, $relationship, $isPrimary = 0) {
        // Verificar duplicado de parentesco exclusivo
        // Se excluye al propio representante para permitir actualizar su propio parentesco
        if ($this->hasExclusiveRelationship($studentId, $relationship, $representativeId)) {
            return ['error' => "Este estudiante ya tiene un representante con parentesco \"{$relationship}\". Solo puede haber uno."];
        }

        // Verificar si ya existe la relación (para decidir INSERT o UPDATE)

        $stmtCheck = $this->db->prepare(
            "SELECT COUNT(*) FROM representatives
            WHERE representative_id = :rep_id AND student_id = :stu_id"
        );
    }
}

```

```

$stmtCheck->execute([':rep_id' => $representativeId, ':stu_id' => $studentId]);
$exists = $stmtCheck->fetchColumn() > 0;

if ($exists) {
    $sql = "UPDATE representatives SET relationship = :relationship, is_primary = :is_primary
    WHERE representative_id = :rep_id AND student_id = :stu_id";
    $stmt = $this->db->prepare($sql);
    $ok = $stmt->execute([
        ':relationship' => $relationship,
        ':is_primary' => $isPrimary,
        ':rep_id' => $representativeId,
        ':stu_id' => $studentId
    ]);
} else {
    $sql = "INSERT INTO representatives (representative_id, student_id, relationship, is_primary)
    VALUES (:rep_id, :stu_id, :relationship, :is_primary)";
    $stmt = $this->db->prepare($sql);
    $ok = $stmt->execute([
        ':rep_id' => $representativeId,
        ':stu_id' => $studentId,
        ':relationship' => $relationship,
        ':is_primary' => $isPrimary
    ]);
}

return $ok ? true : ['error' => 'Error al guardar la relación.'];
}

/**
 * Cambia el tipo (Principal <-> Secundario) de una relación específica.
 * Si se marca como Principal, el otro representante del mismo estudiante pasa a Secundario.
 */
public function togglePrimary($representativeId, $studentId) {
    // Obtener estado actual
    $stmt = $this->db->prepare(
        "SELECT is_primary FROM representatives
        WHERE representative_id = :rep_id AND student_id = :stu_id"
    );
    $stmt->execute([':rep_id' => $representativeId, ':stu_id' => $studentId]);
    $current = $stmt->fetchColumn();

    if ($current === false) return false;

    $newValue = $current ? 0 : 1;

    // Si se va a marcar como Principal, quitar Principal a los demás del mismo estudiante
    if ($newValue == 1) {
        $stmtReset = $this->db->prepare(
            "UPDATE representatives SET is_primary = 0
            WHERE student_id = :stu_id AND representative_id != :rep_id"
        );
        $stmtReset->execute([':stu_id' => $studentId, ':rep_id' => $representativeId]);
    }

    $stmtUpdate = $this->db->prepare(
        "UPDATE representatives SET is_primary = :val
        WHERE representative_id = :rep_id AND student_id = :stu_id"
    );
    return $stmtUpdate->execute([
        ':val' => $newValue,
        ':rep_id' => $representativeId,
        ':stu_id' => $studentId
    ]);
}

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
    ]);
}
}

```

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();
    }
}

```

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);
    $ok = $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']
    ]);
    // Retorna el ID insertado para poder activarlo en la misma conexión
    return $ok ? (int)$this->db->lastInsertId() : 0;
}

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;
}
}

```

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();
    }
}

```

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]);
}
}

```

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, 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.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 ($existing) {
            // SOLUCIÓN: Si ya existe, actualizar y también actualizar el horario
            if ($existing['teacher_id'] != $data['teacher_id']) {
                // Actualizar horario con el nuevo teacher_id
                $updateSchedule = $this->db->prepare(
                    'UPDATE class_schedule
SET teacher_id = :teacher_id
WHERE course_id = :course_id
AND subject_id = :subject_id
AND school_year_id = :school_year_id'
                );
                $updateSchedule->execute([
                    'teacher_id' => $data['teacher_id'],
                    'course_id' => $data['course_id'],
                    'subject_id' => $data['subject_id'],
                    'school_year_id' => $data['school_year_id']
                ]);
            }

            // Actualizar la asignación existente
            $sql = 'UPDATE teacher_assignments
SET teacher_id = :teacher_id
WHERE id = :id';
            $stmt = $this->db->prepare($sql);

```

```

$result = $stmt->execute([
    'teacher_id' => $data['teacher_id'],
    'id' => $existing['id']
]);

return [
    'success' => $result,
    'message' => 'Asignación actualizada correctamente'
];
}

// Si no existe, insertar nueva
$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, teacher_id, course_id, subject_id, school_year_id 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.'
        ];
    }

    // SOLUCIÓN: Antes de eliminar la asignación, actualizar el horario
    if ($assignment) {
        // Poner a NULL los teacher_id en class_schedule para esta materia y curso
        $updateSchedule = $this->db->prepare(
            'UPDATE class_schedule
            SET teacher_id = NULL
            WHERE course_id = :course_id
            AND subject_id = :subject_id
            AND school_year_id = :school_year_id'
        );
        $updateSchedule->execute([
            'course_id' => $assignment['course_id'],
            'subject_id' => $assignment['subject_id'],
            'school_year_id' => $assignment['school_year_id']
        ]);
    }

    $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. El horario se ha actualizado.'
    ];
}

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();
}
}

```

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) {
    // 1. Eliminar roles

```



```

$this->db->prepare("DELETE FROM user_roles WHERE user_id = :uid")
->execute([':uid' => $userId]);

// 2. Eliminar representaciones
$this->db->prepare("DELETE FROM representatives WHERE representative_id = :rid OR student_id = :sid")
->execute([':rid' => $userId, ':sid' => $userId]);

// 3. Eliminar matrículas
$this->db->prepare("DELETE FROM course_students WHERE student_id = :uid")
->execute([':uid' => $userId]);

// 4. Eliminar entradas del horario donde era el docente asignado
$this->db->prepare("DELETE FROM class_schedule WHERE teacher_id = :uid")
->execute([':uid' => $userId]);

// 5. Eliminar asignaciones docentes
$this->db->prepare("DELETE FROM teacher_assignments WHERE teacher_id = :uid")
->execute([':uid' => $userId]);

// 6. Eliminar asistencias registradas por o para este usuario
$this->db->prepare("DELETE FROM attendances WHERE student_id = :uid OR teacher_id = :uid2")
->execute([':uid' => $userId, ':uid2' => $userId]);

// 7. Eliminar usuario
$stmt = $this->db->prepare("DELETE FROM users WHERE id = :id");
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]);
}

public function findByDni(string $dni, int $excludeId = 0) {
    $sql = "SELECT id FROM users WHERE dni = :dni AND id != :exclude AND is_active = 1";
    $stmt = $this->db->prepare($sql);
    $stmt->execute([':dni' => $dni, ':exclude' => $excludeId]);
    return $stmt->fetch();
}
}

```

HELPERS

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', filemtime($file)),
            'timestamp' => filemtime($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 $file) {
        if (time() - filemtime($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;
}
}
```

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 l=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();
}
}

```

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;
        }
    }
}

```

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($roleNames) {
// Superusuario siempre tiene acceso
if (!empty($_SESSION['is_superadmin'])) return true;
if (!isset($_SESSION['roles'])) return false;

```

```

$roleNames = (array) $roleNames;
return count(array_intersect($roleNames, $_SESSION['roles'])) > 0;
}

```

```

public static function isSuperAdmin() {
return !empty($_SESSION['is_superadmin']);
}

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;
}
}

```

VIEWS

views/attendance/calendar.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>Calendario de Asistencia - EcuAsist</title>
    <style>
        .calendar-grid { display: grid; grid-template-columns: repeat(7, 1fr); gap: 6px; margin-top: 16px; }
        .cal-header { text-align: center; font-size: 0.78rem; font-weight: 600; color: #555; padding: 8px;
background: #f8f9fa; border-radius: 4px; }
        .cal-day { border: 1px solid #e0e0e0; padding: 8px; min-height: 70px; border-radius: 6px; background: #fff;
position: relative; font-size: 0.82rem; }
        .cal-day.empty { background: #f8f9fa; border: none; }
        .cal-day.today { background: #fff8e1; border-color: #ffc107; }
        .cal-day.has-data { background: #f0faf0; border-color: #28a745; }
        .cal-day.weekend { background: #fafafa; color: #bbb; }
        .day-num { font-weight: 700; font-size: 0.9rem; margin-bottom: 4px; }
        .day-summary { font-size: 0.72rem; color: #666; line-height: 1.4; }
        .day-summary.ok { color: #28a745; font-weight: 600; }
        .day-summary.no { color: #dc3545; font-weight: 600; }
        .month-nav { display: flex; justify-content: space-between; align-items: center; margin-bottom: 16px; }
        .month-nav h2 { font-size: 1.1rem; font-weight: 600; color: #333; }
    </style>
</head>
<body>

<?php include BASE_PATH . '/views/partials/navbar.php'; ?>

<div class="breadcrumb">
    <a href="?action=dashboard">■ Inicio</a> &rsquo;
    <a href="?action=view_course_attendance">Asistencia</a> &rsquo;
    Calendario
</div>

<div class="container">

    <!-- Header -->
    <div class="page-header teal">
        <div class="ph-icon">■</div>
        <div>
            <h1>Calendario de Asistencia</h1>

<p>Vista mensual del registro de asistencia por curso</p>
</div>
</div>

    <!-- Selector de curso -->
    <div class="panel" style="margin-bottom: 16px;">
        <div class="form-row">
            <div class="form-group" style="flex: 2;">
                <label>Seleccionar Curso</label>
                <select id="course-select" class="form-control" onchange="loadCalendar()">
                    <option value="">Seleccionar curso...</option>
                    <?php foreach($courses as $course): ?>
                        <option value="<?= $course['id'] ?>"
                            <?= isset($_GET['course_id']) && $_GET['course_id'] == $course['id'] ? 'selected' : '' ?>>
                            <?= htmlspecialchars($course['name']) ?> - <?= ucfirst($course['shift_name']) ?>
                        </option>
                    <?php endforeach; ?>
                </select>
            </div>
        </div>
    </div>
```

```

</div>

<?php if(isset($_GET['course_id'])): ?>

<!-- Navegación del mes -->
<div class="month-nav">
  <a href="?action=attendance_calendar&course_id=<?= (int)$_GET['course_id'] ?>&month=<?=
htmlspecialchars($prevMonth) ?>"
  class="btn btn-outline"><- Anterior</a>
  <h2><?= htmlspecialchars($monthName) ?> <?= $year ?></h2>
  <a href="?action=attendance_calendar&course_id=<?= (int)$_GET['course_id'] ?>&month=<?=
htmlspecialchars($nextMonth) ?>"
  class="btn btn-outline">Siguiente -></a>
</div>

<!-- Calendario -->
<div class="panel" style="padding:16px;">
<div class="calendar-grid">
<?php foreach(['Dom', 'Lun', 'Mar', 'Mié', 'Jue', 'Vie', 'Sáb'] as $d): ?>
<div class="cal-header"><?= $d ?></div>
<?php endforeach; ?>

<?php foreach($calendarDays as $day): ?>
  <?php if(!isset($day['day'])): ?>
    <div class="cal-day empty"></div>
  <?php else: ?>
    <div class="cal-day <?= $day['classes'] ?>">
    <div class="day-num"><?= $day['day'] ?></div>
    <?php if(isset($day['attendance'])): ?>
    <div class="day-summary">
    <span class="ok">>✓ <?= $day['attendance']['presente'] ?></span>
    &nbsp;
    <span class="no">>✗ <?= $day['attendance']['ausente'] ?></span>
    </div>
    <?php endif; ?>
  </div>
  <?php endif; ?>
<?php endforeach; ?>
</div>

<!-- Leyenda -->
<div style="margin-top:16px;display:flex;gap:16px;flex-wrap:wrap;font-size:0.8rem;color:#666;">
  <span><span style="display:inline-block;width:12px;height:12px;background:#f0faf0;border:1px solid
#28a745;border-radius:2px;"></span> Con asistencia</span>
  <span><span style="display:inline-block;width:12px;height:12px;background:#fff8e1;border:1px solid
#ffc107;border-radius:2px;"></span> Hoy</span>
  <span><span style="display:inline-block;width:12px;height:12px;background:#fafafa;border:1px solid
#e0e0e0;border-radius:2px;"></span> Fin de semana</span>
</div>

<?php else: ?>
  <div class="empty-state">
    <div class="icon">■</div>
    <p>Selecciona un curso para ver el calendario.</p>
  </div>
<?php endif; ?>
</div>

<script>
function loadCalendar() {
  const id = document.getElementById('course-select').value;
  if (id) window.location.href = '?action=attendance_calendar&course_id=' + id;
}
</script>
</body>
</html>

```


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>Ver Asistencia del Curso - EcuAsist</title>
</head>
<body>

<?php include BASE_PATH . '/views/partials/navbar.php'; ?>

<div class="breadcrumb">
    <a href="?action=dashboard">■ Inicio</a> &rsquo;
    Asistencia del Curso
</div>

<div class="container">

    <?php if(empty($courses)): ?>
    <div class="alert alert-warning">
        ■■ No tiene cursos asignados.
    <?php if(Security::hasRole('docente')): ?>
    Contacte con la autoridad para que le asignen cursos.
    <?php endif; ?>
    </div>
    <?php else: ?>

    <!-- Header -->
    <div class="page-header blue">
    <div class="ph-icon">■</div>
    <div>
    <h1>Asistencia del Curso</h1>
    <p>Consulta estadísticas de asistencia por curso y período</p>
    </div>
    </div>

    <!-- Filtros -->
    <div class="panel">
    <form method="GET" id="filterForm">

<input type="hidden" name="action" value="view_course_attendance">
    <div class="form-row" style="align-items:flex-end;">
    <div class="form-group" style="flex:2;">
    <label>Curso *</label>
    <select name="course_id" class="form-control" required>
    <option value="">Seleccionar curso...</option>
    <?php foreach($courses as $c):
    $label = htmlspecialchars($c['name']);
    if (!empty($c['is_tutor'])) $label .= ' ■ (Mi curso)';
    ?>
    <option value="<?= $c['id'] ?>" <?= isset($courseId) && $courseId == $c['id'] ? 'selected' : '' ?>>
    <?= $label ?>
    </option>
    <?php endforeach; ?>
    </select>
    </div>
    <div class="form-group">
    <label>Desde</label>
    <input type="date" name="start_date" class="form-control"
    value="<?= htmlspecialchars($startDate ?? date('Y-m-01')) ?>">
    </div>
    <div class="form-group">
    <label>Hasta</label>
    <input type="date" name="end_date" class="form-control"
    value="<?= htmlspecialchars($endDate ?? date('Y-m-d')) ?>">
    </div>
    <div class="form-group" style="flex:0 0 auto;">
    <label>&nbsp;</label>
    </div>
    </div>
    </div>
    </div>
```

```

<button type="submit" class="btn btn-primary" style="width:100%;">■ Consultar</button>
</div>
</div>
</form>
</div>

```

```

<?php if(isset($courseId) && !empty($attendanceData)):
$totalPresente = array_sum(array_column($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'));

```

```

$spct = fn($v) => $totalRegistros > 0 ? round($v / $totalRegistros * 100, 1) : 0;
?>

```

```

<!-- Stat cards -->
<div style="display:flex;gap:12px;margin-bottom:20px;flex-wrap:wrap;">
<div class="stat-card gray">
<div class="number"><?=$totalRegistros ?></div>
<div class="label">Total Registros</div>
</div>
<div class="stat-card green">
<div class="number"><?=$totalPresente ?></div>
<div class="label">Presentes (<?=$spct($totalPresente) ?>%</div>
<div class="progress-bar"><div class="progress-fill" style="width:<?=$spct($totalPresente) ?
>%;background:#28a745;"></div></div>
</div>
<div class="stat-card red">
<div class="number"><?=$totalAusente ?></div>
<div class="label">Ausentes (<?=$spct($totalAusente) ?>%</div>
<div class="progress-bar"><div class="progress-fill" style="width:<?=$spct($totalAusente) ?
>%;background:#dc3545;"></div></div>
</div>
<div class="stat-card yellow">
<div class="number"><?=$totalTardanza ?></div>
<div class="label">Tardanzas (<?=$spct($totalTardanza) ?>%</div>
<div class="progress-bar"><div class="progress-fill" style="width:<?=$spct($totalTardanza) ?
>%;background:#ffc107;"></div></div>
</div>
<div class="stat-card teal">
<div class="number"><?=$totalJustificado ?></div>
<div class="label">Justificados (<?=$spct($totalJustificado) ?>%</div>
<div class="progress-bar"><div class="progress-fill" style="width:<?=$spct($totalJustificado) ?
>%;background:#17a2b8;"></div></div>
</div>
</div>

<!-- Tabla estudiantes -->
<div class="table-wrap">
<div class="table-info">
<span>■ Período: <?=$date('d/m/Y', strtotime($startDate)) ?> al <?=$date('d/m/Y', strtotime($endDate)) ?
></span>
<span><strong><?=$count($attendanceData) ?></strong> estudiantes</span>
</div>
<table>
<thead>
<tr>

```

```

<th>#</th>
<th>Estudiante</th>
<th style="text-align:center;">Total</th>
<th style="text-align:center;">Presentes</th>
<th style="text-align:center;">Ausentes</th>
<th style="text-align:center;">Tardanzas</th>
<th style="text-align:center;">Justificados</th>
<th style="text-align:center;">% Asistencia</th>
</tr>
</thead>
<tbody>
<?php $i = 1; foreach($attendanceData as $data):
$spctRow = $data['total_registros'] > 0

```

```

? round(($data['presente'] / $data['total_registros']) * 100, 1) : 0;
$pctColor = $pctRow >= 90 ? '#28a745' : ($pctRow >= 75 ? '#e6ac00' : '#dc3545');
?>
<tr>
<td style="color:#999;"><?= $i++ ?></td>
<td><strong><?= htmlspecialchars($data['student_name']) ?></strong></td>
<td style="text-align:center;"><?= $data['total_registros'] ?></td>
<td style="text-align:center;"><span class="badge badge-green"><?= $data['presente'] ?></span></td>
<td style="text-align:center;"><span class="badge badge-red"><?= $data['ausente'] ?></span></td>
<td style="text-align:center;"><span class="badge badge-yellow"><?= $data['tardanza'] ?></span></td>
<td style="text-align:center;"><span class="badge badge-teal"><?= $data['justificado'] ?></span></td>
<td style="text-align:center;font-weight:700;color:<?= $pctColor ?>;"><?= $pctRow ?>%</td>
</tr>
<?php endforeach; ?>
</tbody>
</table>
</div>

<?php elseif(isset($courseId)): ?>
<div class="empty-state">
<div class="icon">■</div>
<p>No hay registros de asistencia para el período seleccionado.</p>
</div>
<?php else: ?>
<div class="empty-state">
<div class="icon">■</div>
<p>Seleccione un curso y período para ver la asistencia.</p>
</div>
<?php endif; ?>

</div>
</body>
</html>

```

views/attendance/my_attendance.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>Mi Asistencia - EcuAsist</title>
</head>
<body>

<?php include BASE_PATH . '/views/partials/navbar.php'; ?>

<div class="breadcrumb">
<a href="?action=dashboard">■ Inicio</a> &rsquo; Mi Asistencia
</div>

<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'));
$pctAsist = $total > 0 ? round(($presente / $total) * 100, 1) : 0;
$pct = fn($v) => $total > 0 ? round($v / $total * 100, 1) : 0;
?>

<!-- Header -->
<div class="page-header green">

```

```

<div class="ph-icon">■</div>
<div>
<h1>Mi Asistencia</h1>
<p>Historial completo de tu asistencia escolar</p>
</div>
<div class="ph-actions">
<a href="?action=my_justifications" class="btn btn-outline"
style="color:#fff;border-color:rgba(255,255,255,0.5);">
■ Mis Justificaciones
</a>
</div>

```

```

</div>

```

```

<!-- Stat cards -->
<div style="display:flex;gap:12px;margin-bottom:20px;flex-wrap:wrap;">
<div class="stat-card gray">
<div class="number"><?=$total ?></div>
<div class="label">Total Registros</div>
</div>
<div class="stat-card green">
<div class="number"><?=$pctAsist ?>%</div>
<div class="label">Asistencia (<?=$presente ?> presentes)</div>
<div class="progress-bar"><div class="progress-fill" style="width:<?=$pctAsist ?
>%;background:#28a745;"></div></div>
</div>
<div class="stat-card red">
<div class="number"><?=$ausente ?></div>
<div class="label">Ausencias (<?=$pct($ausente) ?>%</div>
<div class="progress-bar"><div class="progress-fill" style="width:<?=$pct($ausente) ?
>%;background:#dc3545;"></div></div>
</div>
<div class="stat-card yellow">
<div class="number"><?=$tardanza ?></div>
<div class="label">Tardanzas (<?=$pct($tardanza) ?>%</div>
<div class="progress-bar"><div class="progress-fill" style="width:<?=$pct($tardanza) ?
>%;background:#ffc107;"></div></div>
</div>
</div>

```

```

<!-- Tabla -->
<div class="table-wrap">
<div class="table-info">
<span>■ <strong><?=$total ?></strong> registros</span>
</div>
<?php if(empty($attendances)): ?>
<div class="empty-state">
<div class="icon">■</div>
<p>No hay registros de asistencia.</p>
</div>
<?php else: ?>
<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>Acción</th>
</tr>
</thead>
<tbody>
<?php foreach($attendances as $att):
$badgeClass = match($att['status']) {
'presente' => 'badge-green',
'ausente' => 'badge-red',
'tardanza' => 'badge-yellow',
'justificado' => 'badge-teal',

```

```

default => 'badge-gray'
};
$label = match($att['status']) {
'presente' => 'Presente',
'ausente' => 'Ausente',
'tardanza' => 'Tardanza',
'justificado' => 'Justificado',
default => ucfirst($att['status'])
};
?>
<tr>
<td><?= date('d/m/Y', strtotime($att['date'])) ?></td>
<td><?= htmlspecialchars($att['course_name']) ?></td>
<td><?= htmlspecialchars($att['subject_name']) ?></td>
<td style="color:#666;"><?= ucfirst($att['shift_name']) ?></td>
<td style="white-space:nowrap;"><?= htmlspecialchars($att['hour_period']) ?></td>
<td><span class="badge <?= $badgeClass ?>"><?= $label ?></span></td>
<td style="color:#888;font-size:0.82rem;"><?= htmlspecialchars($att['observation']) ?? '-' ?></td>
<td>
<?php if($att['status'] === 'ausente'): ?>
<a href="?action=submit_justification&attendance_id=<?= $att['id'] ?>"
class="btn btn-warning btn-sm">■ Justificar</a>
<?php else: ?>
<span style="color:#ccc;">—</span>
<?php endif; ?>
</td>
</tr>
<?php endforeach; ?>
</tbody>
</table>
<?php endif; ?>
</div>
</body>
</html>

```

views/attendance/register.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>Registrar Asistencia - EcuAsist</title>
</head>
<body>

<?php include BASE_PATH . '/views/partials/navbar.php'; ?>

<div class="breadcrumb">
<a href="?action=dashboard">■ Inicio</a> &rsquo;
<a href="?action=attendance_view">Asistencia</a> &rsquo;
Registrar Asistencia
</div>

<div class="container">

<?php if(isset($_GET['success'])): ?>
<div class="alert alert-success">✓ Asistencia registrada correctamente</div>
<?php endif; ?>
<?php if(isset($_GET['error'])): ?>
<?php if($_GET['error'] == 'future'): ?>
<div class="alert alert-danger">✗ No se puede registrar asistencia de fechas futuras</div>
<?php elseif($_GET['error'] == 'toolate'): ?>
<div class="alert alert-danger">✗ Solo se puede registrar asistencia de hasta 48 horas hábiles atrás</div>
<?php endif; ?>

```

```

<?php endif; ?>

<!-- Header -->
<div class="page-header green">
<div class="ph-icon">■</div>
<div>
<h1>Registrar Asistencia</h1>
<p>Selecciona tu clase del día y toma asistencia de los estudiantes</p>
</div>
</div>

<!-- Clases del día -->
<div class="panel">
<h3>■ Mis Clases de Hoy</h3>
<div class="alert alert-warning" style="margin-bottom:12px;">
■ <strong>Importante:</strong> Puede registrar asistencia de hoy o corregir registros anteriores hasta el
<strong><?= date('d/m/Y', strtotime($maxEditDate)) ?></strong>.
</div>

<?php if(empty($todayClasses)): ?>
<div class="empty-state">
<div class="icon">■</div>
<p>No tienes clases programadas para hoy según tu horario.</p>
</div>
<?php else: ?>
<div id="classes-list" style="display:grid;grid-template-columns:repeat(auto-fill,minmax(280px,1fr));gap:12px;">
<?php foreach($todayClasses as $class):
$jsonCourse = json_encode($class['course_name'], JSON_HEX_TAG | JSON_HEX_APOS);
$jsonSubject = json_encode($class['subject_name'], JSON_HEX_TAG | JSON_HEX_APOS);
?>
<div class="class-card" onclick='selectClass(event, <?= $class["id"] ?>, <?= $jsonCourse ?>, <?= $jsonSubject ?>, <?= $class["period_number"] ?>)'>
<div style="font-size:1.5rem;margin-bottom:6px;">■</div>
<div style="font-weight:700;color:#007bff;font-size:0.95rem;margin-bottom:4px;">
<?= $class['period_number'] ?>ra hora – <?= htmlspecialchars($class['subject_name']) ?>
</div>
<div style="font-size:0.83rem;color:#666;">Curso: <?= htmlspecialchars($class['course_name']) ?></div>
<div style="margin-top:10px;">
<span class="btn btn-primary btn-sm">Tomar Asistencia →</span>
</div>
</div>
<?php endforeach; ?>
</div>
<?php endif; ?>
</div>

<!-- Selector de fecha -->
<div class="panel" id="date-selector" style="display:none;">
<h3>■ Seleccionar Fecha</h3>
<form id="config-form">
<input type="hidden" id="schedule_id" name="schedule_id">
<div class="form-row">

<div class="form-group">
<label>Clase Seleccionada</label>
<input type="text" id="selected_class_info" class="form-control" readonly style="background:#f8f9fa;">
</div>
<div class="form-group">
<label>Fecha</label>
<input type="date" id="date" name="date" class="form-control"
value="<?= date('Y-m-d') ?>"
max="<?= date('Y-m-d') ?>"
min="<?= $minDate ?>" required>
<div class="form-hint">■ La asistencia de hoy podrá editarse hasta el <?= date('d/m/Y',
strtotime($maxEditDate)) ?></div>
</div>
</div>
<button type="button" class="btn btn-primary" onclick="loadStudents()">■ Cargar Estudiantes</button>
</form>
</div>

```

```

<!-- Lista de estudiantes -->
<form method="POST" id="student-list" style="display:none;">
<input type="hidden" name="schedule_id" id="schedule_id_hidden">
<input type="hidden" name="date" id="date_hidden">

<div class="panel">
<h3>■ Lista de Estudiantes</h3>
<div class="table-wrap" style="margin-top:8px;">
<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>
</div>
<div style="margin-top:16px;">
<button type="submit" class="btn btn-success btn-lg">■ Guardar Asistencia</button>

```

```

</div>
</div>
</form>

```

```

</div>

```

```

<style>
.class-card {
background: #fff;
border: 2px solid #e0e0e0;
border-radius: 8px;
padding: 16px;
cursor: pointer;
transition: border-color 0.2s, box-shadow 0.2s;
}
.class-card:hover {
border-color: #007bff;
box-shadow: 0 4px 12px rgba(0,123,255,0.15);
}
.class-card.selected {
border-color: #007bff;
background: #f0f7ff;
}
</style>

```

```

<script>
const minDateAllowed = '<?= $minDate ?>';
let selectedScheduleId = null;

function selectClass(event, scheduleId, courseName, subjectName, periodNumber) {
selectedScheduleId = scheduleId;
document.querySelectorAll('.class-card').forEach(c => c.classList.remove('selected'));
event.currentTarget.classList.add('selected');
document.getElementById('date-selector').style.display = 'block';
document.getElementById('schedule_id').value = scheduleId;
document.getElementById('selected_class_info').value = periodNumber + 'ra hora - ' + subjectName + ' (' +
courseName + ')';
document.getElementById('student-list').style.display = 'none';
document.getElementById('date-selector').scrollIntoView({ behavior: 'smooth' });
}

```

```

document.getElementById('date') && document.getElementById('date').addEventListener('change', function() {
const today = new Date().toISOString().split('T')[0];
if (this.value > today) { alert('No se puede seleccionar una fecha futura'); this.value = today; return; }
if (this.value < 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;

  fetch('?action=get_schedule_info&schedule_id=' + scheduleId)
    .then(r => r.json())
    .then(scheduleData => {
      const formData = new FormData();
      formData.append('course_id', scheduleData.course_id);
      return fetch('?action=get_students', { method: 'POST', body: formData })
        .then(r => r.json())
        .then(students => {
          if (!students.length) { alert('No hay estudiantes matriculados en este curso'); return; }
          return fetch('?action=get_existing_attendance&schedule_id=' + scheduleId + '&date=' + date)
            .then(r => r.json())
            .then(existing => {
              const existingMap = {};
              existing.forEach(a => { existingMap[a.student_id] = { status: a.status, observation: a.observation || '' }; });
              const tbody = document.querySelector('#students-table tbody');
              tbody.innerHTML = '';
              const statuses = ['presente', 'ausente', 'tardanza', 'justificado'];
              students.forEach((student, idx) => {
                const cur = existingMap[student.id] || { status: 'presente', observation: '' };
                const opts = statuses.map(s => `<option value="${s}" ${cur.status===s?'selected':''}>${s.charAt(0).toUpperCase()}${s.slice(1)}</option>`).join('');
                tbody.innerHTML += `<tr>
<td style="color:#999;">${idx+1}</td>
<td><strong>${student.last_name} ${student.first_name}</strong></td>
<td style="color:#666;">${student.dni||'-'}</td>
<td><select name="status_${student.id}" style="padding:5px 8px;border:1px solid #ccc;border-radius:6px;font-size:0.85rem;">${opts}</select></td>
<td><input type="text" name="obs_${student.id}" value="${cur.observation}" placeholder="Opcional" style="width:100%;padding:5px 8px;border:1px solid #ccc;border-radius:6px;font-size:0.85rem;"></td>
</tr>`;
              });
              document.getElementById('student-list').style.display = 'block';
              document.getElementById('student-list').scrollIntoView({ behavior: 'smooth' });
            });
            .catch(err => console.error('Error:', err));
          }
        }
      </script>

</body>
</html>

```

views/attendance/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>Ver Asistencias - EcuAsist</title>
</head>
<body>

<?php include BASE_PATH . '/views/partials/navbar.php'; ?>

<div class="breadcrumb">
  <a href="?action=dashboard">■ Inicio</a> &rsaquo;

```



```

Asistencia &rsquo;
Ver Asistencias
</div>

<div class="container">

    <!-- Header -->
    <div class="page-header" style="background:linear-gradient(135deg,#1565c0,#1976d2);">
    <div class="ph-icon">■</div>
    <div>
    <h1>Ver Asistencias</h1>
    <p>Consulta el registro de asistencia por curso y fecha</p>
    </div>
    </div>

    <!-- Filtros -->
    <div class="panel" style="margin-bottom:20px;">
    <h3 style="margin-bottom:16px;font-size:.95rem;color:#555;">■ Filtros de Búsqueda</h3>
    <form method="POST" id="filterForm">
    <div class="form-row" style="align-items:flex-end;">
    <div class="form-group" style="flex:2;">
    <label>Curso</label>
    <select name="course_id" class="form-control" required>
    <option value="">Seleccionar curso...</option>
    <?php foreach($courses as $course): ?>
    <option value="<?= $course['id'] ?>"

<?= (isset($_POST['course_id']) && $_POST['course_id'] == $course['id']) ? 'selected' : '' ?>>
    <?= htmlspecialchars($course['name']) ?>
    - <?= ucfirst($course['shift_name']) ?>
    </option>
    <?php endforeach; ?>
    </select>
    </div>
    <div class="form-group" style="flex:1;">
    <label>Fecha</label>
    <input type="date" name="date" class="form-control"
    value="<?= isset($_POST['date']) ? htmlspecialchars($_POST['date']) : date('Y-m-d') ?>"
    max="<?= date('Y-m-d') ?>" required>
    </div>
    <div class="form-group" style="flex:0 0 auto;">
    <button type="submit" class="btn btn-primary">■ Buscar</button>
    </div>
    </div>
    </form>
    </div>

    <?php if($_SERVER['REQUEST_METHOD'] === 'POST'): ?>

    <?php if(empty($attendances)): ?>
    <!-- Sin resultados -->
    <div class="empty-state">
    <div class="icon">■</div>
    <p>No se encontraron registros de asistencia para los filtros seleccionados.</p>
    <small style="color:#aaa;">Verifica que el docente haya registrado asistencia para este curso y
    fecha.</small>
    </div>

    <?php else:
    /* ---- Calcular estadísticas ---- */
    $stats = ['presente'=>0,'ausente'=>0,'tardanza'=>0,'justificado'=>0];
    $students = [];
    foreach($attendances as $att) {
    $s = $att['status'];
    if(isset($stats[$s])) $stats[$s]++;
    $students[$att['last_name'].' '.$att['first_name']] = true;
    }
    $total = count($attendances);

$totalStu = count($students);
    $pctPresente = $total > 0 ? round(($stats['presente'] + $stats['justificado']) / $total * 100) : 0;

```

```

$selectedCourse = '';
foreach($courses as $c) {
    if($c['id'] == $_POST['course_id']) {
        $selectedCourse = $c['name'] . ' - ' . ucfirst($c['shift_name']);
        break;
    }
}
?>

<!-- Stat cards -->
<div style="display:grid;grid-template-columns:repeat(5,1fr);gap:16px;margin-bottom:20px;">
<div class="stat-card">
<div class="stat-icon" style="background:#e3f2fd;">■</div>
<div class="stat-value"><?= $total ?></div>
<div class="stat-label">Total registros</div>
</div>
<div class="stat-card">
<div class="stat-icon" style="background:#e8f5e9;">■</div>
<div class="stat-value" style="color:#2e7d32;"><?= $stats['presente'] ?></div>
<div class="stat-label">Presentes</div>
</div>
<div class="stat-card">
<div class="stat-icon" style="background:#ffebee;">■</div>
<div class="stat-value" style="color:#c62828;"><?= $stats['ausente'] ?></div>
<div class="stat-label">Ausentes</div>
</div>
<div class="stat-card">
<div class="stat-icon" style="background:#fffde7;">■</div>
<div class="stat-value" style="color:#f57f17;"><?= $stats['tardanza'] ?></div>
<div class="stat-label">Tardanzas</div>
</div>
<div class="stat-card">
<div class="stat-icon" style="background:#e0f7fa;">■</div>
<div class="stat-value" style="color:#00838f;"><?= $stats['justificado'] ?></div>
<div class="stat-label">Justificados</div>
</div>
</div>

```

```

<!-- Barra de asistencia global -->
<div class="panel" style="margin-bottom:20px;">
<div style="display:flex;justify-content:space-between;align-items:center;margin-bottom:8px;">
<span style="font-size:.9rem;color:#555;">
■ <strong><?= htmlspecialchars($selectedCourse) ?></strong>
&nbsp;&nbsp;&nbsp;|&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;■ <?= date('d/m/Y', strtotime($_POST['date'])) ?>
&nbsp;&nbsp;&nbsp;|&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;■ <?= $totalStu ?> estudiante<?= $totalStu!=1?'s':'' ?>
</span>
<span style="font-weight:700;font-size:1.1rem;color:<?=
$pctPresente>=90?'#2e7d32':($pctPresente>=75?'#f57f17':'#c62828') ?>">
<?= $pctPresente ?>% asistencia efectiva
</span>
</div>
<div style="background:#f0f0f0;border-radius:6px;height:10px;overflow:hidden;">
<div style="height:100%;border-radius:6px;width:<?= $pctPresente ?>%;
background:<?= $pctPresente>=90?'#4caf50':($pctPresente>=75?'#ff9800':'#f44336') ?>;
transition:width .4s;">
</div>
</div>
</div>

<!-- Tabla de resultados -->
<div class="table-wrap">
<div class="table-info">
<span>■ <strong>Registros de Asistencia</strong> – <?= $total ?></span>
<div style="display:flex;gap:8px;align-items:center;">
<input type="text" id="searchTable" class="form-control"
placeholder="■ Buscar estudiante..."
style="width:200px;font-size:13px;"
oninput="filterTable(this.value)">
</div>
</div>
<table id="attendanceTable">

```

```

<thead>
<tr>
<th>#</th>
<th>Estudiante</th>
<th>Asignatura</th>
<th>Docente</th>
<th style="text-align:center;">Hora</th>

<th style="text-align:center;">Estado</th>
<th>Observación</th>
</tr>
</thead>
<tbody>
<?php
$i = 1;
foreach($attendances as $att):
    $s = $att['status'];
    $badgeMap = [
        'presente' => 'badge-green',
        'ausente' => 'badge-red',
        'tardanza' => 'badge-yellow',
        'justificado' => 'badge-teal',
    ];
    $labelMap = [
        'presente' => '✓ Presente',
        'ausente' => '✗ Ausente',
        'tardanza' => '■ Tardanza',
        'justificado' => '■ Justificado',
    ];
    $badge = $badgeMap[$s] ?? 'badge-gray';
    $label = $labelMap[$s] ?? ucfirst($s);
    ?>
    <tr>
    <td style="color:#999;font-size:12px;"><? = $i++ ?></td>
    <td>
    <strong><? = htmlspecialchars($att['last_name'] . ' ' . $att['first_name']) ?></strong>
    </td>
    <td style="color:#555;"><? = htmlspecialchars($att['subject_name']) ?></td>
    <td style="font-size:13px;color:#666;"><? = htmlspecialchars($att['teacher_name']) ?></td>
    <td style="text-align:center;">
    <span style="font-size:13px;background:#f0f0f0;padding:3px 8px;border-radius:4px;">
    <? = htmlspecialchars($att['hour_period']) ?>
    </span>
    </td>
    <td style="text-align:center;">
    <span class="badge <? = $badge ?>"><? = $label ?></span>
    </td>
    <td style="font-size:13px;color:#777;">

<? = $att['observation'] ? htmlspecialchars($att['observation']) : '<span style="color:#ccc;">—</span>' ?>
    </td>
</tr>
<?php endforeach; ?>
</tbody>
</table>
</div>

<!-- Leyenda -->
<div style="margin-top:12px;display:flex;gap:12px;flex-wrap:wrap;">
<span class="badge badge-green">✓ Presente</span>
<span class="badge badge-red">✗ Ausente</span>
<span class="badge badge-yellow">■ Tardanza</span>
<span class="badge badge-teal">■ Justificado</span>
</div>

<?php endif; ?>
<?php endif; ?>

</div>

<script>
function filterTable(q) {

```

```

q = q.toLowerCase();
const rows = document.querySelectorAll('#attendanceTable tbody tr');
rows.forEach(r => {
  const name = r.cells[1] ? r.cells[1].textContent.toLowerCase() : '';
  r.style.display = name.includes(q) ? '' : 'none';
});
}
</script>

</body>
</html>

```

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">
  <title>Editar Curso - EcuAsist</title>
  <style>
    * { margin: 0; padding: 0; box-sizing: border-box; }
    body { font-family: Arial, sans-serif; background: #f4f4f4; }
    .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); }
    .card-header { background: #007bff; color: white; padding: 15px 20px; border-radius: 8px 8px 0 0; margin:
-30px -30px 25px -30px; }
    .card-header h4 { margin: 0; font-size: 18px; }
    .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; font-size: 14px; }
    .btn { padding: 10px 20px; border: none; border-radius: 4px; cursor: pointer; font-size: 14px;
text-decoration: none; display: inline-block; }
    .btn-success { background: #28a745; color: white; }
    .btn-secondary { background: #6c757d; color: white; }
    .alert-danger { background: #f8d7da; color: #721c24; padding: 10px; border-radius: 4px; margin-bottom:
15px; }
    .d-flex { display: flex; justify-content: space-between; margin-top: 20px; }
  </style>
</head>
<body>
  <?php include BASE_PATH . '/views/partials/navbar.php'; ?>

  <div class="container">
    <div class="card">
      <div class="card-header">
        <h4>■ Editar Curso: <?= htmlspecialchars($course['name']) ?></h4>
      </div>

      <?php if (!empty($errors)): ?>
      <div class="alert-danger">
        <?php foreach ($errors as $error): ?>
        <div><?= htmlspecialchars($error) ?></div>
      </div>
      <?php endforeach; ?>
    </div>
    <?php endif; ?>

    <form method="POST">

      <div class="form-group">
        <label>Nivel Educativo</label>
        <select name="education_type" id="education_type" required onchange="updateGradeLevels()">
          <option value="">Seleccionar tipo...</option>
          <option value="inicial">■ Educación Inicial</option>

```

```

<option value="egb">■ Educación General Básica (EGB)</option>
<option value="bgu">■ Bachillerato General Unificado (BGU)</option>
<option value="bt">■ Bachillerato Técnico (BT)</option>
</select>
</div>

```

```

<div class="form-group">
<label>Grado / Año</label>
<select name="grade_level" id="grade_level" required onchange="onGradeChange()">
<option value="">Seleccione nivel primero...</option>
</select>
</div>

```

```

<div class="form-group" id="group_specialty" style="display:none;">
<label>Figura Profesional</label>
<select name="specialty" id="specialty" onchange="updateCarreras()">
<option value="">Seleccionar figura...</option>
<option value="Informática">Informática</option>
<option value="Administración">Administración</option>
<option value="Contabilidad">Contabilidad</option>
<option value="Comercialización y Ventas">Comercialización y Ventas</option>
<option value="Servicios Hoteleros">Servicios Hoteleros</option>
<option value="Turismo">Turismo</option>
<option value="Electromecánica Automotriz">Electromecánica Automotriz</option>
<option value="Instalaciones Eléctricas">Instalaciones Eléctricas</option>
<option value="Electrónica de Consumo">Electrónica de Consumo</option>
<option value="Mecanizado y Construcciones Metálicas">Mecanizado y Construcciones Metálicas</option>
<option value="Producción Agropecuaria">Producción Agropecuaria</option>
<option value="Acuicultura">Acuicultura</option>

```

```

<option value="Industrialización de Alimentos">Industrialización de Alimentos</option>
<option value="Confección Textil">Confección Textil</option>
<option value="Música">Música</option>
<option value="Artes Plásticas">Artes Plásticas</option>
<option value="Diseño Gráfico">Diseño Gráfico</option>
<option value="Servicios de Belleza">Servicios de Belleza</option>
<option value="Atención Integral en Salud">Atención Integral en Salud</option>
<option value="Mecatrónica">Mecatrónica</option>
<option value="Refrigeración y Climatización">Refrigeración y Climatización</option>
<option value="Construcción">Construcción</option>
<option value="Redes y Telecomunicaciones">Redes y Telecomunicaciones</option>
<option value="Seguridad Industrial">Seguridad Industrial</option>
<option value="Logística y Transporte">Logística y Transporte</option>
</select>
</div>

```

```

<div class="form-group" id="group_carrera" style="display:none;">
<label>Especialidad / Carrera <span style="color:#999; font-weight:normal;">(opcional)</span></label>
<select name="carrera" id="carrera" onchange="generateCourseName()">
<option value="">Sin especificar</option>
</select>
</div>

```

```

<div class="form-group">
<label>Paralelo</label>
<select name="parallel" id="parallel" required onchange="generateCourseName()">
<option value="">Seleccionar...</option>
<?php foreach(range('A', 'J') as $letter): ?>
<option value="<?= $letter ?>" <?= $course['parallel'] == $letter ? 'selected' : '' ?>>
<?= $letter ?>
</option>
<?php endforeach; ?>
</select>
</div>

```

```

<div class="form-group">
<label>Jornada</label>
<select name="shift_id" id="shift_id" required onchange="generateCourseName()">
<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="form-group">
<label>Nombre del Curso <span style="color:#999; font-weight:normal;">(generado
automáticamente)</span></label>
<input type="text" name="name" id="course_name" readonly
style="background: #f0f0f0; font-weight:bold;"
value="<?= htmlspecialchars($course['name']) ?>">
</div>

<div class="d-flex">
<a href="?action=academic" class="btn btn-secondary"><← Cancelar</a>
<button type="submit" class="btn btn-success"><✓ Guardar Cambios</button>
</div>
</form>
</div>
</div>

<script>
// Datos del curso actual para precargar
const currentGradeLevel = <?= json_encode($course['grade_level']) ?>;
const currentShiftId = <?= json_encode((string)$course['shift_id']) ?>;

const gradeLevels = {
  inicial: ['Inicial 1 (0-3 años)', 'Inicial 2 (3-5 años)'],
  egb: ['1.º EGB - Preparatoria', '2.º EGB', '3.º EGB', '4.º EGB', '5.º EGB', '6.º EGB', '7.º EGB', '8.º EGB', '9.º
EGB', '10.º EGB'],
  bgu: ['1.º BGU', '2.º BGU', '3.º BGU'],
  bt: ['1.º BT', '2.º BT', '3.º BT']
};

const egbNocturnaAllowed = ['8.º EGB', '9.º EGB', '10.º EGB'];

const carreras = {
  'Informática': ['Aplicaciones Informáticas', 'Programación de Software', 'Soporte Técnico', 'Sistemas
Microinformáticos y Redes'],
  'Administración': ['Asistencia Administrativa', 'Gestión Empresarial'],
  'Contabilidad': ['Contabilidad', 'Ventas e Información Comercial'],
  'Comercialización y Ventas': ['Ventas e Información Comercial', 'Marketing'],
  'Servicios Hoteleros': ['Hotelería', 'Hospitalidad'],
  'Turismo': ['Turismo', 'Guía Turístico'],
  'Electromecánica Automotriz': ['Electromecánica Automotriz', 'Mecánica Automotriz'],
  'Instalaciones Eléctricas': ['Electricidad', 'Instalaciones Eléctricas'],
  'Electrónica de Consumo': ['Electrónica', 'Mantenimiento Electrónico'],
  'Atención Integral en Salud': ['Atención en Enfermería', 'Auxiliar de Salud'],
  'Producción Agropecuaria': ['Producción Agropecuaria', 'Agroindustria'],
  'Redes y Telecomunicaciones': ['Redes', 'Telecomunicaciones'],
  'Diseño Gráfico': ['Diseño Gráfico', 'Multimedia'],
  'Servicios de Belleza': ['Peluquería', 'Cosmetología']
};

// Detectar tipo educativo desde el grade_level actual
function detectType(grade) {
  if (!grade) return '';
  if (grade.includes('Inicial')) return 'inicial';
  if (grade.includes('EGB')) return 'egb';
  if (grade.includes('BGU')) return 'bgu';
  if (grade.includes('BT')) return 'bt';
  return '';
}

function getNocturnaOption() {
  const opts = document.getElementById('shift_id').options;
  for (let i = 0; i < opts.length; i++) {

```

```

if ((opts[i].getAttribute('data-shift') || '').toLowerCase() === 'nocturna') return opts[i];
}
return null;
}

function updateNocturnaVisibility(type, grade) {
const nocturna = getNocturnaOption();
if (!nocturna) return;
let visible = (type === 'bgu' || type === 'bt' || (type === 'egb' && egbNocturnaAllowed.includes(grade)));
nocturna.style.display = visible ? '' : 'none';
if (!visible && nocturna.selected) {
document.getElementById('shift_id').value = '';
}
}

function updateGradeLevels(pselectGrade) {
const type = document.getElementById('education_type').value;
const gradeSelect = document.getElementById('grade_level');
gradeSelect.innerHTML = '<option value="">Seleccionar grado...</option>';

if (!type) return;

(gradeLevels[type] || []).forEach(function(g) {
const opt = document.createElement('option');
opt.value = g;
opt.textContent = g;
if (pselectGrade && g === pselectGrade) opt.selected = true;
gradeSelect.appendChild(opt);
});

const isBT = (type === 'bt');
document.getElementById('group_specialty').style.display = isBT ? 'block' : 'none';
document.getElementById('group_carrera').style.display = isBT ? 'block' : 'none';
document.getElementById('specialty').required = isBT;

if (!isBT) {
document.getElementById('specialty').value = '';
document.getElementById('carrera').innerHTML = '<option value="">Sin especificar</option>';
}

updateNocturnaVisibility(type, gradeSelect.value);
}

function onGradeChange() {
const type = document.getElementById('education_type').value;
const grade = document.getElementById('grade_level').value;
updateNocturnaVisibility(type, grade);
generateCourseName();
}

function updateCarreras() {
const fig = document.getElementById('specialty').value;

const carreraSelect = document.getElementById('carrera');
carreraSelect.innerHTML = '<option value="">Sin especificar</option>';
if (carreras[fig]) {
carreras[fig].forEach(function(c) {
const opt = document.createElement('option');
opt.value = c; opt.textContent = c;
carreraSelect.appendChild(opt);
});
}
generateCourseName();
}

function generateCourseName() {
const grade = document.getElementById('grade_level').value;
const parallel = document.getElementById('parallel').value;
const shiftSelect = document.getElementById('shift_id');
const shiftOpt = shiftSelect.options[shiftSelect.selectedIndex];
const shiftName = shiftOpt ? (shiftOpt.getAttribute('data-shift') || '') : '';

```

```

const type = document.getElementById('education_type').value;
const specialty = document.getElementById('specialty').value;
const carrera = document.getElementById('carrera').value;

if (!grade || !parallel || !shiftName) return;

let name = grade + ' ' + parallel + ' ';
if (type === 'bt' && specialty) {
  name += ' - ' + specialty;
  if (carrera) name += ' (' + carrera + ')';
}
name += ' - ' + shiftName.charAt(0).toUpperCase() + shiftName.slice(1);
document.getElementById('course_name').value = name;
}

// Extraer figura profesional y carrera desde el nombre del curso
// Formato: "1.º BT "A" - Informática (Soporte Técnico) - Matutina"
function parseSpecialtyFromName(courseName) {
  const result = { specialty: '', carrera: '' };
  // Buscar patrón: - Figura (Carrera) - o - Figura -
  const match = courseName.match(/BT "[A-Z]" - ([^()]+)?(?:\s*\(((\[^\)]+\)\))?\s*-\s*\w+$/);
  if (match) {

```

```

result.specialty = match[1].trim();
result.carrera = match[2] ? match[2].trim() : '';
}
return result;
}

```

```

// PRECARGAR datos del curso al cargar la página
window.addEventListener('DOMContentLoaded', function() {
  const detectedType = detectType(currentGradeLevel);
  if (detectedType) {
    document.getElementById('education_type').value = detectedType;
    updateGradeLevels(currentGradeLevel);
  }
}

```

```

// Preseleccionar jornada
const shiftSelect = document.getElementById('shift_id');
for (let i = 0; i < shiftSelect.options.length; i++) {
  if (shiftSelect.options[i].value === currentShiftId) {
    shiftSelect.selectedIndex = i;
    break;
  }
}

```

```

// Precargar figura profesional y carrera (solo BT)
if (detectedType === 'bt') {
  const parsed = parseSpecialtyFromName(currentGradeLevel + ' - ' + <?= json_encode($course['name']) ?>);

```

```

// Intentar parsear directo del nombre completo del curso
const fullName = <?= json_encode($course['name']) ?>;
const matchFull = fullName.match(/- ([^()]+)?(?:\s*\(((\[^\)]+\)\))?\s*-\s*\w+$/);
if (matchFull) {
  const fig = matchFull[1].trim();
  const car = matchFull[2] ? matchFull[2].trim() : '';

```

```

// Seleccionar figura profesional
const specialtySelect = document.getElementById('specialty');
for (let i = 0; i < specialtySelect.options.length; i++) {
  if (specialtySelect.options[i].value === fig) {
    specialtySelect.selectedIndex = i;
    break;
  }
}

```

```

// Cargar carreras de esa figura y preseleccionar
if (fig && carreras[fig]) {
  const carreraSelect = document.getElementById('carrera');
  carreraSelect.innerHTML = '<option value="">Sin especificar</option>';
  carreras[fig].forEach(function(c) {

```



```

const opt = document.createElement('option');
opt.value = c; opt.textContent = c;
if (c === car) opt.selected = true;
carreraSelect.appendChild(opt);
});
}
}
}

updateNocturnaVisibility(detectedType, currentGradeLevel);
});
</script>
</body>
</html>

```

views/academic/course_enroll.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>Estudiantes - <?= htmlspecialchars($course['name']) ?> - EcuAsist</title>
  <style>
    .two-col-layout {
      display: grid;
      grid-template-columns: 1fr 1.6fr;
      gap: 20px;
      align-items: start;
    }
    @media (max-width: 900px) {
      .two-col-layout { grid-template-columns: 1fr; }
    }
    .student-check-item {
      display: flex; align-items: center; gap: 10px;
      padding: 8px 12px; border: 1px solid #e8e8e8; border-radius: 6px;
      margin-bottom: 6px; cursor: pointer; transition: background .15s;
    }
    .student-check-item:hover { background: #f0f7ff; border-color: #90caf9; }
    .student-check-item input[type="checkbox"] { width: 16px; height: 16px; cursor: pointer; }
    .student-check-item label { margin: 0; font-weight: normal; cursor: pointer; font-size: 13px; flex: 1; }
    .enrolled-row td { font-size: 13px; padding: 8px 10px; }
    .badge-enrolled { background: #28a745; color: white; padding: 3px 9px; border-radius: 10px; font-size:
11px; font-weight: 600; }
    .btn-sm { padding: 5px 10px; font-size: 12px; border: none; border-radius: 4px; cursor: pointer; color:
white; }
    .empty-state { text-align: center; padding: 30px; color: #999; }
    .empty-state .icon { font-size: 40px; margin-bottom: 10px; }
    .section-title {
      font-size: 15px; font-weight: 700; color: #333;
      margin-bottom: 14px; padding-bottom: 10px;
      border-bottom: 2px solid #f0f0f0;
      display: flex; justify-content: space-between; align-items: center;
    }
    .counter-badge {
      background: #e3f2fd; color: #1565c0;
      padding: 2px 10px; border-radius: 10px; font-size: 12px; font-weight: 700;
    }

/* Modal */
    .modal-overlay { display:none; position:fixed; inset:0; background:rgba(0,0,0,.5); z-index:9999;
align-items:center; justify-content:center; }
    .modal-overlay.open { display:flex; }
    .modal-box { background:#fff; border-radius:10px; padding:28px; max-width:420px; width:90%; box-shadow:0
8px 32px rgba(0,0,0,.2); }
    .modal-box h3 { margin:0 0 12px; color:#dc3545; }
    .modal-box p { margin:0 0 20px; color:#555; font-size:14px; }
    .modal-actions { display:flex; gap:10px; justify-content:flex-end; }

```

```

</style>
</head>
<body>

<?php include BASE_PATH . '/views/partials/navbar.php'; ?>

<div class="breadcrumb">
    <a href="?action=dashboard">■ Inicio</a> &rsquo;
    <a href="?action=academic">Configuración Académica</a> &rsquo;
    <?= htmlspecialchars($course['name']) ?> - Estudiantes
</div>

<div class="container">

    <?php if(isset($_GET['enrolled']) && (int)$_GET['enrolled'] > 0): ?>
        <div class="alert alert-success">✓ <?= (int)$_GET['enrolled'] ?> estudiante(s) matriculado(s)
correctamente</div>
    <?php endif; ?>
    <?php if(isset($_GET['errors']) && (int)$_GET['errors'] > 0): ?>
        <div class="alert alert-warning">■ <?= (int)$_GET['errors'] ?> estudiante(s) ya estaban matriculados en
otro curso</div>
    <?php endif; ?>
    <?php if(isset($_GET['unenrolled'])): ?>
        <div class="alert alert-success">✓ Estudiante retirado del curso correctamente</div>
    <?php endif; ?>
    <?php if(isset($_GET['error']) && $_GET['error'] === 'unenroll_failed'): ?>
        <div class="alert alert-warning">✗ Error al retirar el estudiante</div>
    <?php endif; ?>

    <div class="page-header" style="background:linear-gradient(135deg,#1565c0,#1976d2);">
    <div class="ph-icon">■</div>
    <div>
    <h1><?= htmlspecialchars($course['name']) ?></h1>
    <p>Jornada: <?= ucfirst($course['shift_name']) ?> &nbsp;|&nbsp; Gestión de matrículas</p>
</div>

```

```

</div>

<div class="two-col-layout">

    <!-- ■■ PANEL IZQUIERDO: Matricular ■■ -->
    <div class="panel">
        <div class="section-title">
            ■ Matricular Estudiantes
        <span class="counter-badge"><?= count($available) ?> disponibles</span>
        </div>

        <?php if(count($available) > 0): ?>
            <form method="POST" action="?action=enroll_students">
                <input type="hidden" name="course_id" value="<?= $course['id'] ?>">

                <div style="margin-bottom:10px;">
                    <input type="text" id="searchAvailable" placeholder="■ Buscar estudiante..."
                    class="form-control" style="font-size:13px;"
                    oninput="filterStudents()">
                </div>

                <div style="max-height:320px; overflow-y:auto; padding-right:4px;" id="studentList">
                    <?php foreach($available as $s): ?>
                        <div class="student-check-item" data-name="<?= strtolower($s['last_name'].' '.$s['first_name']) ?>">
                            <input type="checkbox" name="student_ids[]"
                            value="<?= $s['id'] ?>" id="s<?= $s['id'] ?>">
                            <label for="s<?= $s['id'] ?>">
                                <strong><?= htmlspecialchars($s['last_name'].' '.$s['first_name']) ?></strong>
                                <?php if(!empty($s['dni'])): ?>
                                    <span style="color:#999;font-size:11px;"> · <?= $s['dni'] ?></span>
                                <?php endif; ?>
                            </label>
                        </div>
                    <?php endforeach; ?>
                </div>
            </form>
        </div>
    </div>

```

```

<div style="display:flex; gap:8px; margin-top:12px;">
<button type="button" onclick="selectAll()" class="btn btn-outline" style="font-size:12px;padding:6px
12px;">
  Seleccionar todos
</button>

```

```

<button type="submit" class="btn btn-success" style="font-size:13px;padding:6px 16px;">
  ✓ Matricular seleccionados
</button>
</div>
</form>
<?php else: ?>
<div class="empty-state">
<div class="icon">■</div>
<p>Todos los estudiantes ya están matriculados en algún curso.</p>
</div>
<?php endif: ?>
</div>

```

```

<!-- ■■ PANEL DERECHO: Matriculados ■■ -->
<div class="panel">
<div class="section-title">
  ■ Estudiantes Matriculados
<span class="counter-badge"><?= count($enrolled) ?></span>
</div>

<?php if(count($enrolled) > 0): ?>
<div style="margin-bottom:10px;">
<input type="text" id="searchEnrolled" placeholder="■ Buscar..."
class="form-control" style="font-size:13px;"
oninput="filterEnrolled()">
</div>
<table style="width:100%;border-collapse:collapse;" id="enrolledTable">
<thead>
<tr style="background:#f8f9fa;">
<th style="padding:8px 10px;font-size:12px;text-align:left;">#</th>
<th style="padding:8px 10px;font-size:12px;text-align:left;">Apellidos y Nombres</th>
<th style="padding:8px 10px;font-size:12px;text-align:left;">Cédula</th>
<th style="padding:8px 10px;font-size:12px;text-align:center;">Acción</th>
</tr>
</thead>
<tbody>
<?php $n=1; foreach($enrolled as $s): ?>
<tr class="enrolled-row" data-name="<?= strtolower($s['last_name']).' '.$s['first_name']) ?>">
<td><?= $n++ ?></td>
<td><strong><?= htmlspecialchars($s['last_name']).' '.$s['first_name']) ?></strong></td>

```

```

<td><?= $s['dni'] ?? '-' ?></td>
<td style="text-align:center;">
<button class="btn-sm" style="background:#dc3545;"
onclick="confirmUnenroll(<?= $s['id'] ?>, '<?= addslashes($s['last_name']).' '.$s['first_name']) ?>')">
  ✕ Retirar
</button>
</td>
</tr>
<?php endforeach: ?>
</tbody>
</table>
<?php else: ?>
<div class="empty-state">
<div class="icon">■</div>
<p>Ningún estudiante matriculado en este curso aún.</p>
</div>
<?php endif: ?>
</div>

</div><!-- /two-col-layout -->

<div style="margin-top:16px;">
<a href="?action=academic" class="btn btn-outline">← Volver a Configuración Académica</a>
</div>
</div>

```

```

<!-- Modal Retirar -->
<div class="modal-overlay" id="unenrollModal">
  <div class="modal-box">
    <h3>■ Retirar Estudiante</h3>
    <p id="unenrollMsg"></p>
    <div class="modal-actions">
      <button class="btn btn-outline" onclick="closeModal()">Cancelar</button>
      <form method="POST" action="?action=unenroll_student" id="unenrollForm" style="display:inline;">
        <input type="hidden" name="student_id" id="unenrollStudentId">
        <input type="hidden" name="course_id" value="<?= $course['id'] ?>">
        <button type="submit" class="btn btn-danger">Sí, Retirar</button>
      </form>
    </div>
  </div>
</div>

</div>

<script>
function confirmUnenroll(studentId, name) {
  document.getElementById('unenrollStudentId').value = studentId;
  document.getElementById('unenrollMsg').innerHTML =
    '¿Retirar a <strong>' + name + '</strong> del curso? Los registros de asistencia se conservarán.';
  document.getElementById('unenrollModal').classList.add('open');
}
function closeModal() {
  document.getElementById('unenrollModal').classList.remove('open');
}
document.getElementById('unenrollModal').addEventListener('click', function(e){
  if(e.target === this) closeModal();
});

function selectAll() {
  document.querySelectorAll('#studentList input[type="checkbox"]').forEach(function(cb){
    if(cb.closest('.student-check-item').style.display !== 'none') cb.checked = true;
  });
}

function filterStudents() {
  var q = document.getElementById('searchAvailable').value.toLowerCase();
  document.querySelectorAll('#studentList .student-check-item').forEach(function(el){
    el.style.display = el.dataset.name.includes(q) ? '' : 'none';
  });
}

function filterEnrolled() {
  var q = document.getElementById('searchEnrolled').value.toLowerCase();
  document.querySelectorAll('#enrolledTable tbody tr.enrolled-row').forEach(function(el){
    el.style.display = el.dataset.name.includes(q) ? '' : 'none';
  });
}
</script>

</body>
</html>

```

views/academic/course_students.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><?= htmlspecialchars($course['name']) ?> - Estudiantes - EcuAsist</title>
  <style>
    .enrolled-table { width:100%; border-collapse:collapse; }
    .enrolled-table th { background:#f8f9fa; padding:10px 12px; font-size:12px; text-align:left;
border-bottom:2px solid #e0e0e0; }

```

```

.enrolled-table td { padding:9px 12px; font-size:13px; border-bottom:1px solid #f0f0f0;
vertical-align:middle; }
.enrolled-table tbody tr:hover { background:#fafafa; }
.badge-count { background:#e3f2fd; color:#1565c0; padding:3px 10px; border-radius:10px; font-size:12px;
font-weight:700; }
.btn-retirar { padding:4px 10px; font-size:12px; background:#dc3545; color:#fff; border:none;
border-radius:4px; cursor:pointer; }
.btn-retirar:hover { background:#c82333; }
.search-box { width:100%; padding:9px 12px; border:1px solid #ddd; border-radius:6px; font-size:13px;
margin-bottom:12px; }

/* ■■ Modal base ■■ */
.modal-overlay { display:none; position:fixed; inset:0; background:rgba(0,0,0,.5); z-index:9999;
align-items:center; justify-content:center; }
.modal-overlay.open { display:flex; }
.modal-box { background:#fff; border-radius:10px; padding:28px; width:90%; box-shadow:0 8px 32px
rgba(0,0,0,.2); }
.modal-header { display:flex; justify-content:space-between; align-items:center; margin-bottom:16px; }
.modal-header h3 { margin:0; font-size:16px; }
.modal-close { background:none; border:none; font-size:20px; cursor:pointer; color:#999; line-height:1; }
.modal-actions { display:flex; gap:10px; justify-content:flex-end; margin-top:20px; }

/* ■■ Modal matricular ■■ */
#modalEnroll .modal-box { max-width:480px; }
.student-check-item { display:flex; align-items:center; gap:10px; padding:8px 10px; border:1px solid
#e8e8e8; border-radius:6px; margin-bottom:5px; cursor:pointer; transition:background .12s; }
.student-check-item:hover { background:#f0f7ff; border-color:#90caf9; }
.student-check-item input[type="checkbox"] { width:15px; height:15px; cursor:pointer; flex-shrink:0; }
.student-check-item label { margin:0; font-weight:normal; font-size:13px; cursor:pointer; flex:1; }
.student-scroll { max-height:280px; overflow-y:auto; padding-right:2px; }
.enroll-toolbar { display:flex; justify-content:space-between; align-items:center; margin-bottom:10px; }
.counter-sel { font-size:12px; color:#555; background:#f0f0f0; padding:3px 10px; border-radius:10px; }

/* ■■ Modal confirmar retirar ■■ */
#modalConfirmUnenroll .modal-box { max-width:420px; }
#modalConfirmUnenroll h3 { color:#dc3545; }

/* ■■ Alertas ■■ */
.flash-success { background:#d4edda; color:#155724; padding:12px 16px; border-radius:6px;
margin-bottom:16px; font-size:14px; }
.flash-warning { background:#fff3cd; color:#856404; padding:12px 16px; border-radius:6px;
margin-bottom:16px; font-size:14px; }

/* ■■ Empty state ■■ */
.empty-state { text-align:center; padding:36px 20px; color:#aaa; }
.empty-state .es-icon { font-size:42px; margin-bottom:10px; }
</style>
</head>
<body>

<?php include BASE_PATH . '/views/partials/navbar.php'; ?>

<div class="breadcrumb">
<a href="?action=dashboard">■ Inicio</a> &rsquo;
<a href="?action=academic">Configuración Académica</a> &rsquo;
<?= htmlspecialchars($course['name']) ?> - Estudiantes
</div>

<div class="container">

<?php if(isset($_GET['enrolled']) && (int)$_GET['enrolled'] > 0): ?>
<div class="flash-success">✓ <?= (int)$_GET['enrolled'] ?> estudiante(s) matriculado(s) correctamente<?=
isset($_GET['errors']) && (int)$_GET['errors'] > 0 ? ' ' . '■ ' . (int)$_GET['errors'] . ' ya estaban en otro
curso' : '' ?></div>
<?php elseif(isset($_GET['errors']) && (int)$_GET['errors'] > 0): ?>
<div class="flash-warning">■ <?= (int)$_GET['errors'] ?> estudiante(s) ya estaban matriculados en otro
curso</div>
<?php endif; ?>
<?php if(isset($_GET['unenrolled'])): ?>
<div class="flash-success">✓ Estudiante retirado correctamente</div>
<?php endif; ?>
<?php if(isset($_GET['error'])): ?>

```

```

<div class="flash-warning">X Error: <?= htmlspecialchars($_GET['error']) ?></div>
<?php endif; ?>

<div class="page-header" style="background:linear-gradient(135deg,#1565c0,#1976d2);">
<div class="ph-icon">■</div>
<div>
<h1><?= htmlspecialchars($course['name']) ?></h1>
<p>Jornada: <?= ucfirst($course['shift_name']) ?> ' ' ?> &nbsp;|&nbsp; Gestión de matrículas</p>
</div>
</div>

```

```

<div class="panel">
<div style="display:flex; justify-content:space-between; align-items:center; margin-bottom:16px;">
<span style="font-size:15px; font-weight:700; color:#333;">
■ Estudiantes Matriculados
<span class="badge-count"><?= count($students) ?></span>
</span>
<?php if(count($availableStudents) > 0): ?>
<button class="btn btn-success" onclick="openEnrollModal()" style="font-size:13px; padding:7px 16px;">
■ Matricular estudiantes
</button>
<?php else: ?>
<span style="font-size:12px; color:#999;">Todos los estudiantes ya están matriculados</span>
<?php endif; ?>
</div>

<?php if(count($students) > 0): ?>
<div style="margin-bottom:10px;">
<input type="text" id="searchEnrolled" class="search-box" placeholder="■ Buscar estudiante matriculado..."
oninput="filterEnrolled()">
</div>
<table class="enrolled-table" id="enrolledTable">
<thead>
<tr>
<th style="width:40px">#</th>
<th>Apellidos y Nombres</th>
<th>Cédula</th>
<th>F. Matrícula</th>
<th style="text-align:center; width:90px">Acción</th>
</tr>
</thead>
<tbody>
<?php $n = 1; foreach($students as $s): ?>
<tr data-name="<?= strtolower($s['last_name']).' '.$s['first_name']) ?>">
<td><?= $n++ ?></td>
<td><strong><?= htmlspecialchars($s['last_name']).' '.$s['first_name']) ?></strong></td>
<td><?= $s['dni'] ?> '-' ?></td>
<td><?= !empty($s['enrollment_date']) ? date('d/m/Y', strtotime($s['enrollment_date'])) : '-' ?></td>
<td style="text-align:center;">
<button class="btn btn-retirar"
onclick="confirmUnenroll(<?= $s['id'] ?>, '<?= addslashes($s['last_name']).' '.$s['first_name']) ?>')">

```

```

X Retirar
</button>
</td>
</tr>
<?php endforeach; ?>
</tbody>
</table>
<?php else: ?>
<div class="empty-state">
<div class="es-icon">■</div>
<p>Ningún estudiante matriculado en este curso aún.</p>
<?php if(count($availableStudents) > 0): ?>
<button class="btn btn-success" onclick="openEnrollModal()" style="margin-top:14px; font-size:13px;">
■ Matricular estudiantes
</button>
<?php endif; ?>
</div>
<?php endif; ?>
</div>

```

```

<div style="margin-top:16px;">
<a href="?action=academic" class="btn btn-outline">← Volver a Configuración Académica</a>
</div>
</div>

<!-- ■■ MODAL: Matricular ■■ -->
<div class="modal-overlay" id="modalEnroll">
<div class="modal-box">
<div class="modal-header">
<h3>■ Matricular en <?= htmlspecialchars($course['name']) ?></h3>
<button class="modal-close" onclick="closeEnrollModal()">X</button>
</div>

<input type="text" class="search-box" id="searchAvailable" placeholder="■ Buscar estudiante disponible..."
oninput="filterAvailable()">

<div class="enroll-toolbar">
<button type="button" class="btn btn-outline" style="font-size:12px; padding:5px 12px;"
onclick="selectAllAvailable()">
Seleccionar todos
</button>
<span class="counter-sel" id="selectedCounter">0 seleccionados</span>

</div>

<form method="POST" action="?action=enroll_students" id="enrollForm">
<input type="hidden" name="course_id" value="<?= $course['id'] ?>">
<input type="hidden" name="redirect_to" value="view_course_students">

<div class="student-scroll" id="availableList">
<?php if(count($availableStudents) > 0): ?>
<?php foreach($availableStudents as $s): ?>
<div class="student-check-item" data-name="<?= strtolower($s['last_name']. ' '.$s['first_name']) ?>">
<input type="checkbox" name="student_ids[]" value="<?= $s['id'] ?>" id="av<?= $s['id'] ?>"
onchange="updateCounter()">
<label for="av<?= $s['id'] ?>">
<strong><?= htmlspecialchars($s['last_name']. ' '.$s['first_name']) ?></strong>
<?php if(!empty($s['dni'])): ?>
<span style="color:#999; font-size:11px;"> · <?= $s['dni'] ?></span>
<?php endif; ?>
</label>
</div>
<?php endforeach; ?>
<?php else: ?>
<p style="text-align:center; color:#999; padding:20px 0; font-size:13px;">No hay estudiantes disponibles
para matricular.</p>
<?php endif; ?>
</div>

<div class="modal-actions">
<button type="button" class="btn btn-outline" onclick="closeEnrollModal()">Cancelar</button>
<button type="submit" class="btn btn-success" id="btnEnrollSubmit" disabled style="font-size:13px;">
✓ Matricular seleccionados
</button>
</div>
</form>
</div>
</div>

<!-- ■■ MODAL: Confirmar Retirar ■■ -->
<div class="modal-overlay" id="modalConfirmUnenroll">
<div class="modal-box">
<div class="modal-header">
<h3>■■ Retirar Estudiante</h3>
<button class="modal-close" onclick="closeUnenrollModal()">X</button>

</div>

<p id="unenrollMsg" style="color:#555; font-size:14px; margin-bottom:6px;"></p>
<p style="color:#888; font-size:12px;">Los registros de asistencia se conservarán.</p>
<form method="POST" action="?action=unenroll_student" id="unenrollForm">
<input type="hidden" name="student_id" id="unenrollStudentId">

```

```

<input type="hidden" name="course_id" value="<?= $course['id'] ?>">
<input type="hidden" name="redirect_to" value="view_course_students">
<div class="modal-actions">
<button type="button" class="btn btn-outline" onclick="closeUnenrollModal()">Cancelar</button>
<button type="submit" class="btn btn-danger">Sí, Retirar</button>
</div>
</form>
</div>
</div>

<script>
/* ■■ Modal Matricular ■■ */
function openEnrollModal() {
    document.getElementById('modalEnroll').classList.add('open');
    document.getElementById('searchAvailable').focus();
}
function closeEnrollModal() {
    document.getElementById('modalEnroll').classList.remove('open');
}
document.getElementById('modalEnroll').addEventListener('click', function(e) {
    if (e.target === this) closeEnrollModal();
});

function filterAvailable() {
    var q = document.getElementById('searchAvailable').value.toLowerCase();
    document.querySelectorAll('#availableList .student-check-item').forEach(function(el) {
        el.style.display = el.dataset.name.includes(q) ? '' : 'none';
    });
}

function selectAllAvailable() {
    document.querySelectorAll('#availableList .student-check-item').forEach(function(el) {
        if (el.style.display !== 'none') {
            el.querySelector('input[type="checkbox"]').checked = true;
        }
    });
}

function updateCounter() {
}

function updateCounter() {
    var n = document.querySelectorAll('#availableList input[type="checkbox"]:checked').length;
    document.getElementById('selectedCounter').textContent = n + ' seleccionado' + (n !== 1 ? 's' : '');
    document.getElementById('btnEnrollSubmit').disabled = n === 0;
}

/* ■■ Modal Retirar ■■ */
function confirmUnenroll(studentId, name) {
    document.getElementById('unenrollStudentId').value = studentId;
    document.getElementById('unenrollMsg').innerHTML = '¿Retirar a <strong>' + name + '</strong> del curso?';
    document.getElementById('modalConfirmUnenroll').classList.add('open');
}
function closeUnenrollModal() {
    document.getElementById('modalConfirmUnenroll').classList.remove('open');
}
document.getElementById('modalConfirmUnenroll').addEventListener('click', function(e) {
    if (e.target === this) closeUnenrollModal();
});

/* ■■ Filtro tabla matriculados ■■ */
function filterEnrolled() {
    var q = document.getElementById('searchEnrolled').value.toLowerCase();
    document.querySelectorAll('#enrolledTable tbody tr').forEach(function(tr) {
        tr.style.display = tr.dataset.name.includes(q) ? '' : 'none';
    });
}

/* ■■ Auto-abrir modal si venimos de matricular con ?enrolled ■■ */
<?php if(isset($_GET['enrolled']) && (int)$_GET['enrolled'] === 0 && isset($_GET['errors'])): ?>
    // No hacer nada especial
<?php endif; ?>
</script>

```



```
</body>
</html>
```

views/academic/course_subject_edit.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>Editar Asignatura - EcuAsist</title>
  <style>
    * { margin: 0; padding: 0; box-sizing: border-box; }
    body { font-family: Arial, sans-serif; background: #f4f4f4; }
    .container { max-width: 500px; margin: 40px auto; padding: 0 20px; }
    .card { background: white; padding: 30px; border-radius: 8px; box-shadow: 0 2px 10px rgba(0,0,0,0.1); }
    .card h2 { margin-bottom: 24px; color: #333; }
    .form-group { margin-bottom: 18px; }
    label { display: block; margin-bottom: 6px; font-weight: bold; color: #555; font-size: 14px; }
    input { width: 100%; padding: 10px 12px; border: 1px solid #ddd; border-radius: 4px; font-size: 14px; }
    input:focus { border-color: #007bff; outline: none; box-shadow: 0 0 2px rgba(0,123,255,.2); }
    .btn-row { display: flex; gap: 10px; margin-top: 24px; }
    .btn { padding: 10px 20px; border: none; border-radius: 4px; cursor: pointer; font-size: 14px;
text-decoration: none; display: inline-block; }
    .btn-save { background: #28a745; color: white; }
    .btn-back { background: #6c757d; color: white; }
    .btn:hover { opacity: 0.88; }
  </style>
</head>
<body>

<?php include BASE_PATH . '/views/partials/navbar.php'; ?>

<div class="breadcrumb">
  ■ Inicio &rsquo;
  <a href="?action=academic">Configuración Académica</a> &rsquo;
  <a href="?action=course_subjects&course_id=?= $courseId ?">Asignaturas del Curso</a> &rsquo;
  Editar Asignatura
</div>

<div class="container">
  <div class="card">
    <h2>■ Editar Asignatura</h2>

    <form method="POST" action="?action=edit_course_subject&subject_id=?= $subjectId ?&course_id=?= $courseId ?">

<div class="form-group">
  <label>Nombre de la asignatura *</label>
  <input type="text" name="name"
value="<?= htmlspecialchars($subject['name']) ?>"
required autofocus>
</div>
<div class="form-group">
  <label>Código <span style="font-weight:normal;color:#999;">(opcional)</span></label>
  <input type="text" name="code"
value="<?= htmlspecialchars($subject['code']) ?? ' ' ?>"
placeholder="Ej: MAT">
</div>
<div class="btn-row">
  <button type="submit" class="btn btn-save">■ Guardar cambios</button>
  <a href="?action=course_subjects&course_id=?= $courseId ?" class="btn btn-back">← Cancelar</a>
</div>
</form>
</div>
</div>
```

```
</body>
</html>
```

views/academic/course_subjects.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>Asignaturas del Curso - EcuAsist</title>
    <style>
        * { margin: 0; padding: 0; box-sizing: border-box; }
        body { font-family: Arial, sans-serif; background: #f4f4f4; }
        .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; }
        .course-header { background: linear-gradient(135deg, #007bff, #0056b3); color: white; padding: 20px 30px;
border-radius: 8px; margin-bottom: 20px; }
        .course-header h2 { font-size: 22px; margin-bottom: 6px; }
        .course-header p { font-size: 14px; opacity: 0.9; }
        table { width: 100%; border-collapse: collapse; margin-top: 10px; }
        th, td { padding: 11px 13px; text-align: left; border-bottom: 1px solid #eee; font-size: 14px; }
        th { background: #f8f9fa; font-weight: bold; color: #555; }
        tr:hover td { background: #f0f7ff; }
        .btn { display: inline-block; padding: 5px 12px; border-radius: 4px; border: none; cursor: pointer;
font-size: 12px; text-decoration: none; white-space: nowrap; }
        .btn-edit { background: #ffc107; color: #212529; }
        .btn-delete { background: #dc3545; color: white; }
        .btn-back { background: #6c757d; color: white; padding: 8px 18px; font-size: 14px; }
        .btn-add { background: #28a745; color: white; padding: 10px 20px; font-size: 14px; }
        .btn-assign { background: #e65100; color: white; }
        .btn-unassign { background: #6c757d; color: white; }
        .btn:hover { opacity: 0.88; }
        .alert { padding: 12px 16px; border-radius: 4px; margin-bottom: 16px; font-size: 14px; }
        .alert-success { background: #d4edda; color: #155724; }
        .alert-error { background: #f8d7da; color: #721c24; }
        .form-row { display: flex; gap: 12px; align-items: flex-end; flex-wrap: wrap; }
        .form-group { display: flex; flex-direction: column; flex: 1; min-width: 180px; }
        .form-group label { font-size: 13px; font-weight: bold; color: #555; margin-bottom: 5px; }
        .form-group input, .form-group select { padding: 9px 12px; border: 1px solid #ddd; border-radius: 4px;
font-size: 14px; }
        .badge-count { background: #007bff; color: white; border-radius: 12px; padding: 2px 10px; font-size: 13px;
margin-left: 8px; }
        .empty-msg { text-align: center; color: #999; padding: 30px; }
        .teacher-ok { color: #2e7d32; font-weight: bold; font-size: 13px; }
        .teacher-no { color: #bbb; font-style: italic; font-size: 13px; }
        .modal-overlay { display:none; position:fixed; top:0; left:0; width:100%; height:100%;
background:rgba(0,0,0,.5); align-items:center; justify-content:center; z-index:9999; }
        .modal-overlay.active { display:flex; }

        .modal-box { background:white; padding:28px; border-radius:8px; max-width:420px; width:90%; box-shadow:0 4px
20px rgba(0,0,0,.3); }
        .modal-box h3 { margin-bottom:14px; color:#e65100; }
        .modal-box select { width:100%; padding:10px; border:1px solid #ddd; border-radius:4px; font-size:14px;
margin-bottom:18px; }
        .modal-btns { display:flex; gap:10px; justify-content:flex-end; }
        .modal-btns button { padding:8px 18px; border:none; border-radius:4px; cursor:pointer; font-size:14px; }
        .modal-box input:focus {
outline: none;
border-color: #28a745 !important;
box-shadow: 0 0 0 3px rgba(40, 167, 69, 0.1);
}
        .btn-add {
background: #28a745;
color: white;
border: none;
border-radius: 6px;
```

```

cursor: pointer;
transition: background 0.15s;
}
.btn-add:hover {
background: #218838;
}
</style>
</head>
<body>

<?php include BASE_PATH . '/views/partials/navbar.php'; ?>

<div class="breadcrumb">
  ■ Inicio &rsquo; <a href="?action=academic">Configuración Académica</a> &rsquo; Asignaturas del Curso
</div>

<div class="container">

  <div class="course-header">
    <h2>■ <?= htmlspecialchars($course['name']) ?></h2>
    <p>
      <strong>Nivel:</strong> <?= htmlspecialchars($course['grade_level']) ?> &nbsp;|&nbsp;
      <strong>Paralelo:</strong> <?= htmlspecialchars($course['parallel']) ?> &nbsp;|&nbsp;
      <strong>Jornada:</strong> <?= ucfirst($course['shift_name']) ?> &nbsp;|&nbsp;
      <strong>Año Lectivo:</strong> <?= htmlspecialchars($course['year_name']) ?>
    </p>
  </div>

  <!-- Sistema de Toast -->
  <div id="toast-container" style="position:fixed;top:20px;right:20px;z-index:9999;display:flex;flex-direction:column;gap:10px:pointer-events:none;"></div>

  <!-- Mensajes convertidos a Toast -->
  <?php if (isset($_GET['added'])): ?>
  <script>document.addEventListener('DOMContentLoaded', function() { showToast('■ Asignatura agregada correctamente', 'success'); });</script>
  <?php endif; ?>

  <?php if (isset($_GET['updated'])): ?>
  <script>document.addEventListener('DOMContentLoaded', function() { showToast('✓ Horas de asignatura actualizadas', 'success'); });</script>
  <?php endif; ?>

  <?php if (isset($_GET['removed'])): ?>
  <script>document.addEventListener('DOMContentLoaded', function() { showToast('■ Asignatura quitada del curso', 'warning'); });</script>
  <?php endif; ?>

  <?php if (isset($_GET['assigned'])): ?>
  <script>document.addEventListener('DOMContentLoaded', function() { showToast('■ Docente asignado correctamente', 'success'); });</script>
  <?php endif; ?>

  <?php if (isset($_GET['unassigned'])): ?>
  <script>document.addEventListener('DOMContentLoaded', function() { showToast('■ Docente removido', 'info'); });</script>
  <?php endif; ?>

  <?php if (isset($_GET['error']) && $_GET['error'] === 'has_teacher'): ?>
  <script>document.addEventListener('DOMContentLoaded', function() { showToast('■ No se puede quitar: esta asignatura tiene un docente asignado', 'error'); });</script>
  <?php endif; ?>

  <?php if (isset($_GET['assign_error'])): ?>
  <script>document.addEventListener('DOMContentLoaded', function() { showToast('■ <?= htmlspecialchars(addslashes($_GET['assign_error'])) ?>', 'error'); });</script>
  <?php endif; ?>

  <?php if (isset($_GET['error']) && $_GET['error'] == 'hours_exceeded'): ?>
  <script>document.addEventListener('DOMContentLoaded', function() { showToast('■ <?= htmlspecialchars(addslashes($_GET['error'])) ?> Error al asignar horas', 'error'); });</script>
  <?php endif; ?>

```

```

<?php endif; ?>

<div class="card">

<div style="display:flex; justify-content:space-between; align-items:center; margin-bottom:20px;">
  <h3 style="color:#333; margin:0;">
    Asignaturas y Docentes
    <span class="badge-count"><?= count($subjects) ?></span>
  </h3>
  <button onclick="abrirModalAgregar()" class="btn btn-add" style="padding:10px 20px; font-size:14px;">
    ■ Agregar asignatura
  </button>
</div>

<?php if (empty($subjects)): ?>
<div class="empty-msg">■ Este curso aún no tiene asignaturas asignadas.</div>
<?php else: ?>
<div class="table-info" style="padding:10px 0;">
<div id="total-hours-indicator" style="padding:10px;border-radius:6px;font-weight:bold;"></div>
</div>
<table>
<thead>
<tr>
<th>#</th>
<th>Asignatura</th>
<th>Código</th>
<th style="text-align:center;">Hrs/semana</th>
<th>Docente</th>
<th style="width:220px; text-align:center;">Acciones</th>
</tr>
</thead>
<tbody>
<?php foreach ($subjects as $i => $sub): ?>
<tr>
<td><?= $i + 1 ?></td>
<td><strong><?= htmlspecialchars($sub['name']) ?></strong></td>
<td><?= htmlspecialchars($sub['code']) ?>: '<span style="color:#bbb;">--</span>' ?></td>
<td style="text-align:center;">
<form method="POST" action="?action=set_subject_hours&course_id=<?= $course['id'] ?>"
style="display:inline-flex;align-items:center;gap:4px;">
  <input type="hidden" name="subject_id" value="<?= $sub['id'] ?>">
  <input type="number" name="hours_per_week" value="<?= (int)($sub['hours_per_week'] ?? 1) ?>"
min="1" max="20" style="width:52px;padding:4px 6px;border:1px solid
#ddd;border-radius:4px;font-size:13px;text-align:center;">
  <button type="submit" style="padding:4px 8px;background:#007bff;color:white;border:none;border-radius:4px;font-size:11px;cursor:pointer;">✓</button>
</form>

</td>
<td>
<?php if (!empty($sub['teacher_name'])): ?>
<span class="teacher-ok">■ <?= htmlspecialchars($sub['teacher_name']) ?></span>
<?php else: ?>
<span class="teacher-no">Sin docente</span>
<?php endif; ?>
</td>
<td style="text-align:center; white-space:nowrap;">
<button class="btn btn-assign"
onclick="abrirModal(<?= $sub['id'] ?>, '<?= htmlspecialchars(addslashes($sub['name'])) ?>', <?=
$sub['assignment_id'] ?? 'null' ?>, <?= $sub['teacher_id'] ?? 'null' ?>)">
<?= empty($sub['teacher_name']) ? '■' : '■■' ?>
</button>

<?php if (!empty($sub['teacher_name'])): ?>
<form method="POST" action="?action=unassign_subject_teacher" style="display:inline;">
<input type="hidden" name="assignment_id" value="<?= $sub['assignment_id'] ?>">
<input type="hidden" name="course_id" value="<?= $course['id'] ?>">
<button type="submit" class="btn btn-unassign"
onclick="return confirm('<?php echo '<?php endif; ?>

```

```
<a href="?action=edit_course_subject&subject_id=<?= $sub['id'] ?>&course_id=<?= $course['id'] ?>"
class="btn btn-edit">■</a>
```

```
<form method="POST" action="?action=remove_course_subject" style="display:inline;"
onsubmit="return confirmRemove(event, '<?= htmlspecialchars(addslashes($sub['name'])) ?>')">
<input type="hidden" name="course_id" value="<?= $course['id'] ?>">
<input type="hidden" name="subject_id" value="<?= $sub['id'] ?>">
<button type="submit" class="btn btn-delete">■</button>
</form>
</td>
</tr>
<?php endforeach; ?>
</tbody>
</table>
<?php endif; ?>
</div>
```

```
<!-- Modal: Agregar Asignatura -->
```

```
<div class="modal-overlay" id="modalAgregarAsignatura">
<div class="modal-box" style="max-width:500px;">
<h3 style="color:#28a745; margin-bottom:16px;">■ Agregar nueva asignatura</h3>
<form method="POST" action="?action=course_subjects&course_id=<?= $course['id'] ?>"
id="formAgregarAsignatura">
<div style="margin-bottom:20px;">
<label style="font-size:13px; font-weight:bold; color:#555; display:block; margin-bottom:6px;">Nombre de la
asignatura *</label>
<input type="text" name="new_subject_name" id="new_subject_name" class="form-control"
placeholder="Ej: Emprendimiento" required
style="width:100%; padding:10px 12px; border:1.5px solid #ddd; border-radius:6px; font-size:14px;">
</div>

<div style="margin-bottom:20px;">
<label style="font-size:13px; font-weight:bold; color:#555; display:block; margin-bottom:6px;">
Código <span style="font-weight:normal; color:#999;">(opcional)</span>
</label>
<input type="text" name="new_subject_code" id="new_subject_code" class="form-control"
placeholder="Ej: EMP"
style="width:100%; padding:10px 12px; border:1.5px solid #ddd; border-radius:6px; font-size:14px;">
<small style="color:#888; display:block; margin-top:4px;">Código corto para identificar la
asignatura</small>
</div>

<div class="modal-btns" style="justify-content:flex-end; margin-top:24px;">
<button type="button" onclick="cerrarModalAgregar()" style="background:#6c757d; color:white; padding:10px
20px; border:none; border-radius:6px; cursor:pointer; font-size:14px;">Cancelar</button>
<button type="submit" style="background:#28a745; color:white; padding:10px 20px; border:none;
border-radius:6px; cursor:pointer; font-size:14px; font-weight:600;">✓ Agregar asignatura</button>
</div>
</form>
</div>
</div>
```

```
<!-- Modal: Asignar Docente -->
```

```
<div class="modal-overlay" id="modalAsignar">
<div class="modal-box">
<h3>■ Asignar Docente</h3>
<p id="modalSubjectLabel" style="font-size:15px; color:#333; margin-bottom:14px;"></p>
<form method="POST" action="?action=assign_subject_teacher">
<input type="hidden" name="course_id" value="<?= $course['id'] ?>">
<input type="hidden" name="subject_id" id="modalSubjectId">
<input type="hidden" name="assignment_id" id="modalAssignmentId">
<label style="font-size:13px; font-weight:bold; color:#555; display:block; margin-bottom:6px;">Docente
*</label>
```

```
<select name="teacher_id" id="modalTeacherSelect" required>
<option value="">Seleccionar docente...</option>
<?php foreach ($teachers as $t): ?>
<option value="<?= $t['id'] ?>">
<?= htmlspecialchars($t['last_name'] . ' ' . $t['first_name']) ?>
</option>
<?php endforeach; ?>
</select>
```

```

<div class="modal-btns">
  <button type="button" onclick="cerrarModal()" style="background:#6c757d;color:white;">Cancelar</button>
  <button type="submit" style="background:#e65100;color:white;">✓ Guardar</button>
</div>
</form>
</div>
</div>

  <a href="?action=academic" class="btn btn-back">← Volver a Configuración Académica</a>
</div>

<!-- Sistema de Toast Script -->
<script>
function showToast(message, type = 'success') {
  const container = document.getElementById('toast-container');
  if (!container) return;

  const toast = document.createElement('div');

  const styles = {
    success: { bg: '#d4edda', color: '#155724', border: '#c3e6cb', icon: '■' },
    error: { bg: '#f8d7da', color: '#721c24', border: '#f5c6cb', icon: '■' },
    warning: { bg: '#fff3cd', color: '#856404', border: '#ffeeba', icon: '■' },
    info: { bg: '#d1ecf1', color: '#0c5460', border: '#bee5eb', icon: '■' }
  };

  const s = styles[type] || styles.success;

  toast.style.cssText = `
min-width: 300px;
max-width: 400px;
background: ${s.bg};

color: ${s.color};
border: 1px solid ${s.border};
border-left: 5px solid ${s.color};
border-radius: 8px;
padding: 14px 18px;
font-size: 14px;
font-weight: 500;
box-shadow: 0 4px 16px rgba(0,0,0,0.15);
display: flex;
align-items: center;
gap: 12px;
pointer-events: all;
opacity: 0;
transform: translateX(40px);
transition: opacity 0.3s ease, transform 0.3s ease;
`;

  toast.innerHTML = `
<span style="font-size:18px;">${s.icon}</span>
<span style="flex:1;">${message}</span>
<button onclick="this.parentElement.remove()" style="background:none;border:none;color:${s.color};font-size:16px;cursor:pointer;opacity:0.6;padding:0;">X</button>
`;

  container.appendChild(toast);

  setTimeout(() => {
    toast.style.opacity = '1';
    toast.style.transform = 'translateX(0)';
  }, 10);

  setTimeout(() => {
    toast.style.opacity = '0';
    toast.style.transform = 'translateX(40px)';
    setTimeout(() => toast.remove(), 300);
  }, 4000);
}

function abrirModal(subjectId, subjectName, assignmentId, teacherId) {

```

```

document.getElementById('modalSubjectId').value = subjectId;
document.getElementById('modalSubjectLabel').textContent = '■ ' + subjectName;

document.getElementById('modalAssignmentId').value = assignmentId ?? '';
const sel = document.getElementById('modalTeacherSelect');
sel.value = teacherId ?? '';
document.getElementById('modalAsignar').classList.add('active');
}

function cerrarModal() {
  document.getElementById('modalAsignar').classList.remove('active');
}

document.getElementById('modalAsignar').addEventListener('click', function(e) {
  if (e.target === this) cerrarModal();
});

function confirmRemove(event, nombre) {
  event.preventDefault();
  const m = document.createElement('div');
  m.style.cssText = 'position:fixed;top:0;left:0;width:100%;height:100%;background:rgba(0,0,0,.5);display:flex;
align-items:center;justify-content:center;z-index:9999;';
  m.innerHTML = `<div style="background:white;padding:28px;border-radius:8px;max-width:420px;width:90%;box-sha
dow:0 4px 20px rgba(0,0,0,.3);">
<h3 style="color:#dc3545;margin-bottom:12px;">■ ■ Quitar asignatura</h3>
<p style="color:#555;margin-bottom:18px;">¿Quitar <strong>${nombre}</strong> de este curso?</p>
<div style="display:flex;gap:10px;justify-content:flex-end;">
  <button id="bC" style="padding:8px 18px;background:#6c757d;color:white;border:none;border-radius:4px;cursor:
pointer;">Cancelar</button>
  <button id="bO" style="padding:8px 18px;background:#dc3545;color:white;border:none;border-radius:4px;cursor:
pointer;">Sí, Quitar</button>
</div></div>`;
  document.body.appendChild(m);
  const form = event.target;
  document.getElementById('bO').onclick = () => { document.body.removeChild(m); form.submit(); };
  document.getElementById('bC').onclick = () => document.body.removeChild(m);
  return false;
}

document.addEventListener('DOMContentLoaded', function() {
  calcularTotalHoras();
});

function calcularTotalHoras() {
  let total = 0;
  const horasDisponibles = <?= $this->courseModel->getTotalWeeklyHoursAvailable($course['id']) ?>;
  const horasPorDia = <?= $this->courseModel->getMaxHoursPerDay($course['id']) ?>;

  const diasLaborables = <?= $this->courseModel->getWorkingDaysCount() ?>;

  document.querySelectorAll('input[name="hours_per_week"]').forEach(function(input) {
    total += parseInt(input.value) || 0;
  });

  let indicator = document.getElementById('total-hours-indicator');
  if (!indicator) {
    indicator = document.createElement('div');
    indicator.id = 'total-hours-indicator';
    indicator.style.marginTop = '10px';
    indicator.style.padding = '10px';
    indicator.style.borderRadius = '6px';
    indicator.style.fontWeight = 'bold';

    const tableInfo = document.querySelector('.table-info');
    if (tableInfo) {
      tableInfo.appendChild(indicator);
    }
  }

  indicator.textContent = `Total horas asignadas: ${total} de ${horasDisponibles} disponibles ` +
  `(${horasPorDia} horas/día × ${diasLaborables} días)`;

```

```

if (total > horasDisponibles) {
    indicator.style.backgroundColor = '#f8d7da';
    indicator.style.color = '#721c24';
    indicator.style.border = '1px solid #f5c6cb';
} else if (total === horasDisponibles) {
    indicator.style.backgroundColor = '#d4edda';
    indicator.style.color = '#155724';
    indicator.style.border = '1px solid #c3e6cb';
} else {
    indicator.style.backgroundColor = '#fff3cd';
    indicator.style.color = '#856404';
    indicator.style.border = '1px solid #ffeeba';
}
}

document.querySelectorAll('input[name="hours_per_week"]').forEach(function(input) {

input.addEventListener('change', function() {
    const maxHorasDisponibles = <?=$this->courseModel->getTotalWeeklyHoursAvailable($course['id']) ?>;
    let totalActual = 0;

    document.querySelectorAll('input[name="hours_per_week"]').forEach(function(inp) {
        if (inp !== input) {
            totalActual += parseInt(inp.value) || 0;
        }
    });

    const nuevasHoras = parseInt(this.value) || 0;
    const maxPermitido = maxHorasDisponibles - totalActual;

    if (nuevasHoras > maxPermitido) {
        showToast('■ No puedes asignar ' + nuevasHoras + ' horas. Te quedan solo ' + maxPermitido + ' horas disponibles.', 'warning');
        this.value = maxPermitido;
    }

    calcularTotalHoras();
});

});

function abrirModalAgregar() {
    document.getElementById('modalAgregarAsignatura').classList.add('active');
    document.getElementById('new_subject_name').focus();
}

function cerrarModalAgregar() {
    document.getElementById('modalAgregarAsignatura').classList.remove('active');
    document.getElementById('new_subject_name').value = '';
    document.getElementById('new_subject_code').value = '';
}

document.getElementById('modalAgregarAsignatura').addEventListener('click', function(e) {
    if (e.target === this) cerrarModalAgregar();
});

document.getElementById('formAgregarAsignatura').addEventListener('submit', function(e) {
    const nombre = document.getElementById('new_subject_name').value.trim();
    if (nombre === '') {
        e.preventDefault();
        showToast('■ El nombre de la asignatura es obligatorio', 'error');
    }
});
</script>

</body>
</html>

```



```

<?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>
```

```
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"></script>
</body>
</html>
```

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>
            </div>
        </div>
    </div>
```

```

</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>

```

```

<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>

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"></script>
</body>
</html>

```

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; }
    .container { max-width: 1200px; margin: 24px auto; padding: 0 16px; }
    .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>

<?php include BASE_PATH . '/views/partials/navbar.php'; ?>

<div class="breadcrumb">
  <a href="?action=dashboard">■ Inicio</a> &rsquo;
  <a href="?action=academic">Configuración Académica</a> &rsquo;
  Matricular Estudiantes
</div>

<div class="container">

<div class="page-header" style="background: linear-gradient(135deg, #2e7d32, #388e3c);">
  <div class="ph-icon">■</div>
  <div>
    <h1>Matricular Estudiantes</h1>
    <p>Curso: <strong><?= htmlspecialchars($course['name']) ?></strong> &nbsp;&nbsp;&nbsp; Jornada: <?=
ucfirst($course['shift_name']) ?? '' ?></p>
  </div>
</div>
<?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" action="?action=enroll_students">
<input type="hidden" name="course_id" value="<?= $course['id'] ?>">

<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'] ?>">
<input type="hidden" name="course_id" value="<? $course['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>

```

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; }
    .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; }
/* ■■■ Toast ■■■ */
#ac-toast{
position:fixed;top:16px;right:16px;z-index:99999;
padding:11px 20px;border-radius:9px;font-size:13px;font-weight:600;
box-shadow:0 4px 20px rgba(0,0,0,.2);
opacity:0;transform:translateY(-6px);transition:all .22s;
pointer-events:none;max-width:380px;line-height:1.4;
}
#ac-toast.show{opacity:1;transform:translateY(0);}
#ac-toast.ok {background:#0d3b2e;color:#06d6a0;}
#ac-toast.err {background:#3b0d18;color:#ef476f;}
#ac-toast.inf {background:#1a2035;color:#fff;}
h2 { margin-bottom: 20px; color: #333; }

```

```

/* ■■■ Modal base ■■■ */
.modal-overlay {
display: none; position: fixed; inset: 0;
background: rgba(0,0,0,.55); z-index: 9999;
align-items: center; justify-content: center;
}
.modal-overlay.active { display: flex; }
.modal-box {
background: #fff; border-radius: 12px;
width: 90%; max-width: 540px; max-height: 90vh;
display: flex; flex-direction: column;
box-shadow: 0 10px 40px rgba(0,0,0,.25);
animation: modalIn .18s ease;
}
@keyframes modalIn { from { transform: translateY(-18px); opacity: 0; } to { transform: translateY(0);
opacity: 1; } }
.modal-header {
padding: 20px 24px 16px;
border-bottom: 1px solid #f0f0f0;
display: flex; justify-content: space-between; align-items: center;
flex-shrink: 0;
}
.modal-header h3 { margin: 0; font-size: 17px; color: #1a237e; }
.modal-header p { margin: 4px 0 0; font-size: 12px; color: #888; }
.modal-close { background: none; border: none; font-size: 22px; cursor: pointer; color: #aaa; margin: 0;
padding: 0; line-height: 1; }
.modal-close:hover { color: #333; background: none; }
.modal-body { padding: 20px 24px; overflow-y: auto; flex: 1; }
.modal-footer {
padding: 14px 24px; border-top: 1px solid #f0f0f0;
display: flex; justify-content: flex-end; gap: 10px; flex-shrink: 0;
}
.modal-footer .btn-cancel {
padding: 9px 20px; background: #f5f5f5; color: #555;
border: 1px solid #ddd; border-radius: 6px; cursor: pointer; margin: 0; font-size: 14px;
}
.modal-footer .btn-cancel:hover { background: #e8e8e8; }
.modal-footer .btn-submit {
padding: 9px 22px; background: linear-gradient(135deg,#1a237e,#283593);
color: white; border: none; border-radius: 6px; cursor: pointer;
margin: 0; font-size: 14px; font-weight: 600;
}

```

```

.modal-footer .btn-submit:hover { opacity: .88; }

```

```

/* ■■■ Form fields inside modal ■■■ */
.mf-label { font-size: 12px; font-weight: 600; color: #555; margin-bottom: 5px; display: block; }
.mf-input {
width: 100%; padding: 9px 12px;
border: 1.5px solid #e0e0e0; border-radius: 7px;
font-size: 13px; background: #fafafa; color: #333;
outline: none; transition: border-color .2s;
}

```

[illegible]

[illegible]

```

<th>Tutor</th>
<th>Acciones</th>
</tr>
</thead>
<tbody>
<?php $n=1; foreach($courses as $course): ?>
<tr>
<td><?=$n++ ?></td>
<td><strong><?=$htmlspecialchars($course['name']) ?></strong></td>
<td><span class="badge-shift"><?=$ucfirst($course['shift_name']) ?></span></td>
<td style="white-space:nowrap;">
<?php if(!empty($course['tutor_last'])): ?>
<span style="color:#1565c0;font-size:13px;">

```

```

■ <?=$htmlspecialchars($course['tutor_last'].' '.$course['tutor_first']) ?>
</span>
<button class="btn-tutor-change"
data-course-id="<?=$course['id'] ?>"
data-course-name="<?=$htmlspecialchars($course['name'], ENT_QUOTES) ?>"
data-has-tutor="1"
title="Cambiar tutor"
style="padding:2px 7px;font-size:14px;background:none;border:none;cursor:pointer;color:#f57c00;vertical-align:middle;margin:0;">

```

```

■
</button>
<button class="btn-tutor-remove"
data-course-id="<?=$course['id'] ?>"
data-course-name="<?=$htmlspecialchars($course['name'], ENT_QUOTES) ?>"
title="Quitar tutor"
style="padding:2px 7px;font-size:14px;background:none;border:none;cursor:pointer;color:#dc3545;vertical-align:middle;font-weight:bold;margin:0;">

```

```

X
</button>
<?php else: ?>
<button class="btn-tutor-assign"
data-course-id="<?=$course['id'] ?>"
data-course-name="<?=$htmlspecialchars($course['name'], ENT_QUOTES) ?>"
data-has-tutor="0"
style="padding:3px 10px;font-size:12px;background:#17a2b8;color:white;border:none;border-radius:4px;cursor:pointer;margin:0;">

```

```

■ Asignar Tutor
</button>
<?php endif: ?>
</td>
<td style="white-space:nowrap;">
<button class="action-btn" style="background:#6f42c1;color:white;"
id="btnSub_<?=$course['id'] ?>"
onclick="toggleSubjectsRow(<?=$course['id'] ?>)">
■ Asignaturas
</button>
<button class="action-btn" style="background:#007bff;color:white;"
id="btnEst_<?=$course['id'] ?>"
onclick="toggleStudentsRow(<?=$course['id'] ?>)">
■ Estudiantes
</button>
<a href="?action=manage_schedule&course_id=<?=$course['id'] ?>"
class="action-btn" style="background:#17a2b8;color:white;text-decoration:none;display:inline-flex;align-items:center;">

```

```

■ Horario
</a>
<button class="action-btn" style="background:#ffc107;color:#000;"
onclick="openEditCourseModal(<?=$course['id'] ?>, '<?=$htmlspecialchars(addslashes($course['name'])) ?>', '<?=$htmlspecialchars(addslashes($course['grade_level'])) ?>', '<?=$course['parallel'] ?>', '<?=$course['shift_id'] ?>)">
=■ Editar
</button>
<form method="POST" action="?action=delete_course" style="display:inline;"
onsubmit="return confirmDeleteCourse(event, '<?=$htmlspecialchars(addslashes($course['name'])) ?>'">
<input type="hidden" name="course_id" value="<?=$course['id'] ?>">
<button type="submit" class="action-btn" style="background:#dc3545;color:white;">■ Eliminar</button>
</form>

```

```

</td>
</tr>
<!-- Fila expandible estudiantes -->
<tr id="rowEst_<?=$course['id'] ?>" class="students-panel" style="display:none;">
<td colspan="5">
<div class="sp-inner">
<div class="sp-header">
<span style="font-weight:700;font-size:14px;color:#1a237e;">
■ <?=$htmlspecialchars($course['name']) ?>
&nbsp;<span style="background:#e3f2fd;color:#1565c0;padding:2px
10px;border-radius:10px;font-size:12px;font-weight:700;">
<?=$count($enrollmentsByCourse[$course['id']] ?? []) ?> matriculados
</span>
</span>
<button onclick="openEnrollModal(<?=$course['id'] ?>, '<?=$htmlspecialchars(addslashes($course['name'])),
ENT_QUOTES) ?>'"
style="padding:6px 14px;font-size:12px;background:#28a745;color:#fff;border:none;border-radius:5px;cursor:po
inter;font-weight:600;margin:0;">
■ Matricular nuevo
</button>
</div>
<?php if(!empty($enrollmentsByCourse[$course['id']])): ?>
<input type="text" placeholder="■ Buscar estudiante..." oninput="filterInlineEnrolled(this, <?=$course['id'] ?>)"
style="width:100%;max-width:320px;padding:7px 12px;border:1px solid
#ddd;border-radius:6px;font-size:13px;margin-bottom:10px;">
<table id="tblEst_<?=$course['id'] ?>" style="width:100%;border-collapse:collapse;font-size:13px;table-layo
ut:fixed;">
<colgroup>
<col style="width:32px;">
<col style="width:110px;">
<col style="width:auto;">
<col style="width:220px;">
</colgroup>
<thead>
<tr style="background:#e8eaf6;">
<th style="padding:7px 10px;text-align:left;">#</th>
<th style="padding:7px 10px;text-align:left;">Cédula</th>
<th style="padding:7px 10px;text-align:left;">Apellidos y Nombres</th>
<th style="padding:7px 10px;text-align:center;">Acciones</th>
</tr>
</thead>
<tbody>
<?php $ni=1; foreach($enrollmentsByCourse[$course['id']] as $est): ?>
<tr data-course="<?=$course['id'] ?>" data-name="<?=$strtolower($est['last_name']).' '.$est['first_name']) ?
>" style="border-bottom:1px solid #e8eaf6;">
<td style="padding:7px 10px;"><?=$ni++ ?></td>
<td style="padding:7px 10px;"><?=$est['dni'] ?? '-' ?></td>
<td style="padding:7px 10px;"><strong><?=$htmlspecialchars($est['last_name']).' '.$est['first_name']) ?
></strong></td>
<td style="padding:7px 10px;text-align:center;white-space:nowrap;">
<button onclick="openRepModal(<?=$est['id'] ?>, '<?=$htmlspecialchars($est['last_name']).' '
.$est['first_name'], ENT_QUOTES) ?>', <?=$course['id'] ?>)"
style="padding:3px 9px;font-size:12px;background:#17a2b8;color:#fff;border:none;border-radius:4px;cursor:poi
nter;margin:0;">
Representantes
</button>
<button onclick="smConfirmUnenroll(<?=$est['id'] ?>, '<?=$addslashes($est['last_name']).' '
.$est['first_name']) ?>', <?=$course['id'] ?>)"
style="padding:3px 9px;font-size:12px;background:#dc3545;color:#fff;border:none;border-radius:4px;cursor:poi
nter;margin:0;">
X Retirar
</button>
</td>
</tr>
<?php endforeach; ?>
</tbody>
</table>
<?php else: ?>
<div style="text-align:center;padding:20px;color:#aaa;font-size:13px;">■ Ningún estudiante matriculado
aún.</div>

```

```

<?php endif; ?>
</div>
</td>
</tr>
<!-- Fila expandible ASIGNATURAS -->
<tr id="rowSub_<?=$course['id'] ?>" style="display:none;background:#fffde7;">
<td colspan="5" style="padding:0;border-bottom:2px solid #6f42c1;">
<div class="sp-inner">
<div class="sp-header">
<span style="font-weight:700;font-size:14px;color:#4a148c;">
■ <?=$course['name'] ?>
&nbsp;  <span style="background:#ede7f6;color:#4a148c;padding:2px
10px;border-radius:10px;font-size:12px;font-weight:700;">
<?=$count($subjectsByCourse[$course['id']] ?? []) ?> asignatura(s)
</span>
<?php
$totalAsig = array_sum(array_column($subjectsByCourse[$course['id']] ?? [], 'hours_per_week'));
$maxHorasDia = (strpos($course['grade_level'],'BT')!==false) ? 8 : 7;
// Calcular días laborables desde sesión/institución (igual que Course.php)
$diasSem = 5; // default
try {
$dbTmp = new Database();
$stmtTmp = $dbTmp->connect()->prepare("SELECT working_days_list FROM institutions WHERE id = :id");
$stmtTmp->execute([':id' => $_SESSION['institution_id']]);
$instTmp = $stmtTmp->fetch();
if(!empty($instTmp['working_days_list'])) {
$daysTmp = json_decode($instTmp['working_days_list'],true);
if(is_array($daysTmp) && count($daysTmp)) $diasSem = count($daysTmp);
}
} catch(Exception $e) { $diasSem = 5; }
$maxSem = $maxHorasDia * $diasSem;
$horasColor = $totalAsig > $maxSem ? '#dc3545' : ($totalAsig === $maxSem ? '#28a745' : '#e0a800');
$horasIcon = $totalAsig > $maxSem ? '■' : ($totalAsig === $maxSem ? '✓' : '■');
?>
&nbsp;  <span id="hrs-badge-<?=$course['id'] ?>"
data-max="<?=$maxSem ?>"
data-total="<?=$totalAsig ?>"
style="background:#fff3cd;color:<?=$horasColor ?>;padding:2px
10px;border-radius:10px;font-size:12px;font-weight:700;border:1px solid <?=$horasColor ?>;"
title="<?=$maxHorasDia ?> horas/día x <?=$diasSem ?> días = <?=$maxSem ?> horas máximas">
<?=$horasIcon ?> <?=$totalAsig ?>/<?=$maxSem ?> hrs
</span>
</span>
<button onclick="openAddSubjectModal(<?=$course['id'] ?>, '<?=$
htmlspecialchars(addslashes($course['name'])), ENT_QUOTES ?>')"
style="padding:6px 14px;font-size:12px;background:#6f42c1;color:#fff;border:none;border-radius:5px;cursor:po
inter;font-weight:600;margin:0;">
■ Agregar asignatura
</button>
</div>
<?php if(!empty($subjectsByCourse[$course['id']])): ?>
<input type="text" placeholder="■ Buscar asignatura..."
oninput="filterSubjects(this, <?=$course['id'] ?>)"
style="width:100%;max-width:320px;padding:7px 12px;border:1px solid
#ddd;border-radius:6px;font-size:13px;margin-bottom:10px;">
<table id="tblSub_<?=$course['id'] ?>" style="width:100%;border-collapse:collapse;font-size:13px;">

<thead>
<tr style="background:#ede7f6;">
<th style="padding:7px 10px;text-align:left;width:36px;">#</th>
<th style="padding:7px 10px;text-align:left;">Asignatura</th>
<th style="padding:7px 10px;text-align:left;">Código</th>
<th style="padding:7px 10px;text-align:center;width:80px;">Hrs/sem</th>
<th style="padding:7px 10px;text-align:left;">Docente</th>
<th style="padding:7px 10px;text-align:center;width:130px;">Acciones</th>
</tr>
</thead>
<tbody>
<?php $si=1; foreach($subjectsByCourse[$course['id']] as $sub): ?>
<tr data-subname="<?=$strtolower($sub['name']) ?>" style="border-bottom:1px solid #ede7f6;">
<td style="padding:7px 10px;"><?=$si++ ?></td>

```

```
<td style="padding:7px 10px;"><strong><?= htmlspecialchars($sub['name']) ?></strong></td>
<td style="padding:7px 10px;"><?= $sub['code'] ? htmlspecialchars($sub['code']) : '<span
style="color:#bbb;">&#x2014;</span>' ?></td>
<td style="padding:7px 10px;text-align:center;">
<form method="POST" action="?action=set_subject_hours&course_id=<?= $course['id'] ?>"
style="display:inline-flex;align-items:center;gap:4px;">
<input type="hidden" name="subject_id" value="<?= $sub['id'] ?>">
<input type="number" name="hours_per_week" value="<?= (int)($sub['hours_per_week'] ?? 1) ?>"
min="1" max="20"
data-course-id="<?= $course['id'] ?>"
data-original="<?= (int)($sub['hours_per_week'] ?? 1) ?>"
oninput="updateHrsBadge(this)"
style="width:50px;padding:3px 5px;border:1px solid #ddd;border-radius:4px;font-size:12px;text-align:center;"
>
<button type="submit" style="padding:3px 7px;background:#007bff;color:white;border:none;border-radius:4px;font-size:11px;cursor:pointer;margin:0;">&#10003;</button>
</form>
</td>
<td style="padding:7px 10px;">
<?php if(!empty($sub['teacher_name'])): ?>
<span style="color:#2e7d32;font-size:13px;">&#128100; <?= htmlspecialchars($sub['teacher_name']) ?></span>
<?php else: ?>
<span style="color:#bbb;font-style:italic;font-size:12px;">Sin docente</span>
<?php endif; ?>
</td>
<td style="padding:7px 10px;text-align:center;white-space:nowrap;">
<button onclick="abrirModalDocente(<?= $sub['id'] ?>, '<?= htmlspecialchars(addslashes($sub['name'])) ?>',
<?= $sub['assignment_id'] ?? 'null' ?>, <?= $sub['teacher_id'] ?? 'null' ?>, <?= $course['id'] ?>)"
style="padding:3px 8px;font-size:12px;background:#e65100;color:white;border:none;border-radius:4px;cursor:pointer;margin:0;"
title="<?= empty($sub['teacher_name']) ? 'Asignar docente' : 'Cambiar docente' ?>"
<?= empty($sub['teacher_name']) ? '&#128100;' : '&#128260;&#128100;' ?>
</button>
<?php if(!empty($sub['teacher_name'])): ?>
<form method="POST" action="?action=unassign_subject_teacher" style="display:inline;"
onsubmit="return confirmQuitarDocente(event, '<?= htmlspecialchars(addslashes($sub['name'])) ?>')">
<input type="hidden" name="assignment_id" value="<?= $sub['assignment_id'] ?>">
<input type="hidden" name="course_id" value="<?= $course['id'] ?>">
<button type="submit"
style="padding:3px 8px;font-size:12px;background:#6c757d;color:white;border:none;border-radius:4px;cursor:pointer;margin:0;">&#10006;</button>
</form>
<?php endif; ?>
<form method="POST" action="?action=remove_course_subject" style="display:inline;"
onsubmit="return confirmRemoveSubject(event, '<?= htmlspecialchars(addslashes($sub['name'])) ?>'">
<input type="hidden" name="course_id" value="<?= $course['id'] ?>">
<input type="hidden" name="subject_id" value="<?= $sub['id'] ?>">
<button type="submit" style="padding:3px 8px;font-size:12px;background:#dc3545;color:white;border:none;border-radius:4px;cursor:pointer;margin:0;">&#128465;</button>
</form>
</td>
</tr>
<?php endforeach; ?>
</tbody>
</table>
<?php else: ?>
<div style="text-align:center;padding:20px;color:#aaa;font-size:13px;">&#128235; Este curso aun no tiene asignaturas asignadas.</div>
<?php endif; ?>
</div>
</td>
</tr>
<?php endforeach; ?>
</tbody>
</table>
</div>
</div><!-- /container -->
<!--
```

----->

```
<div class="modal-header">
```

</div>

----->

```
<div class="modal-box" style="max-width:580px;">
```

<div>

<p>Las asignaturas se cargan automáticamente según el nivel</p>

```
<button class="modal-close" onclick="closeCourseModal()">X</button>
```

</div>

```
<div class="modal-body">
```

```
<?php
```

```
unset($_SESSION['course_form']);
```

27

```
<script type="application/json" id="_dataCF"><?= json_encode(["education_type"=>$cf["education_type"]??"", "grade_level"=>$cf["grade_level"]??"", "specialty"=>$cf["specialty"]??"", "carrera"=>$cf["carrera"]??""), JSON_HEX_TAG|JSON_HEX_AMP) ?></script>
```



```

<!-- Nivel + Grado -->
<div class="mf-row">
<div>
<label class="mf-label">Nivel Educativo *</label>
<select name="education_type" id="education_type" required onchange="updateGradeLevels()" class="mf-input">
<option value="">Seleccionar...</option>
<option value="inicial" <?= ($cf['education_type']??'')==='inicial'? 'selected':'' ?>>■ Educación
Inicial</option>
<option value="egb" <?= ($cf['education_type']??'')==='egb' ? 'selected':'' ?>>■ EGB</option>
<option value="bgu" <?= ($cf['education_type']??'')==='bgu' ? 'selected':'' ?>>■ BGU</option>
<option value="bt" <?= ($cf['education_type']??'')==='bt' ? 'selected':'' ?>>■ Bachillerato Técnico</option>
</select>
</div>
<div id="group_grade">
<label class="mf-label">Grado / Año *</label>
<select name="grade_level" id="grade_level" required onchange="onGradeChange()" class="mf-input">
<option value="">Selecione nivel primero...</option>

<?php if(!empty($cf['grade_level'])): ?>
<option value="<?= htmlspecialchars($cf['grade_level']) ?>" selected>
<?= htmlspecialchars($cf['grade_level']) ?>
</option>
<?php endif; ?>
</select>
</div>

<!-- BT: Figura + Carrera -->
<div id="group_specialty" style="display:none;" class="mf-group">
<label class="mf-label">Figura Profesional</label>
<select name="specialty" id="specialty" onchange="updateCarreras()" class="mf-input">
<option value="">Seleccionar figura...</option>
<?php foreach(['Informática','Administración','Contabilidad','Comercialización y Ventas','Servicios
Hoteleros','Turismo','Electromecánica Automotriz','Instalaciones Eléctricas','Electrónica de
Consumo','Mecanizado y Construcciones Metálicas','Producción Agropecuaria','Acuicultura','Industrialización
de Alimentos','Confección Textil','Música','Artes Plásticas','Diseño Gráfico','Servicios de
Belleza','Atención Integral en Salud','Mecatrónica','Refrigeración y Climatización','Construcción','Redes y
Telecomunicaciones','Seguridad Industrial','Logística y Transporte'] as $fig): ?>
<option value="<?= $fig ?>"><?= $fig ?></option>
<?php endforeach; ?>
</select>
</div>
<div id="group_carrera" style="display:none;" class="mf-group">
<label class="mf-label">Especialidad / Carrera <span style="color:#aaa;font-weight:400;">(opcional)</span></
label>
<select name="carrera" id="carrera" onchange="generateCourseName()" class="mf-input">
<option value="">Sin especificar</option>
</select>
</div>

<!-- Paralelo + Jornada -->
<div class="mf-row">
<div>
<label class="mf-label">Paralelo *</label>
<select name="parallel" id="parallel" required onchange="generateCourseName()" class="mf-input">
<option value="">Seleccionar...</option>
<?php foreach(range('A','J') as $letter): ?>
<option value="<?= $letter ?>" <?= ($cf['parallel']??'')===$letter?'selected':'' ?>><?= $letter ?></option>
<?php endforeach; ?>
</select>
</div>
<div>
<label class="mf-label">Jornada *</label>
<select name="shift_id" id="shift_id" required onchange="generateCourseName()" class="mf-input">
<option value="">Seleccionar...</option>
<?php foreach($shifts as $shift): ?>
<option value="<?= $shift['id'] ?>" data-shift="<?= $shift['name'] ?>"
<?= ($cf['shift_id']??'')===$shift['id']?'selected':'' ?>>
<?= ucfirst($shift['name']) ?>
</option>
<?php endforeach; ?>

```

```

</select>
</div>
</div>

<!-- Nombre generado -->
<div class="mf-group" style="margin-bottom:0;">
<label class="mf-label">
Nombre del Curso
<span style="font-weight:400;color:#aaa;margin-left:4px;"><- generado automáticamente</span>
</label>
<div style="position:relative;">
<input type="text" name="name" id="course_name" readonly
value="<?= htmlspecialchars($cf['name']) ??">"
class="mf-input"
style="padding-left:34px;font-weight:600;color:#1a237e;background:#eef2ff;">
<span style="position:absolute;left:10px;top:50%;transform:translateY(-50%);font-size:14px;">■</span>
</div>
</div>
</div>
<div class="modal-footer">
<button type="button" class="btn-cancel" onclick="closeCourseModal()">Cancelar</button>
<button type="submit" class="btn-submit">■ Crear Curso</button>
</div>
</form>
</div>
</div>

<!-- ■■ Modal Asignar/Cambiar Tutor ■■ -->
<div id="tutorModal" class="modal-overlay">
<div class="modal-box" style="max-width:420px;">
<div class="modal-header">
<div>
<h3>■ <span id="tutorModalTitle">Asignar Tutor</span></h3>
<p>Curso: <strong id="tutorModalCourse"></strong></p>
</div>
<button class="modal-close" onclick="closeTutorModal()">X</button>
</div>
<form method="POST" action="?action=set_tutor" id="tutorForm">
<div class="modal-body">
<div id="tutorLoadingMsg" style="color:#999;font-size:13px;margin-bottom:12px;display:none;">■ Cargando
docentes...</div>
<div id="tutorNoTeachers" style="display:none;background:#fff3cd;color:#856404;padding:10px;border-radius:6px;
font-size:13px;margin-bottom:12px;">
■■ No hay docentes disponibles. Primero asigna docentes al curso en <strong>Asignaturas</strong>.
</div>
<input type="hidden" name="course_id" id="tutorCourseId">
<div class="mf-group">
<label class="mf-label">Seleccionar Docente</label>
<select name="teacher_id" id="tutorTeacherSelect" class="mf-input">
<option value="">Seleccionar...</option>
</select>
</div>
</div>
<div class="modal-footer">
<button type="button" class="btn-cancel" onclick="closeTutorModal()">Cancelar</button>
<button type="submit" id="tutorSubmitBtn" disabled class="btn-submit">✓ Asignar</button>
</div>
</form>
</div>
</div>

<!-- ■■ Modal Quitar Tutor ■■ -->
<div id="removeTutorModal" class="modal-overlay">
<div class="modal-box" style="max-width:400px;">
<div class="modal-header">
<h3 style="color:#dc3545;">X Quitar Tutor</h3>
<button class="modal-close" onclick="closeRemoveModal()">X</button>
</div>
<div class="modal-body">
<p style="font-size:14px;color:#555;">¿Quitar el tutor del curso <strong

```

```

id="removeTutorCourseName"></strong>?</p>
</div>
<form method="POST" action="?action=remove_tutor">
<input type="hidden" name="course_id" id="removeTutorCourseId">
<div class="modal-footer">

<button type="button" class="btn-cancel" onclick="closeRemoveModal()">Cancelar</button>
<button type="submit" style="padding:9px 22px;background:#dc3545;color:white;border:none;border-radius:6px;cursor:pointer;margin:0;font-size:14px;font-weight:600;">Sí, Quitar</button>
</div>
</form>
</div>
</div>

<!-- ■■ Modal Matricular ■■ -->
<div id="enrollModal" class="modal-overlay" onclick="if(event.target===this)closeEnrollModal()">
<div class="modal-box" style="max-width:460px;">
<div class="modal-header">
<div>
<h3>■■ Matricular Estudiantes</h3>
<p id="emCourseLabel"></p>
</div>
<button class="modal-close" onclick="closeEnrollModal()">X</button>
</div>
<div class="modal-body">
<form method="POST" action="?action=enroll_students" id="emForm">
<input type="hidden" name="course_id" id="emCourseId">
<input type="hidden" name="redirect_to" value="academic">
<div style="display:flex;gap:8px;align-items:center;margin-bottom:10px;">
<input type="text" id="emSearch" placeholder="■■ Buscar..." oninput="filterEmAvail()"
style="flex:1;padding:8px 12px;border:1px solid #ddd;border-radius:6px;font-size:13px;">
<button type="button" onclick="emSelectAll()"
style="padding:7px 13px;background:#e8f5e9;color:#2e7d32;border:1px solid
#a5d6a7;border-radius:6px;font-size:12px;cursor:pointer;white-space:nowrap;margin:0;">
  Seleccionar todos
</button>
</div>
<div style="margin-bottom:8px;">
<span id="emCounter" style="font-size:12px;background:#f0f0f0;color:#555;padding:2px
10px;border-radius:10px;">0 seleccionados</span>
</div>
<div id="emList" style="max-height:260px;overflow-y:auto;padding-right:2px;"></div>
<div style="display:flex;justify-content:flex-end;gap:10px;margin-top:16px;padding-top:12px;border-top:1px
solid #f0f0f0;">
<button type="button" onclick="closeEnrollModal()"
style="padding:8px 18px;background:#f5f5f5;color:#555;border:1px solid
#ddd;border-radius:6px;cursor:pointer;margin:0;">Cancelar</button>
<button type="submit" id="emSubmit" disabled
style="padding:8px 18px;background:#28a745;color:#fff;border:none;border-radius:6px;cursor:pointer;font-weig
ht:600;margin:0;">
  ✓ Matricular seleccionados
</button>
</div>
</div>
</form>
</div>
</div>
</div>

<!-- ■■ Modal Confirmar Retirar ■■ -->
<div id="smUnenrollModal" class="modal-overlay" onclick="if(event.target===this)closeSmUnenroll()">
<div class="modal-box" style="max-width:420px;">
<div class="modal-header">
<h3 style="color:#dc3545;">■■ Retirar Estudiante</h3>
<button class="modal-close" onclick="closeSmUnenroll()">X</button>
</div>
<div class="modal-body">
<p id="smUnenrollMsg" style="color:#555;font-size:14px;margin-bottom:6px;"></p>
<p style="color:#999;font-size:12px;">Los registros de asistencia se conservarán.</p>
</div>
<form method="POST" action="?action=unenroll_student" id="smUnenrollForm">
<input type="hidden" name="student_id" id="smUnenrollStudentId">

```

```
<input type="hidden" name="course_id" id="smUnenrollCourseId">
<input type="hidden" name="redirect_to" value="academic">
<div class="modal-footer">
  <button type="button" class="btn-cancel" onclick="closeSmUnenroll()">Cancelar</button>
  <button type="submit" style="padding:9px 22px;background:#dc3545;color:white;border:none;border-radius:6px;cursor:pointer;margin:0;font-size:14px;font-weight:600;">Sí, Retirar</button>
</div>
</form>
</div>
</div>
```

[illegible]

```
<script type="application/json" id="_dataAvailable"><?=json_encode(array_values($availableStudents),
JSON_HEX_TAG|JSON_HEX_AMP) ?></script>
<script type="application/json" id="_dataReps"><?=json_encode(array_values($representatives ?? []),
JSON_HEX_TAG|JSON_HEX_AMP) ?></script>
<script type="application/json" id="_dataRepsByStudent"><?=json_encode($repsByStudent ?? [],
JSON_HEX_TAG|JSON_HEX_AMP) ?></script>
<script>
```

```
/* \u2500\u2500 Modales SY y Curso \u2500\u2500 */
function openSYModal() { document.getElementById('modalsY').classList.add('active'); }
```

```
function closeSYModal() { document.getElementById('modalSY').classList.remove('active');  
document.getElementById('formSY').reset(); }  
function openCourseModal() { document.getElementById('modalCourse').classList.add('active'); }  
function closeCourseModal(){ document.getElementById('modalCourse').classList.remove('active'); }
```

```
/* Si hubo error de duplicado, reabrir modal de curso autom\u00e9ticamente */
<?php if(isset($_GET['error']) && in_array($_GET['error'], ['course_duplicate','no_active_year']) && !
empty($cf)): ?>
document.addEventListener('DOMContentLoaded', function(){ openCourseModal(); });
<?php endif; ?>
```

```

/* \u2500\u2500 Grade levels (curso) \u2500\u2500 */
const gradeLevels = {
  inicial: ['Inicial 1 (0-3 a\u00f1os)', 'Inicial 2 (3-5 a\u00f1os)'],
  egb: ['1.\u00ba EGB - Preparatoria', '2.\u00ba EGB', '3.\u00ba EGB', '4.\u00ba EGB', '5.\u00ba EGB', '6.\u00ba EGB', '7.\u00ba EGB', '8.\u00ba EGB', '9.\u00ba EGB', '10.\u00ba EGB'],
  bgu: ['1.\u00ba BGU', '2.\u00ba BGU', '3.\u00ba BGU'],
  bt: ['1.\u00ba BT', '2.\u00ba BT', '3.\u00ba BT']
};

const egbNocturnaAllowed = ['8.\u00ba EGB', '9.\u00ba EGB', '10.\u00ba EGB'];

const carreras = {
  'Inform\u00e1tica': ['Aplicaciones Inform\u00e1ticas', 'Programaci\u00f3n de Software', 'Soporte T\u00e9cnico', 'Sistemas Microinform\u00e1ticos y Redes'],
  'Administraci\u00f3n': ['Asistencia Administrativa', 'Gesti\u00f3n Empresarial'],
  'Contabilidad': ['Contabilidad', 'Ventas e Informaci\u00f3n Comercial'],
  'Comercializaci\u00f3n y Ventas': ['Ventas e Informaci\u00f3n Comercial', 'Marketing'],
  'Servicios Hoteleros': ['Hoteler\u00eda', 'Hospitalidad'],
  'Turismo': ['Turismo', 'Gu\u00eda Tur\u00edstica'],
  'Electromec\u00e1nica Automotriz': ['Electromec\u00e1nica Automotriz', 'Mec\u00e1nica Automotriz'],
  'Instalaciones El\u00e9ctricas': ['Electricidad', 'Instalaciones El\u00e9ctricas'],
  'Electr\u00f3nica de Consumo': ['Electr\u00f3nica', 'Mantenimiento Electr\u00f3nico'],
  'Atenci\u00f3n Integral en Salud': ['Atenci\u00f3n en Enfermer\u00eda', 'Auxiliar de Salud'],
  'Producci\u00f3n Agropecuaria': ['Producci\u00f3n Agropecuaria', 'Agroindustria'],
  'Redes y Telecomunicaciones': ['Redes', 'Telecomunicaciones'],
  'Dise\u00f1o Gr\u00e1fico': ['Dise\u00f1o Gr\u00e1fico', 'Multimedia'],
  'Servicios de Belleza': ['Peluquer\u00eda', 'Cosmetolog\u00eda']
};

```

```
function getNocturnaOption() {
  const opts = document.getElementById('shift_id').options;
  for (let i = 0; i < opts.length; i++) {
    if ((opts[i].getAttribute('data-shift')||'').toLowerCase()=== 'nocturna') return opts[i];
  }
  return null;
}
```

```

}
function updateNocturnaVisibility(type, grade) {
  const n = getNocturnaOption(); if (!n) return;
  let visible = (type==='bgu' || type==='bt') || (type==='egb' && egbNocturnaAllowed.includes(grade));
  n.style.display = visible ? '' : 'none';
  if (!visible && n.selected) { document.getElementById('shift_id').value=''; generateCourseName(); }
}
function updateGradeLevels() {
  const type = document.getElementById('education_type').value;
  const sel = document.getElementById('grade_level');
  sel.innerHTML = '<option value="">Seleccionar grado...</option>';
  document.getElementById('course_name').value = '';
  if (!type) return;
  (gradeLevels[type] || []).forEach(function(g) { const o=document.createElement('option');
  o.value=o.textContent=g; sel.appendChild(o); });
  const isBT = type==='bt';
  document.getElementById('group_specialty').style.display = isBT ? 'block' : 'none';
  document.getElementById('group_carrera').style.display = isBT ? 'block' : 'none';
  document.getElementById('specialty').required = isBT;
  if (!isBT) { document.getElementById('specialty').value='';
  document.getElementById('carrera').innerHTML='<option value="">Sin especificar</option>'; }
  updateNocturnaVisibility(type, '');
  generateCourseName();
}
function onGradeChange() {
  updateNocturnaVisibility(document.getElementById('education_type').value,
  document.getElementById('grade_level').value);
  generateCourseName();
}
function updateCarreras() {
  const fig = document.getElementById('specialty').value;
  const sel = document.getElementById('carrera');
  sel.innerHTML = '<option value="">Sin especificar</option>';
  (carreras[fig] || []).forEach(function(c) { const o=document.createElement('option'); o.value=o.textContent=c;
  sel.appendChild(o); });
  generateCourseName();
}
function generateCourseName() {
  const grade = document.getElementById('grade_level').value;
  const parallel = document.getElementById('parallel').value;
  const shiftSel = document.getElementById('shift_id');
  const shiftOpt = shiftSel.options[shiftSel.selectedIndex];
  const shiftName = shiftOpt ? (shiftOpt.getAttribute('data-shift') || '') : '';
  const type = document.getElementById('education_type').value;

  const specialty= document.getElementById('specialty').value;
  const carrera = document.getElementById('carrera').value;
  if (!grade || !parallel || !shiftName) { document.getElementById('course_name').value=''; return; }
  let name = grade + ' ' + parallel + ' ';
  if (type==='bt' && specialty) { name += ' - ' + specialty; if(carrera) name += ' (' + carrera + ')'; }
  name += ' - ' + shiftName.charAt(0).toUpperCase() + shiftName.slice(1);
  document.getElementById('course_name').value = name;
}

/* Restaurar form si volvi con error */
(function restoreForm() {
  var _cfData = JSON.parse(document.getElementById('_dataCF').textContent);
  var savedType = _cfData.education_type || '';
  var savedGrade = _cfData.grade_level || '';
  var savedSpec = _cfData.specialty || '';
  if (!savedType) return;
  document.getElementById('education_type').value = savedType;
  updateGradeLevels();
  setTimeout(function() {
    if (savedGrade) { document.getElementById('grade_level').value=savedGrade; onGradeChange(); }
    if (savedType==='bt' && savedSpec) {
      var s=document.getElementById('specialty'); if(s){s.value=savedSpec;updateCarreras();}
      setTimeout(function(){ var c=document.getElementById('carrera'); if(c) c.value=_cfData.carrera || '';
      generateCourseName(); },60);
    } else { generateCourseName(); }
  },60);
})();

```

```

/* \u2500\u2500 Confirmaciones \u2500\u2500 */
function confirmDeleteCourse(event, courseName) {
    event.preventDefault();
    showConfirmModal('\u26a0\u2014 Eliminar Curso', '#dc3545',
        '\u2014 seguro de eliminar el curso <strong>'+courseName+'</strong>?',
        '<p style="font-size:13px;background:#fff3cd;color:#856404;padding:10px;border-radius:4px;margin-top:8px;">\u26a0\u2014 Se desvincular\u2014 autom\u2014ticamente a todos los estudiantes matriculados y docentes asignados.</p>',
        'S\u2014, Eliminar', event.target);
    return false;
}

function confirmDelete(event, yearName) {
    event.preventDefault();
    showConfirmModal('\u26a0\u2014 Eliminar A\u2014 Lectivo', '#dc3545',
        '\u2014 seguro de eliminar el a\u2014 lectivo <strong>'+yearName+'</strong>?',
        '<p style="font-size:13px;color:#666;margin-top:8px;"><strong>Nota:</strong> No se puede eliminar si tiene cursos asociados.</p>',
        'S\u2014, Eliminar', event.target);
    return false;
}

function confirmSY(event, accion, nombre) {
    event.preventDefault();
    var esActivar = accion==='activar';
    var color = esActivar ? '#28a745' : '#6c757d';
    var desc = esActivar
        ? 'Se desactivar\u2014 el a\u2014 lectivo actual y se activar\u2014 <strong>'+nombre+'</strong>.'
        : '\u2014Seguro que deseas desactivar <strong>'+nombre+'</strong>?';
    showConfirmModal((esActivar ? '\u2713 Activar' : '\u2298 Desactivar') + ' A\u2014 Lectivo', color, desc, '',
        esActivar ? 'S\u2014, Activar' : 'S\u2014, Desactivar', event.target);
    return false;
}

function showConfirmModal(title, color, msg, extra, btnTxt, form) {
    var m = document.createElement('div');
    m.style.cssText='position:fixed;inset:0;background:rgba(0,0,0,.55);display:flex;align-items:center;justify-content:center;z-index:99999;';
    m.innerHTML='<div style="background:#fff;border-radius:12px;padding:28px;max-width:480px;width:90%;box-shadow:0 10px 40px rgba(0,0,0,.25);">'
        + '<h3 style="margin:0 0 12px;color: '+color+';">'+title+'</h3>'
        + '<p style="color:#555;font-size:14px;">'+msg+'</p>'+extra
        + '<div style="display:flex;gap:10px;justify-content:flex-end;margin-top:20px;">'
        + '<button id="__cfCancel" style="padding:9px 20px;background:#f5f5f5;color:#555;border:1px solid #ddd;border-radius:6px;cursor:pointer;">Cancelar</button>'
        + '<button id="__cfOk" style="padding:9px 20px;background: '+color+';color:white;border:none;border-radius:6px;cursor:pointer;font-weight:600;">'+btnTxt+'</button>'
        + '</div></div>';
    document.body.appendChild(m);
    m.querySelector('#__cfOk').onclick = function(){ document.body.removeChild(m); form.submit(); };
    m.querySelector('#__cfCancel').onclick = function(){ document.body.removeChild(m); };
    m.onclick = function(e){ if(e.target===m) document.body.removeChild(m); };
}

/* \u2500\u2500 Tutor \u2500\u2500 */
document.addEventListener('click', function(e) {
    var btn = e.target.closest('.btn-tutor-assign,.btn-tutor-change,.btn-tutor-remove');
    if (!btn) return;
    var courseId=btn.dataset.courseId, courseName=btn.dataset.courseName;
    if (btn.classList.contains('btn-tutor-remove')) { openRemoveModal(courseId,courseName); }
    else { openTutorModal(courseId,courseName,btn.dataset.hasTutor==='1'); }
});

function openTutorModal(courseId,courseName,hasTutor){
    document.getElementById('tutorCourseId').value=courseId;

    document.getElementById('tutorModalCourse').textContent=courseName;
    document.getElementById('tutorModalTitle').textContent=hasTutor?'Cambiar Tutor':'Asignar Tutor';
    document.getElementById('tutorTeacherSelect').innerHTML='<option value="">Seleccionar...</option>';
    document.getElementById('tutorLoadingMsg').style.display='block';
    document.getElementById('tutorNoTeachers').style.display='none';
    document.getElementById('tutorSubmitBtn').disabled=true;
    document.getElementById('tutorModal').classList.add('active');
    fetch('?action=get_course_teachers&course_id='+courseId)

```

```

.then(function(r){return r.json();})
.then(function(teachers){
document.getElementById('tutorLoadingMsg').style.display='none';
if(!teachers||!teachers.length){document.getElementById('tutorNoTeachers').style.display='block';return;}
var sel=document.getElementById('tutorTeacherSelect');
teachers.forEach(function(t){var o=document.createElement('option');o.value=t.teacher_id;o.textContent=t.teacher_name;sel.appendChild(o);});
document.getElementById('tutorSubmitBtn').disabled=false;
})
.catch(function(){
var msg=document.getElementById('tutorLoadingMsg');
msg.textContent='\u2717 Error al cargar docentes.'; msg.style.color='#dc3545';
});
}
function closeTutorModal(){ document.getElementById('tutorModal').classList.remove('active'); }
function openRemoveModal(courseId,courseName){
document.getElementById('removeTutorCourseId').value=courseId;
document.getElementById('removeTutorCourseName').textContent=courseName;
document.getElementById('removeTutorModal').classList.add('active');
}
function closeRemoveModal(){ document.getElementById('removeTutorModal').classList.remove('active'); }
document.getElementById('tutorModal').addEventListener('click',function(e){if(e.target===this)closeTutorModal();});
document.getElementById('removeTutorModal').addEventListener('click',function(e){if(e.target===this)closeRemoveModal();});

/* \u2500\u2500 Estudiantes expandible \u2500\u2500 */
function toggleStudentsRow(courseId) {
var row=document.getElementById('rowEst_'+courseId);
var btn=document.getElementById('btnEst_'+courseId);
if (!row) return;
var isOpen = row.style.display!='none';
// Cerrar todos (estudiantes Y asignaturas)
document.querySelectorAll('[id^="rowEst_"]',[id^="rowSub_"]').forEach(function(r){r.style.display='none';});
document.querySelectorAll('[id^="btnEst_"]').forEach(function(b){b.style.background='#007bff';});

document.querySelectorAll('[id^="btnSub_"]').forEach(function(b){b.style.background='#6f42c1';});
if (!isOpen){ row.style.display=''; btn.style.background='#0056b3'; }
}

function normalize(str){ return str.toLowerCase().normalize('NFD').replace(/[\u0300-\u036f]/g,''); }
function filterInlineEnrolled(input,courseId){
var q=normalize(input.value);
document.querySelectorAll('#tblEst_'+courseId+' tbody tr').forEach(function(tr){
tr.style.display=normalize(tr.dataset.name||'').includes(q)?'':'none';
});
}

/* \u2500\u2500 Matricular \u2500\u2500 */
var smAvailableStudents = JSON.parse(document.getElementById("_dataAvailable").textContent);
var smRepresentatives = JSON.parse(document.getElementById("_dataReps").textContent);
var smRepsByStudent = JSON.parse(document.getElementById("_dataRepsByStudent").textContent);
function openEnrollModal(courseId,courseName){
document.getElementById('emCourseId').value=courseId;
document.getElementById('emCourseLabel').textContent=courseName;
document.getElementById('emSearch').value='';
document.getElementById('emCounter').textContent='0 seleccionados';
document.getElementById('emSubmit').disabled=true;
var html='';
smAvailableStudents.forEach(function(s){
var name=(s.last_name||'')+ ' '(s.first_name||'');
var dni=s.dni?'<span style="color:#aaa;font-size:11px;">\u00b7 '+s.dni+'</span>':'';
html+='\<div class="emItem" data-name="'+normalize(name)+' " '
+'style="display:flex;align-items:center;gap:10px;padding:8px 10px;border:1px solid #e8e8e8;border-radius:6px;margin-bottom:5px;cursor:pointer;" '
+'onclick="emToggleCheck(this)">'
+'<input type="checkbox" name="student_ids[]" value="'+s.id+' " '
+'onclick="event.stopPropagation();" onchange="emUpdateCounter()" '
+'style="width:15px;height:15px;cursor:pointer;flex-shrink:0;">'
+'<span style="font-size:13px;"><strong>'+name+'</strong>'+dni+'</span></div>';
});
if (!html) html='\<p style="text-align:center;color:#aaa;padding:20px;font-size:13px;">No hay estudiantes disponibles para matricular.</p>';

```

```

document.getElementById('emList').innerHTML=html;
document.getElementById('enrollModal').classList.add('active');
}
function closeEnrollModal(){ document.getElementById('enrollModal').classList.remove('active'); }
function filterEmAvalil(){
    var q=normalize(document.getElementById('emSearch').value);
document.querySelectorAll('#emList .emItem').forEach(function(el){el.style.display=normalize(el.dataset.name|'|').includes(q)?':none':''});
}
function emSelectAll(){
    document.querySelectorAll('#emList .emItem').forEach(function(el){if(el.style.display!='none')el.querySelector('input[type="checkbox"]').checked=true;});
    emUpdateCounter();
}
function emToggleCheck(div){ var cb=div.querySelector('input[type="checkbox"]');
if(cb){cb.checked=!cb.checked;emUpdateCounter();} }
function emUpdateCounter(){
    var n=document.querySelectorAll('#emList input[type="checkbox"]:checked').length;
    document.getElementById('emCounter').textContent=n+' seleccionado'+(n!=1?'s':'');
    document.getElementById('emSubmit').disabled=(n===0);
}

/* \u2500\u2500 Retirar estudiante \u2500\u2500 */
function smConfirmUnenroll(studentId,name,courseId){
    document.getElementById('smUnenrollStudentId').value=studentId;
    document.getElementById('smUnenrollCourseId').value=courseId;
    document.getElementById('smUnenrollMsg').innerHTML='\u00bfRetirar a <strong>'+name+'</strong> del curso?';
    document.getElementById('smUnenrollModal').classList.add('active');
}
function closeSmUnenroll(){ document.getElementById('smUnenrollModal').classList.remove('active'); }

/* ESC cierra modales */
document.addEventListener('keydown',function(e){
    if(e.key=='Escape'){
        closeSYModal(); closeCourseModal(); closeTutorModal();
        closeRemoveModal(); closeEnrollModal(); closeSmUnenroll();
    }
});
</script>

```

```
<!-- MODAL: EDITAR AÑO LECTIVO -->
<div id="modalEditSY" class="modal-overlay" onclick="if(event.target===this)closeEditSYModal()">
  <div class="modal-box" style="max-width:540px;">
    <div class="modal-header">
      <div>
        <h3>■ Editar Año Lectivo</h3>
        <p id="editSYSubtitle"></p>
      </div>
      <button class="modal-close" onclick="closeEditSYModal()">X</button>
    </div>
    <form method="POST" id="formEditSY">
      <div class="modal-body">
        <div class="mf-group">
          <label class="mf-label">Nombre del Año Lectivo *</label>
          <input type="text" name="name" id="editSYName" class="mf-input" required placeholder="Ej: 2025-2026">
          <small style="color:#aaa;font-size:11px;">Formato recomendado: AAAA-AAAA</small>
        </div>
        <div class="mf-row">
          <div>
            <label class="mf-label">Fecha de Inicio *</label>
            <input type="date" name="start_date" id="editSYStart" class="mf-input" required>
          </div>
          <div>
            <label class="mf-label">Fecha de Fin *</label>
            <input type="date" name="end_date" id="editSYEnd" class="mf-input" required>
          </div>
        </div>
      </div>
    </form>
  </div>
</div>
```



```

</div>
<div id="editSYActiveNote" style="display:none;background:#e8f5e9;color:#2e7d32;border-radius:7px;padding:10px 14px;font-size:13px;margin-top:4px;">
  ■ Este es el <strong>año lectivo activo</strong> actualmente.
</div>
</div>
<div class="modal-footer">
  <button type="button" class="btn-cancel" onclick="closeEditSYModal()">Cancelar</button>
  <button type="submit" class="btn-submit">■ Guardar Cambios</button>
</div>
</form>
</div>
</div>

```

```

<div class="modal-header">
<div>
<h3>=■ Editor Curso</h3>
<p id="editCourseSubtitle" style="color:#888;font-size:12px;"></p>
</div>
<button class="modal-close" onclick="closeEditCourseModal()">X</button>
</div>
<form method="POST" id="formEditCourse">
<div class="modal-body">
<!-- Nivel + Grado -->
<div class="mf-row">
<div>
<label class="mf-label">Nivel Educativo *</label>
<select name="education_type" id="editEduType" required onchange="editUpdateGradeLevels()" class="mf-input">
<option value="">Seleccionar...</option>
<option value="inicial">■ Educación Inicial</option>
<option value="egb">■ EGB</option>
<option value="bgu">■ BGU</option>
<option value="bt">■ Bachillerato Técnico</option>
</select>
</div>
<div>
<label class="mf-label">Grado / Año *</label>
<select name="grade_level" id="editGradeLevel" required onchange="editOnGradeChange()" class="mf-input">
<option value="">Seleccione nivel primero...</option>
</select>
</div>
</div>
<!-- BT Especialidad -->
<div id="editGroupSpecialty" style="display:none;" class="mf-group">
<label class="mf-label">Figura Profesional</label>
<select name="specialty" id="editSpecialty" onchange="editUpdateCarreras()" class="mf-input">
<option value="">Seleccionar figura...</option>
<?php foreach(['Informática','Administración','Contabilidad','Comercialización y Ventas','Servicios
Hoteleros','Turismo','Electromecánica Automotriz','Instalaciones Eléctricas','Electrónica de
Consumo','Mecanizado y Construcciones Metálicas','Producción Agropecuaria','Acuicultura','Industrialización
de Alimentos','Confección Textil','Música','Artes Plásticas','Diseño Gráfico','Servicios de
Belleza','Atención Integral en Salud','Mecatrónica','Refrigeración y Climatización','Construcción','Redes y
Telecomunicaciones','Seguridad Industrial','Logística y Transporte'] as $fig): ?>
<option value="<?= $fig ?>"><?= $fig ?></option>
<?php endforeach; ?>
</select>
</div>
<div id="editGroupCarrera" style="display:none;" class="mf-group">
<label class="mf-label">Especialidad / Carrera <span style="color:#aaa;font-weight:400;">(opcional)</span></label>
<select name="carrera" id="editCarrera" onchange="editGenerateName()" class="mf-input">
<option value="">Sin especificar</option>
</select>
</div>

```

```

<!-- Paralelo + Jornada -->
<div class="mf-row">
<div>
<label class="mf-label">Paralelo *</label>
<select name="paralelo" id="editParalelo" required onchange="editGenerateName()" class="mf-input">
<option value="">Seleccionar...</option>
<?php foreach(range('A','J') as $l): ?>
<option value="<?= $l ?>"><?= $l ?></option>
<?php endforeach; ?>
</select>
</div>
<div>
<label class="mf-label">Jornada *</label>
<select name="shift_id" id="editShiftId" required onchange="editGenerateName()" class="mf-input">
<option value="">Seleccionar...</option>
<?php foreach($shifts as $shift): ?>
<option value="<?= $shift['id'] ?>" data-shift="<?= $shift['name'] ?>"><?= ucfirst($shift['name']) ?
></option>
<?php endforeach; ?>
</select>
</div>
</div>
<!-- Nombre generado -->
<div class="mf-group" style="margin-bottom:0;">
<label class="mf-label">Nombre del Curso <span style="font-weight:400;color:#aaa;"><- generado
automáticamente</span></label>
<div style="position:relative;">
<input type="text" name="name" id="editCourseName" readonly class="mf-input"
style="padding-left:34px;font-weight:600;color:#1a237e;background:#eef2ff;">
<span style="position:absolute;left:10px;top:50%;transform:translateY(-50%);font-size:14px;">■</span>
</div>
</div>
<div class="modal-footer">
<button type="button" class="btn-cancel" onclick="closeEditCourseModal()">Cancelar</button>
<button type="submit" class="btn-submit">■ Guardar Cambios</button>
</div>
</form>

```

```

</div>
</div>

<script>
/* \u2550\u2550 Editar A\u00f1o Lectivo \u2550\u2550 */
function openEditSYModal(id, name, startDate, endDate, isActive) {
document.getElementById('formEditSY').action = '?action=edit_school_year&id=' + id;
document.getElementById('editSYName').value = name;
document.getElementById('editSYStart').value = startDate;
document.getElementById('editSYEnd').value = endDate;
document.getElementById('editSYSubtitle').textContent = name;
document.getElementById('editSYActiveNote').style.display = isActive ? 'block' : 'none';
document.getElementById('modalEditSY').classList.add('active');
}
function closeEditSYModal() { document.getElementById('modalEditSY').classList.remove('active'); }
document.getElementById('modalEditSY').addEventListener('click', function(e){ if(e.target===this)
closeEditSYModal(); });

/* \u2550\u2550 Editar Curso \u2550\u2550 */
const editGradeLevels = {
  inicial: ['Inicial 1 (0-3 a\u00f1os)', 'Inicial 2 (3-5 a\u00f1os)'],
  egb: ['1.\u00ba EGB - Preparatoria', '2.\u00ba EGB', '3.\u00ba EGB', '4.\u00ba EGB', '5.\u00ba EGB', '6.\u00ba EGB', '7.\u00ba EGB', '8.\u00ba EGB', '9.\u00ba EGB', '10.\u00ba EGB'],
  bgu: ['1.\u00ba BGU', '2.\u00ba BGU', '3.\u00ba BGU'],
  bt: ['1.\u00ba BT', '2.\u00ba BT', '3.\u00ba BT']
};
const editEgbNocturna = ['8.\u00ba EGB', '9.\u00ba EGB', '10.\u00ba EGB'];
const editCarrerasMap = {
  'Inform\u00e1tica': ['Aplicaciones Inform\u00e1ticas', 'Programaci\u00f3n de Software', 'Soporte
T\u00e9cnico', 'Sistemas Microinform\u00e1ticos y Redes'],
  'Administraci\u00f3n': ['Asistencia Administrativa', 'Gesti\u00f3n Empresarial'],
  'Contabilidad': ['Contabilidad', 'Ventas e Informaci\u00f3n Comercial'],
  'Comercializaci\u00f3n y Ventas': ['Ventas e Informaci\u00f3n Comercial', 'Marketing'],

```

```

'Servicios Hoteleros': ['Hoteler\u00eda', 'Hospitalidad'],
'Turismo': ['Turismo', 'Gu\u00eda Tur\u00edstica'],
'Electromec\u00e9nica Automotriz': ['Electromec\u00e9nica Automotriz', 'Mec\u00e9nica Automotriz'],
'Instalaciones El\u00e9ctricas': ['Electricidad', 'Instalaciones El\u00e9ctricas'],
'Electr\u00f3nica de Consumo': ['Electr\u00f3nica', 'Mantenimiento Electr\u00f3nico'],
'Atenci\u00f3n Integral en Salud': ['Atenci\u00f3n en Enfermer\u00eda', 'Auxiliar de Salud'],
'Producci\u00f3n Agropecuaria': ['Producci\u00f3n Agropecuaria', 'Agroindustria'],
'Redes y Telecomunicaciones': ['Redes', 'Telecomunicaciones'],
'Dise\u00f1o Gr\u00e1fico': ['Dise\u00f1o Gr\u00e1fico', 'Multimedia'],
'Servicios de Belleza': ['Peluquer\u00eda', 'Cosmetolog\u00eda']

```

```
};
```

```

function editDetectType(grade) {
    if (!grade) return '';
    if (grade.includes('Inicial')) return 'inicial';
    if (grade.includes('EGB')) return 'egb';
    if (grade.includes('BGU')) return 'bgu';
    if (grade.includes('BT')) return 'bt';
    return '';
}

function editGetNocturna() {
    var opts = document.getElementById('editShiftId').options;
    for (var i=0; i<opts.length; i++) if ((opts[i].getAttribute('data-shift')||'').toLowerCase()=='nocturna')
return opts[i];
    return null;
}

function editUpdateNocturna(type, grade) {
    var n = editGetNocturna(); if (!n) return;
    var v = (type=='bgu' || type=='bt') || (type=='egb' && editEgbNocturna.includes(grade));
    n.style.display = v ? '' : 'none';
    if (!v && n.selected) { document.getElementById('editShiftId').value=''; editGenerateName(); }
}

function editUpdateGradeLevels(pselect) {
    var type = document.getElementById('editEduType').value;
    var sel = document.getElementById('editGradeLevel');
    sel.innerHTML = '<option value="">Seleccionar grado...</option>';
    if (!type) return;
    (editGradeLevels[type]||[]).forEach(function(g){
        var o=document.createElement('option'); o.value=o.textContent=g;
        if (pselect && g==pselect) o.selected=true;
        sel.appendChild(o);
    });
    var isBT = type=='bt';
    document.getElementById('editGroupSpecialty').style.display = isBT?'block':'none';
    document.getElementById('editGroupCarrera').style.display = isBT?'block':'none';
    document.getElementById('editSpecialty').required = isBT;
    if (!isBT) { document.getElementById('editSpecialty').value='';
document.getElementById('editCarrera').innerHTML='<option value="">Sin especificar</option>'; }
    editUpdateNocturna(type, sel.value);
    editGenerateName();
}

function editOnGradeChange() {

```

```

    editUpdateNocturna(document.getElementById('editEduType').value,
document.getElementById('editGradeLevel').value);
    editGenerateName();
}

function editUpdateCarreras(pselect) {
    var fig = document.getElementById('editSpecialty').value;
    var sel = document.getElementById('editCarrera');
    sel.innerHTML = '<option value="">Sin especificar</option>';
    (editCarrerasMap[fig]||[]).forEach(function(c){
        var o=document.createElement('option'); o.value=o.textContent=c;
        if (pselect && c==pselect) o.selected=true;
        sel.appendChild(o);
    });
    editGenerateName();
}

function editGenerateName() {
    var grade = document.getElementById('editGradeLevel').value;
    var parallel= document.getElementById('editParallel').value;

```

[illegible]

```

<div class="modal-body">
<div class="mf-group">
<label class="mf-label">Nombre de la Asignatura *</label>
<input type="text" name="new_subject_name" id="addSubjectName" class="mf-input"
required placeholder="Ej: Matematicas, Lenguaje..."
oninput="autoGenerateCode(this.value)">
</div>
<div class="mf-group" style="margin-bottom:0px;">
<label class="mf-label">Codigo <span style="color:#aaa;font-weight:400;">(opcional)</span></label>
<input type="text" name="new_subject_code" id="addSubjectCode" class="mf-input"
placeholder="Ej: MAT, LEN...">
</div>
</div>
<div class="modal-footer">
<button type="button" class="btn-cancel" onclick="closeAddSubjectModal()">Cancelar</button>
<button type="submit" class="btn-submit" style="background:linear-gradient(135deg,#6f42c1,#5a32a3);">Agregar</button>
</div>
</form>
</div>
</div>

```

```

<!-- MODAL: ASIGNAR DOCENTE A ASIGNATURA -->
<div id="modalDocente" class="modal-overlay" onclick="if(event.target===this)cerrarModalDocente()">
<div class="modal-box" style="max-width:420px;">
<div class="modal-header">
<div>
<h3 style="color:#e65100;">Asignar Docente</h3>
<p id="modalDocenteSubject" style="color:#888;font-size:12px;"></p>
</div>
<button class="modal-close" onclick="cerrarModalDocente()"></button>
</div>
<form method="POST" action="?action=assign_subject_teacher" id="formDocente">

```

```

<div class="modal-body">
<input type="hidden" name="course_id" id="modalDocenteCourseId">
<input type="hidden" name="subject_id" id="modalDocenteSubjectId">
<input type="hidden" name="assignment_id" id="modalDocenteAssignId">
<div class="mf-group" style="margin-bottom:0px;">
<label class="mf-label">Seleccionar Docente *</label>
<select name="teacher_id" id="modalDocenteSelect" required class="mf-input">
<option value="">Seleccionar docente...</option>
<?php foreach($teachers as $t): ?>
<option value="<?=$t['id']"><?=$t['last_name']. ' '.$t['first_name']">
</option>
<?php endforeach; ?>
</select>
</div>
</div>
<div class="modal-footer">
<button type="button" class="btn-cancel" onclick="cerrarModalDocente()">Cancelar</button>
<button type="submit" class="btn-submit" style="background:#e65100;">Guardar</button>
</div>
</form>
</div>
</div>

```

```

<script>
/* \u2500\u2500 Asignaturas expandible \u2500\u2500 */
function toggleSubjectsRow(courseId) {
var row = document.getElementById('rowSub_' + courseId);
var btn = document.getElementById('btnSub_' + courseId);
if (!row) return;
var isOpen = row.style.display !== 'none';
// Cerrar todos (asignaturas Y estudiantes)
document.querySelectorAll('[id^="rowSub_"]',[id^="rowEst_"]').forEach(function(r){ r.style.display='none'; });
document.querySelectorAll('[id^="btnSub_"]').forEach(function(b){ b.style.background='#6f42c1'; });
document.querySelectorAll('[id^="btnEst_"]').forEach(function(b){ b.style.background='#007bff'; });
}

```

```

if (!isOpen) {
    row.style.display = '';
    btn.style.background = '#4a0e8f';
}
}

```

```

function filterSubjects(input, courseId) {
    var q = normalize(input.value);
    document.querySelectorAll('#tblSub_' + courseId + ' tbody tr').forEach(function(tr){
        tr.style.display = normalize(tr.dataset.subname || '').includes(q) ? '' : 'none';
    });
}

```

```

/* \u2500\u2500 Modal agregar asignatura \u2500\u2500 */
function openAddSubjectModal(courseId, courseName) {
    document.getElementById('formAddSubject').action = '?action=course_subjects&course_id=' + courseId;
    document.getElementById('addSubjectCourseName').textContent = courseName;
    document.getElementById('addSubjectName').value = '';
    document.getElementById('addSubjectCode').value = '';
    document.getElementById('addSubjectName').oninput = function(){ autoGenerateCode(this.value); };
    document.getElementById('modalAddSubject').classList.add('active');
    setTimeout(function(){ document.getElementById('addSubjectName').focus(); }, 150);
}

```

```

function closeAddSubjectModal() {
    document.getElementById('modalAddSubject').classList.remove('active');
}
document.getElementById('modalAddSubject').addEventListener('click', function(e){ if(e.target===this)
closeAddSubjectModal(); });

```

```

/* \u2500\u2500 Modal docente \u2500\u2500 */
function abrirModalDocente(subjectId, subjectName, assignmentId, teacherId, courseId) {
    document.getElementById('modalDocenteSubjectId').value = subjectId;
    document.getElementById('modalDocenteCourseId').value = courseId;
    document.getElementById('modalDocenteAssignId').value = assignmentId || '';
    document.getElementById('modalDocenteSubject').textContent = subjectName;
    var sel = document.getElementById('modalDocenteSelect');
    sel.value = teacherId || '';
    document.getElementById('modalDocente').classList.add('active');
}

```

```

function cerrarModalDocente() {
    document.getElementById('modalDocente').classList.remove('active');
}
document.getElementById('modalDocente').addEventListener('click', function(e){ if(e.target===this)
cerrarModalDocente(); });

```

```

/* \u2500\u2500 Confirmar quitar asignatura \u2500\u2500 */
function confirmRemoveSubject(event, nombre) {
    event.preventDefault();

```

```

showConfirmModal(
    'Quitar asignatura', '#dc3545',
    'Quitar <strong>' + nombre + '</strong> de este curso?',
    '', 'Si, Quitar', event.target
);
return false;
}

```

```

function confirmQuitarDocente(event, nombre) {
    event.preventDefault();
    showConfirmModal(
        'Quitar Docente', '#6c757d',
        'Quitar el docente asignado a <strong>' + nombre + '</strong>?',
        '', 'Sí, Quitar', event.target
    );
    return false;
}

```

```

/* ESC para nuevos modales */
document.addEventListener('keydown', function(e){
    if (e.key === 'Escape') { closeAddSubjectModal(); cerrarModalDocente(); }
});

```

```

/* Auto-Generar código de asignatura */
function autoGenerateCode(nombre) {
  var SKIP = ['y','e','o','a','de','del','la','las','el','los','en','con','por','para','sin','al','un','una','unos','unas','que','se'];
  var palabras = nombre.trim().normalize('NFD').replace(/■-■/g,'').split(/\s+/).filter(Boolean);
  var significativas = palabras.filter(function(p){ return SKIP.indexOf(p.toLowerCase()) === -1; });
  var code = '';
  if (significativas.length === 0) {
    code = '';
  } else if (significativas.length === 1) {
    code = significativas[0].replace(/^[a-zA-Z0-9]/g,'').substring(0, 3);
  } else {
    code = significativas.map(function(p){ return p.replace(/^[a-zA-Z0-9]/g,'').charAt(0); }).join('');
  }
  code = code.toUpperCase();
  document.getElementById('addSubjectCode').value = code;
}
</script>

```

```
<script>
/* \u2500\u2500 Auto-reabrir panel tras POST \u2500\u2500 */
(function(){
var openStudents = <? = isset($_GET['open_students']) ? (int)$_GET['open_students'] : 'null' ?>;
var openSubjects = <? = isset($_GET['open_subjects']) ? (int)$_GET['open_subjects'] : 'null' ?>;
if (openStudents) {
var row = document.getElementById('rowEst_' + openStudents);
var btn = document.getElementById('btnEst_' + openStudents);
if (row) { row.style.display = ''; }
if (btn) { btn.style.background = '#0056b3'; }
}
if (openSubjects) {
var row = document.getElementById('rowSub_' + openSubjects);
var btn = document.getElementById('btnSub_' + openSubjects);
if (row) { row.style.display = ''; }
if (btn) { btn.style.background = '#4a0e8f'; }
}
})();
</script>
```

```
<!-- MODAL: REPRESENTANTES DEL ESTUDIANTE (con script ANTES) -->
```

```
<!-- Script de la función openRepModal (DEFINIDO PRIMERO) -->
<script>
function openRepModal(studentId, studentName, courseId) {
  console.log('openRepModal called', studentId, studentName, courseId); // Para debugging

  // Verificar que los datos existen
  if (!studentId || !studentName || !courseId) {
    console.error('Faltan parámetros en openRepModal');
    return;
  }

  // Asignar valores a los campos del modal
  document.getElementById('repStudentId').value = studentId;
```

```
document.getElementById('repCourseId').value = courseId;
document.getElementById('repModalStudentName').textContent = studentName;
document.getElementById('repSelect').value = '';
document.getElementById('repRelationship').value = '';

// Cargar representantes actuales
var reps = [];
if (typeof smRepsByStudent !== 'undefined' && smRepsByStudent) {
    reps = smRepsByStudent[String(studentId)] || [];
}

// Auto-marcar Principal si no tiene representantes, Secundario si ya tiene
```

```

var chk = document.getElementById('repIsPrimary');
if (chk) {
    chk.checked = (reps.length === 0);
}

var html = '';
if (reps.length === 0) {
    html = '<p style="color:#aaa;font-size:13px;text-align:center;padding:12px 0;">Sin representantes asignados.</p>';
} else {
    html = '<p style="font-size:13px;font-weight:600;color:#333;margin-bottom:10px;">Representantes actuales:</p>';
    reps.forEach(function(r) {
        var badge = r.is_primary
        ? '<span style="background:#17a2b8;color:#fff;font-size:11px;padding:2px 8px;border-radius:10px;margin-left:6px;">Principal</span>'
        : '<span style="background:#e9ecef;color:#555;font-size:11px;padding:2px 8px;border-radius:10px;margin-left:6px;">Secundario</span>';
        var removeUrl = '?action=remove_rep_from_academic&rep_id=' + r.id
        + '&student_id=' + studentId + '&course_id=' + courseId;
        html += '<div style="display:flex;justify-content:space-between;align-items:center;">'
        + '<div>'
        + '<span style="font-size:13px;font-weight:600;">■ ' + (r.last_name || '') + ' ' + (r.first_name || '') + ' '
        + '</span>'
        + badge
        + '<br><span style="font-size:11px;color:#888;">' + (r.relationship || '') + '</span>'
        + '</div>'
        + '<a href="' + removeUrl + '" onclick="return confirm(\'¿Quitar representante?\')"'
        + ' style="padding:4px 10px;font-size:12px;background:#dc3545;color:white;border-radius:4px;text-decoration:none;">Quitar</a>'
        + '</div>';
    });
}

```

```

var currentList = document.getElementById('repCurrentList');
if (currentList) {
    currentList.innerHTML = html;
} else {
    console.error('Elemento repCurrentList no encontrado');
}

```

```

// Mostrar el modal
var modal = document.getElementById('modalRep');
if (modal) {
    modal.classList.add('active');
} else {
    console.error('Modal no encontrado');
}

```

```

function closeRepModal() {
    var modal = document.getElementById('modalRep');
    if (modal) {
        modal.classList.remove('active');
    }
}

```

```

// Event listeners para cerrar modal
document.addEventListener('DOMContentLoaded', function() {
    var modal = document.getElementById('modalRep');
    if (modal) {
        modal.addEventListener('click', function(e) {
            if (e.target === this) closeRepModal();
        });
    }
});

```

```

document.addEventListener('keydown', function(e) {
    if (e.key === 'Escape') {
        closeRepModal();
    }
});

```



```

}
});
</script>

```

```

<!-- Luego el HTML del modal -->
<div id="modalRep" class="modal-overlay">
  <div class="modal-box" style="max-width:560px;">
    <div class="modal-header">
      <div>
        <h3 style="color:#17a2b8;">■■■■■ Representantes</h3>
        <p id="repModalStudentName" style="color:#888;font-size:12px;"></p>
      </div>
      <button class="modal-close" onclick="closeRepModal()">X</button>
    </div>
    <div class="modal-body">
      <!-- Lista de representantes actuales -->
      <div id="repCurrentList" style="margin-bottom:18px;"></div>

      <!-- Formulario nueva asignacion -->
      <div style="border-top:1px solid #f0f0f0;padding-top:16px;">
        <p style="font-size:13px;font-weight:600;color:#333;margin-bottom:12px;">■ Asignar representante</p>
        <form method="POST" action="?action=assign_rep_from_academic" id="formAssignRep">
          <input type="hidden" name="student_id" id="repStudentId">
          <input type="hidden" name="course_id" id="repCourseId">
          <div class="mf-row">
            <div>
              <label class="mf-label">Representante *</label>
              <select name="representative_id" id="repSelect" required class="mf-input">
                <option value="">Seleccionar...</option>
                <?php foreach($representatives ?? [] as $rep): ?>
                  <option value="<?= $rep['id'] ?>">
                    <?= htmlspecialchars($rep['last_name']. ' '.$rep['first_name']) ?>
                    <?= $rep['dni'] ? ' ('.$rep['dni'].')' : '' ?>
                  </option>
                <?php endforeach; ?>
              </select>
            </div>
            <div>
              <label class="mf-label">Parentesco *</label>
              <select name="relationship" id="repRelationship" required class="mf-input">
                <option value="">Seleccionar...</option>
                <?php foreach(['Padre','Madre','Tutor Legal','Abuelo/a','Tío/a','Hermano/a','Otro'] as $rel): ?>
                  <option value="<?= $rel ?>"><?= $rel ?></option>
                <?php endforeach; ?>
              </select>
            </div>
          </div>
          <div style="margin-bottom:14px;">
            <label style="display:flex;align-items:center;gap:8px;cursor:pointer;font-weight:normal;font-size:13px;">
              <input type="checkbox" name="is_primary" id="repIsPrimary" value="1"
                style="width:16px;height:16px;accent-color:#17a2b8;">
              <span>
                <strong>Representante Principal</strong>
                <span style="color:#888;font-size:11px;display:block;">Recibe notificaciones prioritarias</span>
              </span>
            </label>
          </div>
          <button type="submit"
            style="width:100%;padding:9px;background:#17a2b8;color:white;border:none;border-radius:6px;font-size:13px;font-weight:600;cursor:pointer;margin:0;">
            ✓ Asignar Representante
          </button>
        </form>
      </div>
    </div>
    <div class="modal-footer">
      <button type="button" class="btn-cancel" onclick="closeRepModal()">Cerrar</button>
    </div>
  </div>
</div>

```

```
<div id="ac-toast"></div>
```

```
<script>  
/* ■■ Toast académica ████████████████████████ */  
function acToast(msg, type, dur){  
    var t = document.getElementById('ac-toast');  
    t.innerHTML = msg;  
    t.className = 'show ' + (type||'ok');  
    clearTimeout(t._t);  
    t._t = setTimeout(function(){ t.className=''; }, dur||4000);  
}  
  
(function(){  
    var p = new URLSearchParams(location.search);
```

```
// Mensajes ok
var ok = {
  'course_success' : function(){
    var msg = '✓ Curso creado correctamente';
    if(p.get('subjects_loaded')>0) msg += ' – Se pre-cargaron <strong>' + p.get('subjects_loaded') + '</strong> asignaturas';
    else if(p.get('subjects_loaded')!==null) msg += ' – Asignaturas ya estaban registradas';
    return msg;
  },
  'course_updated' : '✓ Curso actualizado correctamente',
  'course_deleted' : '✓ Curso eliminado correctamente',
  'subject_success' : '✓ Asignatura creada correctamente',
  'subject_updated' : '✓ Asignatura actualizada correctamente',
  'subject_deleted' : '✓ Asignatura eliminada correctamente',
  'removed' : '✓ Asignatura quitada del curso',
  'sy_created' : '✓ Año lectivo creado correctamente',
  'sy_updated' : '✓ Año lectivo actualizado correctamente',
  'sy_deleted' : '✓ Año lectivo eliminado correctamente',
  'sy_activated' : '✓ Año lectivo activado correctamente',
  'sy_deactivated' : 'Año lectivo desactivado',
  'hours_updated' : '✓ Horas actualizadas – el horario fue ajustado automáticamente',
  'rep_assigned' : '✓ Representante asignado correctamente',
  'rep_removed' : '✓ Representante eliminado correctamente',
};

var errMap = {
  'no_active_year' : 'No hay un año lectivo activo',
  'course_not_found' : 'Curso no encontrado',
  'course_has_students' : 'No se puede eliminar: el curso tiene estudiantes matriculados',
  'course_has_assignments' : 'No se puede eliminar: el curso tiene asignaciones docentes',
  'course_duplicate' : 'Ya existe un curso con ese nivel, paralelo y jornada',
  'subject_not_found' : 'Asignatura no encontrada',
  'subject_has_assignments' : 'No se puede eliminar: la asignatura tiene asignaciones docentes',
  'cannot_delete_active' : 'No se puede eliminar el año lectivo activo',
  'has_courses' : 'No se puede eliminar: el año lectivo tiene cursos asociados',
  'year_not_found' : 'Año lectivo no encontrado',
  'hours_exceeded' : p.get('msg') || 'No se pueden asignar esas horas: supera el límite semanal',
  'hours_full' : '■ Horario lleno: ' + (p.get('available')||'') + ' horas semanales ya están distribuidas. No se puede agregar otra asignatura.',
};

var fired = false;
for(var k in ok){
  if(p.get(k) !== null){
    var msg = typeof ok[k] === 'function' ? ok[k]() : ok[k];
    var type = k === 'sy_deactivated' ? 'inf' : 'ok';
    acToast(msg, type);
    fired = true; break;
  }
}
if(!fired && p.get('rep_error')){
  acToast('✗ ' + p.get('rep_error'), 'err'); fired = true;
}
if(!fired && p.get('error')){
  var em = errMap[p.get('error')] || ('✗ ' + p.get('error'));
  acToast('✗ ' + em, 'err'); fired = true;
}
if(fired){
```

```

var u = new URL(location.href);
['course_success','course_updated','course_deleted','subject_success','subject_updated',
'subject_deleted','removed','sy_created','sy_updated','sy_deleted','sy_activated',
'sy_deactivated','hours_updated','rep_assigned','rep_removed','rep_error',
'error','subjects_loaded','msg','available','assigned'].forEach(function(k){ u.searchParams.delete(k); });
history.replaceState(null,'',u);
}
})();
</script>
<script>
function updateHrsBadge(input) {
  var courseId = input.dataset.courseId;
  var original = parseInt(input.dataset.original) || 1;
  var newVal = parseInt(input.value) || 0;
  var badge = document.getElementById('hrs-badge-' + courseId);
  if (!badge) return;

  var max = parseInt(badge.dataset.max);
  var total = parseInt(badge.dataset.total);

  // Recalcular: quitar el valor original de este input y sumar el nuevo
  var newTotal = total - original + newVal;

  var icon, color;
  if (newTotal > max) {
    icon = '■'; color = '#dc3545';

} else if (newTotal === max) {
  icon = '✓'; color = '#28a745';
} else {
  icon = '■'; color = '#e0a800';
}

  badge.textContent = icon + ' ' + newTotal + '/' + max + ' hrs';
  badge.style.color = color;
  badge.style.borderColor = color;

  // Marcar el input visualmente si va a exceder
  input.style.borderColor = newTotal > max ? '#dc3545' : '#ddd';
}
</script>
</body>
</html>

```

views/academic/school_year_create.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>Crear Año Lectivo - EcuAsist</title>
</head>
<body>

<?php include BASE_PATH . '/views/partials/navbar.php'; ?>

<div class="breadcrumb">
  <a href="?action=dashboard">■ Inicio</a> &rsquo;
  <a href="?action=academic">Configuración Académica</a> &rsquo;
  Crear Año Lectivo
</div>

<div class="container" style="max-width:600px;">

  <div class="page-header" style="background:linear-gradient(135deg,#1a237e,#283593);">
  <div class="ph-icon">■</div>
</div>

```

```

<h1>Crear Año Lectivo</h1>
<p>Define el período académico</p>
</div>
</div>

<?php if(!empty($errors)): ?>
<div class="alert alert-danger">
<?php foreach($errors as $e): ?>
<div>■■■ <?= htmlspecialchars($e) ?></div>
<?php endforeach; ?>
</div>
<?php endif; ?>

<div class="panel">
<form method="POST" action="?action=create_school_year">

<div class="form-group">

```

```

<label>Nombre del Año Lectivo *</label>
<input type="text" name="name" class="form-control" required
value="<?= htmlspecialchars($_POST['name']) ?? '' ?>"
placeholder="Ej: 2025-2026">
<small style="color:#888;">Formato recomendado: AAAA-AAAA</small>
</div>

<div class="form-row">
<div class="form-group">
<label>Fecha de Inicio *</label>
<input type="date" name="start_date" class="form-control" required
value="<?= htmlspecialchars($_POST['start_date']) ?? '' ?>">
</div>
<div class="form-group">
<label>Fecha de Fin *</label>
<input type="date" name="end_date" class="form-control" required
value="<?= htmlspecialchars($_POST['end_date']) ?? '' ?>">
</div>
</div>

<div class="form-group" style="margin-top:8px;">
<label style="display:flex;align-items:center;gap:10px;cursor:pointer;font-weight:normal;">
<input type="checkbox" name="is_active" value="1"
<?= isset($_POST['is_active']) ? 'checked' : '' ?>
style="width:18px;height:18px;accent-color:#28a745;">
<span>
<strong>Marcar como año lectivo activo</strong><br>
<small style="color:#888;">Solo puede haber un año lectivo activo a la vez</small>
</span>
</label>
</div>

<div style="display:flex;gap:10px;margin-top:20px;">
<button type="submit" class="btn btn-success">■ Crear Año Lectivo</button>
<a href="?action=academic" class="btn btn-outline">Cancelar</a>
</div>

</form>
</div>

```

```

</div>
</body>
</html>

```

views/academic/school_year_edit.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>Editar Año Lectivo - EcuAsist</title>
</head>
<body>

<?php include BASE_PATH . '/views/partials/navbar.php'; ?>

<div class="breadcrumb">
  <a href="?action=dashboard">■ Inicio</a> &rsquo;
  <a href="?action=academic">Configuración Académica</a> &rsquo;
  Editar Año Lectivo
</div>

<div class="container" style="max-width:600px;">

  <div class="page-header" style="background:linear-gradient(135deg,#e65100,#ef6c00);">
    <div class="ph-icon">■</div>
    <div>
      <h1>Editar Año Lectivo</h1>
      <p><?= htmlspecialchars($year['name']) ?></p>
    </div>
  </div>

  <?php if(!empty($errors)): ?>
  <div class="alert alert-danger">
    <?php foreach($errors as $e): ?>
    <div>■■ <?= htmlspecialchars($e) ?></div>
    <?php endforeach; ?>
  </div>
  <?php endif; ?>

  <div class="panel">
    <form method="POST" action="?action=edit_school_year&id=<?= $year['id'] ?>">

      <div class="form-group">

<label>Nombre del Año Lectivo *</label>
<input type="text" name="name" class="form-control" required
value="<?= htmlspecialchars($_POST['name']) ?? $year['name'] ?>"
placeholder="Ej: 2025-2026">
</div>

<div class="form-row">
<div class="form-group">
<label>Fecha de Inicio *</label>
<input type="date" name="start_date" class="form-control" required
value="<?= htmlspecialchars($_POST['start_date']) ?? $year['start_date'] ?>">
</div>
<div class="form-group">
<label>Fecha de Fin *</label>
<input type="date" name="end_date" class="form-control" required
value="<?= htmlspecialchars($_POST['end_date']) ?? $year['end_date'] ?>">
</div>
</div>

<?php if($year['is_active']): ?>
<div class="alert alert-info" style="margin-top:8px;font-size:13px;">
  ■ Este es el <strong>año lectivo activo</strong> actualmente. Para cambiarlo, activa otro año lectivo.
</div>
<?php endif; ?>

<div style="display:flex;gap:10px;margin-top:20px;">
<button type="submit" class="btn btn-success">■ Guardar Cambios</button>
<a href="?action=academic" class="btn btn-outline">Cancelar</a>
</div>

</form>
</div>
</body>
</html>

```

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">■ Editar 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>
    </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>

```

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 { background: #f4f6f9; padding-top: 0; }
.container { max-width: 860px; }
.form-section { background: #fff; border: 1px solid #e8ecf0; border-radius: 10px; padding: 22px 26px;
margin-bottom: 18px; }
.form-section h5 { color: #1a237e; border-bottom: 2px solid #e8ecf0; padding-bottom: 10px; margin-bottom:
18px; font-size: 15px; font-weight: 700; }
.required::after { content: " *"; color: #dc3545; }
.form-control:focus { border-color: #1a237e; box-shadow: 0 0 0 3px rgba(26,35,126,0.1); }
.form-control.is-valid { border-color: #28a745; }
.form-control.is-invalid { border-color: #dc3545; }
.feedback-icon { position: absolute; right: 10px; top: 50%; transform: translateY(-50%); font-size: 14px;
pointer-events: none; }
.input-wrap { position: relative; }
.field-hint { font-size: 11.5px; color: #888; margin-top: 3px; }
.field-error { font-size: 12px; color: #dc3545; margin-top: 3px; display: none; }
.field-error.show { display: block; }
.doc-toggle { display: flex; align-items: center; gap: 10px; margin-bottom: 10px; }
.doc-toggle label { margin: 0; font-size: 13px; color: #555; cursor: pointer; user-select: none; }
</style>
</head>
<body>

<?php include BASE_PATH . '/views/partials/navbar.php'; ?>

<div class="breadcrumb">
<a href="?action=dashboard">■ Inicio</a> &rsquo;
<a href="?action=users">Gestión de Usuarios</a> &rsquo;
Crear Usuario
</div>

<div class="container mt-4">

<!-- Page header -->
<div style="background:linear-gradient(135deg,#1a237e,#3949ab);color:#fff;border-radius:10px;
padding:18px 24px;margin-bottom:20px;display:flex;align-items:center;gap:14px;">
<span style="font-size:28px;">■</span>
<div>
<h1 style="font-size:1.2rem;font-weight:800;margin:0;line-height:1.2;">Crear Nuevo Usuario</h1>
<p style="font-size:12px;opacity:.8;margin:3px 0 0 0;">Completa los datos para registrar un nuevo usuario en
el sistema</p>
</div>
<a href="?action=users" style="margin-left:auto;padding:7px 16px;background:rgba(255,255,255,.18);
color:#fff;text-decoration:none;border-radius:7px;font-size:12px;
border:1px solid rgba(255,255,255,.3);">← Volver</a>
</div>

```

```
<?php if (!empty($errors)): ?>
<div class="alert alert-danger">
<strong>■ Errores:</strong>
<ul class="mb-0 mt-1">
<?php foreach ($errors as $e): ?>
<li><?= htmlspecialchars($e) ?></li>
<?php endforeach; ?>
</ul>
</div>
<?php endif; ?>
```

```
<!-- ■■ Datos de Acceso ████████████████████████████████████████████████████████████ -->
<div class="form-section">
<h5>■■ Datos de Acceso</h5>
<div class="row g-3">
<div class="col-md-6">
<label class="form-label required">Nombre de Usuario</label>
<input type="text" name="username" id="username" class="form-control"
value="<?= htmlspecialchars($_POST['username']) ?? '' ?>"
autocomplete="off" required>
<div class="field-hint">Sin espacios ni caracteres especiales</div>
<div class="field-error" id="err-username"></div>
</div>
```

[illegible]

[illegible]

[illegible][illegible]


```

}

// Contraseñas
var p1 = document.getElementById('password').value;
var p2 = document.getElementById('confirm_password').value;
if (p1.length < 6) {
  setInvalid('password', null, 'err-password', 'Mínimo 6 caracteres'); ok = false;
}
if (p1 !== p2) {
  setInvalid('confirm_password', null, 'err-confirm', 'Las contraseñas no coinciden'); ok = false;
}

// Cédula (solo si no es extranjero y hay valor)
var esExtranjero = document.getElementById('es_extranjero').checked;
if (!esExtranjero) {
  var ced = document.getElementById('cedula').value.trim();
  if (ced !== '' && !validarCedula(ced)) {
    setInvalid('cedula', 'icon-cedula', 'err-cedula', 'Cédula inválida - verifica los dígitos'); ok = false;
  }
}

// Teléfono (si hay valor)
var tel = document.getElementById('phone').value.trim();
if (tel !== '' && !validarTelefono(tel).ok) {
  setInvalid('phone', 'icon-phone', 'err-phone',
    'Formato inválido. Celular: 09XXXXXXXXX • Fijo: 0XXXXXXXXX'); ok = false;
}

if (!ok) e.preventDefault();
});
</script>

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"></script>
</body>
</html>

```

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">■ Editar 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>
```

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: 10px; }
th, td { padding: 10px 12px; text-align: left; border-bottom: 1px solid #ddd; font-size: 13px; }
th { background: #f8f9fa; font-weight: 700; font-size: 12px; text-transform: uppercase; letter-spacing: .04em; color: #555; }
.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:#ff6b6b; }
select, button { padding:8px 12px; 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:15px; }
.error { background:#f8d7da; color:#721c24; padding:10px; border-radius:4px; margin-bottom:15px; }
.info { background:#d1ecf1; color:#0c5460; padding:10px; border-radius:4px; margin-bottom:15px; }
.btn-sm { padding:5px 10px; font-size:12px; margin:2px; }
.action-buttons { white-space:nowrap; }

/* ■■ Buscador ■■ */
.search-bar {
display: flex; gap: 10px; align-items: center; margin-bottom: 16px;
}
.search-bar input {
flex: 1; padding: 9px 14px; border: 1.5px solid #e0e0e0; border-radius: 8px;
font-size: 13px; outline: none; transition: border-color .2s;
}
.search-bar input:focus { border-color: #1a237e; }
#searchClear { display:none; background:#f5f5f5; color:#555; border:1px solid #ddd; border-radius:8px; padding:9px 12px; cursor:pointer; font-size:13px; }

/* ■■ Modal base ■■ */
.modal-overlay {
display:none; position:fixed; inset:0;
background:rgba(0,0,0,.55); z-index:9999;
align-items:center; justify-content:center;
}
.modal-overlay.active { display:flex; }
.modal-box {
background:#fff; border-radius:12px;
width:90%; max-width:620px; max-height:92vh;
display:flex; flex-direction:column;
box-shadow:0 10px 40px rgba(0,0,0,.25);
animation:modalIn .18s ease;
}
@keyframes modalIn { from{transform:translateY(-16px);opacity:0} to{transform:translateY(0);opacity:1} }
.modal-header {
padding:20px 24px 14px; border-bottom:1px solid #f0f0f0;
display:flex; justify-content:space-between; align-items:flex-start; flex-shrink:0;
}
.modal-header h3 { margin:0; font-size:16px; }
.modal-header p { margin:4px 0 0; font-size:12px; color:#888; }
```

```

.modal-close { background:none; border:none; font-size:22px; cursor:pointer; color:#aaa; margin:0;
padding:0; line-height:1; }
.modal-close:hover { color:#333; background:none; }
.modal-body { padding:20px 24px; overflow-y:auto; flex:1; }
.modal-footer {
padding:14px 24px; border-top:1px solid #f0f0f0;
display:flex; justify-content:flex-end; gap:10px; flex-shrink:0;
}
.btn-cancel { padding:9px 20px; background:#f5f5f5; color:#555; border:1px solid #ddd; border-radius:6px;
cursor:pointer; margin:0; font-size:13px; }
.btn-cancel:hover { background:#e8e8e8; }
.btn-submit { padding:9px 22px; background:linear-gradient(135deg,#1a237e,#283593); color:white;
border:none; border-radius:6px; cursor:pointer; margin:0; font-size:13px; font-weight:600; }
.btn-submit:hover { opacity:.88; }

/* ■■ Form fields ■■ */
.mf-label { font-size:12px; font-weight:600; color:#555; margin-bottom:5px; display:block; }
.mf-input {
width:100%; padding:9px 12px;
border:1.5px solid #e0e0e0; border-radius:7px;
font-size:13px; background:#fafafa; color:#333;
outline:none; transition:border-color .2s;
}

.mf-input:focus { border-color:#1a237e; background:#fff; }
.mf-row { display:grid; grid-template-columns:1fr 1fr; gap:14px; margin-bottom:14px; }
.mf-row3 { display:grid; grid-template-columns:1fr 1fr 1fr; gap:12px; margin-bottom:14px; }
.mf-group { margin-bottom:14px; }
.mf-hint { font-size:11px; color:#999; margin-top:4px; }
.mf-section { font-size:12px; font-weight:700; color:#1a237e; text-transform:uppercase;
letter-spacing:.06em; margin:16px 0 10px; padding-bottom:6px; border-bottom:2px solid #e8eaf6; }
.mf-note { background:#fff8e1; border-left:3px solid #ffc107; padding:8px 12px; border-radius:0 6px 6px 0;
font-size:12px; color:#795548; margin-bottom:14px; }
.mf-error { font-size:11px; color:#dc3545; margin-top:3px; display:none; }
.mf-error.show { display:block; }

/* Roles checkboxes */
.roles-grid { display:grid; grid-template-columns:repeat(3,1fr); gap:8px; }
.role-chip {
display:flex; align-items:center; gap:7px;
padding:7px 10px; border:1.5px solid #e0e0e0; border-radius:7px;
cursor:pointer; transition:all .15s; font-size:12px; font-weight:600;
background:#fafafa; user-select:none;
}
.role-chip:hover { border-color:#1a237e; background:#f0f4ff; }
.role-chip input[type=checkbox] { display:none; }
.role-chip.checked { background:#e8eaf6; border-color:#3949ab; color:#1a237e; }
.role-chip .chip-dot { width:8px; height:8px; border-radius:50%; background:#ccc; flex-shrink:0; }
.role-chip.checked .chip-dot { background:#3949ab; }

/* ■■ Toast ■■ */
#um-toast{
position:fixed;top:16px;right:16px;z-index:99999;
padding:11px 20px;border-radius:9px;font-size:13px;font-weight:600;
box-shadow:0 4px 20px rgba(0,0,0,.2);
opacity:0;transform:translateY(-6px);transition:all .22s;
pointer-events:none;max-width:340px;
}
#um-toast.show{opacity:1;transform:translateY(0);}
#um-toast.ok {background:#0d3b2e;color:#06d6a0;}
#um-toast.err {background:#3b0d18;color:#ef476f;}
#um-toast.inf {background:#1a2035;color:#fff;}

/* No results */
#noResults { display:none; text-align:center; padding:30px; color:#aaa; font-size:14px; }
</style>
</head>
<body>
<?php include BASE_PATH . '/views/partials/navbar.php'; ?>
<div class="breadcrumb">
<a href="?action=dashboard">■ Inicio</a> &rsquo; Gestión de Usuarios

```



```

</div>

<?php
// Recuperar datos de sesión para modales (errores)
$modalCreateData = $_SESSION['modal_create_data'] ?? null;
$modalCreateErrors = $_SESSION['modal_create_errors'] ?? null;
$modalEditData = $_SESSION['modal_edit_data'] ?? null;
$modalEditErrors = $_SESSION['modal_edit_errors'] ?? null;
$openModal = $_SESSION['open_modal'] ?? null;
unset($_SESSION['modal_create_data'], $_SESSION['modal_create_errors'],
    $_SESSION['modal_edit_data'], $_SESSION['modal_edit_errors'],
    $_SESSION['open_modal']);
?>

<div class="container">
    <div class="page-header" style="background:linear-gradient(135deg,#2e7d32,#388e3c);">
        <div class="ph-icon">■</div>
        <div>
            <h1>Gestión de Usuarios</h1>
            <p>Administración de docentes, estudiantes y representantes</p>
        </div>
    </div>

    <div class="card">
        <!-- Toast mensajes (se dispara via JS abajo) -->

        <!-- Header + Crear -->
        <div style="display:flex;justify-content:space-between;align-items:center;margin-bottom:16px;">
            <h2 style="font-size:16px;color:#333;">■ Gestión de Usuarios</h2>
            <button onclick="openCreateModal()" style="padding:9px 18px;background:linear-gradient(135deg,#1a237e,#283593);color:white;border:none;border-radius:7px;font-size:13px;font-weight:600;cursor:pointer;">
                + Crear Usuario
            </button>
        </div>

        <!-- Filtro por rol -->
        <?php
        $roleIcons = ['docente'=>'■■■■','estudiante'=>'■■■■','inspector'=>'■','autoridad'=>'■■','representante'=>'■■■■■■■■'];
        $roleColors = [
            'docente' => ['active'=>'background:#007bff;color:white;border-color:#007bff;', '
            outline'=>'color:#007bff;border-color:#007bff;'],
            'estudiante' => ['active'=>'background:#28a745;color:white;border-color:#28a745;', '
            outline'=>'color:#28a745;border-color:#28a745;'],
            'inspector' => ['active'=>'background:#fd7e14;color:white;border-color:#fd7e14;', '
            outline'=>'color:#fd7e14;border-color:#fd7e14;'],
            'autoridad' => ['active'=>'background:#6f42c1;color:white;border-color:#6f42c1;', '
            outline'=>'color:#6f42c1;border-color:#6f42c1;'],
            'representante'=> ['active'=>'background:#20c997;color:white;border-color:#20c997;', '
            outline'=>'color:#20c997;border-color:#20c997;'],
        ];
        $currentFilter = $_GET['filter_role'] ?? '';
        ?>
        <div style="display:flex;gap:8px;flex-wrap:wrap;margin-bottom:14px;">
            <a href="?action=users" style="padding:6px 14px;border-radius:20px;text-decoration:none;font-size:12px;font-weight:600;border:2px solid;<?= $currentFilter==='' ? 'background:#343a40;color:white;border-color:#343a40;' : 'color:#343a40;border-color:#343a40;background:white;' ?>">■ Todos</a>
            <?php foreach($roles as $role):
                $rn=$role['name']; $ico=$roleIcons[$rn]??'■';
                $col=$roleColors[$rn]??['active'=>'background:#007bff;color:white;border-color:#007bff;', 'outline'=>'color:#007bff;border-color:#007bff;'];
                $isActive=($currentFilter=== $rn); ?>
                <a href="?action=users&filter_role=<?= $rn ?>" style="padding:6px 14px;border-radius:20px;text-decoration:none;font-size:12px;font-weight:600;border:2px solid;<?= $isActive?$col['active']:$col['outline'].'background:white;' ?>">
                    <?= $ico ?> <?= ucfirst($rn) ?>
                </a>
            <?php endforeach; ?>
        </div>

        <!-- Buscador -->
        <div class="search-bar">

```

```



```

[illegible]

```

<div id="modalCreate" class="modal-overlay <?= $openModal==='create'? 'active':'' ?>">
  <div class="modal-box" style="max-width:640px;">
    <div class="modal-header">
      <div>
        <h3 style="color:#1a237e;">■ Crear Nuevo Usuario</h3>
        <p>Completa los datos para registrar un usuario</p>
      </div>
      <button class="modal-close" onclick="closeCreateModal()">X</button>
    </div>

    <div class="modal-body">

      <?php if($modalCreateErrors): ?>
      <div class="error" style="margin-bottom:14px;">
        <strong>■ Errores:</strong><ul style="margin:6px 0 0 18px;">
          <?php foreach($modalCreateErrors as $e): ?><li><?= htmlspecialchars($e) ?></li><?php endforeach; ?>
        </ul>
      </div>
      <?php endif; ?>

      <form method="POST" action="?action=create_user_modal" id="formCreate" novalidate>

        <div class="mf-section">■ Datos de Acceso</div>
        <div class="mf-row">
          <div>
            <label class="mf-label">Usuario *</label>
            <input type="text" name="username" class="mf-input" required autocomplete="off"
            value="<?= htmlspecialchars($modalCreateData['username'] ?? '') ?>">
            <div class="mf-hint">Sin espacios ni caracteres especiales</div>
          </div>
          <div>
            <label class="mf-label">Correo Electrónico *</label>
            <input type="email" name="email" class="mf-input" required autocomplete="off"
            value="<?= htmlspecialchars($modalCreateData['email'] ?? '') ?>">
          </div>
        </div>
        <div class="mf-row">
          <div>
            <label class="mf-label">Contraseña *</label>
            <input type="password" name="password" id="cp_password" class="mf-input" minlength="6" required>
            <div class="mf-hint">Mínimo 6 caracteres</div>
          </div>
          <div>
            <label class="mf-label">Confirmar Contraseña *</label>
            <input type="password" name="confirm_password" id="cp_confirm" class="mf-input" minlength="6" required>
            <div class="mf-error" id="cp_confirm_err">Las contraseñas no coinciden</div>
          </div>
        </div>

        <div class="mf-section">■■■ Datos Personales</div>
      </div>
    </div>
  </div>

```

```
<span id="c_icon_cedula" style="position:absolute;right:10px;top:50%;transform:translateY(-50%);font-size:13px;pointer-events:none;"></span>
</div>
<div id="c_err_cedula" class="mf-hint" style="color:#999;">Cédula: 10 dígitos · Pasaporte:
mayúsculas+números, máx 12 car.</div>
</div>
<div>
<label class="mf-label">Teléfono</label>
<div style="position:relative;">
<input type="text" name="phone" id="c_phone" class="mf-input" maxlength="10"
placeholder="09XXXXXXX o 02XXXXXXX"
inputmode="tel"
oninput="this.value=this.value.replace(/[^\d]/g,''); validatePhone('c_phone','c_icon_phone','c_err_phone')"
value="<?=>= htmlspecialchars($modalCreateData['phone']) ?? ' ')" ?>">
<span id="c_icon_phone" style="position:absolute;right:10px;top:50%;transform:translateY(-50%);font-size:13px;pointer-events:none;"></span>
</div>
<div id="c_err_phone" class="mf-hint" style="color:#999;">Celular: 10 dígitos (09...) · Fijo: 9 dígitos
(02-07...)</div>
</div>
</div>

<div class="mf-section">■ Roles (opcional)</div>
<div class="roles-grid">
```

```
<form method="POST" action="?action=edit_user_modal" id="formEdit" novalidate>
<input type="hidden" name="user_id" id="editUserId"
value="<?= htmlspecialchars($modalEditData['user_id'] ?? '') ?>"

<div class="mf-section">■ Datos de Acceso</div>
```

```

<div class="mf-row">
<div>
<label class="mf-label">Usuario *</label>
<input type="text" name="username" id="editUsername" class="mf-input" required
value="<?>= htmlspecialchars($modalEditData['username'] ?? '') ?>">
</div>
<div>
<label class="mf-label">Correo Electrónico *</label>
<input type="email" name="email" id="editEmail" class="mf-input" required
value="<?>= htmlspecialchars($modalEditData['email'] ?? '') ?>">
</div>
</div>
<div class="mf-note">■■■ Deja la contraseña en blanco si no deseas cambiarla</div>
<div class="mf-row">
<div>
<label class="mf-label">Nueva Contraseña</label>
<input type="password" name="password" id="ep_password" class="mf-input" minlength="6">
</div>
<div>
<label class="mf-label">Confirmar Nueva Contraseña</label>
<input type="password" name="confirm_password" id="ep_confirm" class="mf-input" minlength="6">
<div class="mf-error" id="ep_confirm_err">Las contraseñas no coinciden</div>
</div>
</div>

<div class="mf-section">■■■ Datos Personales</div>
<div class="mf-row">
<div>
<label class="mf-label">Nombres *</label>
<input type="text" name="first_name" id="editFirstName" class="mf-input" required
value="<?>= htmlspecialchars($modalEditData['first_name'] ?? '') ?>">
</div>
<div>
<label class="mf-label">Apellidos *</label>
<input type="text" name="last_name" id="editLastName" class="mf-input" required
value="<?>= htmlspecialchars($modalEditData['last_name'] ?? '') ?>">
</div>
</div>
<div class="mf-row">
<div>
<label class="mf-label">Cédula / Pasaporte</label>
<div style="position:relative;">
<input type="text" name="dni" id="editDni" class="mf-input" maxlength="20"
placeholder="Cédula (10 dígitos) o Pasaporte"
oninput="this.value=this.value.toUpperCase(); autoDetectDocEdit()">
<span id="e_icon_cedula" style="position:absolute;right:10px;top:50%;transform:translateY(-50%);font-size:13px;pointer-events:none;"></span>
</div>
<div id="e_err_cedula" class="mf-hint" style="color:#999;">Cédula: 10 dígitos · Pasaporte: mayúsculas+números, máx 12 car.</div>
</div>
<div>
<label class="mf-label">Teléfono</label>
<div style="position:relative;">
<input type="text" name="phone" id="editPhone" class="mf-input" maxlength="10"
placeholder="09XXXXXXXXX o 02XXXXXXXX"
inputmode="tel"
oninput="this.value=this.value.replace(/^[^d]/g,''); validatePhone('editPhone','e_icon_phone','e_err_phone')">
<span id="e_icon_phone" style="position:absolute;right:10px;top:50%;transform:translateY(-50%);font-size:13px;pointer-events:none;"></span>
</div>
<div id="e_err_phone" class="mf-hint" style="color:#999;">Celular: 10 dígitos (09...) · Fijo: 9 dígitos (02-07...)</div>
</div>
</div>
</form>
</div>
<div class="modal-footer">
<button type="button" class="btn-cancel" onclick="closeEditModal()">Cancelar</button>
<button type="submit" form="formEdit" class="btn-submit" id="editSubmitBtn">✓ Guardar Cambios</button>

```

```
</div>
</div>
</div>
```

```
<!-- Data JSON de usuarios para editar sin fetch -->
```

```
<script type="application/json" id="_usersData">
```

```
<?= json_encode(
    array_map(fn($u) => [
```

```
'id' => $u['id'],
'username' => $u['username'],
'email' => $u['email'],
'first_name' => $u['first_name'],
'last_name' => $u['last_name'],
'dni' => $u['dni'] ?? '',
'phone' => $u['phone'] ?? '',
], array_values($users)),
JSON_HEX_TAG|JSON_HEX_AMP
)>>
```

</script>

```
<script>
```

```
var usersData = JSON.parse(document.getElementById('_usersData').textContent);
```

[illegible]

```
function umToast(msg, type, duration){
  var t = document.getElementById('um-toast');
  t.textContent = msg;
  t.className = 'show ' + (type||'ok');
  clearTimeout(t._t);
  t._t = setTimeout(function(){ t.className=''; }, duration||3800);
}
```

```
(function(){
var p = new URLSearchParams(location.search);
var msgs = {
'created': ['✓ Usuario creado correctamente', 'ok'],
'updated': ['✓ Usuario actualizado correctamente', 'ok'],
'deleted': ['✓ Usuario eliminado correctamente', 'ok'],
'deactivated': ['■ Usuario desactivado (tiene registros de asistencia)', 'inf'],
'success': ['✓ Rol asignado correctamente', 'ok'],
'removed': ['✓ Rol eliminado correctamente', 'ok'],
};
// dni_warn se muestra junto al ok, secuencial
if(p.get('dni_warn')){
setTimeout(function(){ umToast('■ Cédula guardada pero no pasa la validación - verificala luego', 'inf',
6000); }, 200);
var u2 = new URL(location.href); u2.searchParams.delete('dni_warn');
history.replaceState(null, '', u2);
}
var errMsgs = {
```

```
'has_assignments': 'X No se puede eliminar el rol docente: tiene asignaciones activas',
'not_found': 'X Usuario no encontrado',
'self_delete': 'X No puedes eliminar tu propia cuenta',
'admin_protected': 'X El usuario administrador no puede ser eliminado',
'delete_failed': 'X Error al eliminar el usuario',
};
var fired = false;
for(var k in msgs){
if(p.get(k)!=null){ umToast(msgs[k][0], msgs[k][1]); fired=true; break; }
}
if(!fired && p.get('error')){
var em = errMsgs[p.get('error')] || ('X ' + p.get('error'));
umToast(em, 'err'); fired=true;
}
if(fired){
var u = new URL(location.href);
['created','updated','deleted','deactivated','success','removed','error'].forEach(function(k){
u.searchParams.delete(k); });
history.replaceState(null,'',u);
}
})();
```

```
/* ■■■ Buscador ■■■ */
function normalize(s){ return (s||' ').normalize('NFD').replace(/[\u0300-\u036f]/g,'').toLowerCase(); }
function filterUsers(){
    var q = normalize(document.getElementById('userSearch').value.trim());
    document.getElementById('searchClear').style.display = q ? '' : 'none';
    var rows = document.querySelectorAll('#usersTable tbody tr[data-search]');
    var visible = 0;
    rows.forEach(function(r){
        var match = !q || normalize(r.dataset.search).includes(q);
        r.style.display = match ? '' : 'none';
        if(match) visible++;
    });
    document.getElementById('noResults').style.display = (visible===0 && q) ? '' : 'none';
}

function clearSearch(){
    document.getElementById('userSearch').value = '';
    filterUsers();
}

/* ■■ Modal Crear ■■■ */
function openCreateModal(){ document.getElementById('modalCreate').classList.add('active'); }
function closeCreateModal(){ document.getElementById('modalCreate').classList.remove('active'); }

/* ■■ Modal Editar ■■■ */
function openEditModal(userId){
    var u = usersData.find(function(x){ return x.id == userId; });
    if(!u) return;
    document.getElementById('editUserId').value = u.id;
    document.getElementById('editUsername').value = u.username;
    document.getElementById('editEmail').value = u.email;
    document.getElementById('editFirst_name').value = u.first_name;
    document.getElementById('editLast_name').value = u.last_name;
    var dni = u.dni || '';
    document.getElementById('editDni').value = dni;
    var ph = u.phone || '';
    document.getElementById('editPhone').value = ph;
    if(ph) validatePhone('editPhone','e_icon_phone','e_err_phone');
    else { document.getElementById('e_icon_phone').textContent='';
document.getElementById('e_err_phone').textContent='Celular: 10 dígitos (09...) · Fijo: 9 dígitos (02-07...)' ; document.getElementById('e_err_phone').style.color='#999'; }
    if(dni) autoDetectDoc(dni, 'e_icon_cedula', 'e_err_cedula');
    else setDocState('e_icon_cedula','e_err_cedula','','Cédula: 10 dígitos · Pasaporte: mayúsculas+números, máx 12 car.',' #999');
    document.getElementById('editModalSubtitle').textContent = u.last_name + ', ' + u.first_name;
    document.getElementById('ep_password').value = '';
    document.getElementById('ep_confirm').value = '';
    document.getElementById('ep_confirm_err').classList.remove('show');
    document.getElementById('modalEdit').classList.add('active');
}

function closeEditModal(){ document.getElementById('modalEdit').classList.remove('active'); }

/* ■■ Cerrar con ESC / clic fuera ■■■ */
document.addEventListener('keydown', function(e){ if(e.key==='Escape'){ closeCreateModal();
closeEditModal(); document.getElementById('modalConfirm').classList.remove('active'); _pendingForm=null; }}};
document.getElementById('modalCreate').addEventListener('click', function(e){ if(e.target===this)
closeCreateModal(); });
document.getElementById('modalEdit').addEventListener('click', function(e){ if(e.target===this)
closeEditModal(); });

/* ■■ Validación contraseña confirmar ■■■ */
document.getElementById('cp_confirm').addEventListener('input', function(){
    var match = this.value === document.getElementById('cp_password').value;
    document.getElementById('cp_confirm_err').classList.toggle('show', !match && this.value!=='');
});
document.getElementById('ep_confirm').addEventListener('input', function(){

var match = this.value === document.getElementById('ep_password').value;
    document.getElementById('ep_confirm_err').classList.toggle('show', !match && this.value!=='');
});

/* ■■ Role chips ■■■ */
```


[illegible]

[illegible]

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>

```

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>
<style>
.rep-block { background:#fff; border:1px solid #e0e0e0; border-radius:8px; padding:16px;
margin-bottom:12px; }
.rep-block-header { display:flex; justify-content:space-between; align-items:center; margin-bottom:12px; }
.rep-name { font-size:0.95rem; font-weight:700; }
.rep-email { font-size:0.8rem; color:#888; }
#toast-container {
position: fixed !important;
top: 20px !important;
right: 20px !important;
z-index: 99999 !important;
display: flex !important;
flex-direction: column;
gap: 10px;
pointer-events: none;
}
.toast {
min-width: 300px;
max-width: 420px;
padding: 14px 18px;
border-radius: 8px;
font-size: 0.88rem;
font-weight: 500;
box-shadow: 0 4px 20px rgba(0,0,0,0.2);
display: flex;
align-items: center;
gap: 10px;
pointer-events: all;
opacity: 0;
transform: translateX(40px);
transition: opacity 0.3s ease, transform 0.3s ease;
}

```

```

.toast.show { opacity: 1; transform: translateX(0); }
.toast.hide { opacity: 0; transform: translateX(40px); }

.toast-success { background: #d4edda; border-left: 5px solid #28a745; color: #155724; }
.toast-danger { background: #f8d7da; border-left: 5px solid #dc3545; color: #721c24; }
.toast-close {
margin-left: auto; background: none; border: none;
cursor: pointer; font-size: 1rem; color: inherit;
opacity: 0.6; padding: 0; line-height: 1;
}
.toast-close:hover { opacity: 1; }
.btn-toggle-primary {
font-size: 0.75rem; padding: 3px 10px; border-radius: 6px; cursor: pointer; border: none;
transition: background 0.2s; text-decoration: none; display: inline-block;
}
.btn-toggle-on { background: #007bff; color: #fff; }
.btn-toggle-off { background: #e9ecef; color: #555; }
.btn-toggle-on:hover { background: #0056b3; }
.btn-toggle-off:hover { background: #dee2e6; }
.actions-cell { display: flex; gap: 6px; align-items: center; flex-wrap: wrap; }
</style>
</head>
<body>

<?php include BASE_PATH . '/views/partials/navbar.php'; ?>

<div class="breadcrumb">
  <a href="?action=dashboard">■ Inicio</a> &rsquo;
  Gestión de Representantes
</div>

<div class="container">

  <?php
  $toastMsg = '';
  $toastType = 'success';
  if (!empty($errorMsg)) { $toastMsg = '✗ ' . $errorMsg; $toastType = 'danger'; }
  elseif (isset($_GET['success'])) { $toastMsg = '✓ Relación asignada correctamente'; }
  elseif (isset($_GET['removed'])) { $toastMsg = '✓ Relación eliminada correctamente'; }
  elseif (isset($_GET['toggled'])) { $toastMsg = '✓ Tipo de representante actualizado'; }
  elseif (isset($_GET['error'])) { $toastMsg = '✗ Error al procesar la solicitud'; $toastType = 'danger'; }
  ?>

<!-- Header -->
<div class="page-header blue">
  <div class="ph-icon">■</div>
  <div>
    <h1>Gestión de Representantes</h1>
    <p>Vincula representantes con sus estudiantes</p>
  </div>
  <div style="margin-left: auto;">
    <button class="btn btn-success" onclick="openModalNuevaRelacion()">■ Nueva Relación</button>
  </div>
</div>

<!-- Filtros -->
<div class="filters" style="margin-bottom: 12px;">
  <h3 style="font-size: 0.85rem; font-weight: 600; color: #555; margin-bottom: 8px;">■ Buscar</h3>
  <div class="filter-grid">
    <div>
      <label style="font-size: 0.78rem; color: #666; display: block; margin-bottom: 4px;">Representante</label>
      <input type="text" id="filterRep" class="form-control" placeholder="Nombre..." oninput="applyFilters()">
    </div>
    <div>
      <label style="font-size: 0.78rem; color: #666; display: block; margin-bottom: 4px;">Estudiante</label>
      <input type="text" id="filterStu" class="form-control" placeholder="Nombre..." oninput="applyFilters()">
    </div>
    <div>
      <label style="font-size: 0.78rem; color: #666; display: block; margin-bottom: 4px;">Curso</label>
      <select id="filterCourse" class="form-control" onchange="applyFilters()">
        <option value="">Todos los cursos...</option>
      </select>
    </div>
  </div>
</div>

```

```

</div>
</div>
<button class="btn btn-outline btn-sm" style="margin-top:8px;" onclick="clearFilters()">■ Limpiar</button>
</div>

```

```

<!-- Relaciones -->
<div id="rep-list">
<?php
$hasRelations = false;
foreach($representatives as $rep):
$children = $this->representativeModel->getStudentsByRepresentative($rep['id']);

```

```

if(!count($children)) continue;
$hasRelations = true;
?>
<div class="rep-block"
data-repname="<? = htmlspecialchars($rep['last_name'] . ' ' . $rep['first_name']) ?>"
<div class="rep-block-header">
<div>
<div class="rep-name">■ <? = htmlspecialchars($rep['last_name'] . ' ' . $rep['first_name']) ?></div>
<div class="rep-email"><? = htmlspecialchars($rep['email']) ?></div>
</div>
<span class="badge badge-blue"><? = count($children) ?> representado(s)</span>
</div>
<div class="table-wrap" style="margin-bottom:0;">
<table style="table-layout:fixed;width:100%;">
<colgroup>
<col style="width:25%;">
<col style="width:13%;">
<col style="width:32%;">
<col style="width:15%;">
<col style="width:15%;">
</colgroup>
<thead>
<tr>
<th>Estudiante</th>
<th>Parentesco</th>
<th>Curso</th>
<th>Tipo</th>
<th>Acciones</th>
</tr>
</thead>
<tbody>
<?php foreach($children as $child): ?>
<tr class="student-row"
data-student="<? = htmlspecialchars($child['last_name'] . ' ' . $child['first_name']) ?>"
data-course="<? = htmlspecialchars($child['course_name'] ?? '') ?>"
<td><strong><? = htmlspecialchars($child['last_name'] . ' ' . $child['first_name']) ?></strong></td>
<td><? = htmlspecialchars($child['relationship']) ?></td>
<td>
<?php if($child['course_name']): ?>
<span class="badge badge-teal"><? = htmlspecialchars($child['course_name']) ?></span>

```

```

<?php else: ?>
<span style="color:#ccc;">Sin curso</span>
<?php endif: ?>
</td>
<td>
<?php if($child['is_primary']): ?>
<span class="badge badge-blue">■ Principal</span>
<?php else: ?>
<span class="badge badge-red">■ Secundario</span>
<?php endif: ?>
</td>
<td>
<div class="actions-cell">
<!-- Toggle Principal / Secundario -->
<a href="?action=toggle_primary_representative&rep_id=<? = $rep['id'] ?>&student_id=<? = $child['id'] ?>"
class="btn-toggle-primary btn-toggle-on"
title="<? = $child['is_primary'] ? 'Cambiar a Secundario' : 'Cambiar a Principal' ?>"
<? = $child['is_primary'] ? '■' : '■' ?>
</a>

```

```

<!-- Editar -->
<button class="btn btn-warning btn-sm"
onclick="openEdit(<?= $rep['id'] ?>, <?= $child['id'] ?>,
'<?= addslashes(htmlspecialchars($rep['last_name']) . ' ' . $rep['first_name'])) ?>',
'<?= addslashes(htmlspecialchars($child['last_name']) . ' ' . $child['first_name'])) ?>',
'<?= addslashes(htmlspecialchars($child['relationship'])) ?>',
<?= $child['is_primary'] ? 1 : 0 ?>)"
title="Editar relación">■</button>
<!-- Eliminar -->
<button class="btn btn-danger btn-sm"
onclick="confirmRemove(<?= $rep['id'] ?>, <?= $child['id'] ?>,
'<?= addslashes(htmlspecialchars($rep['last_name']) . ' ' . $rep['first_name'])) ?>',
'<?= addslashes(htmlspecialchars($child['last_name']) . ' ' . $child['first_name'])) ?>',
title="Eliminar relación">X</button>
</div>
</td>
</tr>
<?php endforeach; ?>
</tbody>
</table>
</div>

```

```

</div>
<?php endforeach; ?>

<?php if(!$hasRelations): ?>
<div class="empty-state">
<div class="icon">■</div>
<p>No hay relaciones representante-estudiante registradas.</p>
<button class="btn btn-success" style="margin-top:12px;" onclick="openModalNuevaRelacion()">■ Crear primera
relación</button>
</div>
<?php endif; ?>
</div>

```

```

<!-- ===== MODAL: Nueva Relación ===== -->
<div class="modal-overlay" id="modalNuevaRelacion">
<div class="modal-box" style="max-width:480px;">
<h3>■ Nueva Relación</h3>
<form method="POST" style="margin-top:16px;">
<div class="form-group">
<label>Representante *</label>
<select name="representative_id" class="form-control" required>
<option value="">Seleccionar representante...</option>
<?php foreach($representatives as $rep): ?>
<option value="<?= $rep['id'] ?>">
<?= htmlspecialchars($rep['last_name']) . ' ' . $rep['first_name']) ?>
(<?= htmlspecialchars($rep['dni']) ?? 'Sin cédula') ?>)
</option>
<?php endforeach; ?>
</select>
</div>

<div class="form-group">
<label>Parentesco *</label>
<select name="relationship" class="form-control" required>
<option value="">Seleccionar...</option>
<?php foreach(['Padre','Madre','Tutor Legal','Abuelo/a','Tío/a','Hermano/a','Otro'] as $rel): ?>
<option><?= $rel ?></option>
<?php endforeach; ?>
</select>

```

```

<small style="color:#888;font-size:0.77rem;">
■ Solo puede haber un Padre y una Madre por estudiante.
</small>
</div>

<div class="form-group">
<label>Estudiante *</label>
<select name="student_id" class="form-control" required>

```

```

<option value="">Seleccionar estudiante...</option>
<?php foreach($students as $s): ?>
<option value="<?=$s['id'] ?>">
<?=$s['last_name'] . ' ' . $s['first_name'] ?>
</option>
<?php endforeach; ?>
</select>
</div>

<div class="form-group">
<label style="display:flex;align-items:center;gap:8px;cursor:pointer;">
<input type="checkbox" name="is_primary" value="1"> Representante Principal
</label>
<small style="color:#888;font-size:0.77rem;">
El representante principal recibe las notificaciones prioritarias.
</small>
</div>

<div class="modal-actions">
<button type="button" class="btn btn-outline" onclick="closeModal('modalNuevaRelacion')">Cancelar</button>
<button type="submit" class="btn btn-success">■ Asignar Relación</button>
</div>
</form>
</div>
</div>

<!-- Modal confirmar eliminar -->
<div class="modal-overlay" id="modalRemove">
<div class="modal-box" style="max-width:460px;">
<h3>■ Eliminar Relación</h3>
<div id="modalRemoveBody" style="margin:12px 0;color:#555;font-size:0.88rem;"></div>
<div class="modal-actions">
<button class="btn btn-outline" onclick="closeModal('modalRemove')">Cancelar</button>
<a id="confirmRemoveLink" href="#" class="btn btn-danger">✗ Sí, Eliminar</a>
</div>
</div>
</div>

<!-- Modal editar relación -->
<div class="modal-overlay" id="modalEdit">
<div class="modal-box" style="max-width:460px;">
<h3>■ Editar Relación</h3>
<form method="POST" id="formEdit">
<input type="hidden" name="representative_id" id="editRepId">
<input type="hidden" name="student_id" id="editStuId">
<div style="margin:12px 0;padding:10px;background:#f8f9fa;border-radius:6px;font-size:0.88rem;">
<strong>Representante:</strong> <span id="editRepName"></span><br>
<strong>Estudiante:</strong> <span id="editStuName"></span>
</div>
<div class="form-group">
<label>Parentesco *</label>
<select name="relationship" id="editRelationship" class="form-control" required>
<option>Padre</option>
<option>Madre</option>
<option>Tutor Legal</option>
<option>Abuelo/a</option>
<option>Tío/a</option>
<option>Hermano/a</option>
<option>Otro</option>
</select>
</div>
<div class="form-group">
<label style="display:flex;align-items:center;gap:8px;cursor:pointer;">
<input type="checkbox" name="is_primary" id="editIsPrimary" value="1"> Representante Principal
</label>
</div>
<div class="modal-actions">
<button type="button" class="btn btn-outline" onclick="closeModal('modalEdit')">Cancelar</button>
<button type="submit" class="btn btn-success">✓ Guardar Cambios</button>
</div>
</form>

```


[illegible]

[illegible]

views/representatives/my_children.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 Representados - EcuAsist</title>
  <style>
    .children-grid { display: grid; grid-template-columns: repeat(auto-fill, minmax(300px, 1fr)); gap: 16px; }
    .child-card { background: #fff; border: 1px solid #e0e0e0; border-radius: 10px; padding: 20px; border-top:
4px solid #007bff; }
    .child-card h3 { font-size: 1rem; font-weight: 700; color: #007bff; margin-bottom: 16px; }
    .child-info { display: flex; flex-direction: column; gap: 8px; }
    .child-row { display: flex; font-size: 0.85rem; padding: 6px 0; border-bottom: 1px solid #f0f0f0; }
    .child-row:last-child { border-bottom: none; }
    .child-label { font-weight: 600; color: #888; width: 100px; flex-shrink: 0; }
    .child-val { color: #333; }
  </style>
</head>
<body>

<?php include BASE_PATH . '/views/partials/navbar.php'; ?>

<div class="breadcrumb">
  <a href="?action=dashboard">■ Inicio</a> &rsquo;
  Mis Representados
</div>

<div class="container">

  <!-- Header -->
  <div class="page-header teal">
    <div class="ph-icon">■■■■■■■■■</div>
    <div>
      <h1>Mis Representados</h1>
      <p>Estudiantes vinculados a tu cuenta como representante</p>
    </div>
  </div>

  <?php if(empty($children)): ?>
  <div class="empty-state">

<div class="icon">■</div>
<p>No tiene estudiantes asignados como representante.</p>
<small style="color:#bbb;">Contacte con la autoridad para vincular a sus representados.</small>
</div>
<?php else: ?>
<div class="children-grid">
<?php foreach($children as $child): ?>
<div class="child-card">
<h3>■ <?= htmlspecialchars($child['last_name']) . ' ' . $child['first_name'] ?>
<?php if($child['is_primary']): ?>
<span class="badge badge-yellow" style="font-size:0.7rem;margin-left:6px;">Principal</span>
<?php endif; ?>
</h3>
<div class="child-info">
<div class="child-row">
<span class="child-label">Cédula</span>
<span class="child-val"><?= htmlspecialchars($child['dni']) ?? 'No registrada' ?></span>
</div>
<div class="child-row">
<span class="child-label">Parentesco</span>
<span class="child-val"><?= htmlspecialchars($child['relationship']) ?></span>
</div>
<div class="child-row">
<span class="child-label">Curso</span>
<span class="child-val"><?= htmlspecialchars($child['course_name']) ?? 'Sin asignar' ?></span>
</div>
<div class="child-row">
```

```

<span class="child-label">Jornada</span>
<span class="child-val"><?=$child['shift_name'] ? ucfirst($child['shift_name']) : '-' ?></span>
</div>
</div>
<div style="margin-top:16px;">
<a href="?action=child_attendance&student_id=<?=$child['id'] ?>" class="btn btn-primary">
■ Ver Asistencia
</a>
</div>
</div>
<?php endforeach; ?>
</div>
<?php endif; ?>

</div>
</body>
</html>

```

views/tutor/course_attendance.php

```

<?php
// views/tutor/course_attendance.php

$statusLabels = [
    'presente' => ['label' => 'Presente', 'color' => '#28a745', 'bg' => '#d4edda'],
    'ausente' => ['label' => 'Ausente', 'color' => '#dc3545', 'bg' => '#f8d7da'],
    'tardanza' => ['label' => 'Tardanza', 'color' => '#ffc107', 'bg' => '#fff3cd'],
    'justificado' => ['label' => 'Justificado', 'color' => '#17a2b8', 'bg' => '#d1ecf1'],
];

$pct = function($part, $total) {
    return $total > 0 ? round(($part / $total) * 100, 1) : 0;
};

$hasFilters = !empty($filters['subject_id']) || !empty($filters['student_id'])
    || !empty($filters['status']) || !empty($filters['start_date'])
    || !empty($filters['end_date']);
?>

<!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>
    <style>
        * { box-sizing: border-box; margin: 0; padding: 0; }
        body { font-family: -apple-system, BlinkMacSystemFont, 'Segoe UI', sans-serif; background: #f4f6f9; color: #333; font-size: 14px; }
        .breadcrumb { padding: 10px 24px; background: #fff; border-bottom: 1px solid #e0e0e0; font-size: 0.83rem; color: #888; }
        .breadcrumb a { color: #007bff; text-decoration: none; }
        .container { max-width: 1200px; margin: 24px auto; padding: 0 16px; }

        /* Page header estándar */
        .page-header { background: linear-gradient(135deg, #007bff, #0056b3); color: #fff; border-radius: 10px; padding: 20px 24px; margin-bottom: 20px; display: flex; align-items: center; gap: 16px; }
        .page-header .ph-icon { font-size: 2.2rem; }
        .page-header h1 { font-size: 1.3rem; font-weight: 700; }
        .page-header p { font-size: 0.85rem; opacity: 0.85; margin-top: 4px; }

        /* Stats */
        #stats-section { display: flex; gap: 12px; margin-bottom: 20px; flex-wrap: wrap; }
        .stat-card { background: #fff; border: 1px solid #e0e0e0; border-radius: 8px; padding: 14px 18px; flex: 1; min-width: 120px; border-top: 4px solid #ddd; }
        .stat-card .number { font-size: 1.8rem; font-weight: 700; }
        .stat-card .label { font-size: 0.75rem; color: #888; margin-top: 2px; text-transform: uppercase; letter-spacing: 0.5px; }

        .stat-card.total { border-top-color: #6c757d; } .stat-card.total .number { color: #6c757d; }
        .stat-card.presente { border-top-color: #28a745; } .stat-card.presente .number { color: #28a745; }

```

```

.stat-card.ausente { border-top-color:#dc3545; } .stat-card.ausente .number { color:#dc3545; }
.stat-card.tardanza { border-top-color:#ffc107; } .stat-card.tardanza .number { color:#e6ac00; }
.stat-card.justif { border-top-color:#17a2b8; } .stat-card.justif .number { color:#17a2b8; }
.progress-bar { height:8px; background:#e9ecef; border-radius:4px; overflow:hidden; margin-top:6px; }
.progress-fill { height:100%; border-radius:4px; transition:width 0.4s ease; }

/* Top ausencias */
.top-absences { background:#fff; border:1px solid #e0e0e0; border-radius:8px; padding:16px;
margin-bottom:20px; }
.top-absences h3 { font-size:0.9rem; font-weight:600; margin-bottom:12px; color:#dc3545; }
.absence-item { display:flex; align-items:center; gap:10px; padding:6px 0; border-bottom:1px solid #f0f0f0; }
.absence-item:last-child { border-bottom:none; }
.absence-name { flex:1; font-size:0.85rem; }
.absence-count { font-weight:700; color:#dc3545; font-size:0.95rem; min-width:28px; text-align:right; }
.absence-bar { flex:2; }

/* Filtros */
.filters { background:#fff; border:1px solid #e0e0e0; border-radius:8px; padding:16px; margin-bottom:16px; }
.filters h3 { font-size:0.85rem; font-weight:600; color:#555; margin-bottom:12px; }
.filter-grid { display:grid; grid-template-columns:repeat(auto-fill,minmax(180px,1fr)); gap:10px; }
.filter-grid select,
.filter-grid input { width:100%; padding:7px 10px; border:1px solid #ccc; border-radius:6px;
font-size:0.85rem; transition:border-color 0.2s; }
.filter-grid select:focus,
.filter-grid input:focus { border-color:#007bff; outline:none; box-shadow:0 0 2px rgba(0,123,255,0.15); }
.filter-grid select.on,
.filter-grid input.on { border-color:#007bff; background:#f0f7ff; }
.filter-banner { background:#fff3cd; border:1px solid #ffc107; border-radius:6px; padding:7px 12px;
font-size:0.82rem; color:#856404; margin-bottom:10px; display:flex; justify-content:space-between;
align-items:center; }
.filter-banner a { color:#856404; font-weight:700; text-decoration:none; cursor:pointer; }
.btn { padding:7px 16px; border-radius:6px; border:none; cursor:pointer; font-size:0.85rem;
font-weight:500; text-decoration:none; display:inline-flex; align-items:center; gap:5px; transition:opacity
0.15s; }
.btn:hover { opacity:0.85; }
.btn-outline { background:#fff; color:#555; border:1px solid #ccc; }

/* Tabla */
.table-section { position:relative; }
.loading-overlay { display:none; position:absolute; inset:0; background:rgba(255,255,255,0.75);
border-radius:8px; z-index:5; align-items:center; justify-content:center; }
.loading-overlay.on { display:flex; }
.spinner { width:28px; height:28px; border:3px solid #e0e0e0; border-top-color:#007bff; border-radius:50%;
animation:spin 0.7s linear infinite; }
@keyframes spin { to { transform:rotate(360deg); } }
.table-wrap { background:#fff; border:1px solid #e0e0e0; border-radius:8px; overflow:hidden; }

.table-info { padding:12px 16px; border-bottom:1px solid #e0e0e0; font-size:0.83rem; color:#666;
display:flex; justify-content:space-between; align-items:center; }
table { width:100%; border-collapse:collapse; }
thead th { background:#f8f9fa; padding:10px 12px; text-align:left; font-size:0.8rem; font-weight:600;
color:#555; border-bottom:2px solid #e0e0e0; white-space:nowrap; }
tbody td { padding:9px 12px; border-bottom:1px solid #f0f0f0; font-size:0.85rem; }
tbody tr:hover { background:#f8fbff; }
tbody tr:last-child td { border-bottom:none; }
.status-badge { display:inline-block; padding:2px 10px; border-radius:12px; font-size:0.75rem;
font-weight:600; }
.empty-state { text-align:center; padding:50px 20px; color:#999; }
.empty-state .icon { font-size:2.5rem; margin-bottom:10px; }
</style>
</head>
<body>

<?php include BASE_PATH . '/views/partials/navbar.php'; ?>

<div class="breadcrumb">
<a href="?action=dashboard">■ Inicio</a> &rsquo;
<a href="?action=dashboard">Mi Dashboard</a> &rsquo;
Asistencia de Mi Curso
</div>

```

```
<div class="container">

<div class="page-header" style="background:linear-gradient(135deg,#1565c0,#1976d2);">
<div class="ph-icon">&#x26A0;/div>
<div>
<h1>Asistencia de Mi Curso - <?= htmlspecialchars($course['name']) ?></h1>
<p>
Jornada: <?= htmlspecialchars($course['shift_name']) ?> &nbsp;&nbsp;&nbsp;;
Nivel: <?= htmlspecialchars($course['grade_level']) ?> &nbsp;&nbsp;&nbsp;;
Paralelo: <?= htmlspecialchars($course['parallel']) ?>
</p>
</div>
</div>

<!-- Stats - actualizado por AJAX -->
<div id="stats-section">
<?php if ($stats && $stats['total'] > 0): ?>
<div class="stat-card total"><div class="number"><?= $stats['total'] ?></div><div class="label">Total registros</div></div>
<div class="stat-card presente">
<div class="number"><?= $stats['presente'] ?></div>
<div class="label">Presentes (<%= $pct($stats['presente'], $stats['total']) %>)</div>
<div class="progress-bar"><div class="progress-fill" style="width:<%= $pct($stats['presente'], $stats['total']) %>;background:#28a745;"></div></div>
</div>
<div class="stat-card ausente">
<div class="number"><%= $stats['ausente'] ?></div>
<div class="label">Ausentes (<%= $pct($stats['ausente'], $stats['total']) %>)</div>
<div class="progress-bar"><div class="progress-fill" style="width:<%= $pct($stats['ausente'], $stats['total']) %>;background:#dc3545;"></div></div>
</div>
<div class="stat-card tardanza">
<div class="number"><%= $stats['tardanza'] ?></div>
<div class="label">Tardanzas (<%= $pct($stats['tardanza'], $stats['total']) %>)</div>
<div class="progress-bar"><div class="progress-fill" style="width:<%= $pct($stats['tardanza'], $stats['total']) %>;background:#ffc107;"></div></div>
</div>
<div class="stat-card justificif">
<div class="number"><%= $stats['justificado'] ?></div>
<div class="label">Justificados (<%= $pct($stats['justificado'], $stats['total']) %>)</div>
<div class="progress-bar"><div class="progress-fill" style="width:<%= $pct($stats['justificado'], $stats['total']) %>;background:#17a2b8;"></div></div>
</div>
<?php endif; ?>
</div>

<!-- Top ausencias (estático) -->
<?php if (!empty($topAbsences)): ?>
<div class="top-absences">
<h3>&#x26A0&#x26A0 Top estudiantes con más ausencias</h3>
<?php $maxAbs = $topAbsences[0]['total_ausencias'] ?? 1;
foreach ($topAbsences as $i => $row): ?>
<div class="absence-item">
<span style="font-size:0.75rem;color:#999;min-width:18px;"><%= $i+1 ?>.</span>
<span class="absence-name"><?= htmlspecialchars($row['student_name']) ?></span>
<div class="absence-bar"><div class="progress-bar"><div class="progress-fill" style="width:<%= round(($row['total_ausencias']/ $maxAbs)*100) %>;background:#dc3545;"></div></div></div>
<span class="absence-count"><%= $row['total_ausencias'] ?></span>
</div>
<?php endforeach; ?>
</div>
<?php endif; ?>

<!-- Filtros - con name para fallback, interceptados por JS -->
<div class="filters">
```

```

<div class="filter-grid">
<div>
<label style="font-size:0.78rem;color:#666;display:block;margin-bottom:4px;">Asignatura</label>
<select name="subject_id" id="f_subject" onchange="runFilter()" class="<?!= !empty($filters['subject_id']) ? 'on' : '' ?>">
<option value="">- Todas -</option>
<?php foreach ($subjects as $s): ?>
<option value="<?!= $s['id'] ?>" <?!= ($filters['subject_id'] ?? '') == $s['id'] ? 'selected' : '' ?>>
<?!= htmlspecialchars($s['name']) ?>
</option>
<?php endforeach; ?>
</select>
</div>
<div>
<label style="font-size:0.78rem;color:#666;display:block;margin-bottom:4px;">Estudiante</label>
<select name="student_id" id="f_student" onchange="runFilter()" class="<?!= !empty($filters['student_id']) ? 'on' : '' ?>">
<option value="">- Todos -</option>
<?php foreach ($students as $st): ?>
<option value="<?!= $st['id'] ?>" <?!= ($filters['student_id'] ?? '') == $st['id'] ? 'selected' : '' ?>>
<?!= htmlspecialchars($st['full_name']) ?>
</option>
<?php endforeach; ?>
</select>
</div>
<div>
<label style="font-size:0.78rem;color:#666;display:block;margin-bottom:4px;">Estado</label>
<select name="status" id="f_status" onchange="runFilter()" class="<?!= !empty($filters['status']) ? 'on' : '' ?>">
<option value="">- Todos -</option>
<option value="presente" <?!= ($filters['status'] ?? '') === 'presente' ? 'selected' : '' ?>
>>Presente</option>
<option value="ausente" <?!= ($filters['status'] ?? '') === 'ausente' ? 'selected' : '' ?>>Ausente</option>
<option value="tardanza" <?!= ($filters['status'] ?? '') === 'tardanza' ? 'selected' : '' ?>
>>Tardanza</option>
<option value="justificado" <?!= ($filters['status'] ?? '') === 'justificado' ? 'selected' : '' ?>
>>Justificado</option>
</select>
</div>
<div>
<label style="font-size:0.78rem;color:#666;display:block;margin-bottom:4px;">Desde</label>
<input type="date" name="start_date" id="f_start" onchange="runFilter()"
value="<?!= htmlspecialchars($filters['start_date'] ?? '') ?>"
class="<?!= !empty($filters['start_date']) ? 'on' : '' ?>">
</div>
<div>
<label style="font-size:0.78rem;color:#666;display:block;margin-bottom:4px;">Hasta</label>
<input type="date" name="end_date" id="f_end" onchange="runFilter()"
value="<?!= htmlspecialchars($filters['end_date'] ?? '') ?>"
class="<?!= !empty($filters['end_date']) ? 'on' : '' ?>">
</div>
</div>

<!-- Tabla - actualizada por AJAX -->
<div class="table-section">
<div class="loading-overlay" id="loader"><div class="spinner"></div></div>
<div class="table-wrap">
<div class="table-info">
<span>■ <strong id="result-count"><?!= count($attendances) ?></strong> registros encontrados</span>
<span id="filter-tag" style="color:#856404;font-size:0.8rem;display:<?!= $hasFilters ? 'inline' : 'none' ?>
>,">■ Filtros activos</span>
</div>
<div id="table-body">
<?php if (empty($attendances)): ?>
<div class="empty-state"><div class="icon">■</div><p>No hay registros de asistencia.</p></div>
<?php else: ?>
<table>
<thead><tr>
<th>Fecha</th><th>Hora</th><th>Estudiante</th>
<th>Asignatura</th><th>Docente</th><th>Estado</th><th>Observación</th>

```

```

</tr></thead>
<tbody>
<?php foreach ($attendances as $a):
$sl = $statusLabels[$a['status']] ?? ['label'=>$a['status'],'color'=>'#666','bg'=>'#eee'];
?>
<tr>
<td><? date('d/m/Y', strtotime($a['date'])) ?></td>
<td style="white-space:nowrap;"><?= htmlspecialchars($a['hour_period']) ?></td>
<td><?= htmlspecialchars($a['student_name']) ?></td>
<td><?= htmlspecialchars($a['subject_name']) ?></td>
<td style="color:#666;"><?= htmlspecialchars($a['teacher_name']) ?></td>

<td><span class="status-badge" style="background:<? $sl['bg'] ?>;color:<? $sl['color'] ?>;"><? $sl['label'] ?></span></td>
<td style="color:#888;font-size:0.82rem;"><?= htmlspecialchars($a['observation']) ?? '-' ?></td>
</tr>
<?php endforeach; ?>
</tbody>
</table>
<?php endif; ?>
</div>
</div>
</div>

<script>
var SL = {
  'presente': { label:'Presente', color:'#28a745', bg:'#d4edda' },
  'ausente': { label:'Ausente', color:'#dc3545', bg:'#f8d7da' },
  'tardanza': { label:'Tardanza', color:'#ffc107', bg:'#fff3cd' },
  'justificado': { label:'Justificado', color:'#17a2b8', bg:'#d1ecf1' },
};

function getVals() {
  return {
    subject_id: document.getElementById('f_subject').value,
    student_id: document.getElementById('f_student').value,
    status: document.getElementById('f_status').value,
    start_date: document.getElementById('f_start').value,
    end_date: document.getElementById('f_end').value,
  };
}

function hasActive(v) {
  return v.subject_id || v.student_id || v.status || v.start_date || v.end_date;
}

function runFilter() {
  var v = getVals();

  // Marcar campos activos visualmente

  ['f_subject','f_student','f_status','f_start','f_end'].forEach(function(id){
    var el = document.getElementById(id);
    el.classList.toggle('on', !!el.value);
  });

  // Mostrar/ocultar banner y tag
  var active = hasActive(v);
  document.getElementById('filter-banner').style.display = active ? 'flex' : 'none';
  document.getElementById('filter-tag').style.display = active ? 'inline' : 'none';

  // Spinner
  document.getElementById('loader').classList.add('on');

  // Construir URL para el endpoint AJAX
  var p = new URLSearchParams({ action: 'tutor_course_attendance_ajax' });
  if (v.subject_id) p.set('subject_id', v.subject_id);
  if (v.student_id) p.set('student_id', v.student_id);
  if (v.status) p.set('status', v.status);

```



```

if (v.start_date) p.set('start_date', v.start_date);
if (v.end_date) p.set('end_date', v.end_date);

fetch('? ' + p.toString(), { credentials: 'same-origin' })
.then(function(r){ return r.json(); })
.then(function(data){
renderStats(data.stats, data.total);
renderTable(data.attendances);
document.getElementById('loader').classList.remove('on');
})
.catch(function(){
document.getElementById('loader').classList.remove('on');
});
});
}

function renderStats(s, total) {
var sec = document.getElementById('stats-section');
if (!s) return;
var t = parseInt(total) || 0;
var pre = parseInt(s.presente) || 0;
var aus = parseInt(s.ausente) || 0;
var tar = parseInt(s.tardanza) || 0;

var jus = parseInt(s.justificado) || 0;
function pct(v){ return t > 0 ? Math.round(v/t*100)/10 : 0; }
function bar(v,c){ return '<div class="progress-bar"><div class="progress-fill"
style="width: '+pct(v)+'%;background: '+c+';"></div></div>'; }
sec.innerHTML =
'<div class="stat-card total"><div class="number">'+t+'</div><div class="label">Total
registros</div></div>' +
'<div class="stat-card presente"><div class="number">'+pre+'</div><div class="label">Presentes
(' +pct(pre)+'%)</div>'+bar(pre,'#28a745')+'</div>' +
'<div class="stat-card ausente"><div class="number">'+aus+'</div><div class="label">Ausentes
(' +pct(aus)+'%)</div>'+bar(aus,'#dc3545')+'</div>' +
'<div class="stat-card tardanza"><div class="number">'+tar+'</div><div class="label">Tardanzas
(' +pct(tar)+'%)</div>'+bar(tar,'#ffc107')+'</div>' +
'<div class="stat-card justif"><div class="number">'+jus+'</div><div class="label">Justificados
(' +pct(jus)+'%)</div>'+bar(jus,'#17a2b8')+'</div>';
}

function renderTable(rows) {
document.getElementById('result-count').textContent = rows.length;
if (!rows.length) {
document.getElementById('table-body').innerHTML =
'<div class="empty-state"><div class="icon">■</div><p>No se encontraron registros con los filtros
aplicados.</p></div>';
return;
}
var html = '<table><thead><tr><th>Fecha</th><th>Hora</th><th>Estudiante</th><th>Asignatura</th><th>Docente</
th><th>Estado</th><th>Observación</th></tr></thead><tbody>';
rows.forEach(function(a){
var sl = SL[a.status] || { label:a.status, color:'#666', bg:'#eee' };
var d = a.date ? a.date.substr(8,2)+'/'+a.date.substr(5,2)+'/'+a.date.substr(0,4) : '';
html += '<tr>' +
'<td>'+d+'</td>' +
'<td style="white-space:nowrap;">'+esc(a.hour_period)+'</td>' +
'<td>'+esc(a.student_name)+'</td>' +
'<td>'+esc(a.subject_name)+'</td>' +
'<td style="color:#666;">'+esc(a.teacher_name)+'</td>' +
'<td><span class="status-badge" style="background: '+sl.bg+';color: '+sl.color+';">'+sl.label+'</span></td>' +
'<td style="color:#888;font-size:0.82rem;">'+(a.observation ? esc(a.observation) : '-')+'</td>' +
'</tr>';
});
html += '</tbody></table>';
document.getElementById('table-body').innerHTML = html;
}

function esc(s){
if (!s) return '';
return String(s).replace(/&/g, '&amp;').replace(/</g, '&lt;').replace(/>/g, '&gt;').replace(/"/g, '&quot;');
}

```

```
function clearFilters(){
  ['f_subject','f_student','f_status'].forEach(function(id){ document.getElementById(id).value = ''; });
  document.getElementById('f_start').value = '';
  document.getElementById('f_end').value = '';
  runFilter();
}
</script>

</body>
</html>
```

views/tutor/dashboard.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 Tutor - EcuAsist</title>
</head>
<body>

<?php include BASE_PATH . '/views/partials/navbar.php'; ?>

<div class="breadcrumb">
  <a href="?action=dashboard">■ Inicio</a> &rsquo; Dashboard Tutor
</div>

<div class="container">

  <?php
  $pct = ($stats['total'] > 0)
  ? round(($stats['presente'] + $stats['justificado']) / $stats['total'] * 100)
  : 0;
  $todayPct = ($todayStats['total'] > 0)
  ? round(($todayStats['presente'] + $todayStats['justificado']) / $todayStats['total'] * 100)
  : 0;
  ?>

  <!-- Header -->
  <div class="page-header" style="background:linear-gradient(135deg,#1565c0,#1976d2);">
  <div class="ph-icon">■</div>
  <div>
    <h1>Dashboard – Tutor</h1>
    <p><?= htmlspecialchars($course['name']) ?> &nbsp;&nbsp;&nbsp;<?= count($students) ?> estudiantes</p>
  </div>
  <div class="ph-actions">
    <a href="?action=tutor_search_students" class="btn" style="background:rgba(255,255,255,0.2);color:#fff;border:1px solid rgba(255,255,255,0.5);">■ Buscar Estudiantes</a>
    <a href="?action=tutor_course_attendance" class="btn" style="background:rgba(255,255,255,0.2);color:#fff;border:1px solid rgba(255,255,255,0.5);margin-left:8px;">■ Ver Asistencias</a>
  </div>
</div>

<!-- Métricas globales del curso -->
<div style="display:grid;grid-template-columns:repeat(5,1fr);gap:16px;margin-bottom:20px;">
  <div class="stat-card">
    <div class="stat-icon" style="background:#e3f2fd;">■</div>
    <div class="stat-value"><?= $stats['total'] ?? 0 ?></div>
    <div class="stat-label">Total registros</div>
  </div>
  <div class="stat-card">
    <div class="stat-icon" style="background:#e8f5e9;">■</div>
    <div class="stat-value" style="color:#2e7d32;"><?= $stats['presente'] ?? 0 ?></div>
    <div class="stat-label">Presentes</div>
  </div>
</div>
```

```

<div class="stat-icon" style="background:#ffebee;">■</div>
<div class="stat-value" style="color:#c62828;"><?= $stats['ausente'] ?? 0 ?></div>
<div class="stat-label">Ausentes</div>
</div>
<div class="stat-card">
<div class="stat-icon" style="background:#fffde7;">■</div>
<div class="stat-value" style="color:#f57f17;"><?= $stats['tardanza'] ?? 0 ?></div>
<div class="stat-label">Tardanzas</div>
</div>
<div class="stat-card">
<div class="stat-icon" style="background:#e0f7fa;">■</div>
<div class="stat-value" style="color:#00838f;"><?= $stats['justificado'] ?? 0 ?></div>
<div class="stat-label">Justificados</div>
</div>
</div>

<!-- Grid: asistencia general + hoy -->
<div style="display:grid;grid-template-columns:1fr 1fr;gap:20px;margin-bottom:20px;">

<!-- Asistencia general -->
<div class="panel">
<h3 style="font-size:.95rem;margin-bottom:12px;color:#555;">■ Asistencia General del Curso</h3>
<div style="display:flex;justify-content:space-between;margin-bottom:6px;">
<span style="font-size:.85rem;color:#666;">Efectividad (presentes + justificados)</span>
<strong style="color:<?= $pct>=90?'#2e7d32':($pct>=75?'#f57f17':'#c62828') ?>"><?= $pct ?>%</strong>
</div>
<div style="background:#f0f0f0;border-radius:6px;height:12px;overflow:hidden;margin-bottom:16px;">

<div style="height:100%;border-radius:6px;width:<?= $pct ?>%;background:<?= $pct>=90?'#4caf50':($pct>=75?'#ff9800':'#f44336') ?>;"></div>
</div>
<?php if($stats['total'] > 0): ?>
<div style="display:grid;grid-template-columns:1fr 1fr;gap:8px;font-size:13px;">
<?php
$bars = [
['presente', '#4caf50', '✓ Presente'],
['ausente', '#f44336', '✗ Ausente'],
['tardanza', '#ff9800', '■ Tardanza'],
['justificado', '#00acc1', '■ Justificado'],
];
foreach($bars as [$key, $color, $label]):
$n = $stats[$key] ?? 0;
$pct_bar = round($n / $stats['total'] * 100);
?>
<div>
<div style="display:flex;justify-content:space-between;margin-bottom:3px;">
<span><?= $label ?></span>
<span style="font-weight:700;color:<?= $color ?>"><?= $n ?> (<?= $pct_bar ?>%)</span>
</div>
<div style="background:#f0f0f0;border-radius:4px;height:6px;">
<div style="height:100%;border-radius:4px;width:<?= $pct_bar ?>%;background:<?= $color ?>;"></div>
</div>
</div>
<?php endforeach; ?>
</div>
<?php else: ?>
<p style="color:#aaa;font-size:13px;text-align:center;margin-top:8px;">Sin registros de asistencia aún</p>
<?php endif; ?>
</div>

<!-- Hoy -->
<div class="panel">
<h3 style="font-size:.95rem;margin-bottom:12px;color:#555;">■ Hoy - <?= date('d/m/Y') ?></h3>
<?php if($todayStats['total'] > 0): ?>
<div style="display:flex;justify-content:space-between;margin-bottom:6px;">
<span style="font-size:.85rem;color:#666;">Asistencia del día</span>
<strong style="color:<?= $todayPct>=90?'#2e7d32':($todayPct>=75?'#f57f17':'#c62828') ?>"><?= $todayPct ?>%</strong>
</div>
<div style="background:#f0f0f0;border-radius:6px;height:12px;overflow:hidden;margin-bottom:16px;">

```

```

<div style="height:100%;border-radius:6px;width:<?= $todayPct ?>%;background:<?=
$todayPct>=90?'#4caf50':($todayPct>=75?'#ff9800':'#f44336') ?>;"></div>
</div>
<div style="display:flex;gap:8px;justify-content:center;flex-wrap:wrap;">
<div style="text-align:center;padding:10px 14px;background:#e8f5e9;border-radius:8px;">
<div style="font-size:1.4rem;font-weight:700;color:#2e7d32;"><?= $todayStats['presente'] ?></div>
<div style="font-size:11px;color:#666;">Presentes</div>
</div>
<div style="text-align:center;padding:10px 14px;background:#ffebee;border-radius:8px;">
<div style="font-size:1.4rem;font-weight:700;color:#c62828;"><?= $todayStats['ausente'] ?></div>
<div style="font-size:11px;color:#666;">Ausentes</div>
</div>
<div style="text-align:center;padding:10px 14px;background:#fffde7;border-radius:8px;">
<div style="font-size:1.4rem;font-weight:700;color:#f57f17;"><?= $todayStats['tardanza'] ?></div>
<div style="font-size:11px;color:#666;">Tardanzas</div>
</div>
<div style="text-align:center;padding:10px 14px;background:#e0f7fa;border-radius:8px;">
<div style="font-size:1.4rem;font-weight:700;color:#00838f;"><?= $todayStats['justificado'] ?></div>
<div style="font-size:11px;color:#666;">Justificados</div>
</div>
</div>
<?php else: ?>
<div style="text-align:center;padding:30px 0;color:#aaa;">
<div style="font-size:2rem;margin-bottom:8px;">■</div>
<p style="font-size:13px;">No hay registros de asistencia hoy</p>
</div>
<?php endif; ?>
</div>
</div>

<!-- Grid: tendencia + top ausencias -->
<div style="display:grid;grid-template-columns:1fr 1fr;gap:20px;margin-bottom:20px;">

<!-- Tendencia 7 días -->
<div class="panel">
<h3 style="font-size:.95rem;margin-bottom:16px;color:#555;">■ Últimos 7 días</h3>
<?php if(!empty($trend)): ?>
<div style="display:flex;align-items:flex-end;gap:8px;height:80px;">
<?php foreach($trend as $day):
$dp = ($day['total'] > 0) ? round($day['presente'] / $day['total'] * 100) : 0;
$h = max(4, round($dp * 0.8));
$col = $dp>=90?'#4caf50':($dp>=75?'#ff9800':'#f44336');
?>
<div style="flex:1;display:flex;flex-direction:column;align-items:center;gap:4px;">
<span style="font-size:10px;color:#999;"><?= $dp ?>%</span>
<div style="width:100%;height:<?= $h ?>px;background:<?= $col ?>;border-radius:3px 3px 0 0;" title="<?=
date('d/m', strtotime($day['date'])) ?>: <?= $dp ?>%"></div>
<span style="font-size:10px;color:#999;"><?= date('d/m', strtotime($day['date'])) ?></span>
</div>
<?php endforeach; ?>
</div>
<?php else: ?>
<p style="color:#aaa;font-size:13px;text-align:center;padding:20px 0;">Sin datos de los últimos 7 días</p>
<?php endif; ?>
</div>

<!-- Top ausencias -->
<div class="panel">
<div style="display:flex;justify-content:space-between;align-items:center;margin-bottom:12px;">
<h3 style="font-size:.95rem;color:#555;">■■■ Estudiantes con más ausencias</h3>
<a href="?action=tutor_search_students" style="font-size:12px;color:#007bff;text-decoration:none;">Ver
todos →</a>
</div>
<?php if(!empty($topAbsences)): ?>
<?php foreach($topAbsences as $i => $row): ?>
<div style="display:flex;align-items:center;gap:10px;padding:8px 0;border-bottom:1px solid #f5f5f5;">
<span style="font-size:11px;color:#999;width:16px;text-align:right;"><?= $i+1 ?>.</span>
<div style="flex:1;font-size:13px;color:#333;"><?= htmlspecialchars($row['student_name']) ?></div>
<span class="badge <?= $row['total_ausencias']>=10?'badge-red':($row['total_ausencias']>=5?'badge-yellow':'b
adge-gray') ?>">
<?= $row['total_ausencias'] ?> faltas

```

```

</span>
</div>
<?php endforeach; ?>
<?php else: ?>
<p style="color:#aaa;font-size:13px;text-align:center;padding:20px 0;">■ Sin ausencias registradas</p>
<?php endif; ?>
</div>
</div>

<!-- Lista de estudiantes del curso -->
<div class="table-wrap">
<div class="table-info">
<span>■ <strong>Estudiantes del Curso</strong> - <?= count($students) ?></span>

<a href="?action=tutor_search_students" class="btn btn-primary btn-sm">■ Buscar estudiante</a>
</div>
<?php if(empty($students)): ?>
<div class="empty-state" style="padding:30px;">
<div class="icon">■</div>
<p>No hay estudiantes matriculados en este curso.</p>
</div>
<?php else: ?>
<table>
<thead>
<tr><th>#</th><th>Estudiante</th></tr>
</thead>
<tbody>
<?php $i = 1; foreach($students as $s): ?>
<tr>
<td style="color:#999;font-size:12px;"><?= $i++ ?></td>
<td><?= htmlspecialchars($s['full_name']) ?></td>
</tr>
<?php endforeach; ?>
</tbody>
</table>
<?php endif; ?>
</div>
</div>
</body>
</html>

```

views/tutor/no_tutor.php

```

<?php // views/tutor/no_tutor.php ?>
<!DOCTYPE html>
<html lang="es">
<head>
<meta charset="UTF-8">
<title>Asistencia de Mi Curso - EcuAsist</title>
<style>
* { margin:0; padding:0; box-sizing:border-box; }
body { font-family: Arial, sans-serif; background:#f4f4f4; }
.container { max-width:1200px; margin:24px auto; padding:0 16px; }
.page-header { background:linear-gradient(135deg,#1565c0,#1976d2); color:#fff; border-radius:10px;
padding:20px 24px; margin-bottom:20px; display:flex; align-items:center; gap:16px; }
.page-header .ph-icon { font-size:2.2rem; }
.page-header h1 { font-size:1.3rem; font-weight:700; }
.page-header p { font-size:0.85rem; opacity:0.85; margin-top:4px; }
.empty-state { background:white; border-radius:10px; padding:60px 20px; text-align:center; box-shadow:0 2px
10px rgba(0,0,0,0.08); }
.empty-state .icon { font-size:3.5rem; margin-bottom:16px; }
.empty-state h2 { font-size:1.3rem; color:#333; margin-bottom:8px; }
.empty-state p { color:#888; font-size:14px; margin-bottom:24px; }
.btn-back { display:inline-block; padding:10px 22px; background:#007bff; color:#fff; border-radius:6px;
text-decoration:none; font-size:14px; }
.btn-back:hover { background:#0056b3; }
</style>
</head>

```

```

<body>

<?php include BASE_PATH . '/views/partials/navbar.php'; ?>

<div class="breadcrumb">
  <a href="?action=dashboard">■ Inicio</a> &rsquo; Asistencia de Mi Curso
</div>

<div class="container">

  <div class="page-header">
    <div class="ph-icon">■</div>
    <div>
      <h1>Asistencia de Mi Curso</h1>
      <p>Vista del docente tutor sobre la asistencia de su curso</p>
    </div>
  </div>

  <div class="empty-state">
    <div class="icon">■■■</div>
    <h2>No eres tutor de ningún curso</h2>
    <p>Actualmente no tienes asignado un curso como tutor en el año lectivo activo.</p>
    <a href="?action=dashboard" class="btn-back">← Volver al inicio</a>
  </div>

</div>
</body>
</html>

```

views/tutor/search_students.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>Buscar Estudiantes - EcuAsist</title>
</head>
<body>

<?php include BASE_PATH . '/views/partials/navbar.php'; ?>

<div class="breadcrumb">
  <a href="?action=dashboard">■ Inicio</a> &rsquo;
  <a href="?action=tutor_dashboard">Dashboard Tutor</a> &rsquo;
  Buscar Estudiantes
</div>

<div class="container">

  <!-- Header -->
  <div class="page-header" style="background:linear-gradient(135deg,#1565c0,#1976d2);">
    <div class="ph-icon">■</div>
    <div>
      <h1>Buscar Estudiantes</h1>
      <p><?= htmlspecialchars($course['name']) ?></p>
    </div>
    <div class="ph-actions">
      <a href="?action=tutor_dashboard" class="btn" style="background:rgba(255,255,255,0.2);color:#fff;border:1px solid rgba(255,255,255,0.5);">← Dashboard</a>
    </div>
  </div>

  <!-- Buscador -->
  <div class="panel" style="margin-bottom:20px;">
    <form method="GET" action="">
      <input type="hidden" name="action" value="tutor_search_students">
      <div class="form-row" style="align-items:flex-end;">

```

```

<div class="form-group" style="flex:1;">
<label>Nombre, apellido o cédula</label>
<input type="text" name="q" class="form-control"

```

```

placeholder="Ej: García, 1750..."
value="<?= htmlspecialchars($_GET['q']) ?? '' ?>"
autofocus>
</div>
<div class="form-group" style="flex:0 0 auto; display:flex; gap:8px;">
<button type="submit" class="btn btn-primary">■ Buscar</button>
<?php if(!empty($_GET['q'])): ?>
<a href="?action=tutor_search_students" class="btn btn-outline">X Limpiar</a>
<?php endif; ?>
</div>
</div>
</form>
</div>

```

```

<!-- Resultados -->
<?php if(isset($_GET['q']) && empty($students)): ?>
<div class="empty-state">
<div class="icon">■</div>
<p>No se encontraron estudiantes con "<?= htmlspecialchars($_GET['q']) ?>"</p>
</div>

<?php else: ?>
<div class="table-wrap">
<div class="table-info">
<span>
■ <strong><?= isset($_GET['q']) && $_GET['q'] !== '' ? 'Resultados' : 'Todos los estudiantes' ?></strong>
- <?= count($students) ?> encontrado<?= count($students)!=1?'s':'' ?>
<?php if(!empty($_GET['q'])): ?>
para "<strong><?= htmlspecialchars($_GET['q']) ?></strong>"
<?php endif; ?>
</span>
</div>
<?php if(empty($students)): ?>
<div class="empty-state" style="padding:30px;">
<div class="icon">■</div>
<p>No hay estudiantes matriculados en este curso.</p>
</div>
<?php else: ?>
<table>
<thead>

```

```

<tr>
<th>#</th>
<th>Estudiante</th>
<th>Cédula</th>
<th>Email</th>
<th>Teléfono</th>
<th style="text-align:center;">Clases</th>
<th style="text-align:center;">■</th>
<th style="text-align:center;">■</th>
<th style="text-align:center;">■</th>
<th style="text-align:center;">% Asistencia</th>
</tr>
</thead>
<tbody>
<?php $i = 1; foreach($students as $s):
$total = (int)($s['total_clases'] ?? 0);
$pres = (int)($s['presentes'] ?? 0);
$ause = (int)($s['ausentes'] ?? 0);
$tard = (int)($s['tardanzas'] ?? 0);
$just = (int)($s['justificados'] ?? 0);
$pct = $total > 0 ? round(($pres + $just) / $total * 100) : 0;
$pctColor = $pct >= 90 ? '#2e7d32' : ($pct >= 75 ? '#f57f17' : '#c62828');
$pctBg = $pct >= 90 ? '#e8f5e9' : ($pct >= 75 ? '#fff8e1' : '#ffebee');
?>
<tr>
<td style="color:#999;font-size:12px;"><?= $i++ ?></td>
<td><strong><?= htmlspecialchars($s['last_name'] . ' ' . $s['first_name']) ?></strong></td>

```

```

<td style="font-size:13px;color:#666;"><?= htmlspecialchars($s['dni']) ?? '-' ?></td>
<td style="font-size:13px;color:#666;"><?= htmlspecialchars($s['email']) ?? '-' ?></td>
<td style="font-size:13px;color:#666;"><?= htmlspecialchars($s['phone']) ?? '-' ?></td>
<td style="text-align:center;font-size:13px;"><?= $total ?></td>
<td style="text-align:center;">
<span style="font-weight:700;color:#2e7d32;"><?= $pres ?></span>
</td>
<td style="text-align:center;">
<span style="font-weight:700;color:#c62828;"><?= $ause ?></span>
</td>
<td style="text-align:center;">
<span style="font-weight:700;color:#f57f17;"><?= $tard ?></span>
</td>

<td style="text-align:center;">
<?php if($total > 0): ?>
<span style="background:<?= $pctBg ?>;color:<?= $pctColor ?>;font-weight:700;padding:3px
10px;border-radius:12px;font-size:13px;">
<?= $pct ?>%
</span>
<?php else: ?>
<span style="color:#ccc;font-size:12px;">Sin datos</span>
<?php endif; ?>
</td>
</tr>
<?php endforeach; ?>
</tbody>
</table>
<?php endif; ?>
</div>
<?php endif; ?>
</div>
</body>
</html>

```

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>
<style>
.badge-pending { background:#ffc107; color:#000; padding:5px 12px; border-radius:12px; font-size:12px;
font-weight:bold; }
.badge-approved { background:#28a745; color:white; padding:5px 12px; border-radius:12px; font-size:12px;
font-weight:bold; }
.badge-rejected { background:#dc3545; color:white; padding:5px 12px; border-radius:12px; font-size:12px;
font-weight:bold; }
.stat-box { background:white; border-radius:8px; padding:20px; text-align:center;
box-shadow:0 2px 8px rgba(0,0,0,.08); flex:1; }
.stat-box .num { font-size:28px; font-weight:700; }
.stat-box .lbl { font-size:12px; color:#888; margin-top:4px; }
</style>
</head>
<body>

<?php include BASE_PATH . '/views/partials/navbar.php'; ?>

<div class="breadcrumb">
<a href="?action=dashboard">■ Inicio</a> &rsquo; Mis Justificaciones
</div>

<div class="container">

<?php if(isset($_GET['success'])): ?>

```



```

<div class="alert alert-success">✓ Justificación enviada correctamente. Está pendiente de revisión.</div>
<?php endif; ?>

<div class="page-header" style="background:linear-gradient(135deg,#e65100,#ef6c00);display:flex;justify-content:space-between;align-items:center;">
<div style="display:flex;align-items:center;gap:12px;">
<div class="ph-icon">■</div>
<div>
<h1>Mis Justificaciones</h1>
<p>Historial de justificaciones enviadas</p>
</div>
</div>
<a href="?action=submit_justification"

```

```

style="padding:9px 18px;background:white;color:#e65100;text-decoration:none;
border-radius:6px;font-weight:600;font-size:13px;white-space:nowrap;">
■ Nueva Justificación
</a>
</div>

<?php if(empty($justifications)): ?>
<div class="empty-state">
<div class="icon">■</div>
<p>No tienes justificaciones enviadas aún.</p>
<p style="font-size:13px;color:#aaa;margin-top:6px;">Puedes justificar ausencias desde "Mi Asistencia".</p>
<a href="?action=my_attendance" class="btn btn-primary" style="margin-top:16px;">← Ir a Mi Asistencia</a>
</div>

<?php else: ?>

<!-- Resumen -->
<div style="display:flex;gap:16px;margin-bottom:20px;flex-wrap:wrap;">
<div class="stat-box">
<div class="num" style="color:#f57c00;">
<?= count(array_filter($justifications, fn($j) => $j['status'] === 'pendiente')) ?>
</div>
<div class="lbl">■ Pendientes</div>
</div>
<div class="stat-box">
<div class="num" style="color:#2e7d32;">
<?= count(array_filter($justifications, fn($j) => $j['status'] === 'aprobado')) ?>
</div>
<div class="lbl">✓ Aprobadas</div>
</div>
<div class="stat-box">
<div class="num" style="color:#c62828;">
<?= count(array_filter($justifications, fn($j) => $j['status'] === 'rechazado')) ?>
</div>
<div class="lbl">✗ Rechazadas</div>
</div>
</div>

<!-- Tabla -->
<div class="panel">

```

```

<table>
<thead>
<tr>
<th>#</th>
<th>Fecha</th>
<th>Motivo</th>
<th>Documento</th>
<th>Estado</th>
<th>Envío</th>
<th>Revisado por</th>
<th>Observaciones</th>
</tr>
</thead>
<tbody>
<?php $i = 1; foreach($justifications as $just): ?>
<tr>
<td><?= $i++ ?></td>

```

```

<td><?= date('d/m/Y', strtotime($just['attendance_date'])) ?></td>
<td><?= htmlspecialchars($just['reason']) ?></td>
<td>
<?php if($just['document_path']): ?>
<a href="<?= BASE_URL ?>/<?= ltrim($just['document_path'],'/') ?>"
target="_blank"
style="padding:5px 10px;background:#007bff;color:white;border-radius:4px;
text-decoration:none;font-size:12px;display:inline-block;">
■ Ver
</a>
<?php else: ?>
<span style="color:#aaa;font-size:12px;">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 style="font-size:12px;"><?= date('d/m/Y H:i', strtotime($just['submitted_at'])) ?></td>
<td style="font-size:12px;">
<?php if($just['reviewed_by']): ?>
<?= htmlspecialchars($just['reviewer_name']) ?><br>
<span style="color:#888;"><?= date('d/m/Y H:i', strtotime($just['reviewed_at'])) ?></span>
<?php else: ?>
<span style="color:#aaa;">—</span>
<?php endif; ?>
</td>
<td style="font-size:12px;">
<?= $just['review_notes'] ?> htmlspecialchars($just['review_notes']) : '<span style="color:#aaa;">—</span>' ?
>
</td>
</tr>
<?php endforeach; ?>
</tbody>
</table>
</div>

<div style="margin-top:16px;">
<a href="?action=my_attendance" class="btn btn-outline">← Volver a Mi Asistencia</a>
</div>

<?php endif; ?>

</div>
</body>
</html>

```

views/justifications/pending.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>Justificaciones Pendientes - EcuAsist</title>
</head>
<body>

<?php include BASE_PATH . '/views/partials/navbar.php'; ?>

<div class="breadcrumb">
<a href="?action=dashboard">■ Inicio</a> &rsaquo; Justificaciones Pendientes

```

```

</div>

<div class="container">

    <?php if(isset($_GET['success'])): ?>
    <div class="alert alert-success">✓ Justificación procesada correctamente</div>
    <?php endif; ?>

    <div class="page-header" style="background:linear-gradient(135deg,#1b5e20,#2e7d32);">
    <div class="ph-icon">■</div>
    <div>
    <h1>Justificaciones Pendientes</h1>
    <p><?= count($justifications) ?> solicitud<?= count($justifications)!=1?'es':'' ?> por revisar</p>
    </div>
    </div>

    <?php if(empty($justifications)): ?>
    <div class="empty-state">
    <div class="icon">■</div>
    <p>No hay justificaciones pendientes de revisión.</p>
    </div>

    <?php else: ?>
    <div class="table-wrap">
    <table>
    <thead>

    <tr>
    <th>#</th>
    <th>Estudiante</th>
    <th>Curso</th>
    <th>Período de ausencia</th>
    <th style="text-align:center;">Días Lab.</th>
    <th style="text-align:center;">Revisión</th>
    <th>Enviado por</th>
    <th>Fecha solicitud</th>
    <th>Acciones</th>
    </tr>
    </thead>
    <tbody>
    <?php $i = 1; foreach($justifications as $just):
    $days = (int)($just['working_days'] ?? 1);
    $canApprove = $just['can_approve'] ?? 'inspector';

    if ($canApprove === 'tutor') {
    $approverLabel = '■ Tutor';
    $approverColor = 'badge-green';
    } elseif ($canApprove === 'inspector') {
    $approverLabel = '■ Inspector';
    $approverColor = 'badge-yellow';
    } else {
    $approverLabel = '■■ Autoridad';
    $approverColor = 'badge-red';
    }

    // Período de ausencia
    if (!empty($just['date_from'])) {
    $dateFrom = date('d/m/Y', strtotime($just['date_from']));
    $dateTo = !empty($just['date_to']) && $just['date_to'] !== $just['date_from']
    ? date('d/m/Y', strtotime($just['date_to']))
    : null;
    $period = $dateTo ? "$dateFrom al $dateTo" : $dateFrom;
    } elseif (!empty($just['attendance_date'])) {
    $period = date('d/m/Y', strtotime($just['attendance_date']));
    } else {
    $period = '-';
    }

    ?>
    <tr>
    <td style="color:#999;font-size:12px;"><?= $i++ ?></td>
    <td><strong><?= htmlspecialchars($just['student_name']) ?></strong></td>

```

```

<td style="font-size:13px;"><?= htmlspecialchars($just['course_name'] ?? '-') ?></td>
<td style="font-size:13px;"><?= $period ?></td>
<td style="text-align:center;">
<span style="font-weight:700;color:<?= $days<=3?'#2e7d32':'#f57f17' ?>">
<?= $days ?>
</span>
</td>
<td style="text-align:center;">
<span class="badge <?= $approverColor ?>"><?= $approverLabel ?></span>
</td>
<td style="font-size:13px;color:#666;"><?= htmlspecialchars($just['submitted_by_name']) ?></td>
<td style="font-size:12px;color:#999;"><?= date('d/m/Y H:i', strtotime($just['created_at'])) ?></td>
<td>
<button class="btn btn-primary btn-sm"
onclick="openReview(
<?= $just['id'] ?>,
<?= json_encode($just['reason'] ?? '') ?>,
<?= json_encode($just['document_path'] ?? '') ?>,
<?= json_encode($period) ?>,
<?= $days ?>
)">■ Revisar</button>
</td>
</tr>
<?php endforeach; ?>
</tbody>
</table>
</div>
<?php endif; ?>
</div>

<!-- Modal revisar -->
<div class="modal-overlay" id="modalReview">
<div class="modal-box" style="max-width:520px;">
<h3 style="margin-bottom:16px;">■ Revisar Justificación</h3>

<div style="background:#f8f9fa;border-radius:6px;padding:12px;margin-bottom:14px;font-size:13px;">
<div><strong>Período:</strong> <span id="rv-period"></span></div>
<div style="margin-top:4px;"><strong>Días laborables:</strong> <span id="rv-days"></span></div>
</div>

<div style="margin-bottom:12px;">
<strong style="font-size:13px;">Motivo:</strong>
<p id="rv-reason" style="margin-top:6px;font-size:14px;color:#444;line-height:1.5;"></p>
</div>

<div id="rv-doc" style="margin-bottom:14px;display:none;">
<strong style="font-size:13px;">Documento adjunto:</strong>
<div style="margin-top:6px;">
<a id="rv-doc-link" href="#" target="_blank" class="btn btn-info btn-sm">■ Ver documento</a>
</div>
</div>

<form method="POST" action="?action=review_justification">
<input type="hidden" name="justification_id" id="rv-id">
<div class="form-group">
<label style="font-size:13px;">Observaciones <span style="color:#999;font-weight:400;">(opcional)</span></label>
<textarea name="notes" class="form-control" rows="3" placeholder="Notas para el estudiante..."></textarea>
</div>
<div class="modal-actions">
<button type="button" class="btn btn-outline" onclick="closeModal('modalReview')">Cancelar</button>
<button type="submit" name="review_action" value="reject" class="btn btn-danger">✗ Rechazar</button>
<button type="submit" name="review_action" value="approve" class="btn btn-success">✓ Aprobar</button>
</div>
</form>
</div>
</div>

<script>
function openReview(id, reason, docPath, period, days) {
document.getElementById('rv-id').value = id;

```

```

document.getElementById('rv-period').textContent = period;
document.getElementById('rv-days').textContent = days + ' día' + (days !== 1 ? 's' : '');
document.getElementById('rv-reason').textContent = reason || '(sin descripción)';

var docBlock = document.getElementById('rv-doc');

if (docPath) {
  document.getElementById('rv-doc-link').href = '<?= BASE_URL ?>/' + docPath;
  docBlock.style.display = 'block';
} else {
  docBlock.style.display = 'none';
}
document.getElementById('modalReview').classList.add('on');
}
function openModal(id) { document.getElementById(id).classList.add('on'); }
function closeModal(id) { document.getElementById(id).classList.remove('on'); }
document.querySelectorAll('.modal-overlay').forEach(function(m) {
  m.addEventListener('click', function(e){ if(e.target === m) closeModal(m.id); });
});
</script>

</body>
</html>

```

views/justifications/reviewed.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>Justificaciones Revisadas - EcuAsist</title>
  <style>
    .badge-aprobado { background:#d4edda; color:#155724; padding:4px 10px; border-radius:12px; font-size:12px;
font-weight:bold; }
    .badge-rechazado{ background:#f8d7da; color:#721c24; padding:4px 10px; border-radius:12px; font-size:12px;
font-weight:bold; }
    .filter-bar { display:flex; gap:8px; flex-wrap:wrap; margin-bottom:20px; }
    .filter-bar a { padding:6px 16px; border-radius:20px; text-decoration:none; font-size:13px;
font-weight:600; border:2px solid; transition:all .15s; }
    .modal { display:none; position:fixed; top:0; left:0; width:100%; height:100%;
background:rgba(0,0,0,.5); z-index:1000; align-items:center; justify-content:center; }
    .modal.open { display:flex; }
    .modal-content { background:white; padding:30px; border-radius:8px; max-width:560px;
width:90%; box-shadow:0 4px 20px rgba(0,0,0,.3); position:relative; }
    .modal-close { position:absolute; top:12px; right:16px; font-size:24px;
cursor:pointer; color:#888; line-height:1; }
    .modal-close:hover { color:#333; }
  </style>
</head>
<body>

<?php include BASE_PATH . '/views/partials/navbar.php'; ?>

<div class="breadcrumb">
  <a href="?action=dashboard">■ Inicio</a> &rsquo; Justificaciones Revisadas
</div>

<div class="container">

  <?php if(isset($_GET['success'])): ?>
  <div class="alert alert-success">✓ Acción realizada correctamente</div>
  <?php endif; ?>

  <div class="page-header" style="background:linear-gradient(135deg,#1b5e20,#388e3c);display:flex;justify-cont
ent:space-between;align-items:center;">
    <div style="display:flex;align-items:center;gap:12px;">
      <div class="ph-icon">■</div>

```

```

<div>
<h1>Justificaciones Revisadas</h1>
<p>Historial de justificaciones aprobadas y rechazadas</p>
</div>
</div>
<a href="?action=pending_justifications"
style="padding:9px 18px;background:white;color:#1b5e20;text-decoration:none;
border-radius:6px;font-weight:600;font-size:13px;white-space:nowrap;">
■ Ver Pendientes
</a>
</div>

<!-- Filtros -->
<div class="filter-bar">
<a href="?action=reviewed_justifications"
style="<?=( $filter==='all') ? 'background:#343a40;color:white;border-color:#343a40;' : '
background:white;color:#343a40;border-color:#343a40;' ?>">
■ Todas (<?=( count($justifications) ?> )
</a>
<a href="?action=reviewed_justifications&filter=aprobado"
style="<?=( $filter==='aprobado') ? 'background:#28a745;color:white;border-color:#28a745;' : '
background:white;color:#28a745;border-color:#28a745;' ?>">
■ Aprobadas
</a>
<a href="?action=reviewed_justifications&filter=rechazado"
style="<?=( $filter==='rechazado') ? 'background:#dc3545;color:white;border-color:#dc3545;' : '
background:white;color:#dc3545;border-color:#dc3545;' ?>">
■ Rechazadas
</a>
</div>

<?php if(empty($justifications)): ?>
<div class="empty-state">
<div class="icon">■</div>
<p>No hay justificaciones revisadas con este filtro.</p>
</div>
<?php else: ?>
<div class="panel">
<table>
<thead>
<tr>
<th>#</th>
<th>Estudiante</th>

<th>Curso</th>
<th>Asignatura</th>
<th>Fecha Falta</th>
<th>Enviado por</th>
<th>Estado</th>
<th>Revisado por</th>
<th>Acciones</th>
</tr>
</thead>
<tbody>
<?php $i = 1; foreach($justifications as $j):
$jData = htmlspecialchars(json_encode([
'student' => $j['student_name'],
'reason' => $j['reason'],
'notes' => $j['review_notes'] ?? '',
'doc' => $j['document_path'] ?? '',
'status' => $j['status'],
]), ENT_QUOTES);
?>
<tr>
<td><?=( $i++ ?></td>
<td><?=( htmlspecialchars($j['student_name']) ?></td>
<td><?=( htmlspecialchars($j['course_name'] ?? '-') ?></td>
<td><?=( htmlspecialchars($j['subject_name']) ?></td>
<td><?=( date('d/m/Y', strtotime($j['attendance_date'])) ?></td>
<td><?=( htmlspecialchars($j['submitted_by_name']) ?></td>
<td>
<?php if($j['status'] === 'aprobado'): ?>

```

```

<span class="badge-aprobado">■ Aprobada</span>
<?php else: ?>
<span class="badge-rechazado">■ Rechazada</span>
<?php endif; ?>
</td>
<td><?= htmlspecialchars($j['reviewer_name'] ?? '-') ?></td>
<td>
<button class="btn btn-primary"
style="padding:5px 12px;font-size:12px;"
onclick="verDetalle('<?= $jData ?>')">■ Ver</button>
</td>
</tr>

<?php endforeach; ?>
</tbody>
</table>
</div>
<?php endif; ?>

</div>

<!-- Modal detalle -->
<div id="modal" class="modal" onclick="if(event.target===this)cerrarModal()">
  <div class="modal-content">
    <span class="modal-close" onclick="cerrarModal()">&times;</span>
    <h3 style="margin-bottom:16px;">■ Detalle de Justificación</h3>
    <div id="modal-body"></div>
  </div>
</div>

<script>
function verDetalle(jsonStr) {
  const d = JSON.parse(jsonStr);
  const badge = d.status === 'aprobado'
    ? '<span style="background:#d4edda;color:#155724;padding:3px 10px;border-radius:10px;">■ Aprobada</span>'
    : '<span style="background:#f8d7da;color:#721c24;padding:3px 10px;border-radius:10px;">■ Rechazada</span>';

  let html = `<p><strong>Estudiante:</strong> ${d.student}</p>`;
  html += `<p style="margin-top:10px;"><strong>Estado:</strong> ${badge}</p>`;
  html += `<p style="margin-top:10px;"><strong>Motivo:</strong></p><p
style="margin-top:4px;color:#555;">${d.reason}</p>`;

  if (d.notes) {
    html += `<p style="margin-top:10px;"><strong>Observación del revisor:</strong></p><p
style="margin-top:4px;color:#555;">${d.notes}</p>`;
  }

  if (d.doc) {
    html += `<p style="margin-top:14px;"><strong>Documento:</strong>
<a href="<?= BASE_URL ?>/${d.doc}" target="_blank"
style="color:#007bff;">■ Ver archivo adjunto</a></p>`;
  } else {
    html += `<p style="margin-top:14px;color:#999;"><em>Sin documento adjunto</em></p>`;
  }
}

document.getElementById('modal-body').innerHTML = html;
document.getElementById('modal').classList.add('open');
}

function cerrarModal() {
  document.getElementById('modal').classList.remove('open');
}
</script>

</body>
</html>

```

```

<?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>Justificar Ausencia - EcuAsist</title>
  <style>
    .reason-grid { display: grid; grid-template-columns: 1fr 1fr; gap: 10px; margin-bottom: 12px; }
    .reason-option { display: flex; align-items: center; gap: 8px; padding: 10px 14px;
    border: 2px solid #e0e0e0; border-radius: 8px; cursor: pointer;
    transition: all .2s; font-size: 14px; background: #fff; }
    .reason-option:hover { border-color: #007bff; background: #f0f7ff; }
    .reason-option input[type=radio] { accent-color: #007bff; width: 16px; height: 16px; flex-shrink: 0; }
    .reason-option.selected { border-color: #007bff; background: #e8f2ff; font-weight: 600; }

    .absence-item { display: flex; align-items: center; gap: 12px; padding: 12px 16px;
    border: 2px solid #e0e0e0; border-radius: 8px; cursor: pointer;
    transition: all .2s; margin-bottom: 8px; background: #fff; }
    .absence-item:hover { border-color: #007bff; background: #f0f7ff; }
    .absence-item.checked { border-color: #007bff; background: #e8f2ff; }
    .absence-item input[type=checkbox] { accent-color: #007bff; width: 18px; height: 18px; flex-shrink: 0; }
    .absence-date { font-weight: 700; font-size: 15px; color: #333; min-width: 95px; }
    .absence-meta { font-size: 13px; color: #666; flex: 1; }
    .absence-hour { display: inline-block; font-size: 12px; background: #f0f0f0;
    padding: 2px 8px; border-radius: 4px; color: #555; margin: 2px 2px 0 0; }
    .select-all-bar { display: flex; align-items: center; gap: 10px;
    padding: 8px 0; margin-bottom: 10px; border-bottom: 1px solid #eee; }
    #otro-block, #detail-block { display: none; margin-top: 6px; }
    .counter-badge { display: inline-block; background: #007bff; color: #fff;
    border-radius: 12px; padding: 3px 12px; font-size: 13px; font-weight: 700; }
    .approver-note { padding: 10px 14px; border-radius: 6px; font-size: 13px;
    margin-top: 10px; display: none; }
    .step-title { font-size: .95rem; color: #555; margin-bottom: 14px; font-weight: 600; }
  </style>
</head>
<body>

<?php include BASE_PATH . '/views/partials/navbar.php'; ?>

<div class="breadcrumb">
  <a href="?action=dashboard">■ Inicio</a> &rsquo;
  <a href="?action=my_attendance">Mi Asistencia</a> &rsquo;
  Justificar Ausencia
</div>

<div class="container" style="max-width:720px;">

  <?php if(isset($_GET['error'])): ?>
  <div class="alert alert-danger">
    <?= ['no_absences' => 'Debes seleccionar al menos una ausencia.'][$_GET['error']] ?? 'Error al enviar.' ?>
  </div>
  <?php endif; ?>

  <div class="page-header" style="background:linear-gradient(135deg,#e65100,#ef6c00);">
  <div class="ph-icon">■</div>
  <div>
    <h1>Justificar Ausencia</h1>
    <p>Selecciona los días que deseas justificar</p>
  </div>
</div>

  <?php if(empty($absences)): ?>
  <div class="empty-state">
    <div class="icon">■</div>
    <p>No tienes ausencias pendientes de justificar.</p>
    <a href="?action=my_attendance" class="btn btn-primary" style="margin-top:12px;">← Volver</a>
  </div>

  <?php else: ?>

  <div class="alert alert-info">

```



```
■ Selecciona uno o varios días. <strong>Hasta 3 días</strong> → revisa el <strong>Docente Tutor</strong>.
<strong>Más de 3</strong> → revisa <strong>Inspector o Autoridad</strong>.
</div>
```

```
<form method="POST" enctype="multipart/form-data" id="justForm">
```

```
<?php if(Security::hasRole('representante')): ?>
```

```
<input type="hidden" name="student_id" value="<?= isset($_GET['student_id']) ? (int)$_GET['student_id'] : ' '
?>">
```

```
<?php endif; ?>
```

```
<!-- ■■ PASO 1: Seleccionar días ■■ -->
```

```
<div class="panel" style="margin-bottom:16px;">
```

```
<div style="display:flex;justify-content:space-between;align-items:center;margin-bottom:12px;">
```

```
<p class="step-title" style="margin:0;">■ Paso 1 – Días de ausencia a justificar</p>
```

```
<span id="selected-count" class="counter-badge">0 seleccionados</span>
```

```
</div>
```

```
<div class="select-all-bar">
```

```
<label style="display:flex;align-items:center;gap:8px;cursor:pointer;font-size:13px;color:#555;">
```

```
<input type="checkbox" id="select-all" onchange="toggleAll(this)"
```

```
style="width:16px;height:16px;accent-color:#007bff;">
```

```
Seleccionar todos (<?= count($absences) ?> días)
```

```
</label>
```

```
</div>
```

```
<?php
```

```
// Agrupar registros por fecha (un día puede tener varias horas)
```

```
$byDate = [];
```

```
foreach ($absences as $a) {
```

```
$byDate[$a['date']][] = $a;
```

```
}
```

```
$days = ['Domingo','Lunes','Martes','Miércoles','Jueves','Viernes','Sábado'];
```

```
foreach ($byDate as $date => $rows):
```

```
$ids = implode(',', array_column($rows, 'id'));
```

```
$dayLabel = $days[date('w', strtotime($date))];
```

```
$dateLabel= date('d/m/Y', strtotime($date));
```

```
?>
```

```
<div class="absence-item" onclick="toggleCheck(this)">
```

```
<input type="checkbox" name="attendance_ids[]" value="<?= $ids ?>"
```

```
onchange="updateCount(); event.stopPropagation();">
```

```
<div class="absence-date"><?= $dateLabel ?></div>
```

```
<div class="absence-meta">
```

```
<div style="font-weight:600;color:#444;margin-bottom:4px;"><?= $dayLabel ?></div>
```

```
<?php foreach($rows as $r): ?>
```

```
<span class="absence-hour">Hora <?= $r['hour_period'] ?> · <?= htmlspecialchars($r['subject_name']) ?
></span>
```

```
<?php endforeach; ?>
```

```
</div>
```

```
</div>
```

```
<?php endforeach; ?>
```

```
<!-- Aviso quién aprueba -->
```

```
<div class="approver-note" id="note-tutor" style="background:#e8f5e9;color:#2e7d32;">
```

```
■ Con <strong id="lbl-t">0</strong> día(s), revisará el <strong>Docente Tutor</strong>.
```

```
</div>
```

```
<div class="approver-note" id="note-insp" style="background:#fff8e1;color:#f57f17;">
```

```
■■ Con <strong id="lbl-i">0</strong> día(s), revisará el <strong>Inspector o Autoridad</strong>.
```

```
</div>
```

```
</div>
```

```
<!-- ■■ PASO 2: Motivo ■■ -->
```

```
<div class="panel" style="margin-bottom:16px;">
```

```
<p class="step-title">■ Paso 2 – Motivo de la ausencia</p>
```

```
<div class="reason-grid">
```

```
<?php foreach([
```

```
['■','Enfermedad o malestar'],
```

```
['■','Cita médica o examen'],
```

```
['■■■■■■','Fallecimiento familiar'],
```

```
['■■','Trámites legales / judiciales'],
```

```

[ '■', 'Accidente o emergencia'],
[ '■■', 'Causas climáticas'],
[ '■', 'Evento deportivo o académico oficial'],
[ '☞■', 'Viaje o traslado familiar'],
[ '■', 'Otro motivo'],
] as [$icon, $label]):
$val = ($label === 'Otro motivo') ? 'Otro' : $label;
?>
<label class="reason-option" onclick="selectReason(this, '<?= addslashes($val) ?>')">
<input type="radio" name="reason_type" value="<?= htmlspecialchars($val) ?>" required>
<span style="font-size:18px;"><?= $icon ?></span>
<?= htmlspecialchars($label) ?>
</label>
<?php endforeach; ?>
</div>
<div id="otro-block">
<textarea name="reason" id="reason-text" class="form-control" rows="3"
placeholder="Describe el motivo de la ausencia..."></textarea>
</div>
<div id="detail-block">
<input type="text" name="reason" id="reason-detail" class="form-control"
placeholder="Detalle adicional (opcional)...">
</div>
</div>

<!-- ■■ PASO 3: Documento ■■ -->
<div class="panel" style="margin-bottom:20px;">
<p class="step-title">■ Paso 3 – Documento de respaldo <span
style="color:#999;font-weight:400;">(opcional)</span></p>
<input type="file" name="document" class="form-control" accept=".pdf,.jpg,.jpeg,.png">
<small style="color:#888;display:block;margin-top:4px;">PDF, JPG o PNG – máximo 5MB</small>
</div>

<div style="display:flex;gap:10px;">
<button type="submit" class="btn btn-success" id="submitBtn" disabled>■ Enviar Justificación</button>
<a href="?action=my_attendance" class="btn btn-outline">Cancelar</a>
</div>

</form>
<?php endif; ?>
</div>

<script>
function toggleCheck(item) {
    var cb = item.querySelector('input[type=checkbox]');
    cb.checked = !cb.checked;
    item.classList.toggle('checked', cb.checked);
    updateCount();
}

function toggleAll(master) {
    document.querySelectorAll('.absence-item input[type=checkbox]').forEach(function(cb) {
        cb.checked = master.checked;
        cb.closest('.absence-item').classList.toggle('checked', master.checked);
    });
    updateCount();
}

function updateCount() {
    var checked = document.querySelectorAll('.absence-item input[type=checkbox]:checked');
    var n = checked.length;

    document.getElementById('selected-count').textContent = n + (n === 1 ? ' seleccionado' : ' seleccionados');

    var noteTutor = document.getElementById('note-tutor');
    var noteInsp = document.getElementById('note-insp');

    if (n === 0) {
        noteTutor.style.display = 'none';
        noteInsp.style.display = 'none';
    }
}

```

```

    } else if (n <= 3) {
        document.getElementById('lbl-t').textContent = n;
        noteTutor.style.display = 'block';
        noteInsp.style.display = 'none';
    } else {
        document.getElementById('lbl-i').textContent = n;
        noteTutor.style.display = 'none';
        noteInsp.style.display = 'block';
    }

    // Estado del master checkbox
    var all = document.querySelectorAll('.absence-item input[type=checkbox]');
    var masterCb = document.getElementById('select-all');
    masterCb.indeterminate = (n > 0 && n < all.length);
    masterCb.checked = (n > 0 && n === all.length);

    checkReady();
}

function selectReason(el, val) {
    document.querySelectorAll('.reason-option').forEach(function(r){ r.classList.remove('selected'); });
    el.classList.add('selected');
    el.querySelector('input[type=radio]').checked = true;

    var otroBlock = document.getElementById('otro-block');
    var detailBlock = document.getElementById('detail-block');
    var reasonText = document.getElementById('reason-text');
    var reasonDetail = document.getElementById('reason-detail');

    if (val === 'Otro') {
        otroBlock.style.display = 'block';
        detailBlock.style.display = 'none';
    }

    reasonText.required = true;
    reasonText.name = 'reason';
    reasonDetail.name = '';
    } else {
        otroBlock.style.display = 'none';
        detailBlock.style.display = 'block';
        reasonText.required = false;
        reasonText.name = '';
        reasonDetail.name = 'reason';
    }
    checkReady();
}

function checkReady() {
    var hasAbsence = document.querySelectorAll('.absence-item input:checked').length > 0;
    var hasReason = document.querySelector('input[name=reason_type]:checked');
    document.getElementById('submitBtn').disabled = !(hasAbsence && hasReason);
}
</script>

</body>
</html>

```

views/schedule/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>Gestión de Horarios - EcuAsist</title>
</head>
<body>

<?php include BASE_PATH . '/views/partials/navbar.php'; ?>

```

```

<div class="breadcrumb">
<a href="?action=dashboard">■ Inicio</a> &rsquo;
Gestión de Horarios
</div>

<div class="container">

<!-- Header -->
<div class="page-header purple">
<div class="ph-icon">■■</div>
<div>
<h1>Horarios de Clases</h1>
<p>Configure el horario semanal por curso</p>
</div>
</div>

<!-- Tabla de cursos -->
<div class="table-wrap">
<div class="table-info">
<span>■ <strong><?= count($courses) ?></strong> cursos disponibles</span>
</div>
<?php if(empty($courses)): ?>
<div class="empty-state">
<div class="icon">■</div>
<p>No hay cursos registrados.</p>
</div>
<?php else: ?>
<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 $course): ?>
<tr>
<td><strong><?= htmlspecialchars($course['name']) ?></strong></td>
<td><?= htmlspecialchars($course['grade_level']) ?></td>
<td><?= htmlspecialchars($course['parallel']) ?></td>
<td><?= ucfirst($course['shift_name']) ?></td>
<td>
<a href="?action=manage_schedule&course_id=<?= $course['id'] ?>"
class="btn btn-primary btn-sm">
■■ Configurar Horario
</a>
</td>
</tr>
</tbody>
<?php endforeach; ?>
</table>
<?php endif; ?>
</div>

</div>
</body>
</html>

```

views/schedule/manage.php

[illegible]

```

if(!empty($_SESSION['institution_id'])) {
    try {
        $db2 = new Database();
        $instRow = $db2->connect()->prepare("SELECT working_days_list FROM institutions WHERE id = :id");
        $instRow->execute([':id' => $_SESSION['institution_id']]);
        $inst = $instRow->fetch();
        if(!empty($inst['working_days_list'])) {
            $parsed = json_decode($inst['working_days_list'], true);
            if(is_array($parsed) && count($parsed)) $workingDays = $parsed;
        }
        catch (PDOException $e) {
            // Columna working_days_list no existe aun, usar dias por defecto
        }
    }
}

```

```

$dayLabels = ['lunes'=>'Lunes','martes'=>'Martes','miercoles'=>'Miércoles',
    'jueves'=>'Jueves','viernes'=>'Viernes','sabado'=>'Sábado'];

```

```

$isTecnico = strpos($course['grade_level'],'BT') !== false
    || strpos($course['grade_level'],'Técnico') !== false
    || ($course['education_type'] ?? '') === 'bt';
$maxHours = $isTecnico ? 8 : 7;
$recreoAfter = 4;

```

```

// Index schedule grid
$grid = [];
foreach($schedule as $cls) {
    $grid[$cls['day_of_week']][$cls['period_number']] = $cls;
}

```

```

// Count assigned hours per subject
$assignedCount = []; // subject_id -> count
foreach($schedule as $cls) {
    $assignedCount[$cls['subject_id']] = ($assignedCount[$cls['subject_id']] ?? 0) + 1;
}
?>

```

```

<!DOCTYPE html>
<html lang="es">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Horario - <?= htmlspecialchars($course['name']) ?></title>
    <style>
        :root {
            --ink: #1a2035;
            --muted: #6b7a99;
            --bg: #eef0f7;
            --surface: #ffffff;
            --border: #ddelee;
            --accent: #4361ee;
            --success: #06d6a0;
            --danger: #ef476f;
            --warn: #ffd166;
            --recreo-bg:#f8f9fc;
            --cell-h: 70px;
            --label-w: 110px;
            --radius: 12px;
            --shadow: 0 2px 16px rgba(67,97,238,.09);
        }
        *{margin:0;padding:0;box-sizing:border-box;}
        body{font-family:'Segoe UI',system-ui,sans-serif;background:var(--bg);color:var(--ink);}

        .wrap{max-width:1440px;margin:0 auto;padding:20px 16px 60px;}

        .crumb{font-size:12px;color:var(--muted);margin-bottom:14px;}
        .crumb a{color:var(--accent);text-decoration:none;}

        /* Page header */
        .ph{background:linear-gradient(135deg,#1a237e 0%,#4361ee 100%);
            border-radius:var(--radius);padding:18px 24px;margin-bottom:18px;
            display:flex;align-items:center;gap:14px;flex-wrap:wrap;

```



```

}

/* Schedule table */
.st {
border-collapse:collapse;
width:100%;
table-layout:fixed;
}

/* Sticky hour column */
.st thead th:first-child,
.st tbody td:first-child {
position:sticky;
left:0;
z-index:2;
}

```

```

/* Day column fixed width */
.st colgroup col.day-col { width:160px; }
.st colgroup col.hr-col { width:80px; }

/* Header */
.st thead th {
background:#1a237e; color:#fff;
font-size:12px; font-weight:800;
text-transform:uppercase; letter-spacing:.06em;
height:46px; text-align:center; padding:0 10px;
border-right:1px solid rgba(255,255,255,.15);
white-space:nowrap;
}
.st thead th:first-child {
background:#0d1545; width:80px; min-width:80px;
border-right:3px solid rgba(255,255,255,.2);
}

/* Period label - sticky */
.pl {
background:#f0f2fa;
border-right:3px solid #c5cae9;
border-bottom:1px solid #ddeeee;
height:82px; vertical-align:middle;
text-align:center; padding:0;
width:80px; min-width:80px;
}
.pl .n {
font-size:26px; font-weight:900; color:#9fa8da;
display:block; line-height:1;
}
.pl .l {
font-size:10px; color:var(--muted); margin-top:2px;
font-weight:600; letter-spacing:.04em;
}

/* Recreo */
.rl {

```

```

background:#fafbfe; border-right:3px solid #c5cae9;
border-bottom:1px solid #ddeeee;
height:34px; vertical-align:middle; text-align:center;
position:sticky; left:0; z-index:2;
}
.rl span {
font-size:10px; font-weight:800; color:#b0bce8;
text-transform:uppercase; letter-spacing:.1em;
}
.rc {
background:#fafbfe;
border-right:1px solid #e8eaf6;
border-bottom:1px solid #ddeeee;
height:34px;
}

```

```

/* Schedule cells */
.sc {
border-right:1px solid #ddelee;
border-bottom:1px solid #ddelee;
height:82px; vertical-align:top;
padding:6px; position:relative;
transition:background .12s;
width:160px; min-width:160px;
cursor:pointer;
}
.sc:hover { background:#f0f3fd; }
.sc.over { background:#e8effe; outline:2px dashed var(--accent); }
.sc.occ { cursor:default; }
.sc.occ:hover { background:transparent; }

/* Class card */
.cc {
height:100%; border-radius:8px; padding:6px 10px;
display:flex; flex-direction:column; justify-content:center;
border-left:4px solid transparent; position:relative; overflow:hidden;
}
.cc .s {
font-size:13px; font-weight:800; line-height:1.25;
white-space:nowrap; overflow:hidden; text-overflow:ellipsis;
}

.cc .t {
font-size:11px; margin-top:3px; opacity:.7;
white-space:nowrap; overflow:hidden; text-overflow:ellipsis;
}
.cc .x {
position:absolute; top:4px; right:4px;
width:18px; height:18px; border-radius:50%;
border:none; cursor:pointer; font-size:11px;
display:flex; align-items:center; justify-content:center;
background:rgba(0,0,0,.15); color:#fff;
opacity:0; transition:opacity .12s;
}
.cc:hover .x { opacity:1; }
.cc .x:hover { background:var(--danger); }

/* Empty hint */
.eh {
height:100%; display:flex; align-items:center; justify-content:center;
color:#d0d5ea; font-size:20px; border-radius:8px;
border:2px dashed transparent; transition:all .12s;
}
.sc:hover .eh { border-color:#c5cae9; color:#9fa8da; }

/* Legend */
.leg {
display:flex; gap:12px; flex-wrap:wrap;
padding:10px 16px; border-top:2px solid var(--border);
font-size:11px; color:var(--muted);
}
.leg-i { display:flex; align-items:center; gap:5px; }
.leg-d { width:10px; height:10px; border-radius:50%; }

/* Modals */
.mo { display:none; position:fixed; inset:0; background:rgba(10,15,40,.55);
z-index:8000; align-items:center; justify-content:center; }
.mo.on { display:flex; }
.mb { background:#fff; border-radius:14px; padding:26px;
max-width:380px; width:92%; box-shadow:0 10px 50px rgba(0,0,0,.2); }
.mb h3 { font-size:1rem; font-weight:800; margin-bottom:8px; }

.mb p { color:var(--muted); font-size:13px; margin-bottom:18px; line-height:1.55; }
.ma { display:flex; gap:8px; justify-content:flex-end; }
.btn { padding:8px 18px; border:none; border-radius:7px; font-size:13px;
font-weight:700; cursor:pointer; transition:opacity .12s; }
.btn:hover { opacity:.85; }

```


[illegible]

```

<tr>
<td class="pl">
<span class="n"><?=$p ?></span>
<span class="l"><?=$p ?>a hora</span>

```

```

</td>
<?php foreach($workingDays as $day):
$cls = $grid[$day][$p] ?? null; ?>
<td class="sc <?=$cls ? 'occ' : '' ?>"
data-day="<?=$day ?>"
data-period="<?=$p ?>"
<?=$cls ? 'data-sid="'. $cls['id']. ' "' : '' ?>
ondragover="dov(event)"
ondragleave="dlv(event)"
ondrop="drp(event)"
onclick="cellClick(this)">
<?php if($cls): ?>
<div class="cc" data-subject-id="<?=$cls['subject_id'] ?>"
draggable="true"
ondragstart="cardDragStart(event, <?=$cls['id'] ?>, '<?=$cls['subject_name'] ?>'"
ondragend="cardDragEnd(event)"
style="cursor:grab;">
<button class="x" onclick="openDel(event,<?=$cls['id'] ?>,<?=$cls['subject_name'] ?>','<?=$dayLabels[$day] ?? ucfirst($day) ?>','<?=$p ?>)">x</button>
<span class="s"><?=$cls['subject_name'] ?></span>
<span class="t">■ <?=$cls['teacher_name'] ?? 'Sin docente' ?></span>
</div>
<?php else: ?>
<div class="eh">■</div>
<?php endif; ?>
</td>
<?php endforeach; ?>
</tr>

<?php endfor; ?>
</tbody>
</table>
</div>
<div class="leg" id="leg"></div>
</div>
</div>

<!-- Toast -->
<div id="toast"></div>

```

```

<!-- Delete modal -->
<div class="mo" id="moDel">
<div class="mb">
<h3>■ Eliminar Clase</h3>
<p id="delTxt"></p>
<div class="ma">
<button class="btn btn-g" onclick="closeMo('moDel')">Cancelar</button>
<form method="POST" action="?action=delete_schedule_class" style="display:inline;" id="delFrm">
<input type="hidden" name="schedule_id" id="delSid">
<input type="hidden" name="course_id" value="<?=$course['id'] ?>">
<button type="submit" class="btn btn-d">Eliminar</button>
</form>
</div>
</div>
</div>

<!-- Assign modal -->
<div class="mo" id="moAsgn">
<div class="mb">
<h3>■ Asignar Materia</h3>
<p id="asgnLbl" style="margin-bottom:10px;"></p>

```

```

<select id="asgnSel">
<option value="">Seleccionar materia...</option>
</select>
<div id="asgnTch" style="font-size:12px;color:var(--muted);margin-bottom:14px;min-height:16px;"></div>
<div class="ma">
<button class="btn btn-g" onclick="closeAsgn()">Cancelar</button>
<button class="btn btn-p" id="asgnOk" onclick="confirmAsgn()" disabled>Asignar</button>
</div>
</div>
</div>

```

```

<!-- Hidden save form -->
<form method="POST" action="?action=manage_schedule&course_id=<?=$course['id'] ?>" id="saveFrm"
style="display:none;">
  <input type="hidden" name="subject_id" id="sf_sub">
  <input type="hidden" name="teacher_id" id="sf_tch">
  <input type="hidden" name="day_of_week" id="sf_day">
  <input type="hidden" name="period_number" id="sf_per">
</form>

```

```

<script>
const CID = <?=(int)$course['id'] ?>;
const DLABELS = <?=json_encode(array_combine($workingDays, array_map(fn($d)=>$dayLabels[$d]?ucfirst($d),
$workingDays))) ?>;
const ASSIGNED = <?=json_encode($assignedCount) ?>; // {subjectId: count}

// Rich subject color palette
const PAL = [
  {bg:'#e8f4fd',bd:'#5b9bd5',tx:'#0d4f8b'},
  {bg:'#fef9e7',bd:'#f4c430',tx:'#7d5a00'},
  {bg:'#eafaf1',bd:'#52be80',tx:'#1a5c36'},
  {bg:'#fddedec',bd:'#e74c3c',tx:'#7b241c'},
  {bg:'#f5eef8',bd:'#a569bd',tx:'#5b2c6f'},
  {bg:'#eaf4fb',bd:'#48c9b0',tx:'#0e6655'},
  {bg:'#fef5e7',bd:'#e59866',tx:'#7e5109'},
  {bg:'#f4ecf7',bd:'#c39bd3',tx:'#6c3483'},
  {bg:'#e8f6f3',bd:'#76d7c4',tx:'#148f77'},
  {bg:'#fdfeff',bd:'#85929e',tx:'#2e4057'},
  {bg:'#f9f3e3',bd:'#d4ac0d',tx:'#7d6608'},
  {bg:'#ebf5fb',bd:'#3498db',tx:'#1a5276'},
];
const cmap = {};
let subjects = [];
let dragSub = null;
let selSub = null; // mobile selected
let asgnTarget = null;

fetch('?action=get_course_subjects_schedule&course_id=' + CID)
  .then(r => r.json())
  .then(data => {
    subjects = data;
    subjects.forEach((s,i) => cmap[s.subject_id] = i % PAL.length);

    const list = document.getElementById('chipsList');
    list.innerHTML = '';
    if (!subjects.length) {
      list.innerHTML = '<div class="no-sub">Sin materias.<br>Ve a ■ Asignaturas.</div>';
      return;
    }
  })

```

```

subjects.forEach(s => {
  const sid = String(s.subject_id);
  const p = PAL[cmap[sid]];
  const hrs = parseInt(s.hours_per_week) || 1;
  const done = parseInt(ASSIGNED[sid]) || 0;
  const pct = Math.min(100, Math.round(done / hrs * 100));
  const full = done >= hrs;
  const barCls = full ? 'hrs-done' : (done > 0 ? 'hrs-partial' : '');

  const chip = document.createElement('div');
  chip.className = 'chip' + (full ? ' full' : '');

```


[illegible]

[illegible]

[illegible]

views/institution/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>Configuración de Institución - EcuAsist</title>
  <style>
    /* ■■ Jornadas toggle ■■ */
    .shifts-grid { display: flex; gap: 12px; flex-wrap: wrap; margin-top: 4px; }
    .shift-card {
      display: flex; align-items: center; gap: 10px;
      padding: 12px 20px; border-radius: 10px; cursor: pointer;
      border: 2px solid #e0e0e0; background: #f9f9f9;
      transition: all .2s; user-select: none; min-width: 150px;
    }
    .shift-card:hover { border-color: #90caf9; background: #e3f2fd; }
    .shift-card.active { border-color: #1976d2; background: #e3f2fd; }
    .shift-card .shift-icon { font-size: 22px; }
    .shift-card .shift-info { flex: 1; }
    .shift-card .shift-name { font-weight: 600; font-size: 14px; color: #333; }
    .shift-card .shift-status { font-size: 11px; margin-top: 2px; }
    .shift-card.active .shift-status { color: #1565c0; }
    .shift-card:not(.active) .shift-status { color: #999; }
    .shift-card .shift-toggle {
      width: 36px; height: 20px; border-radius: 10px; position: relative;
      background: #ccc; transition: background .2s; flex-shrink: 0;
    }
    .shift-card.active .shift-toggle { background: #1976d2; }
    .shift-card .shift-toggle::after {
      content: ''; position: absolute; top: 2px; left: 2px;
      width: 16px; height: 16px; border-radius: 50%; background: #fff;
      transition: left .2s; box-shadow: 0 1px 3px rgba(0,0,0,.2);
    }
    .shift-card.active .shift-toggle::after { left: 18px; }
    .shift-saving { opacity: .5; pointer-events: none; }

    /* ■■ Logo preview ■■ */
    .logo-wrap { display: flex; align-items: center; gap: 20px; flex-wrap: wrap; }
    .logo-preview {
      width: 100px; height: 100px; border-radius: 8px; border: 2px dashed #ddd;
      display: flex; align-items: center; justify-content: center;
      background: #fafafa; overflow: hidden; flex-shrink: 0;
    }
    .logo-preview img { max-width: 100%; max-height: 100%; object-fit: contain; }
    .logo-preview .no-logo { font-size: 32px; color: #ccc; }
    .logo-upload-btn {
      display: inline-block; padding: 8px 16px; background: #f0f7ff;
      border: 1.5px solid #90caf9; border-radius: 6px; cursor: pointer;
      font-size: 13px; color: #1565c0; transition: all .2s;
    }
    .logo-upload-btn:hover { background: #e3f2fd; }
    #logo-input { display: none; }
  </style>
</head>
<body>

<?php include BASE_PATH . '/views/partials/navbar.php'; ?>

<div class="breadcrumb">
  <a href="?action=dashboard">■ Inicio</a> &rsquo; Configuración de Institución
</div>

<div class="container">
```



```

<?php if(isset($_GET['success'])): ?>
<div class="alert alert-success">✓ Información actualizada correctamente</div>
<?php endif; ?>

<div class="page-header" style="background:linear-gradient(135deg,#1a237e,#283593);">
<div class="ph-icon">■</div>
<div>
<h1>Configuración de Institución</h1>
<p>Datos generales, jornadas y logotipo</p>
</div>
</div>

<form method="POST" action="?action=update_institution" enctype="multipart/form-data" id="instForm">
<!-- Respaldo: logo actual en caso de que no se suba archivo nuevo -->
<input type="hidden" name="current_logo_path" value="<?= htmlspecialchars($institution['logo_path']) ?>">

```

```

<div style="display:grid;grid-template-columns:1fr 1fr;gap:20px;margin-bottom:20px;">

<!-- ■■ Panel izquierdo: datos ■■ -->
<div>
<div class="panel" style="margin-bottom:16px;">
<h3 style="font-size:.95rem;color:#555;margin-bottom:14px;">■ Datos de la Institución</h3>
<div class="form-group">
<label>Nombre de la institución *</label>
<input type="text" name="name" class="form-control" required
value="<?= htmlspecialchars($institution['name']) ?>">
</div>
<div class="form-row">
<div class="form-group">
<label>Código AMIE</label>
<input type="text" name="amie_code" class="form-control"
value="<?= htmlspecialchars($institution['amie_code']) ?>"
placeholder="17H01988">
</div>
<div class="form-group">
<label>Teléfono</label>
<input type="text" name="phone" class="form-control"
value="<?= htmlspecialchars($institution['phone']) ?>"
placeholder="02-2345678">
</div>
</div>
<div class="form-group">
<label>Email institucional</label>
<input type="email" name="email" class="form-control"
value="<?= htmlspecialchars($institution['email']) ?>"
placeholder="info@institucion.edu.ec">
</div>
<div class="form-group">
<label>Sitio web</label>
<input type="text" name="website" class="form-control" id="website"
value="<?= htmlspecialchars($institution['website']) ?>"
placeholder="www.institucion.edu.ec"
onblur="autoHttps(this)">
</div>
<div class="form-group">

```

```

<label>Nombre del Director/Rector</label>
<input type="text" name="director_name" class="form-control"
value="<?= htmlspecialchars($institution['director_name']) ?>"
placeholder="MSc. Nombre Apellido">
</div>
</div>

```

```

<div class="panel" style="margin-bottom:16px;">
<h3 style="font-size:.95rem;color:#555;margin-bottom:14px;">■ Ubicación</h3>
<div class="form-row">
<div class="form-group">
<label>Provincia</label>
<select name="province" id="province" class="form-control" onchange="loadCities()">
<option value="">Seleccionar...</option>

```

```

<?php foreach(getProvinces() as $prov): ?>
<option value="<?= $prov ?>" <?= ($institution['province'] ?? '') === $prov ? 'selected' : '' ?>>
<?= $prov ?>
</option>
<?php endforeach; ?>
</select>
</div>
<div class="form-group">
<label>Ciudad</label>
<select name="city" id="city" class="form-control">
<option value="<?= htmlspecialchars($institution['city'] ?? '') ?>">
<?= htmlspecialchars($institution['city'] ?? 'Seleccionar provincia...') ?>
</option>
</select>
</div>
</div>
<div class="form-group">
<label>Dirección</label>
<input type="text" name="address" class="form-control"
value="<?= htmlspecialchars($institution['address'] ?? '') ?>"
placeholder="Av. Principal 123">
</div>
</div>
</div>
<!-- ■■ Panel derecho: logo + jornadas ■■ -->

```

```

<div>
<!-- Logo -->
<div class="panel" style="margin-bottom:16px;">
<h3 style="font-size:.95rem;color:#555;margin-bottom:14px;">■■ Logo Institucional</h3>
<div class="logo-wrap">
<div class="logo-preview" id="logoPreview">
<?php
// BASE_URL viene de env.php (local o producción)
$logoUrl = '';
if (!empty($institution['logo_path'])) {
$imgFile = ltrim(str_replace('uploads/', '', $institution['logo_path']), '/');
$logoUrl = BASE_URL . '/img.php?f=' . urlencode($imgFile) . '&v=' . time();
}
?>
<?php if($logoUrl): ?>

<?php else: ?>
<span class="no-logo" id="noLogoIcon">■</span>
<?php endif; ?>
</div>
<div>
<label for="logo-input" class="logo-upload-btn">■ Seleccionar imagen</label>
<input type="file" name="logo" id="logo-input" accept=".jpg,.jpeg,.png,.gif,.webp"
onchange="previewLogo(this)">
<p style="font-size:12px;color:#999;margin-top:6px;">JPG, PNG o WebP – máx. 2MB</p>
<?php if($logoUrl): ?>
<p style="font-size:12px;color:#2e7d32;margin-top:4px;">✓ Logo actual cargado</p>
<?php endif; ?>
</div>
</div>
</div>

<!-- Jornadas toggle -->
<div class="panel">
<h3 style="font-size:.95rem;color:#555;margin-bottom:6px;">■ Jornadas de la Institución</h3>
<p style="font-size:12px;color:#999;margin-bottom:14px;">Clic para activar o desactivar cada jornada</p>

<?php
$shiftIcons = [
'matutina' => ['■', 'Mañana'],
'vespertina' => ['■', 'Tarde'],
'nocturna' => ['■', 'Noche'],
];
?>

```

```

<div class="shifts-grid" id="shiftsGrid">
  <?php foreach($allShifts as $shift):
    $isActive = in_array($shift['id'], $assignedShiftIds);
    $icon = $shiftIcons[strtolower($shift['name'])][0] ?? '■';
    $timeLabel= $shiftIcons[strtolower($shift['name'])][1] ?? ucfirst($shift['name']);
    ?>
    <div class="shift-card <?= $isActive ? 'active' : '' ?>"
      id="shift-<?= $shift['id'] ?>"
      onclick="toggleShift(<?= $shift['id'] ?>, this)"
      title="Clic para <?= $isActive ? 'desactivar' : 'activar' ?>">
      <span class="shift-icon"><?= $icon ?></span>
      <div class="shift-info">
        <div class="shift-name"><?= ucfirst($shift['name']) ?></div>
        <div class="shift-status"><?= $isActive ? '✓ Activa' : 'Inactiva' ?></div>
      </div>
      <div class="shift-toggle"></div>
    </div>
  <?php endforeach; ?>
</div>

<div id="shift-msg" style="font-size:12px;margin-top:10px;min-height:18px;"></div>
</div>
<!-- Días laborables -->
<div class="panel" style="margin-top:16px;">
  <h3 style="font-size:.95rem;color:#555;margin-bottom:6px;">■ Días Laborables</h3>
  <p style="font-size:12px;color:#999;margin-bottom:14px;">Selecciona los días que aplican al horario de
clases</p>
  <?php
    $savedDays = !empty($institution['working_days_list'])
    ? json_decode($institution['working_days_list'], true)
    : ['lunes','martes','miercoles','jueves','viernes'];
    $allDays = [
      'lunes' => 'Lunes',
      'martes' => 'Martes',
      'miercoles'=> 'Miércoles',
      'jueves' => 'Jueves',
      'viernes' => 'Viernes',
      'sabado' => 'Sábado',
    ];
    ?>
    <div style="display:flex;flex-wrap:wrap;gap:8px;">
      <?php foreach($allDays as $key => $label):
        $isChecked = in_array($key, $savedDays);
        $borderClr = $isChecked ? '#1976d2' : '#ddd';
        $bgClr = $isChecked ? '#e3f2fd' : '#fafafa';
        ?>
        <label style="display:flex;align-items:center;gap:7px;padding:7px 14px;border:1.5px solid <?= $borderClr ?
>;border-radius:7px;background:<?= $bgClr ?>;cursor:pointer;font-size:13px;font-weight:600;color:#333;transit
ion:all .15s;" class="day-label">
          <input type="checkbox" name="working_days[]" value="<?= $key ?>"
            <?= $isChecked ? 'checked' : '' ?>
          <div style="font-size:13px;font-weight:600;color:#333;flex-grow:1;">
            <?= $label ?>
          </div>
        </label>
      <?php endforeach; ?>
    </div>

    <div style="display:flex;gap:10px;">
      <button type="submit" class="btn btn-success">■ Guardar Cambios</button>
      <a href="?action=dashboard" class="btn btn-outline">Cancelar</a>
    </div>

  </form>
</div>
<script>

```



```
?>

</body>
</html>
```

views/stats/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>Estadísticas - EcuAsist</title>
</head>
<body>

<?php include BASE_PATH . '/views/partials/navbar.php'; ?>

<div class="breadcrumb">
  <a href="?action=dashboard">■ Inicio</a> &rsquo; Estadísticas
</div>

<div class="container">

  <!-- Header -->
  <div class="page-header purple">
    <div class="ph-icon">■</div>
    <div>
      <h1>Estadísticas de Asistencia</h1>
      <p>Análisis y métricas del período seleccionado</p>
    </div>
  </div>

  <!-- Filtro de fechas -->
  <div class="filters">
    <h3>■ Período de análisis</h3>
    <form method="GET">
      <input type="hidden" name="action" value="stats">
      <div class="filter-grid">
        <div class="filter-group">
          <label>Fecha Inicio</label>
          <input type="date" name="start_date" value="<?= $_GET['start_date'] ?? date('Y-m-01') ?>">
        </div>
        <div class="filter-group">
          <label>Fecha Fin</label>
          <input type="date" name="end_date" value="<?= $_GET['end_date'] ?? date('Y-m-d') ?>">
        </div>
      </div>
      <div class="filter-group" style="display:flex;align-items:flex-end;">
        <button type="submit" class="btn btn-primary" style="width:100%;">Filtrar</button>
      </div>
    </form>
  </div>

  <!-- Stats globales -->
  <?php
    $total = $stats['overall']['total_records'] ?? 0;
    $pres = $stats['overall']['presente'] ?? 0;
    $aus = $stats['overall']['ausente'] ?? 0;
    $tard = $stats['overall']['tardanza'] ?? 0;
    $just = $stats['overall']['justificado'] ?? 0;
    $pct = $total > 0 ? round(($pres / $total) * 100, 1) : 0;
  ?>
  <div class="stats-row">
    <div class="stat-card gray">
      <div class="number"><?= $total ?></div>
      <div class="label">Total Registros</div>
```

```

</div>
<div class="stat-card green">
<div class="number"><?=$pres ?></div>
<div class="label">Presentes (<?=$pct ?>%</div>
<div class="progress-bar"><div class="progress-fill" style="width:<?=$pct ?
>%;background:#28a745;"></div></div>
</div>
<div class="stat-card red">
<div class="number"><?=$aus ?></div>
<div class="label">Ausentes (<?=$total > 0 ? round(($aus/$total)*100,1) : 0 ?>%</div>
<div class="progress-bar"><div class="progress-fill" style="width:<?=$total>0?round(($aus/$total)*100):0 ?
>%;background:#dc3545;"></div></div>
</div>
<div class="stat-card yellow">
<div class="number"><?=$tard ?></div>
<div class="label">Tardanzas</div>
</div>
<div class="stat-card teal">
<div class="number"><?=$just ?></div>
<div class="label">Justificados</div>
</div>

```

```

</div>

```

```

<!-- Asistencia por curso -->
<div class="panel">
<h3 class="blue">■ Asistencia por Curso</h3>
<div class="table-wrap" style="margin-top:8px;">
<table>
<thead>
<tr>
<th>Curso</th>
<th>Total</th>
<th>Presentes</th>
<th>Ausentes</th>
<th>% Asistencia</th>
<th style="min-width:120px;">Indicador</th>
</tr>
</thead>
<tbody>
<?php foreach($stats['by_course'] as $course):
$color = $course['percentage'] >= 90 ? '#28a745' : ($course['percentage'] >= 75 ? '#ffc107' : '#dc3545');
?>
<tr>
<td><strong><?=$course['course_name'] ?></strong></td>
<td><?=$course['total'] ?></td>
<td><?=$course['presente'] ?></td>
<td><?=$course['ausente'] ?></td>
<td><span style="font-weight:700;color:<?=$color ?> "><?=$course['percentage'] ?>%</span></td>
<td>
<div class="progress-bar" style="height:10px;">
<div class="progress-fill" style="width:<?=$course['percentage'] ?>%;background:<?=$color ?> "></div>
</div>
</td>
</tr>
<?php endforeach; ?>
</tbody>
</table>
</div>
</div>

<!-- Top 10 ausencias -->

```

```

<div class="panel">
<h3 class="red">■ Top 10 – Estudiantes con Más Ausencias</h3>
<?php if(!empty($stats['by_student'])): ?>
<?php $maxAbs = $stats['by_student'][0]['total_absences'] ?? 1; ?>
<div style="margin-top:8px;">
<?php foreach($stats['by_student'] as $i => $student): ?>
<div style="display:flex;align-items:center;gap:10px;padding:8px 0;border-bottom:1px solid #f0f0f0;">
<span style="font-size:0.75rem;color:#999;min-width:20px;text-align:right;"><?=$i+1 ?>.</span>
<span style="flex:1;font-size:0.88rem;"><strong><?=$student['student_name'] ?></strong>

```

```

        <span style="color:#888;font-size:0.8rem;margin-left:6px;"><?= htmlspecialchars($student['course_name']) ?
    ></span>
    </span>
    <div style="flex:2;">
        <div class="progress-bar" style="height:8px;">
            <div class="progress-fill" style="width:<?= round(($student['total_absences']/ $maxAbs)*100) ?
        >%;background:#dc3545;"></div>
        </div>
        <div>
            <span style="font-weight:700;color:#dc3545;min-width:28px;text-align:right;"><?= $student['total_absences'] ?
            ></span>
        </div>
    </div>
    <?php endforeach; ?>
</div>
<?php else: ?>
<p style="color:#999;font-size:0.88rem;padding:12px 0;">No hay datos de ausencias en este período.</p>
<?php endif; ?>
</div>

<!-- Tendencia diaria -->
<div class="panel">
<h3 class="blue">■ Tendencia Diaria</h3>
<div class="table-wrap" style="margin-top:8px;">
    <table>
    <thead>
    <tr>
    <th>Fecha</th>
    <th>Registros</th>
    <th>Presentes</th>
    <th>% Asistencia</th>
    <th style="min-width:120px;">Indicador</th>
    </tr>
    </thead>
    <tbody>

<?php foreach($stats['by_day'] as $day):
    $dc = $day['percentage'] >= 90 ? '#28a745' : ($day['percentage'] >= 75 ? '#ffc107' : '#dc3545');
    ?>
    <tr>
    <td><?= date('d/m/Y', strtotime($day['date'])) ?></td>
    <td><?= $day['total'] ?></td>
    <td><?= $day['presente'] ?></td>
    <td><span style="font-weight:600;color:<?= $dc ?>;"><?= $day['percentage'] ?>%</span></td>
    <td>
        <div class="progress-bar" style="height:8px;">
        <div class="progress-fill" style="width:<?= $day['percentage'] ?>%;background:<?= $dc ?>;"></div>
        </div>
    </td>
    </tr>
    <?php endforeach; ?>
</tbody>
</table>
</div>
</div>
</body>
</html>

```

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>
</head>

```



```

<body>

<?php include BASE_PATH . '/views/partials/navbar.php'; ?>

<div class="breadcrumb">
    ■ Inicio &rsquo; Dashboard
</div>

<div class="container">

    <!-- Header de bienvenida -->
    <div class="page-header" style="background: linear-gradient(135deg, #667eea, #764ba2);">
        <div class="ph-icon">■</div>
        <div>
            <h1>Bienvenido, <?= htmlspecialchars($_SESSION['first_name'] . ' ' . $_SESSION['last_name']) ?></h1>

        </div>
    </div>

    <!-- ■■ Stats: Autoridad ■■ -->
    <?php if(Security::hasRole('autoridad') && isset($stats['total_students'])): ?>
    <div class="stats-row">
        <div class="stat-card blue">
            <div class="number"><?= $stats['total_students'] ?></div>
            <div class="label">Estudiantes</div>
        </div>
        <div class="stat-card green">
            <div class="number"><?= $stats['total_teachers'] ?></div>
            <div class="label">Docentes</div>
        </div>
        <div class="stat-card orange">
            <div class="number"><?= $stats['total_courses'] ?></div>

<div class="label">Cursos</div>
</div>
<?php if(!empty($stats['today_attendance']['total']) && $stats['today_attendance']['total'] > 0): ?>
    <div class="stat-card teal">
        <div class="number"><?= round(($stats['today_attendance']['presente'] /
$stats['today_attendance']['total']) * 100, 1) ?>%</div>
        <div class="label">Asistencia Hoy</div>
        <div class="progress-bar">
            <div class="progress-fill" style="width:<?= round(($stats['today_attendance']['presente'] /
$stats['today_attendance']['total']) * 100) ?>%;background:#17a2b8;"></div>
        </div>
    </div>
<?php endif; ?>
</div>
<?php endif; ?>

    <!-- ■■ Stats: Docente ■■ -->
    <?php if(Security::hasRole('docente') && isset($stats['my_courses'])): ?>
    <div class="stats-row">
        <div class="stat-card blue">
            <div class="number"><?= $stats['my_courses'] ?></div>
            <div class="label">Mis Cursos</div>
        </div>
        <div class="stat-card green">
            <div class="number"><?= $stats['my_students'] ?></div>
            <div class="label">Mis Estudiantes</div>
        </div>
        <?php if(!empty($stats['tutor_course'])): ?>
        <div class="stat-card purple">
            <div style="font-size:1.4rem;margin-bottom:4px;">■</div>
            <div class="number" style="font-size:1rem;"><?= htmlspecialchars($stats['tutor_course']) ?></div>
            <div class="label">Mi Curso como Tutor</div>
        </div>
    <?php endif; ?>
</div>
<?php endif; ?>

    <!-- ■■ Stats: Estudiante ■■ -->
    <?php if(Security::hasRole('estudiante') && isset($stats['my_attendance'])): ?>

```

```

<?php $pctAsis = $stats['my_attendance']['total'] > 0 ? round(($stats['my_attendance']['presente'] /
$stats['my_attendance']['total']) * 100, 1) : 0; ?>
<div class="stats-row">
<div class="stat-card green">

<div class="number"><?= $pctAsis ?>%</div>
<div class="label">Mi Asistencia (30 días)</div>
<div class="progress-bar"><div class="progress-fill" style="width:<?= $pctAsis ?
>%;background:#28a745;"></div></div>
<div class="stat-card red">
<div class="number"><?= $stats['my_attendance']['ausente'] ?></div>
<div class="label">Ausencias</div>
</div>
<div class="stat-card yellow">
<div class="number"><?= $stats['my_attendance']['tardanza'] ?></div>
<div class="label">Tardanzas</div>
</div>
</div>
<?php endif; ?>

<!-- ■■ Stats: Representante ■■ -->
<?php if(Security::hasRole('representante') && isset($stats['my_children'])): ?>
<div class="stats-row">
<div class="stat-card blue">
<div class="number"><?= $stats['my_children'] ?></div>
<div class="label">Representados</div>
</div>
</div>
<?php endif; ?>

<!-- ■■ Menú de accesos rápidos ■■ -->
<div class="section-title">Accesos rápidos</div>
<div style="display:grid;grid-template-columns:repeat(auto-fill,minmax(240px,1fr));gap:14px;margin-bottom:24
px;">

<?php if(Security::hasRole(['docente','autoridad'])): ?>
<a href="?action=attendance_register" class="menu-card green">
<div class="mc-icon">■</div>
<div class="mc-title">Registrar Asistencia</div>
<div class="mc-desc">Tomar asistencia de estudiantes</div>
</a>
<?php endif; ?>

<?php if(Security::hasRole('docente')): ?>
<a href="?action=tutor_course_attendance" class="menu-card blue">
<div class="mc-icon">■</div>

<div class="mc-title">Asistencia de Mi Curso</div>
<div class="mc-desc">Ver asistencia del curso que tutoro</div>
</a>
<?php endif; ?>

<?php if(Security::hasRole(['docente','inspector','autoridad'])): ?>
<a href="?action=attendance_view" class="menu-card blue">
<div class="mc-icon">■</div>
<div class="mc-title">Ver Asistencias</div>
<div class="mc-desc">Consultar registros de asistencia</div>
</a>
<a href="?action=attendance_calendar" class="menu-card orange">
<div class="mc-icon">■</div>
<div class="mc-title">Calendario</div>
<div class="mc-desc">Vista mensual de asistencias</div>
</a>
<?php endif; ?>

<?php if(Security::hasRole('estudiante')): ?>
<a href="?action=my_attendance" class="menu-card blue">
<div class="mc-icon">■</div>
<div class="mc-title">Mi Asistencia</div>
<div class="mc-desc">Ver mi historial de asistencia</div>
</a>

```

```

<a href="?action=my_justifications" class="menu-card green">
<div class="mc-icon">■</div>
<div class="mc-title">Mis Justificaciones</div>
<div class="mc-desc">Justificar ausencias</div>
</a>
<?php endif; ?>

<?php if(Security::hasRole(['autoridad','inspector'])): ?>
<a href="?action=pending_justifications" class="menu-card orange">
<div class="mc-icon">■</div>
<div class="mc-title">Revisar Justificaciones</div>
<div class="mc-desc">Aprobar o rechazar justificaciones pendientes</div>
</a>
<?php endif; ?>

<?php if(Security::hasRole('autoridad')): ?>

```

```

<a href="?action=stats" class="menu-card purple">
<div class="mc-icon">■</div>
<div class="mc-title">Estadísticas</div>
<div class="mc-desc">Análisis y métricas de asistencia</div>
</a>
<a href="?action=reports" class="menu-card purple">
<div class="mc-icon">■</div>
<div class="mc-title">Reportes</div>
<div class="mc-desc">Generar informes PDF y Excel</div>
</a>
<a href="?action=users" class="menu-card blue">
<div class="mc-icon">■</div>
<div class="mc-title">Gestión de Usuarios</div>
<div class="mc-desc">Administrar usuarios y roles</div>
</a>
<a href="?action=academic" class="menu-card teal">
<div class="mc-icon">■</div>
<div class="mc-title">Configuración Académica</div>
<div class="mc-desc">Cursos, materias, matrículas</div>
</a>
<a href="?action=assignments" class="menu-card green">
<div class="mc-icon">■</div>
<div class="mc-title">Asignaciones Docentes</div>
<div class="mc-desc">Gestionar docentes y tutores</div>
</a>
<a href="?action=manage_representatives" class="menu-card orange">
<div class="mc-icon">■■■■■■■■</div>
<div class="mc-title">Gestión Representantes</div>
<div class="mc-desc">Asignar representantes a estudiantes</div>
</a>
<a href="?action=backups" class="menu-card dark">
<div class="mc-icon">■</div>
<div class="mc-title">RespalDOS</div>
<div class="mc-desc">Gestionar respaldos de base de datos</div>
</a>
<?php endif; ?>

<?php if(Security::hasRole('representante')): ?>
<a href="?action=my_children" class="menu-card purple">
<div class="mc-icon">■■■■■■■■</div>

```

```

<div class="mc-title">Mis Representados</div>
<div class="mc-desc">Ver estudiantes y su asistencia</div>
</a>
<?php endif; ?>

```

```

</div>
</div>

```

```

<style>
/* Menu cards del dashboard */
.menu-card {
display: flex;
flex-direction: column;
background: #fff;

```

```

border: 1px solid #e0e0e0;
border-left: 4px solid #007bff;
border-radius: 8px;
padding: 20px;
text-decoration: none;
color: #333;
transition: transform 0.2s, box-shadow 0.2s;
cursor: pointer;
}
.menu-card:hover {
transform: translateY(-3px);
box-shadow: 0 6px 20px rgba(0,0,0,0.12);
text-decoration: none;
}
.menu-card .mc-icon { font-size: 1.8rem; margin-bottom: 8px; }
.menu-card .mc-title { font-size: 0.95rem; font-weight: 700; margin-bottom: 4px; }
.menu-card .mc-desc { font-size: 0.8rem; color: #888; }

.menu-card.blue { border-left-color: #007bff; } .menu-card.blue .mc-title { color: #007bff; }
.menu-card.green { border-left-color: #28a745; } .menu-card.green .mc-title { color: #28a745; }
.menu-card.orange { border-left-color: #fd7e14; } .menu-card.orange .mc-title { color: #fd7e14; }
.menu-card.purple { border-left-color: #6f42c1; } .menu-card.purple .mc-title { color: #6f42c1; }
.menu-card.teal { border-left-color: #17a2b8; } .menu-card.teal .mc-title { color: #17a2b8; }
.menu-card.dark { border-left-color: #343a40; } .menu-card.dark .mc-title { color: #343a40; }
</style>
</body>
</html>

```

views/partials/navbar.php

```

<?php
// views/partials/navbar.php

$current_page = $_GET['action'] ?? 'dashboard';

$notifCount = 0;
if (isset($_SESSION['user_id'])) {
require_once BASE_PATH . '/models/Notification.php';
$notifDb = new Database();
$notifModel = new Notification($notifDb);
$notifCount = $notifModel->getUnreadCount($_SESSION['user_id']);
}

// Determinar qué sección está activa para auto-expandirla
$activeSection = '';
if (in_array($current_page, ['attendance_register', 'attendance_view', 'attendance_calendar', 'my_attendance', 'tutor_course_attendance'])) $activeSection = 'asistencia';
elseif (in_array($current_page, ['my_justifications', 'pending_justifications', 'reviewed_justifications']))
$activeSection = 'justificaciones';
elseif (in_array($current_page, ['institution', 'academic', 'users', 'assignments', 'tutor_management', 'manage_representatives', 'schedules', 'backups'])) $activeSection = 'administracion';
elseif (in_array($current_page, ['reports', 'stats'])) $activeSection = 'reportes';
?>
<style>
@import url('https://fonts.googleapis.com/css2?family=Plus+Jakarta+Sans:wght@400;500;600;700&display=swap');

:root {
--nav-bg: #1e40af;
--nav-accent: #60a5fa;
--nav-hover: rgba(96,165,250,0.18);
--nav-text: #e2e8f0;
--nav-muted: #bfdbfe;
--nav-border: rgba(255,255,255,0.12);
--nav-dd-bg: #1d4ed8;
--nav-dd-hover: #2563eb;
--nav-height: 52px;
}
*, ::before, ::after { box-sizing: border-box; }

```

```
/* ■■■ Barra ■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■ */
.ec-nav {
background: var(--nav-bg);
height: var(--nav-height);
}

display: flex; align-items: center;
position: sticky; top: 0; z-index: 1000;
box-shadow: 0 1px 0 var(--nav-border), 0 4px 20px rgba(0,0,0,0.25);
font-family: 'Plus Jakarta Sans', system-ui, sans-serif;
}

.ec-nav__inner {
max-width: 1400px; margin: 0 auto;
padding: 0 20px; width: 100%;
display: flex; align-items: center; gap: 4px;
}

/* ■■■ Brand ■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■ */
.ec-brand { display: flex; align-items: center; gap: 8px; text-decoration: none; margin-right: 12px;
flex-shrink: 0; }
.ec-brand__logo {
width: 28px; height: 28px; background: #fff; border-radius: 7px;
display: flex; align-items: center; justify-content: center;
font-size: 14px; font-weight: 800; color: var(--nav-bg); letter-spacing: -0.5px;
}
.ec-brand__name { font-size: 14px; font-weight: 700; color: #fff; letter-spacing: -0.3px; white-space: nowrap; }
.ec-brand__name span { color: #bfdbfe; }

/* ■■■ Hamburger ■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■ */
.ec-hamburger {
display: none;
flex-direction: column; justify-content: center; align-items: center;
width: 38px; height: 38px; border-radius: 8px;
cursor: pointer; border: none; background: none;
gap: 5px; flex-shrink: 0; padding: 0;
}
.ec-hamburger span {
display: block; width: 20px; height: 2px;
background: #fff; border-radius: 2px;
transition: transform 0.25s, opacity 0.2s; transform-origin: center;
}
.ec-hamburger:hover { background: var(--nav-hover); }
.ec-hamburger.open span:nth-child(1) { transform: translateY(7px) rotate(45deg); }
.ec-hamburger.open span:nth-child(2) { opacity: 0; transform: scaleX(0); }
.ec-hamburger.open span:nth-child(3) { transform: translateY(-7px) rotate(-45deg); }

/* ■■■ Menú desktop ■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■ */
.ec-menu { display: flex; align-items: center; gap: 1px; flex: 1; }
.ec-item { position: relative; }
.ec-item__btn {
display: flex; align-items: center; gap: 5px;
padding: 0 10px; height: var(--nav-height);
color: var(--nav-text); text-decoration: none;
font-size: 13px; font-weight: 500; white-space: nowrap;
cursor: pointer; border: none; background: none;
transition: background 0.15s, color 0.15s; position: relative;
}
.ec-item__btn:hover, .ec-item:hover > .ec-item__btn { background: var(--nav-hover); color: #fff; }
.ec-item__btn.active { color: #fff; background: rgba(255,255,255,0.15); border-radius: 6px; }
.ec-item__icon { font-size: 14px; line-height: 1; }
.ec-item__caret { font-size: 9px; color: var(--nav-muted); margin-left: 1px; transition: transform 0.2s; }
.ec-item:hover .ec-item__caret { transform: rotate(180deg); }

/* ■■■ Dropdown desktop ■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■ */
.ec-dropdown {
display: none; position: absolute;
top: 100%; left: 0; min-width: 210px;
background: var(--nav-dd-bg); border: 1px solid var(--nav-border);
border-radius: 0 0 10px 10px; box-shadow: 0 8px 32px rgba(0,0,0,0.3);
padding: 4px; z-index: 200;
}
```

```
/* Punte invisible para no perder hover al bajar el mouse */  
margin-top: 0;  
}  
.ec-item:hover .ec-dropdown { display: block; }  
/* Puente invisible que cubre el espacio entre btn y dropdown */  
.ec-item_btn::before {  
content: ''; position: absolute;  
bottom: -8px; left: 0; right: 0; height: 8px;  
background: transparent;  
}  
.ec-dropdown a {  
display: flex; align-items: center; gap: 9px; padding: 9px 12px;  
color: var(--nav-text); text-decoration: none; font-size: 13px;  
border-radius: 7px; transition: background 0.12s; border-left: 2px solid transparent;  
}  
.ec-dropdown a:hover { background: var(--nav-dd-hover); color: #fff; }  
.ec-dropdown a.active { background: rgba(96,165,250,0.2); color: #fff; border-left-color: #fff; font-weight:  
600; }  
  
.ec-dropdown_divider { height: 1px; background: var(--nav-border); margin: 4px 8px; }  
  
/* ■■■ Right desktop ■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■ */  
.ec-right { display: flex; align-items: center; gap: 2px; margin-left: auto; flex-shrink: 0; }  
.ec-icon-btn {  
position: relative; display: flex; align-items: center; justify-content: center;  
width: 36px; height: 36px; border-radius: 8px;  
color: var(--nav-text); text-decoration: none; font-size: 16px; transition: background 0.15s;  
}  
.ec-icon-btn:hover { background: var(--nav-hover); color: #fff; }  
.ec-notif-badge {  
position: absolute; top: 3px; right: 3px;  
background: #ef4444; color: #fff; border-radius: 50%;  
min-width: 16px; height: 16px; font-size: 10px; font-weight: 700;  
display: flex; align-items: center; justify-content: center;  
border: 2px solid var(--nav-bg); line-height: 1;  
}  
.ec-user-wrap { position: relative; display: flex; align-items: center; }  
.ec-user-btn { display: flex; align-items: center; padding: 4px; border-radius: 50%; text-decoration: none;  
transition: background 0.15s; }  
.ec-user-btn:hover { background: var(--nav-hover); }  
.ec-user-avatar {  
width: 30px; height: 30px; border-radius: 50%;  
background: rgba(255,255,255,0.22);  
display: flex; align-items: center; justify-content: center;  
font-size: 11px; font-weight: 700; color: #fff; flex-shrink: 0;  
border: 1.5px solid rgba(255,255,255,0.4);  
}  
.ec-user-tooltip {  
display: none; position: absolute; top: calc(100% + 8px); right: 0;  
background: #1e3a8a; color: #e2e8f0; font-size: 12px; font-weight: 500;  
padding: 6px 12px; border-radius: 7px; white-space: nowrap;  
box-shadow: 0 4px 16px rgba(0,0,0,0.3); border: 1px solid rgba(255,255,255,0.12);  
pointer-events: none; font-family: 'Plus Jakarta Sans', system-ui, sans-serif; z-index: 300;  
}  
.ec-user-tooltip::before {  
content: ''; position: absolute; top: -5px; right: 10px;  
width: 8px; height: 8px; background: #1e3a8a;  
border-left: 1px solid rgba(255,255,255,0.12);  
border-top: 1px solid rgba(255,255,255,0.12);  
transform: rotate(45deg);  
}  
  
.ec-user-wrap:hover .ec-user-tooltip { display: block; }  
.ec-logout-btn {  
display: flex; align-items: center; gap: 5px; padding: 6px 10px; border-radius: 8px;  
color: #fca5a5; text-decoration: none; font-size: 13px; font-weight: 500;  
transition: background 0.15s, color 0.15s; white-space: nowrap;  
}  
.ec-logout-btn:hover { background: rgba(239,68,68,0.15); color: #fecaca; }  
.ec-divider-v { width: 1px; height: 22px; background: var(--nav-border); margin: 0 4px; }  
  
/* ■■■ Breadcrumb ■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■ */
```


[illegible]


```

<?php if(Security::hasRole(['docente','autoridad'])): ?>
<a href="?action=attendance_register" class="<?=$current_page==='attendance_register'? 'active':'' ?>">■
Registrar Asistencia</a>
<?php endif; ?>
<?php if(Security::hasRole('docente')): ?>
<a href="?action=tutor_course_attendance" class="<?=$current_page==='tutor_course_attendance'? 'active':'' ?
">■ Asistencia de Mi Curso</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; ?>

```

```

<?php if(Security::hasRole(['estudiante','autoridad','inspector'])): ?>
<div class="ec-item">
<span class="ec-item__btn <?=$current_page,['my_justifications','pending_justifications','reviewed_
justifications'])?'active':'' ?>">
<span class="ec-item__icon">■</span> Justificaciones <span class="ec-item__caret">▼</span>
</span>
<div class="ec-dropdown">
<?php if(Security::hasRole('estudiante')): ?>
<a href="?action=my_justifications" class="<?=$current_page==='my_justifications'? 'active':'' ?>">■ Mis
Justificaciones</a>
<?php endif; ?>
<?php if(Security::hasRole(['autoridad','inspector'])): ?>
<a href="?action=pending_justifications" class="<?=$current_page==='pending_justifications'? 'active':'' ?
">■ Revisar Justificaciones</a>
<a href="?action=reviewed_justifications" class="<?=$current_page==='reviewed_justifications'? 'active':'' ?
">■ Historial Revisadas</a>
<?php endif; ?>
</div>
</div>
<?php endif; ?>

```

```

<?php if(Security::hasRole('autoridad')): ?>
<div class="ec-item">
<span class="ec-item__btn <?=$current_page,['institution','academic','users','assignments','tutor_
management','manage_representatives','schedules','backups'])?'active':'' ?>">
<span class="ec-item__icon">■</span> Administración <span class="ec-item__caret">▼</span>
</span>
<div class="ec-dropdown">
<a href="?action=institution" class="<?=$current_page==='institution'? 'active':'' ?>">■ Configuración
General</a>
<a href="?action=academic" class="<?=$current_page==='academic'? 'active':'' ?>">■ Configuración
Académica</a>
<a href="?action=users" class="<?=$current_page==='users'? 'active':'' ?>">■ Gestión de Usuarios</a>
<div class="ec-dropdown__divider"></div>
<a href="?action=manage_representatives" class="<?=$current_page==='manage_representatives'? 'active':'' ?
">■ Representantes</a>
<div class="ec-dropdown__divider"></div>
<a href="?action=schedules" class="<?=$current_page==='schedules'? 'active':'' ?>">■ Horarios de Clases</a>
<a href="?action=backups" class="<?=$current_page==='backups'? 'active':'' ?>">■ Respaldos del Sistema</a>
</div>
</div>

```

```

<div class="ec-item">
<span class="ec-item__btn <?=$current_page,['reports','stats'])?'active':'' ?>">
<span class="ec-item__icon">■</span> Reportes <span class="ec-item__caret">▼</span>
</span>
<div class="ec-dropdown">
<a href="?action=reports" class="<?=$current_page==='reports'? 'active':'' ?>">■ Generar Reportes</a>

```

[illegible]

```

<a href="?action=notifications" class="ec-icon-btn" style="position:relative;display:flex;"
title="Notificaciones">
■
<?php if ($_notifCount > 0): ?>
<span class="ec-notif-badge" id="notif-badge-mob"><? $ _notifCount > 99 ? '99+' : $_notifCount ?></span>
<?php endif; ?>
</a>

<a href="?action=profile" class="ec-icon-btn" style="display:flex;" title="Mi perfil">■</a>
</div>
</div>

<!-- Inicio (sin sub-menú) -->
<a href="?action=dashboard" class="ec-mob-direct <? $current_page==='dashboard'? 'active':'' ?>">
<span class="ec-mob-direct__icon">■</span> Inicio
</a>

<!-- ■■ Asistencia ■■ -->
<?php if(Security::hasRole(['docente','autoridad','inspector','estudiante'])): ?>
<div class="ec-acc" data-section="asistencia">
<div class="ec-acc__head <? $ _activeSection==='asistencia'? 'has-active':'' ?>">
<span class="ec-acc__head__icon">■</span>
<span class="ec-acc__head__label">Asistencia</span>
<span class="ec-acc__head__arrow">▼</span>
</div>
<div class="ec-acc__body" style="display:none;">
<?php if(Security::hasRole(['docente','autoridad'])): ?>
<a href="?action=attendance_register" class="ec-acc__link <? $current_page==='attendance_register'? 'active':'' ?>"><span class="ec-acc__link__icon">■</span> Registrar Asistencia</a>
<?php endif; ?>
<?php if(Security::hasRole('docente')): ?>
<a href="?action=tutor_course_attendance" class="ec-acc__link <? $current_page==='tutor_course_attendance'? 'active':'' ?>"><span class="ec-acc__link__icon">■</span> Asistencia de Mi Curso</a>
<?php endif; ?>
<?php if(Security::hasRole(['docente','inspector','autoridad'])): ?>
<a href="?action=attendance_view" class="ec-acc__link <? $current_page==='attendance_view'? 'active':'' ?>"><span class="ec-acc__link__icon">■</span> Ver Asistencias</a>
<a href="?action=attendance_calendar" class="ec-acc__link <? $current_page==='attendance_calendar'? 'active':'' ?>"><span class="ec-acc__link__icon">■</span> Calendario</a>
<?php endif; ?>
<?php if(Security::hasRole('estudiante')): ?>
<a href="?action=my_attendance" class="ec-acc__link <? $current_page==='my_attendance'? 'active':'' ?>"><span class="ec-acc__link__icon">■</span> Mi Asistencia</a>
<?php endif; ?>
</div>
</div>
<?php endif; ?>

<!-- ■■ Justificaciones ■■ -->
<?php if(Security::hasRole(['estudiante','autoridad','inspector'])): ?>
<div class="ec-acc" data-section="justificaciones">
<div class="ec-acc__head <? $ _activeSection==='justificaciones'? 'has-active':'' ?>">
<span class="ec-acc__head__icon">■</span>
<span class="ec-acc__head__label">Justificaciones</span>
<span class="ec-acc__head__arrow">▼</span>
</div>
<div class="ec-acc__body" style="display:none;">
<?php if(Security::hasRole('estudiante')): ?>
<a href="?action=my_justifications" class="ec-acc__link <? $current_page==='my_justifications'? 'active':'' ?>"><span class="ec-acc__link__icon">■</span> Mis Justificaciones</a>
<?php endif; ?>
<?php if(Security::hasRole(['autoridad','inspector'])): ?>
<a href="?action=pending_justifications" class="ec-acc__link <? $current_page==='pending_justifications'? 'active':'' ?>"><span class="ec-acc__link__icon">■</span> Revisar Justificaciones</a>
<a href="?action=reviewed_justifications" class="ec-acc__link <? $current_page==='reviewed_justifications'? 'active':'' ?>"><span class="ec-acc__link__icon">■</span> Historial Revisadas</a>

```

```

<?php endif; ?>
</div>
</div>
<?php endif; ?>

<!-- ■■ Administración ■■ -->
<?php if(Security::hasRole('autoridad')): ?>
<div class="ec-acc" data-section="administracion">
<div class="ec-acc__head <?= $_activeSection==='administracion'? 'has-active':' ' ?>">
<span class="ec-acc__head__icon">■■</span>
<span class="ec-acc__head__label">Administración</span>
<span class="ec-acc__head__arrow">▼</span>
</div>
<div class="ec-acc__body" style="display:none;">
<a href="?action=institution" class="ec-acc__link <?= $current_page==='institution'? 'active':' ' ?>"><span
class="ec-acc__link__icon">■</span> Configuración General</a>
<a href="?action=academic" class="ec-acc__link <?= $current_page==='academic'? 'active':' ' ?>"><span
class="ec-acc__link__icon">■</span> Configuración Académica</a>
<a href="?action=users" class="ec-acc__link <?= $current_page==='users'? 'active':' ' ?>"><span
class="ec-acc__link__icon">■</span> Gestión de Usuarios</a>
<a href="?action=manage_representatives" class="ec-acc__link <?=
$current_page==='manage_representatives'? 'active':' ' ?>"><span class="ec-acc__link__icon">■■■■■■■■</span>
Representantes</a>
<a href="?action=schedules" class="ec-acc__link <?= $current_page==='schedules'? 'active':' ' ?>"><span
class="ec-acc__link__icon">■</span> Horarios de Clases</a>
<a href="?action=backups" class="ec-acc__link <?= $current_page==='backups'? 'active':' ' ?>"><span
class="ec-acc__link__icon">■</span> RespalDOS del Sistema</a>
</div>
</div>

<!-- ■■ Reportes ■■ -->
<div class="ec-acc" data-section="reportes">
<div class="ec-acc__head <?= $_activeSection==='reportes'? 'has-active':' ' ?>">
<span class="ec-acc__head__icon">■</span>
<span class="ec-acc__head__label">Reportes</span>
<span class="ec-acc__head__arrow">▼</span>
</div>
<div class="ec-acc__body" style="display:none;">
<a href="?action=reports" class="ec-acc__link <?= $current_page==='reports'? 'active':' ' ?>"><span
class="ec-acc__link__icon">■</span> Generar Reportes</a>
<a href="?action=stats" class="ec-acc__link <?= $current_page==='stats'? 'active':' ' ?>"><span
class="ec-acc__link__icon">■</span> Estadísticas</a>
</div>
</div>
<?php endif; ?>

<!-- Mis Representados (sin sub-menú) -->
<?php if(Security::hasRole('representante')): ?>
<a href="?action=my_children" class="ec-mob-direct <?= $current_page==='my_children'? 'active':' ' ?>">
<span class="ec-mob-direct__icon">■■■■■■■■</span> Mis Representados
</a>
<?php endif; ?>

<!-- Logout -->
<a href="?action=logout" class="ec-mob-logout">
<svg width="16" height="16" viewBox="0 0 24 24" fill="none" stroke="currentColor" stroke-width="2.2"
stroke-linecap="round" stroke-linejoin="round"><path d="M9 21H5a2 2 0 0 1-2-2V5a2 2 0 0 1 2-2h4"/><polyline
points="16 17 21 12 16 7"/><line x1="21" y1="12" x2="9" y2="12"/></svg>
Cerrar Sesión
</a>

</div><!-- /#ec-mobile-drawer -->

<script>
(function () {
/* ■■ Polling notificaciones ■■ */
var nb = document.getElementById('notif-badge');
var nbm = document.getElementById('notif-badge-mob');
function updateBadges(n) {
[nb, nbm].forEach(function(b) {
if (!b) return;

```

```

if (n > 0) { b.textContent = n > 99 ? '99+' : n; b.style.display = 'flex'; }
else b.style.display = 'none';
});
}
setInterval(function() {
fetch('?action=notifications_unread_json', { credentials: 'same-origin' })
.then(function(r){ return r.json(); })
.then(function(d){ if (typeof d.count !== 'undefined') updateBadges(d.count); })
.catch(function(){});
}, 30000);

```

```

/* ■■ Drawer ■■ */
var btn = document.getElementById('ec-hamburger');
var drawer = document.getElementById('ec-mobile-drawer');
if (!btn || !drawer) return;

function openDrawer() {
drawer.style.display = 'block';
btn.classList.add('open');
btn.setAttribute('aria-expanded', 'true');
document.body.style.overflow = 'hidden';
}
function closeDrawer() {
drawer.style.display = 'none';
btn.classList.remove('open');
btn.setAttribute('aria-expanded', 'false');
document.body.style.overflow = '';
}

btn.addEventListener('click', function() {
drawer.style.display === 'none' || !drawer.style.display ? openDrawer() : closeDrawer();
});

/* Cerrar al navegar */
drawer.querySelectorAll('a').forEach(function(a) {
a.addEventListener('click', closeDrawer);
});

/* Escape */
document.addEventListener('keydown', function(e) {
if (e.key === 'Escape') closeDrawer();
});

/* Cerrar si se pasa a desktop */
window.addEventListener('resize', function() {
if (window.innerWidth > 992) closeDrawer();
});

/* ■■ Acordeón ■■ */

```

```

var accs = drawer.querySelectorAll('.ec-acc');
accs.forEach(function(acc) {
var head = acc.querySelector('.ec-acc__head');
var body = acc.querySelector('.ec-acc__body');
if (!head || !body) return;

/* Auto-expandir si contiene el ítem activo */
if (head.classList.contains('has-active')) {
body.style.display = 'block';
acc.classList.add('is-open');
}

head.addEventListener('click', function() {
var isOpen = acc.classList.contains('is-open');

/* Cerrar todos los demás (comportamiento acordeón) */
accs.forEach(function(other) {
if (other !== acc) {
other.classList.remove('is-open');
var ob = other.querySelector('.ec-acc__body');
if (ob) ob.style.display = 'none';
}
}
);
}
});

```

```

}
});

/* Toggle el actual */
if (isOpen) {
  acc.classList.remove('is-open');
  body.style.display = 'none';
} else {
  acc.classList.add('is-open');
  body.style.display = 'block';
}
});
});
})();
</script>

```

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>
</head>
<body>

<?php include BASE_PATH . '/views/partials/navbar.php'; ?>

<div class="breadcrumb">
  <a href="?action=dashboard">■ Inicio</a> &rsquo; Respaldos
</div>

<div class="container">

  <!-- Mensajes flash -->
  <?php if(isset($_GET['backup_created'])): ?>
  <div class="alert alert-success">✓ Respaldo creado correctamente</div>
  <?php endif; ?>
  <?php if(isset($_GET['cleanup'])): ?>
  <div class="alert alert-info">■■ <?= (int)$_GET['cleanup'] ?> respaldo(s) antiguo(s) eliminado(s)</div>
  <?php endif; ?>
  <?php if(isset($_GET['deleted'])): ?>
  <div class="alert alert-success">✓ Respaldo eliminado correctamente</div>
  <?php endif; ?>

  <!-- Header -->
  <div class="page-header dark">
  <div class="ph-icon">■</div>
  <div>
  <h1>Gestión de Respaldos</h1>
  <p>Crea y descarga copias de seguridad de la base de datos</p>
  </div>
  <div class="ph-actions">
  <button class="btn btn-success" onclick="openModal('modalCreate')">+ Crear Respaldo</button>
  <?php if(count($backups) > 0): ?>
  <button class="btn btn-warning" onclick="openModal('modalCleanup')">■■ Limpiar Antiguos</button>

<?php endif; ?>
</div>
</div>

  <!-- Info -->
  <div class="alert alert-info">
  <strong>■■ Información:</strong>
  Los respaldos se crean en formato SQL · Se recomienda descargar semanalmente · Los respaldos >30 días
  pueden limpiarse para liberar espacio
  </div>

```

```

<!-- Tabla de respaldos -->
<div class="table-wrap">
<div class="table-info">
<span>■ <strong><?= count($backups) ?></strong> respaldos disponibles</span>
</div>

<?php if(empty($backups)): ?>
<div class="empty-state">
<div class="icon">■</div>
<p>No hay respaldos disponibles. Crea uno usando el botón superior.</p>
</div>
<?php else: ?>
<table>
<thead>
<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 style="color:#999;"><?= $counter++ ?></td>
<td><strong><?= htmlspecialchars($backup['filename']) ?></strong></td>
<td><?= htmlspecialchars($backup['date']) ?></td>
<td><span class="badge badge-gray"><?= htmlspecialchars($backup['size']) ?></span></td>
<td style="white-space:nowrap;">
<a href="?action=download_backup&file=<?= urlencode($backup['filename']) ?>" class="btn btn-success btn-sm">■
■ Descargar</a>
<button class="btn btn-danger btn-sm" onclick="confirmDelete('<?= addslashes(htmlspecialchars($backup['filename'])) ?>')">■■ Eliminar</button>
</td>
</tr>
<?php endforeach; ?>
</tbody>
</table>
<?php endif; ?>
</div>

<!-- Modal Crear Respaldo -->
<div class="modal-overlay" id="modalCreate">
<div class="modal-box">
<h3 style="color:#007bff;">■ Crear Respaldo</h3>
<p>¿Crear un nuevo respaldo de la base de datos?<br>
<span class="form-hint">Este proceso puede tardar unos segundos dependiendo del tamaño de la base de
datos.</span></p>
<div class="modal-actions">
<button class="btn btn-outline" onclick="closeModal('modalCreate')">Cancelar</button>
<button class="btn btn-primary" onclick="window.location.href='?action=create_backup'">Sí, Crear</button>
</div>
</div>
</div>

<!-- Modal Limpiar Antiguos -->
<div class="modal-overlay" id="modalCleanup">
<div class="modal-box">
<h3 style="color:#e6ac00;">■■ Limpiar Respaldos Antiguos</h3>
<p>¿Eliminar todos los respaldos con más de <strong>30 días</strong> de antigüedad?<br>
<span class="form-hint">■■ Esta acción no se puede deshacer. Descarga los importantes antes de
limpiar.</span></p>
<div class="modal-actions">
<button class="btn btn-outline" onclick="closeModal('modalCleanup')">Cancelar</button>
<button class="btn btn-warning" onclick="window.location.href='?action=cleanup_backups'">Sí,
Limpiar</button>
</div>
</div>

```

```

</div>
</div>

<!-- Modal Eliminar Respaldo -->
<div class="modal-overlay" id="modalDelete">

<div class="modal-box">
<h3 style="color:#dc3545;">■ ■ Eliminar Respaldo</h3>
<p>¿Está seguro de eliminar el respaldo <strong id="deleteFileName"></strong>?<br>
<span class="form-hint">Esta acción no se puede deshacer.</span></p>
<div class="modal-actions">
<button class="btn btn-outline" onclick="closeModal('modalDelete')">Cancelar</button>
<button class="btn btn-danger" id="deleteConfirmBtn">Sí, Eliminar</button>
</div>
</div>
</div>

<script>
function openModal(id) { document.getElementById(id).classList.add('on'); }
function closeModal(id) { document.getElementById(id).classList.remove('on'); }

function confirmDelete(filename) {
document.getElementById('deleteFileName').textContent = filename;
document.getElementById('deleteConfirmBtn').onclick = function() {
window.location.href = '?action=delete_backup&file=' + encodeURIComponent(filename);
};
openModal('modalDelete');
}

document.querySelectorAll('.modal-overlay').forEach(function(m) {
m.addEventListener('click', function(e) { if(e.target === m) closeModal(m.id); });
});
</script>

</body>
</html>

```

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>

```

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 - EcuAsistencia</title>
    <style>
        * { box-sizing: border-box; margin: 0; padding: 0; }
        body { font-family: -apple-system, BlinkMacSystemFont, 'Segoe UI', sans-serif;
            background: linear-gradient(135deg, #667eea 0%, #764ba2 100%);
            min-height: 100vh; display: flex; align-items: center; justify-content: center; }
        .login-card { background: #fff; border-radius: 14px; box-shadow: 0 12px 40px rgba(0,0,0,0.2);
            max-width: 420px; width: 100%; padding: 40px; }
        .login-logo { text-align: center; margin-bottom: 28px; }
        .login-logo .icon { font-size: 3.5rem; margin-bottom: 12px; }
        .login-logo h2 { font-size: 1.5rem; font-weight: 700; color: #333; }
        .login-logo p { color: #888; font-size: 0.88rem; margin-top: 4px; }
        .form-group { margin-bottom: 18px; }
        .form-group label { display: block; margin-bottom: 6px; font-size: 0.85rem; font-weight: 600; color: #444; }
        .form-group input { width: 100%; padding: 10px 14px; border: 1px solid #ddd; border-radius: 8px;
            font-size: 0.9rem; transition: border-color 0.2s; }
        .form-group input:focus { border-color: #667eea; outline: none; box-shadow: 0 0 0 3px
            rgba(102,126,234,0.15); }
        .btn-login { width: 100%; padding: 12px; background: linear-gradient(135deg, #667eea, #764ba2);
            color: #fff; border: none; border-radius: 8px; font-size: 0.95rem; font-weight: 600;
            cursor: pointer; transition: opacity 0.2s; margin-top: 4px; }
        .btn-login:hover { opacity: 0.9; }
        .alert { padding: 10px 14px; border-radius: 8px; margin-bottom: 18px; font-size: 0.85rem; }
        .alert-danger { background: #f8d7da; color: #721c24; border: 1px solid #f5c6cb; }
        .alert-success { background: #d4edda; color: #155724; border: 1px solid #c3e6cb; }
        .login-footer { text-align: center; margin-top: 20px; }
        .login-footer a { color: #667eea; text-decoration: none; font-size: 0.85rem; }
        .login-footer a:hover { text-decoration: underline; }
        .divider { border: none; border-top: 1px solid #f0f0f0; margin: 20px 0; }
        .demo-hint { background: #f8f9fa; border-radius: 8px; padding: 10px 14px; font-size: 0.78rem; color: #888;
            text-align: center; }
        .demo-hint strong { color: #555; }
    </style>
</head>
<body>
    <div class="login-card">
        <div class="login-logo">
            <div class="icon">■</div>
            <h2>EcuAsistencia</h2>
            <p>Sistema de Asistencia Escolar</p>
        </div>

        <?php if(isset($error)): ?>
        <div class="alert alert-danger">■■ <?= htmlspecialchars($error) ?></div>
        <?php endif; ?>

        <?php if(isset($_GET['registered'])): ?>

```

```

<div class="alert alert-success">✓ Registro exitoso. Ahora puede iniciar sesión.</div>
<?php endif; ?>

<?php if(isset($_GET['reset'])): ?>
<div class="alert alert-success">✓ Contraseña restablecida correctamente.</div>
<?php endif; ?>

<form method="POST">
<?php Security::generateToken(); ?>
<input type="hidden" name="csrf_token" value="<?= $_SESSION['csrf_token'] ?>">

<div class="form-group">
<label>■ Usuario o Correo Electrónico</label>
<input type="text" name="email" required autofocus placeholder="Ingrese usuario o email">
</div>

<div class="form-group">
<label>■ Contraseña</label>
<input type="password" name="password" required placeholder="Ingrese contraseña">
</div>

<button type="submit" class="btn-login">■ Iniciar Sesión</button>
</form>

```

```

<div class="login-footer" style="margin-top:16px;">
<a href="?action=forgot">¿Olvidó su contraseña?</a>
</div>

<hr class="divider">

<div class="demo-hint">
■■ Usuario de prueba: <strong>admin</strong> / <strong>password</strong>
</div>
</div>
</body>
</html>

```

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>

```

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>

```

views/profile/change_password.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>Cambiar Contraseña - EcuAsist</title>
</head>
<body>

<?php include BASE_PATH . '/views/partials/navbar.php'; ?>

<div class="breadcrumb">
  <a href="?action=dashboard">■ Inicio</a> &rsaquo;
  <a href="?action=profile">Mi Perfil</a> &rsaquo;
  Cambiar Contraseña
</div>

<div class="container" style="max-width:500px;">

  <!-- Header -->
  <div class="page-header dark">

```

```

<div class="ph-icon">■</div>
<div>
<h1>Cambiar Contraseña</h1>
<p>Actualiza tu contraseña de acceso</p>
</div>
</div>

<?php if(isset($error)): ?>
<div class="alert alert-danger"><?= htmlspecialchars($error) ?></div>
<?php endif; ?>

<div class="panel">
<form method="POST">
<div class="form-group">
<label>Contraseña Actual *</label>
<input type="password" name="current_password" class="form-control" required
placeholder="Ingresa tu contraseña actual">
</div>

<div class="form-group">
<label>Nueva Contraseña *</label>
<input type="password" name="new_password" class="form-control" minlength="6" required
placeholder="Mínimo 6 caracteres">
</div>

<div class="form-group">
<label>Confirmar Nueva Contraseña *</label>
<input type="password" name="confirm_password" class="form-control" minlength="6" required
placeholder="Repite la nueva contraseña">
</div>

<div style="display:flex;gap:10px;margin-top:24px;">
<button type="submit" class="btn btn-primary">■ Cambiar Contraseña</button>
<a href="?action=profile" class="btn btn-outline">Cancelar</a>
</div>
</form>
</div>

</div>
</body>
</html>

```

views/profile/edit.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>Editar Perfil - EcuAsist</title>
</head>
<body>

<?php include BASE_PATH . '/views/partials/navbar.php'; ?>

<div class="breadcrumb">
<a href="?action=dashboard">■ Inicio</a> &rsquo;
<a href="?action=profile">Mi Perfil</a> &rsquo;
Editar
</div>

<div class="container" style="max-width:600px;">

<!-- Header -->
<div class="page-header blue">
<div class="ph-icon">■</div>
<div>

```

```

<h1>Editar Perfil</h1>
<p>Actualiza tu información personal</p>
</div>
</div>

<?php if(isset($error)): ?>
<div class="alert alert-danger"><?= htmlspecialchars($error) ?></div>
<?php endif; ?>

<div class="panel">
<form method="POST">
<div class="form-row">
<div class="form-group">
<label>Nombres *</label>
<input type="text" name="first_name" class="form-control"
value="<?= htmlspecialchars($user['first_name']) ?>" required>
</div>
<div class="form-group">
<label>Apellidos *</label>
<input type="text" name="last_name" class="form-control"
value="<?= htmlspecialchars($user['last_name']) ?>" required>
</div>
</div>

<div class="form-group">
<label>Email *</label>
<input type="email" name="email" class="form-control"
value="<?= htmlspecialchars($user['email']) ?>" required>
</div>

<div class="form-group">
<label>Teléfono</label>
<input type="text" name="phone" class="form-control"
value="<?= htmlspecialchars($user['phone']) ?? '' ?>" placeholder="Ej: 0987654321">
</div>

<div style="display:flex;gap:10px;margin-top:24px;">
<button type="submit" class="btn btn-success">■ Guardar Cambios</button>
<a href="?action=profile" class="btn btn-outline">Cancelar</a>
</div>
</form>
</div>

</div>
</body>
</html>

```

views/profile/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>Mi Perfil - EcuAsist</title>
<style>
.avatar { width: 80px; height: 80px; border-radius: 50%; background: linear-gradient(135deg, #0277bd, #0288d1); display: flex; align-items: center; justify-content: center; color: #fff; font-size: 2rem; font-weight: 700; flex-shrink: 0; }
.profile-header { display: flex; align-items: center; gap: 20px; margin-bottom: 24px; }
.info-row { display: flex; padding: 14px 0; border-bottom: 1px solid #f0f0f0; font-size: 0.9rem; }
.info-row:last-child { border-bottom: none; }
.info-label { font-weight: 600; color: #666; width: 150px; flex-shrink: 0; }
.info-value { color: #333; flex: 1; }
</style>
</head>
<body>

```

```

<?php include BASE_PATH . '/views/partials/navbar.php'; ?>

<div class="breadcrumb">
  <a href="?action=dashboard">■ Inicio</a> &rsquo;
  Mi Perfil
</div>

<div class="container" style="max-width:800px;">

  <?php if(isset($_GET['success'])): ?>
  <div class="alert alert-success">✓ Perfil actualizado correctamente</div>
  <?php endif; ?>
  <?php if(isset($_GET['password_changed'])): ?>
  <div class="alert alert-success">✓ Contraseña cambiada correctamente</div>
  <?php endif; ?>

  <!-- Header -->
  <div class="page-header" style="background:linear-gradient(135deg,#0277bd,#0288d1);">
  <div class="ph-icon">■</div>
  <div>
  <h1>Mi Perfil</h1>
  <p>Información de tu cuenta</p>

</div>
<div class="ph-actions">
  <a href="?action=edit_profile" class="btn" style="background:rgba(255,255,255,0.2);color:#fff;border:1px
solid rgba(255,255,255,0.6);">■ Editar</a>
  <a href="?action=change_password" class="btn" style="background:rgba(255,255,255,0.2);color:#fff;border:1px
solid rgba(255,255,255,0.6);margin-left:8px;">■ Contraseña</a>
</div>
</div>

<div class="panel">
<div class="profile-header">
<div class="avatar">
  <?= strtoupper(substr($user['first_name'], 0, 1) . substr($user['last_name'], 0, 1)) ?>
</div>
<div>
  <h2 style="font-size:1.3rem;font-weight:700;"><?= htmlspecialchars($user['first_name']) . ' ' .
$user['last_name'] ?></h2>
  <p style="color:#888;margin-top:2px;">@<?= htmlspecialchars($user['username']) ?></p>
</div>
</div>

<div class="info-row">
<span class="info-label">Email</span>
<span class="info-value"><?= htmlspecialchars($user['email']) ?></span>
</div>
<div class="info-row">
<span class="info-label">Cédula</span>
<span class="info-value"><?= htmlspecialchars($user['dni']) ?? 'No registrada' ?></span>
</div>
<div class="info-row">
<span class="info-label">Teléfono</span>
<span class="info-value"><?= htmlspecialchars($user['phone']) ?? 'No registrado' ?></span>
</div>
<div class="info-row">
<span class="info-label">Roles</span>
<span class="info-value">
  <?php foreach($roles as $role): ?>
  <span class="badge badge-green" style="margin-right:4px;"><?= ucfirst(htmlspecialchars($role)) ?></span>
  <?php endforeach; ?>
</span>
</div>
<div class="info-row">
<span class="info-label">Miembro desde</span>
<span class="info-value" style="color:#888;"><?= date('d/m/Y', strtotime($user['created_at'])) ?></span>
</div>
</div>

```

```
</div>
</body>
</html>
```

views/assignments/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>Asignaciones Docentes - 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; }
    label { display: block; margin-bottom: 5px; font-weight: bold; color: #333; font-size: 13px; }
    select { width: 100%; padding: 9px 10px; border: 1px solid #ddd; border-radius: 4px; font-size: 14px; }
    table { width: 100%; border-collapse: collapse; margin-top: 15px; }
    th, td { padding: 11px 13px; text-align: left; border-bottom: 1px solid #eee; font-size: 14px; }
    th { background: #f8f9fa; font-weight: bold; color: #555; }
    tr:hover td { background: #f9f9f9; }
    .btn-danger { background: #dc3545; color: white; padding: 5px 12px; font-size: 12px; border: none;
border-radius: 4px; cursor: pointer; }
    .btn-danger:hover { background: #c82333; }
    .btn-clear { padding: 9px 16px; background: #6c757d; color: white; text-decoration: none; border-radius:
4px; font-size: 13px; display: inline-block; }
    .info-banner { background: #e8f4fd; border-left: 4px solid #007bff; padding: 14px 18px; border-radius: 4px;
margin-bottom: 20px; font-size: 14px; color: #0056b3; }
    .alert-success { background: #d4edda; color: #155724; padding: 12px 16px; border-radius: 4px;
margin-bottom: 16px; font-size: 14px; }
    .alert-error { background: #f8d7da; color: #721c24; padding: 12px 16px; border-radius: 4px; margin-bottom:
16px; font-size: 14px; }
    .empty-msg { text-align: center; color: #999; padding: 30px; }
    .filter-grid { display: grid; grid-template-columns: repeat(3,1fr) auto; gap: 14px; align-items: flex-end; }
    @media(max-width:700px) { .filter-grid { grid-template-columns: 1fr 1fr; } }
  </style>
</head>
<body>

<?php include BASE_PATH . '/views/partials/navbar.php'; ?>

<div class="breadcrumb">
  <a href="?action=dashboard">■ Inicio</a> &rsquo; Asignaciones Docentes
</div>

<div class="container">

  <div class="page-header" style="background:linear-gradient(135deg,#e65100,#f57c00);">

<div class="ph-icon">■</div>
  <div>
    <h1>Asignaciones Docente – Materia</h1>
    <p>Consulta global de todas las asignaciones del año lectivo activo</p>
  </div>
  <div class="ph-actions">
    <a href="?action=academic" style="padding:8px 16px;background:rgba(255,255,255,0.2);color:white;text-decorat
ion:none;border-radius:4px;font-size:13px;">
      ■ Ir a Cursos
    </a>
    <a href="?action=tutor_management" style="padding:8px 16px;background:rgba(255,255,255,0.2);color:white;text
-decoration:none;border-radius:4px;font-size:13px;">
      ■■■ Tutores
    </a>
  </div>
</div>
</div>
```

```

<?php if(isset($_GET['removed'])): ?>
<div class="alert-success">✓ Asignación eliminada correctamente.</div>
<?php endif; ?>
<?php if(isset($_GET['error'])): ?>
<div class="alert-error">✗ <?= htmlspecialchars($_GET['error']) ?></div>
<?php endif; ?>

<div class="info-banner">
■ Para <strong>asignar</strong> un docente a una materia, ve a
<a href="?action=academic" style="color:#0056b3;font-weight:bold;">Configuración Académica</a>
→ selecciona un curso → botón <strong>■ Asignaturas</strong> → botón <strong>■ Asignar</strong>.
</div>

<div class="card">
<div style="display:flex;justify-content:space-between;align-items:center;margin-bottom:18px;">
<h2 style="color:#333;font-size:18px;">■ Todas las asignaciones</h2>
<span style="font-size:13px;color:#888;"><?= count($assignments) ?> registradas</span>
</div>

<form method="GET" action="">
<input type="hidden" name="action" value="assignments">
<div class="filter-grid">
<div>
<label>Filtrar por Curso</label>
<select name="filter_course" onchange="this.form.submit()">
<option value="">Todos los cursos</option>
<?php foreach($courses as $c): ?>
<option value="<?= $c['id'] ?>" <?= (!empty($_GET['filter_course']) && $_GET['filter_course'] == $c['id']) ?
'selected' : '' ?>>
<?= htmlspecialchars($c['name']) ?>
</option>
<?php endforeach; ?>
</select>
</div>
<div>
<label>Filtrar por Asignatura</label>
<select name="filter_subject" onchange="this.form.submit()">
<option value="">Todas las asignaturas</option>
<?php foreach($subjects as $s): ?>
<option value="<?= $s['id'] ?>" <?= (!empty($_GET['filter_subject']) && $_GET['filter_subject'] ==
$s['id']) ? 'selected' : '' ?>>
<?= htmlspecialchars($s['name']) ?>
</option>
<?php endforeach; ?>
</select>
</div>
<div>
<label>Filtrar por Docente</label>
<select name="filter_teacher" onchange="this.form.submit()">
<option value="">Todos los docentes</option>
<?php foreach($teachers as $t): ?>
<option value="<?= $t['id'] ?>" <?= (!empty($_GET['filter_teacher']) && $_GET['filter_teacher'] ==
$t['id']) ? 'selected' : '' ?>>
<?= htmlspecialchars($t['last_name'].' '.$t['first_name']) ?>
</option>
<?php endforeach; ?>
</select>
</div>
<?php if(!empty($_GET['filter_course']) || !empty($_GET['filter_subject']) || !
empty($_GET['filter_teacher'])): ?>
<div style="padding-top:18px;">
<a href="?action=assignments" class="btn-clear">✗ Limpiar</a>
</div>
<?php endif; ?>
</div>
</form>

<table>
<thead>

```



```

<tr>
<th>Curso</th>
<th>Asignatura</th>
<th>Docente</th>
<th style="width:110px;text-align:center;">Acción</th>
</tr>
</thead>
<tbody>
<?php
$list = $assignments;
if (!empty($_GET['filter_course'])) {
$id = (int)$_GET['filter_course']; $nm = '';
foreach($courses as $c){ if($c['id']==$id){ $nm=$c['name']; break; } }
$list = array_filter($list, fn($a) => $a['course_name']==$nm);
}
if (!empty($_GET['filter_subject'])) {
$id = (int)$_GET['filter_subject']; $nm = '';
foreach($subjects as $s){ if($s['id']==$id){ $nm=$s['name']; break; } }
$list = array_filter($list, fn($a) => $a['subject_name']==$nm);
}
if (!empty($_GET['filter_teacher'])) {
$id = (int)$_GET['filter_teacher']; $nm = '';
foreach($teachers as $t){ if($t['id']==$id){ $nm=$t['last_name'].' '.$t['first_name']; break; } }
$list = array_filter($list, fn($a) => $a['teacher_name']==$nm);
}
?>
<?php if(empty($list)): ?>
<tr><td colspan="4" class="empty-msg">No hay asignaciones que coincidan con los filtros.</td></tr>
<?php else: ?>
<?php foreach($list as $a): ?>
<tr>
<td><?= htmlspecialchars($a['course_name']) ?></td>
<td><?= htmlspecialchars($a['subject_name']) ?></td>
<td>■ <?= htmlspecialchars($a['teacher_name']) ?></td>
<td style="text-align:center;">
<form method="POST" action="?action=remove_assignment" style="display:inline;">
<input type="hidden" name="assignment_id" value="<?= $a['id'] ?>">
<button type="submit" class="btn-danger"
onclick="return confirmDelete(event, '<?= htmlspecialchars(addslashes($a['teacher_name'])) ?>', '<?=
htmlspecialchars(addslashes($a['course_name'])) ?>', '<?= htmlspecialchars(addslashes($a['subject_name'])) ?>')">
■ ■ Quitar
</button>
</form>
</td>
</tr>
<?php endforeach; ?>
<?php endif; ?>
</tbody>
</table>
</div>
</div>

<script>
function confirmDelete(event, teacher, course, subject) {
event.preventDefault();
const m = document.createElement('div');
m.style.cssText = 'position:fixed;top:0;left:0;width:100%;height:100%;background:rgba(0,0,0,.5);display:flex;
align-items:center;justify-content:center;z-index:9999;';
m.innerHTML = `<div style="background:white;padding:28px;border-radius:8px;max-width:460px;width:90%;box-sha
dow:0 4px 20px rgba(0,0,0,.3);">
<h3 style="color:#dc3545;margin-bottom:14px;">■ ■ Quitar asignación</h3>
<p style="color:#555;margin-bottom:6px;"><strong>Docente:</strong> ${teacher}</p>
<p style="color:#555;margin-bottom:6px;"><strong>Curso:</strong> ${course}</p>
<p style="color:#555;margin-bottom:18px;"><strong>Asignatura:</strong> ${subject}</p>
<div style="display:flex;gap:10px;justify-content:flex-end;">
<button id="bc" style="padding:8px 18px;background:#6c757d;color:white;border:none;border-radius:4px;cursor:
pointer;">Cancelar</button>
<button id="bo" style="padding:8px 18px;background:#dc3545;color:white;border:none;border-radius:4px;cursor:
pointer;">Sí, Quitar</button>
</div></div>`;

```

```

document.body.appendChild(m);
const form = event.target.closest('form');
document.getElementById('bo').onclick = () => { document.body.removeChild(m); form.submit(); };
document.getElementById('bc').onclick = () => document.body.removeChild(m);
return false;
}
</script>
</body>
</html>

```

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="breadcrumb">
    <a href="?action=dashboard">■ Inicio</a> &rsquo; Asignar Docente Tutor
  </div>

  <div class="container">

    <div class="page-header" style="background:linear-gradient(135deg,#6a1b9a,#8e24aa);">
      <div class="ph-icon">■■■■</div>
      <div>
        <h1>Asignar Docente Tutor</h1>
        <p>Designa el tutor responsable de cada curso</p>
      </div>

    </div>

    <?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">
      <div class="header-actions" style="display:flex;justify-content:space-between;align-items:center;margin-bott

```

```

om:20px;">
<h2 style="margin:0;">■■■ Asignar Docente Tutor</h2>
<a href="?action=assignments" style="padding:8px 16px;background:#28a745;color:white;text-decoration:none;border-radius:4px;font-size:13px;">
  ■ Ir a Asignar Docente-Materia
</a>
</div>
<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>

```

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>

```

views/reports/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>Reportes - EcuAsist</title>
</head>
<body>

<?php include BASE_PATH . '/views/partials/navbar.php'; ?>

<div class="breadcrumb">
<a href="?action=dashboard">■ Inicio</a> &rsaquo; Reportes
</div>

<div class="container">

<!-- Header -->
<div class="page-header purple">
<div class="ph-icon">■</div>
<div>
<h1>Generación de Reportes</h1>
<p>Exporta informes de asistencia en PDF o Excel</p>
</div>
</div>

<!-- Configuración -->
<div class="panel">
<h3 class="blue">■■ Configuración del Reporte</h3>
<div class="alert alert-info" style="margin-bottom: 16px;">

```

```

<strong>Tipos disponibles:</strong> Vista Previa (pantalla) · PDF (formato institucional) · Excel (análisis
de datos)
</div>
<form method="POST" id="report-form">
<div class="form-row">
<div class="form-group">
<label>Curso</label>
<select name="course_id" class="form-control" required>
<option value="">- Seleccionar curso -</option>
<?php foreach($courses as $c): ?>
<option value="<?= $c['id'] ?>" <?= isset($course) && $course['id'] == $c['id'] ? 'selected' : '' ?>>
<?= htmlspecialchars($c['name']) ?> - <?= htmlspecialchars($c['shift_name']) ?>
</option>
<?php endforeach; ?>
</select>
</div>
<div class="form-group">
<label>Fecha Inicio</label>
<input type="date" name="start_date" class="form-control" value="<?= $startDate ?? date('Y-m-01') ?>"
required>
</div>
<div class="form-group">
<label>Fecha Fin</label>
<input type="date" name="end_date" class="form-control" value="<?= $endDate ?? date('Y-m-d') ?>" required>
</div>
</div>
<div style="display:flex;gap:10px;margin-top:8px;flex-wrap:wrap;">
<button type="submit" name="preview" class="btn btn-primary">■ Vista Previa</button>
<button type="button" class="btn btn-danger" onclick="generateReport('pdf')" <?= empty($data) ? 'disabled
title="Genera una Vista Previa primero" : '' ?>>
■ Generar PDF
</button>
<button type="button" class="btn btn-success" onclick="generateReport('excel')" <?= empty($data) ? '
disabled title="Genera una Vista Previa primero" : '' ?>>
■ Generar Excel
</button>
</div>
</form>
</div>

<!-- Vista previa -->
<?php if(!empty($data)): ?>
<?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'));
$total = count($data);
$porcentaje = $total > 0 ? round(($totalPresente / $total) * 100, 1) : 0;
?>

<div class="panel">
<h3 class="blue">■ Vista Previa del Reporte</h3>

<div class="alert alert-info" style="margin-bottom:14px;">
<strong>Curso:</strong> <?= htmlspecialchars($course['name']) ?> - <?=
htmlspecialchars($course['shift_name']) ?> &nbsp;|&nbsp;
<strong>Período:</strong> <?= date('d/m/Y', strtotime($startDate)) ?> al <?= date('d/m/Y',
strtotime($endDate)) ?> &nbsp;|&nbsp;
<strong>Total:</strong> <?= $total ?> registros
</div>

<!-- Mini stats -->
<div class="stats-row" style="margin-bottom:16px;">
<div class="stat-card green">
<div class="number"><?= $totalPresente ?></div>
<div class="label">Presentes (<?= $porcentaje ?>%)</div>
<div class="progress-bar"><div class="progress-fill" style="width:<?= $porcentaje ?
>%;background:#28a745;"></div></div>
</div>
<div class="stat-card red">

```



```

<div class="number"><?= $totalAusente ?></div>
<div class="label">Ausentes</div>
</div>
<div class="stat-card yellow">
<div class="number"><?= $totalTardanza ?></div>
<div class="label">Tardanzas</div>
</div>
<div class="stat-card teal">
<div class="number"><?= $totalJustificado ?></div>
<div class="label">Justificados</div>
</div>
</div>

```

```

<div class="table-wrap">
<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
$statusBadge = [
'presente' => 'badge-green',
'ausente' => 'badge-red',
'tardanza' => 'badge-yellow',
'justificado' => 'badge-teal',
];
foreach($data as $row):
$bc = $statusBadge[$row['status']] ?? 'badge-gray';
?>
<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><span class="badge <?= $bc ?>"><?= ucfirst($row['status']) ?></span></td>
<td style="color:#888;"><?= htmlspecialchars($row['observation']) ? : '-' ?></td>
</tr>
<?php endforeach; ?>
</tbody>
</table>
</div>
</div>

```

```

<?php elseif($_SERVER['REQUEST_METHOD'] === 'POST'): ?>
<div class="panel">
<div class="empty-state">
<div class="icon">■</div>
<p>No hay registros de asistencia para el período seleccionado</p>
</div>
</div>
<?php endif; ?>

```

```

</div>

```

```

<script>
function generateReport(type) {
var form = document.getElementById('report-form');
if (!form.checkValidity()) { form.reportValidity(); return; }
var tf = document.createElement('form');

```

```

tf.method = 'POST';
tf.style.display = 'none';
tf.action = type === 'pdf' ? '?action=generate_pdf' : '?action=generate_excel';
['course_id', 'start_date', 'end_date'].forEach(function(n) {

```

```

var i = document.createElement('input');
i.type = 'hidden'; i.name = n;
i.value = form.querySelector('[name="'+n+'"]').value;
tf.appendChild(i);
});
document.body.appendChild(tf);
tf.submit();
document.body.removeChild(tf);
}
</script>

</body>
</html>

```

views/notifications/index.php

```

<?php
// ARCHIVO COMPLETO - Reemplaza TODO el archivo
// views/notifications/index.php

$typeIcons = [
    'info' => '■',
    'success' => '■',
    'warning' => '■',
    'danger' => '■',
    'ausente' => '■',
    'justificacion' => '■',
];
?>
<!DOCTYPE html>
<html lang="es">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Notificaciones - EcuAsist</title>
    <style>
        * { box-sizing: border-box; margin: 0; padding: 0; }
        body { font-family: -apple-system, BlinkMacSystemFont, 'Segoe UI', sans-serif; background: #f4f6f9; color: #333; }

        .breadcrumb { padding: 10px 24px; background: #fff; border-bottom: 1px solid #e0e0e0; font-size: 0.83rem; color: #888; }
        .breadcrumb a { color: #007bff; text-decoration: none; }

        .container { max-width: 760px; margin: 28px auto; padding: 0 16px; }

        .page-title { font-size: 1.4rem; font-weight: 600; margin-bottom: 20px; }
        .page-title span { color: #6c757d; font-size: 0.95rem; font-weight: 400; margin-left: 8px; }

        .alert { padding: 10px 16px; border-radius: 6px; margin-bottom: 16px; font-size: 0.88rem; }
        .alert-success { background: #d4edda; color: #155724; border: 1px solid #c3e6cb; }

        .btn { padding: 7px 14px; border-radius: 6px; border: none; cursor: pointer; font-size: 0.85rem; font-weight: 500; text-decoration: none; display: inline-flex; align-items: center; gap: 5px; transition: opacity 0.15s; }
        .btn:hover { opacity: 0.85; }
        .btn-primary { background: #007bff; color: #fff; }
        .btn-outline { background: #fff; color: #555; border: 1px solid #ccc; }
        .btn-sm { padding: 4px 10px; font-size: 0.78rem; }

        .stats-row { display: flex; gap: 12px; margin-bottom: 16px; flex-wrap: wrap; }
        .stat-card { background: #fff; border: 1px solid #e0e0e0; border-radius: 8px; padding: 12px 18px; flex: 1; min-width: 100px; }
        .stat-card .number { font-size: 1.6rem; font-weight: 700; }
        .stat-card .label { font-size: 0.78rem; color: #888; margin-top: 2px; }
        .stat-card.blue .number { color: #007bff; }

        .notif-list { display: flex; flex-direction: column; gap: 8px; }

```

```
.notif-item {
background: #fff;
border: 1px solid #e0e0e0;
border-radius: 8px;
padding: 14px 16px;
display: flex;
gap: 12px;
align-items: flex-start;
position: relative;
text-decoration: none;
color: inherit;
transition: box-shadow 0.15s, border-color 0.15s;
}

.notif-item:hover { box-shadow: 0 2px 8px rgba(0,0,0,0.08); }
.notif-item.has-link { cursor: pointer; }
.notif-item.has-link:hover { border-color: #007bff; box-shadow: 0 3px 12px rgba(0,123,255,0.15); }
.notif-item.unread { border-left: 4px solid #007bff; background: #f8fbff; }

.notif-dot {
width: 8px; height: 8px; border-radius: 50%;
background: #007bff;
position: absolute; top: 14px; right: 14px;
}

.notif-icon { font-size: 1.5rem; min-width: 36px; text-align: center; line-height: 1; }
.notif-body { flex: 1; min-width: 0; }
.notif-title { font-weight: 600; font-size: 0.92rem; margin-bottom: 3px; }
.notif-message { font-size: 0.85rem; color: #555; line-height: 1.4; }
.notif-meta { display: flex; align-items: center; gap: 10px; margin-top: 6px; flex-wrap: wrap; }
.notif-time { font-size: 0.78rem; color: #999; }
.notif-link-hint { font-size: 0.78rem; color: #007bff; font-weight: 500; }
.badge-new { font-size: 0.72rem; background: #007bff; color: #fff; padding: 1px 7px; border-radius: 10px;
font-weight: 600; }
```

```
<div class="number"><?= $unreadCount ?></div>
<div class="label">No leídas</div>
</div>
<div class="stat-card">
```

```
<div class="number"><?= $total ?></div>
<div class="label">Total</div>
</div>
</div>
```

```
<div style="display:flex; gap:8px; margin-bottom:16px; flex-wrap:wrap;">
<?php if ($unreadCount > 0): ?>
<form method="POST" action="?action=notifications_mark_all" style="margin:0;">
<button type="submit" class="btn btn-primary">■ Marcar todas como leídas</button>
</form>
<?php endif; ?>
<form method="POST" action="?action=notifications_delete_read"
onsubmit="return confirm('¿Eliminar todas las notificaciones leídas?');" style="margin:0;">
<button type="submit" class="btn btn-outline">■ Limpiar leídas</button>
</form>
</div>
```

```
<div class="notif-list">
<?php if (empty($notifications)): ?>
<div class="empty-state">
<div class="icon">■</div>
<p>No tienes notificaciones.</p>
</div>
<?php else: ?>
<?php foreach ($notifications as $n): ?>
<?php
$icon = $typeIcons[$n['type']] ?? '■';
$isNew = !(bool)$n['is_read'];
$hasLink = !empty($n['link']);
$tag = $hasLink ? 'a' : 'div';
$href = $hasLink ? ' href="' . htmlspecialchars($n['link']) . '"' : '';
$css = 'notif-item' . ($isNew ? ' unread' : '') . ($hasLink ? ' has-link' : '');
$onclick = ($hasLink && $isNew) ? ' onclick="markRead(' . $n['id'] . ', this)"' : '';
?>
<<?= $tag ?><?= $href ?> class="<?= $css ?>"<?= $onclick ?>>

<?php if ($isNew): ?><span class="notif-dot"></span><?php endif; ?>

<div class="notif-icon"><?= $icon ?></div>
```

```
<div class="notif-body">
<div class="notif-title"><?= htmlspecialchars($n['title']) ?></div>
<div class="notif-message"><?= htmlspecialchars($n['message']) ?></div>
<div class="notif-meta">
<span class="notif-time">■ <?= date('d/m/Y H:i', strtotime($n['created_at'])) ?></span>
<?php if ($isNew): ?><span class="badge-new">Nueva</span><?php endif; ?>
<?php if ($hasLink): ?><span class="notif-link-hint">■ Clic para ver →</span><?php endif; ?>
</div>
</div>

<div class="notif-actions" onclick="event.stopPropagation();">
<?php if ($isNew): ?>
<a href="?action=notifications_mark_read&id=<?= $n['id'] ?>"
class="btn btn-outline btn-sm" title="Marcar como leída">✓</a>
<?php endif; ?>
<form method="POST" action="?action=notifications_delete" style="margin:0;"
onsubmit="return confirm('¿Eliminar esta notificación?');">
<input type="hidden" name="notification_id" value="<?= $n['id'] ?>">
<button type="submit" class="btn btn-outline btn-sm" title="Eliminar">■■</button>
</form>
</div>

</<?= $tag ?>>
<?php endforeach; ?>
<?php endif; ?>
</div>
```

```

<?php if ($totalPages > 1): ?>
<div class="pagination">
<?php if ($page > 1): ?>
<a href="?action=notifications&page=<?= $page - 1 ?>"><- Anterior</a>
<?php endif; ?>
<?php for ($i = 1; $i <= $totalPages; $i++): ?>
<?php if ($i == $page): ?>
<span class="current"><?= $i ?></span>
<?php else: ?>
<a href="?action=notifications&page=<?= $i ?>"><?= $i ?></a>
<?php endif; ?>
<?php endfor; ?>
<?php if ($page < $totalPages): ?>

<a href="?action=notifications&page=<?= $page + 1 ?>">Siguiete →</a>
<?php endif; ?>
</div>

<?php endif; ?>

</div>

<script>
function markRead(id, el) {
// Marcar como leída en background
fetch('?action=notifications_mark_read&id=' + id, { credentials: 'same-origin' })
.catch(function(){});

// Quitar estilos no-leída visualmente
el.classList.remove('unread');
var dot = el.querySelector('.notif-dot');
if (dot) dot.remove();
var badgeNew = el.querySelector('.badge-new');
if (badgeNew) badgeNew.remove();

// Actualizar campana del navbar
setTimeout(function(){
fetch('?action=notifications_unread_json', { credentials: 'same-origin' })
.then(function(r){ return r.json(); })
.then(function(data){
var b = document.getElementById('notif-badge');
if (b) {
if (data.count > 0) {
b.textContent = data.count > 99 ? '99+' : data.count;
b.style.display = 'flex';
} else {
b.style.display = 'none';
}
}
}).catch(function(){});
}, 200);
}
</script>

</body>

</html>

```

PUBLIC

public/.htaccess

```
<IfModule mod_rewrite.c>
RewriteEngine On
RewriteBase /

RewriteCond %{REQUEST_FILENAME} !-f
RewriteCond %{REQUEST_FILENAME} !-d
RewriteRule ^(.*)$ index.php [QSA,L]
</IfModule>
```

public/img.php

```
<?php
// public/img.php - Sirve imágenes desde uploads/ (fuera del document root)
// Depende de env.php para obtener BASE_PATH correcto en local y producción

require_once __DIR__ . '/../config/env.php';

$uploadsDir = realpath(BASE_PATH . '/uploads');

if (!$uploadsDir) {
    http_response_code(404); exit;
}

$file = $_GET['f'] ?? '';

// Seguridad: bloquear path traversal
$file = str_replace(['..', '\\', '\\0'], '', $file);
$file = ltrim($file, '/');

if (!$file) { http_response_code(400); exit; }

$fullPath = $uploadsDir . DIRECTORY_SEPARATOR . $file;
$realFull = realpath($fullPath);

// Verificar que el archivo está dentro de uploads/
if (!$realFull || strpos($realFull, $uploadsDir) !== 0) {
    http_response_code(403); exit;
}

if (!file_exists($realFull) || !is_file($realFull)) {
    http_response_code(404); exit;
}

$ext = strtolower(pathinfo($realFull, PATHINFO_EXTENSION));
$mimes = [
    'jpg' => 'image/jpeg',
    'jpeg' => 'image/jpeg',
    'png' => 'image/png',
    'gif' => 'image/gif',
    'webp' => 'image/webp',
];

if (!isset($mimes[$ext])) { http_response_code(403); exit; }

header('Content-Type: ' . $mimes[$ext]);
header('Content-Length: ' . filesize($realFull));
header('Cache-Control: public, max-age=86400');
readfile($realFull);
exit;
```

public/index.php

```
<?php

if ($_SERVER['SERVER_NAME'] === 'localhost' || $_SERVER['SERVER_NAME'] === '127.0.0.1') {
    require_once __DIR__ . '/../config/config.php'; // local
} else {
    require_once '/home/ecuasysc/ecuasistencia/config/config.php'; // producción
}

$action = $_GET['action'] ?? 'login';

switch ($action) {
    case 'register':
        require_once BASE_PATH . '/controllers/AuthController.php';
        $auth = new AuthController();
        $auth->register();
        break;

    case 'login':
        require_once BASE_PATH . '/controllers/AuthController.php';
        $auth = new AuthController();
        $auth->login();
        break;

    case 'logout':
        require_once BASE_PATH . '/controllers/AuthController.php';
        $auth = new AuthController();
        $auth->logout();
        break;

    case 'forgot':
        require_once BASE_PATH . '/controllers/AuthController.php';
        $auth = new AuthController();
        $auth->forgotPassword();
        break;

    case 'reset':
        require_once BASE_PATH . '/controllers/AuthController.php';
        $auth = new AuthController();
        $auth->resetPassword();
        break;
```

```
    case 'dashboard':
        require_once BASE_PATH . '/controllers/DashboardController.php';
        $dash = new DashboardController();
        $dash->index();
        break;

    case 'users':
        require_once BASE_PATH . '/controllers/UserController.php';
        $userCtrl = new UserController();
        $userCtrl->index();
        break;

    case 'assign_role':
        require_once BASE_PATH . '/controllers/UserController.php';
        $userCtrl = new UserController();
        $userCtrl->assignRole();
        break;

    case 'attendance_register':
        require_once BASE_PATH . '/controllers/AttendanceController.php';
        $attCtrl = new AttendanceController();
        $attCtrl->register();
        break;
```

```

case 'get_students':
require_once BASE_PATH . '/controllers/AttendanceController.php';
$attCtrl = new AttendanceController();
$attCtrl->getStudents();
break;

case 'attendance_view':
require_once BASE_PATH . '/controllers/AttendanceController.php';
$attCtrl = new AttendanceController();
$attCtrl->view();
break;

case 'my_attendance':
require_once BASE_PATH . '/controllers/AttendanceController.php';
$attCtrl = new AttendanceController();

```

```

$attCtrl->myAttendance();
break;

```

```

case 'academic':
require_once BASE_PATH . '/controllers/AcademicController.php';
$acadCtrl = new AcademicController();
$acadCtrl->index();
break;

```

```

case 'create_course':
require_once BASE_PATH . '/controllers/AcademicController.php';
$acadCtrl = new AcademicController();
$acadCtrl->createCourse();
break;

```

```

case 'create_subject':
require_once BASE_PATH . '/controllers/AcademicController.php';
$acadCtrl = new AcademicController();
$acadCtrl->createSubject();
break;

```

```

case 'set_subject_hours':
require_once BASE_PATH . '/controllers/AcademicController.php';
$controller = new AcademicController();
$controller->setSubjectHours();
break;

```

```

case 'course_subjects':
require_once BASE_PATH . '/controllers/AcademicController.php';
$acadCtrl = new AcademicController();
$acadCtrl->courseSubjects();
break;

```

```

case 'view_course_students':
require_once BASE_PATH . '/controllers/AcademicController.php';
$acadCtrl = new AcademicController();
$acadCtrl->viewCourseStudents();
break;

```

```

case 'edit_course_subject':

```

```

require_once BASE_PATH . '/controllers/AcademicController.php';
$acadCtrl = new AcademicController();
$acadCtrl->editCourseSubject();
break;

```

```

case 'remove_course_subject':
require_once BASE_PATH . '/controllers/AcademicController.php';
$acadCtrl = new AcademicController();
$acadCtrl->removeCourseSubject();
break;

```

```

case 'assign_subject_teacher':
require_once BASE_PATH . '/controllers/AcademicController.php';
$acadCtrl = new AcademicController();
$acadCtrl->assignSubjectTeacher();

```



```

break;

case 'unassign_subject_teacher':
require_once BASE_PATH . '/controllers/AcademicController.php';
$acadCtrl = new AcademicController();
$acadCtrl->unassignSubjectTeacher();
break;

case 'reports':
require_once BASE_PATH . '/controllers/ReportController.php';
$reportCtrl = new ReportController();
$reportCtrl->index();
break;

case 'generate_pdf':
require_once BASE_PATH . '/controllers/ReportController.php';
$reportCtrl = new ReportController();
$reportCtrl->generatePDF();
break;

case 'generate_excel':
require_once BASE_PATH . '/controllers/ReportController.php';
$reportCtrl = new ReportController();
$reportCtrl->generateExcel();
break;

```

```

case 'manage_representatives':
require_once BASE_PATH . '/controllers/RepresentativeController.php';
$repCtrl = new RepresentativeController();
$repCtrl->manageRepresentatives();
break;

case 'my_children':
require_once BASE_PATH . '/controllers/RepresentativeController.php';
$repCtrl = new RepresentativeController();
$repCtrl->myChildren();
break;

case 'child_attendance':
require_once BASE_PATH . '/controllers/RepresentativeController.php';
$repCtrl = new RepresentativeController();
$repCtrl->childAttendance();
break;

case 'assignments':
require_once BASE_PATH . '/controllers/AssignmentController.php';
$assignCtrl = new AssignmentController();
$assignCtrl->index();
break;

case 'create_assignment':
require_once BASE_PATH . '/controllers/AssignmentController.php';
$assignCtrl = new AssignmentController();
$assignCtrl->assign();
break;

case 'set_tutor':
require_once BASE_PATH . '/controllers/AssignmentController.php';
$assignCtrl = new AssignmentController();
$assignCtrl->setTutor();
break;

case 'remove_tutor':
require_once BASE_PATH . '/controllers/AssignmentController.php';
$controller = new AssignmentController();

```

```

$controller->removeTutor();
break;

case 'get_course_teachers':
require_once BASE_PATH . '/controllers/AssignmentController.php';

```

```

$controller = new AssignmentController();
$controller->getCourseTeachers();
break;

case 'remove_assignment':
require_once BASE_PATH . '/controllers/AssignmentController.php';
$assignCtrl = new AssignmentController();
$assignCtrl->remove();
break;

case 'view_course_assignments':
require_once BASE_PATH . '/controllers/AssignmentController.php';
$assignCtrl = new AssignmentController();
$assignCtrl->viewByCourse();
break;

case 'profile':
require_once BASE_PATH . '/controllers/ProfileController.php';
$profileCtrl = new ProfileController();
$profileCtrl->view();
break;

case 'edit_profile':
require_once BASE_PATH . '/controllers/ProfileController.php';
$profileCtrl = new ProfileController();
$profileCtrl->edit();
break;

case 'change_password':
require_once BASE_PATH . '/controllers/ProfileController.php';
$profileCtrl = new ProfileController();
$profileCtrl->changePassword();
break;

case 'search_students':

```

```

require_once BASE_PATH . '/controllers/SearchController.php';
$searchCtrl = new SearchController();
$searchCtrl->searchStudents();
break;

case 'export_student_list':
require_once BASE_PATH . '/controllers/ExportController.php';
$exportCtrl = new ExportController();
$exportCtrl->exportStudentList();
break;

case 'stats':
require_once BASE_PATH . '/controllers/StatsController.php';
$statsCtrl = new StatsController();
$statsCtrl->index();
break;

case 'attendance_calendar':
require_once BASE_PATH . '/controllers/AttendanceController.php';
$attCtrl = new AttendanceController();
$attCtrl->calendar();
break;

case 'submit_justification':
require_once BASE_PATH . '/controllers/JustificationController.php';
$justCtrl = new JustificationController();
$justCtrl->submit();
break;

case 'my_justifications':
require_once BASE_PATH . '/controllers/JustificationController.php';
$justCtrl = new JustificationController();
$justCtrl->myJustifications();
break;

case 'pending_justifications':

```

```
require_once BASE_PATH . '/controllers/JustificationController.php';
$justCtrl = new JustificationController();
$justCtrl->pending();
break;
```

```
case 'reviewed_justifications':
require_once BASE_PATH . '/controllers/JustificationController.php';
$controller = new JustificationController();
$controller->reviewed();
break;
```

```
case 'review_justification':
require_once BASE_PATH . '/controllers/JustificationController.php';
$justCtrl = new JustificationController();
$justCtrl->review();
break;
```

```
case 'backups':
require_once BASE_PATH . '/controllers/BackupController.php';
$backupCtrl = new BackupController();
$backupCtrl->index();
break;
```

```
case 'create_backup':
require_once BASE_PATH . '/controllers/BackupController.php';
$backupCtrl = new BackupController();
$backupCtrl->create();
break;
```

```
case 'download_backup':
require_once BASE_PATH . '/controllers/BackupController.php';
$backupCtrl = new BackupController();
$backupCtrl->download();
break;
```

```
case 'cleanup_backups':
require_once BASE_PATH . '/controllers/BackupController.php';
$backupCtrl = new BackupController();
$backupCtrl->cleanup();
break;
```

```
case 'remove_role':
require_once BASE_PATH . '/controllers/UserController.php';
$userCtrl = new UserController();
```

```
$userCtrl->removeRole();
break;
```

```
case 'get_course_subjects':
require_once BASE_PATH . '/controllers/AttendanceController.php';
$controller = new AttendanceController();
$controller->getCourseSubjects();
break;
```

```
case 'get_teacher_course_subjects':
require_once BASE_PATH . '/controllers/AttendanceController.php';
$controller = new AttendanceController();
$controller->getTeacherCourseSubjects();
break;
```

```
case 'schedules':
require_once BASE_PATH . '/controllers/ScheduleController.php';
$controller = new ScheduleController();
$controller->index();
break;
```

```
case 'manage_schedule':
require_once BASE_PATH . '/controllers/ScheduleController.php';
$controller = new ScheduleController();
$controller->manageCourse();
break;
```

```
case 'delete_schedule_class':
require_once BASE_PATH . '/controllers/ScheduleController.php';
$controller = new ScheduleController();
$controller->deleteClass();
break;
```

```
case 'swap_schedule_class':
require_once BASE_PATH . '/controllers/ScheduleController.php';
$controller = new ScheduleController();
$controller->swapClass();
break;
```

```
case 'move_schedule_class':
```

```
require_once BASE_PATH . '/controllers/ScheduleController.php';
$controller = new ScheduleController();
$controller->moveClass();
break;
```

```
case 'get_schedule_info':
require_once BASE_PATH . '/controllers/AttendanceController.php';
$controller = new AttendanceController();
$controller->getScheduleInfo();
break;
```

```
case 'get_existing_attendance':
require_once BASE_PATH . '/controllers/AttendanceController.php';
$controller = new AttendanceController();
$controller->getExistingAttendance();
break;
```

```
case 'tutor_management':
require_once BASE_PATH . '/controllers/AssignmentController.php';
$controller = new AssignmentController();
$controller->tutorManagement();
break;
```

```
case 'check_course_tutor':
require_once BASE_PATH . '/controllers/AssignmentController.php';
$controller = new AssignmentController();
$controller->checkCourseTutor();
break;
```

```
case 'get_course_subjects_schedule':
require_once BASE_PATH . '/controllers/ScheduleController.php';
$controller = new ScheduleController();
$controller->getCourseSubjectsSchedule();
break;
```

```
case 'check_schedule_conflict':
require_once BASE_PATH . '/controllers/ScheduleController.php';
$controller = new ScheduleController();
$controller->checkScheduleConflict();
break;
```

```
case 'institution':
require_once BASE_PATH . '/controllers/InstitutionController.php';
$controller = new InstitutionController();
$controller->index();
break;
```

```
case 'update_institution':
require_once BASE_PATH . '/controllers/InstitutionController.php';
$controller = new InstitutionController();
$controller->update();
break;
```

```
case 'assign_institution_shift':
require_once BASE_PATH . '/controllers/InstitutionController.php';
$controller = new InstitutionController();
```

```

$controller->assignShift();
break;

case 'remove_institution_shift':
require_once BASE_PATH . '/controllers/InstitutionController.php';
$controller = new InstitutionController();
$controller->removeShift();
break;

case 'toggle_institution_shift':
require_once BASE_PATH . '/controllers/InstitutionController.php';
(new InstitutionController())->toggleShift();
break;

case 'create_user':
require_once BASE_PATH . '/controllers/UserController.php';
$controller = new UserController();
$controller->create();
break;

case 'create_user_modal':
require_once BASE_PATH . '/controllers/UserController.php';
$controller = new UserController();
$controller->createFromModal();

```

```

break;

case 'edit_user_modal':
require_once BASE_PATH . '/controllers/UserController.php';
$controller = new UserController();
$controller->editFromModal();
break;

case 'edit_user':
require_once BASE_PATH . '/controllers/UserController.php';
$controller = new UserController();
$controller->edit();
break;

case 'delete_user':
require_once BASE_PATH . '/controllers/UserController.php';
$controller = new UserController();
$controller->delete();
break;

case 'create_school_year':
require_once BASE_PATH . '/controllers/AcademicController.php';
$controller = new AcademicController();
$controller->createSchoolYear();
break;

case 'edit_school_year':
require_once BASE_PATH . '/controllers/AcademicController.php';
$controller = new AcademicController();
$controller->editSchoolYear();
break;

case 'delete_school_year':
require_once BASE_PATH . '/controllers/AcademicController.php';
$controller = new AcademicController();
$controller->deleteSchoolYear();
break;

case 'activate_school_year':
require_once BASE_PATH . '/controllers/AcademicController.php';

```

```

$controller = new AcademicController();
$controller->activateSchoolYear();
break;

case 'deactivate_school_year':
require_once BASE_PATH . '/controllers/AcademicController.php';

```

```
$controller = new AcademicController();
$controller->deactivateSchoolYear();
break;

case 'edit_course':
require_once BASE_PATH . '/controllers/AcademicController.php';
$controller = new AcademicController();
$controller->editCourse();
break;

case 'delete_course':
require_once BASE_PATH . '/controllers/AcademicController.php';
$controller = new AcademicController();
$controller->deleteCourse();
break;

case 'edit_subject':
require_once BASE_PATH . '/controllers/AcademicController.php';
$controller = new AcademicController();
$controller->editSubject();
break;

case 'delete_subject':
require_once BASE_PATH . '/controllers/AcademicController.php';
$controller = new AcademicController();
$controller->deleteSubject();
break;

case 'enroll_students':
require_once BASE_PATH . '/controllers/AcademicController.php';
$controller = new AcademicController();
$controller->enrollStudents();
break;
```

```
case 'unenroll_student':
require_once BASE_PATH . '/controllers/AcademicController.php';
$controller = new AcademicController();
$controller->unenrollStudent();
break;

case 'delete_backup':
require_once BASE_PATH . '/controllers/BackupController.php';
$controller = new BackupController();
$controller->delete();
break;

case 'remove_representative':
require_once BASE_PATH . '/controllers/RepresentativeController.php';
$controller = new RepresentativeController();
$controller->removeRelation();
break;

case 'assign_rep_from_academic':
require_once BASE_PATH . '/controllers/RepresentativeController.php';
$controller = new RepresentativeController();
$controller->assignFromAcademic();
break;

case 'remove_rep_from_academic':
require_once BASE_PATH . '/controllers/RepresentativeController.php';
$controller = new RepresentativeController();
$controller->removeFromAcademic();
break;

case 'toggle_primary_representative':
require_once BASE_PATH . '/controllers/RepresentativeController.php';
$controller = new RepresentativeController();
$controller->togglePrimary();
break;

case 'edit_representative':
require_once BASE_PATH . '/controllers/RepresentativeController.php';
```

```

$controller = new RepresentativeController();
$controller->editRelation();

break;

case 'notifications':
require_once BASE_PATH . '/controllers/NotificationController.php';
(new NotificationController())->index();
break;

case 'notifications_mark_read':
require_once BASE_PATH . '/controllers/NotificationController.php';
(new NotificationController())->markRead();
break;

case 'notifications_mark_all':
require_once BASE_PATH . '/controllers/NotificationController.php';
(new NotificationController())->markAllRead();
break;

case 'notifications_delete':
require_once BASE_PATH . '/controllers/NotificationController.php';
(new NotificationController())->delete();
break;

case 'notifications_delete_read':
require_once BASE_PATH . '/controllers/NotificationController.php';
(new NotificationController())->deleteRead();
break;

case 'notifications_unread_json':
require_once BASE_PATH . '/controllers/NotificationController.php';
(new NotificationController())->getUnread();
break;

case 'tutor_course_attendance':
require_once BASE_PATH . '/controllers/TutorController.php';
(new TutorController())->courseAttendance();
break;

case 'tutor_course_attendance_ajax':
require_once BASE_PATH . '/controllers/TutorController.php';
(new TutorController())->ajax();

break;

default:
header('Location: ?action=login');
}

```

public/css/global.css

```
/* =====
EcuAsist 2026 - global.css
Sistema de diseño unificado (basado en vista tutor)
===== */

/* Reset y base */
* { box-sizing: border-box; margin: 0; padding: 0; }

body {
font-family: -apple-system, BlinkMacSystemFont, 'Segoe UI', Roboto, sans-serif;
background: #f4f6f9;
color: #333;
font-size: 14px;
line-height: 1.5;
}

/* Breadcrumb */
```

```
.breadcrumb {  
padding: 10px 24px;  
background: #fff;  
border-bottom: 1px solid #e0e0e0;  
font-size: 0.83rem;  
color: #888;  
}  
.breadcrumb a { color: #007bff; text-decoration: none; }  
.breadcrumb a:hover { text-decoration: underline; }  
  
/* ■■■ Contenedor principal ■■■ */  
.container {  
max-width: 1200px;  
margin: 24px auto;  
padding: 0 16px;  
}  
.container-wide {  
max-width: 1400px;  
margin: 24px auto;  
padding: 0 16px;  
}  
  
/* ■■■ Page header (gradiente) ■■■ */  
.page-header {  
background: linear-gradient(135deg, #007bff, #0056b3);  
color: #fff;  
border-radius: 10px;  
padding: 20px 24px;  
margin-bottom: 20px;  
display: flex;  
align-items: center;  
gap: 16px;  
}  
.page-header .ph-icon { font-size: 2.2rem; }  
.page-header h1 { font-size: 1.3rem; font-weight: 700; }  
.page-header p { font-size: 0.85rem; opacity: 0.85; margin-top: 4px; }  
.page-header .ph-actions { margin-left: auto; display: flex; gap: 8px; flex-wrap: wrap; }  
  
/* Variantes de color para page-header */  
.page-header.green { background: linear-gradient(135deg, #28a745, #1e7e34); }  
.page-header.teal { background: linear-gradient(135deg, #17a2b8, #117a8b); }  
.page-header.orange { background: linear-gradient(135deg, #fd7e14, #dc6502); }  
.page-header.purple { background: linear-gradient(135deg, #6f42c1, #533187); }  
.page-header.dark { background: linear-gradient(135deg, #343a40, #23272b); }  
  
/* ■■■ Stat cards ■■■ */  
.stats-row {  
display: flex;  
gap: 12px;  
margin-bottom: 20px;  
flex-wrap: wrap;  
}  
.stat-card {  
background: #fff;  
border: 1px solid #e0e0e0;  
border-radius: 8px;  
padding: 14px 18px;  
flex: 1;  
min-width: 120px;  
border-top: 4px solid dddd;  
}  
.stat-card.number { font-size: 1.8rem; font-weight: 700; }  
.stat-card.label { font-size: 0.75rem; color: #888; margin-top: 2px; text-transform: uppercase; letter-spacing: 0.5px; }  
  
.stat-card.blue { border-top-color: #007bff; } .stat-card.blue.number { color: #007bff; }  
.stat-card.green { border-top-color: #28a745; } .stat-card.green.number { color: #28a745; }  
.stat-card.red { border-top-color: #dc3545; } .stat-card.red.number { color: #dc3545; }  
.stat-card.yellow { border-top-color: #ffc107; } .stat-card.yellow.number { color: #e6ac00; }  
.stat-card.teal { border-top-color: #17a2b8; } .stat-card.teal.number { color: #17a2b8; }
```


[illegible]

[illegible]

[illegible]

[illegible]

```
opacity: 1;
transform: translateX(0);
}
.toast.hide {
  opacity: 0;
  transform: translateX(40px);
}
.toast-success { background: #d4edda; border-left: 4px solid #28a745; color: #155724; }
.toast-danger { background: #f8d7da; border-left: 4px solid #dc3545; color: #721c24; }
.toast-warning { background: #fff3cd; border-left: 4px solid #ffc107; color: #856404; }
.toast-close {
  margin-left: auto;
  background: none;
  border: none;
  cursor: pointer;
  font-size: 1rem;
  color: inherit;
  opacity: 0.6;
  padding: 0;
  line-height: 1;
}
.toast-close:hover { opacity: 1; }
```